scASK: A novel ensemble framework for classifying cell types based on single-cell RNA-seq data

Bo Liu[1,3], Fang-Xiang Wu[2,*] and Xiufen Zou[1,*]

[1] School of Mathematics and Statistics, Wuhan University, Wuhan, 430072, China, [2] Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, S7N 5A9, Canada and [3] School of Mathematics and Statistics, Hubei Minzu University, Enshi, 445000, China

*Corresponding authors: Xiufen Zou xfzou@whu.edu.cn and Fang-Xiang Wu faw341@mail.usask.ca

Key Words: Single-cell, scRNA-seq data, Cell type classification, Data adaptive slicing, Machine learning, Ensemble strategy

ABSTRACT

The Human Cell Atlas (HCA) is a large project that aims to identify all cell types in the human body. The dimension reduction and clustering for identification of cell types from single-cell RNA-sequencing (scRNA-seq) data have become foundational approaches to HCA. The major challenges of current computational analyses are of poor performance on large scale data and sensitive to initial data. We present a new ensemble framework called Adaptive Slice KNNs (scASK) to address the challenges for analysing scRNA-seq data with high dimensionality. scASK consists of three innovational modules, called DAS (Data Adaptive Slicing), MCS (Meta Classifiers Selecting) and EMS (Ensemble Mode Switching), respectively, which facilitate scASK to approximate a bias-variance tradeoff beyond classification. Thirteen real scRNA-seq datasets are used to evaluate the performance of scASK. Compared with five popular classification algorithms, our experimental results indicate that scASK achieves the best accuracy and robustness among all competing methods. In conclusion, adaptive slicing is an effective structural reduction procedure, and meanwhile scASK provides novel and robust ensemble framework especially for classifying cell types based on scRNA-seq data. scASK is publically available at https://github.com/liubo2358/scASKcmd.

INTRODUCTION

The cell is the fundamental unit of living organisms, which is the key to study human biology and diseases (1). The Human Cell Atlas (HCA) project, with the ultimate goal of profiling all human cell types, has huge potential to transform many aspects of fundamental biology and clinical practice (2, 3), and thus attracts more and more attention from the scientific community. In just a few years, the ever-increasing scale of single-cell datasets (4, 5) and the emergence of new technologies and methods (6–8) have pushed single-cell transcriptome to a new level. Identification of subgroups (e.g., cell types or functional cell states) from single-cell data, as the most fundamental question (9, 10) for whole downstream analyses, has became the hot research topic and main driving force in single-cell transcriptome. However, the identification of cell types across datasets is confronting with both data and methodological challenges (3).

As mentioned above, popular methods for dimension reduction in single-cell transcriptome, like principal component analysis (PCA) and t-distributed stochastic neighbour embedding (t-SNE), often suffer from the high dimensionality and high variability in data. Although PCA and t-SNE have been widely used as de facto gold standards for analysing or visualizing single-cell data, these algorithms assume that the underlying data are drawn from a Gaussian or a t-distribution, which does not always hold for scRNA-seq data (11). The discrepancy between the assumed and actual distribution fundamentally limits the accuracy of the resulting predictions (12). In practice, PCA which relies on singular value decomposition (SVD) is always computationally expensive, the traditional PCA computing out all singular values of $m \times n$ matrix has time complexity of $O(min\{mn^2, m^2n\})$ (13). The computational complexity makes it infeasible to apply to very large-scale datasets, for example, when the sample size is more than one million. Moreover, the exact SVD is an algorithm which is difficult to parallelize with high efficiency, due to its matrix factorization nature (14). t-SNE is another popular, but more complicated method for dimension reduction and is particularly well suited for the visualization of high-dimensional data (15). The t-SNE algorithm doesn't always produce similar output on successive runs, for example, and there are additional hyperparameters (such as "perplexity") related to the optimization process, which makes it tricky to interpret. In other words, t-SNE plots can sometimes be mysterious or misleading (16), it is a valuable tool in generating hypotheses and understanding, but does not produce conclusive evidence. Non-negative matrix factorization (NMF) is another matrix decomposition technique widely studied (17–19) in computational biology for dimension reduction (or feature extraction). Benefiting from the nonnegative constraint, the results of NMF are usually easier to interpret than those of SVD. However, the existence, uniqueness, effectiveness of NMF solutions still need to be further studied (20–22).

Paralleling with the dimension reduction, several clustering methods have been developed for identification of novel cell types, such as SNN-cliq (23), RaceID (24), pcaReduce (25), SC3 (26) and SIMLR (27). Most of these clustering methods are based on metrics comparing intra- and inter-cluster similarity without references to external information. As unsupervised learning, the validation for clustering is very difficult in the absence of prior knowledge (28). For example, given a dataset, each clustering method will always find some partitioning, no matter whether the structure exists or not (29). Mathematically, clustering is essentially a heuristic algorithm to detect the subpopulation structure of existing data, different methods usually lead to different clusters, and even for the same method, the choice of the number of clusters K or the initial states of the same set of data may affect the final results. From an algorithmic perspective, clustering is a feasible approach to identify novel cell types from existing data, but is not an ideal way to apply known labels to new data — if only one type of cells in a sample, clustering can do nothing. In contrast, for many samples there exists a hierarchy of cell-types and cell-states, and they may all be of interest (30). Based on the analysis above, as more and more known cell types have been located within the HCA. Therefore, it is appealing to develop more reliable methods to achieve high accuracy of cell type identification for new samples beyond clustering and dimension reduction.

Generally, a better approach of cell type identification for gene expression data with the label information is to use machine learning techniques based on supervised learning, such as classification.

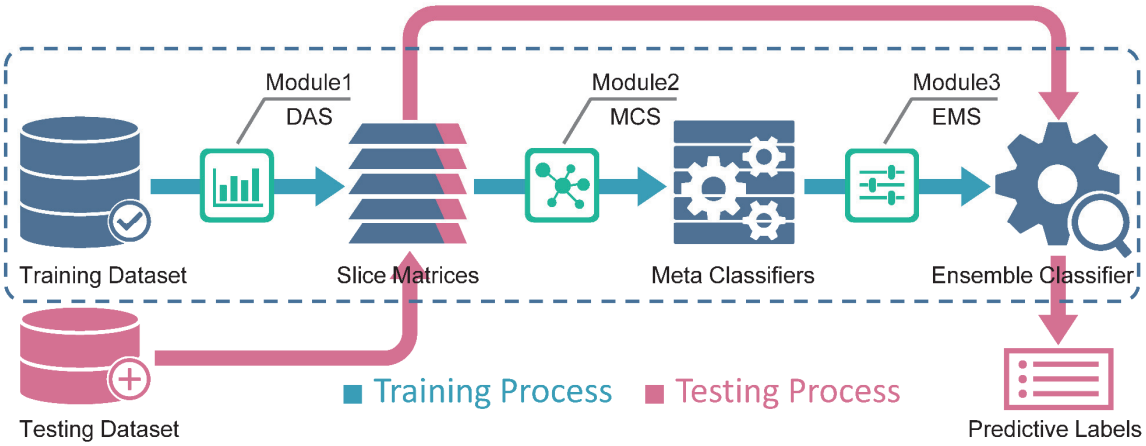


Figure 1. The framework of the proposed scASK. It consists of three modules, i.e., Data Adaptive Slicing (DAS), Meta Classifiers Selecting (MCS) and Ensemble Mode Switching (EMS), respectively. The first module DAS uses novel adaptive slicing procedure to extract latent classification features. The second module MCS selects three kinds of efficient distance measures Correlation, Jaccard and Cosine to construct competitive meta classifiers for diversity and complementarity. The third module EMS applies a new switching strategy for ensemble classification with dual operating mode to enhance the accuracy and the robustness of the final prediction. For evaluating the real performance of the ensemble classifier, we have reserved a portion of raw dataset as "testing dataset" independent of the training process, then calculate the overall accuracy by comparing predictive labels with true labels.

In the past few years, numerous classification methods from single-cell data have been proposed (31–35), which were focusing mostly on specific tissues (e.g., bone marrow and neurons) or specific phenomenon of data (e.g., dropout and cross-dataset). Several classification algorithms were employed by the methods above, such as Naive Bayes (NB), Decision Trees (DT), Support Vector Machine (SVM) and k Nearest Neighbours (kNN). Regardless of underlying data distribution and tuning parameters, all of these classifiers can make effective classifications on specific datasets. However, high dimensionality and high variability as intrinsic characteristics of single-cell data are always obstacles for all computational techniques. Classification of cell types from single-cell data with label information, which guarantees the high accuracy as well as the high robustness on most of datasets, is a very challenging problem in the current study. To the best of our knowledge, there is no generic method available that performs well for both the objectives on scRNA-seq data.

Due to the nature of scRNA-seq data, there are some challenges in single-cell data analysis. The first notorious characteristic of single-cell data is high dimensionality. In a typical single-cell dataset, the number of cells or genes can easily reach or exceed 10^4, while new experimental techniques have raised the upper limit of cell counts to 10^6 and are still growing. The second notorious characteristic of single-cell data is high sparsity. In some single-cell datasets, the proportion of zeroes is as high as 80%-90%. Unfortunately, even in non-zero data, there are a lot of high levels of technical noise and confounding factors. Based on the above facts, we need to fully consider the intrinsic characteristics of single-cell data when developing new methods. Specifically, how to fully extract the classification information hidden in high-dimensional data, while at the same time reducing the sensitivity of the algorithm to missing values and noise as much as possible.

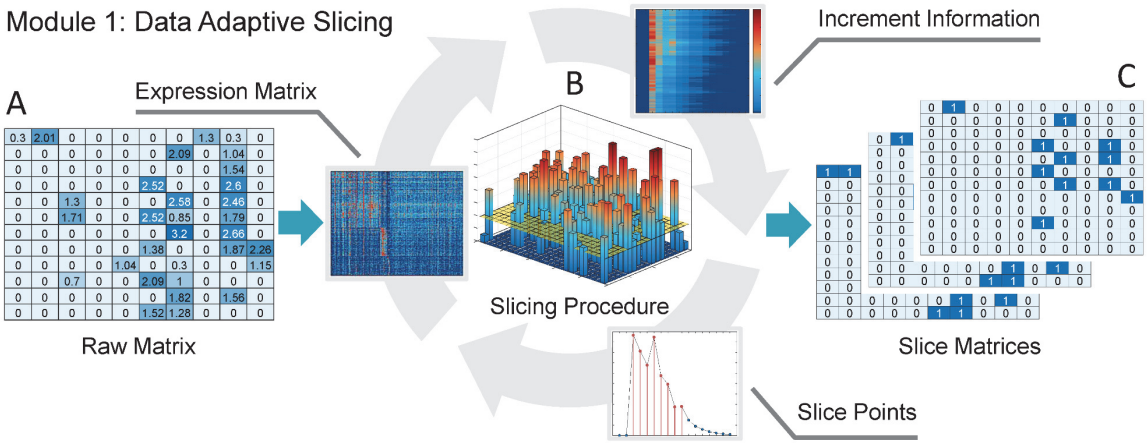


Figure 2. Module 1 of the scASK: Data Adaptive Slicing. (**A**) The raw matrix of scRNA-seq data with rows representing cells and columns representing genes, which measures the distribution of expression levels for each gene across a population of cells. (**B**) scASK stretches the data matrix along Z axis to three-dimensional space according to the range of gene expression values, and then slices the raw matrix to many slice matrices on selected slice points with the aid of quantifying the cumulation of binary switching in expression state between all slice matrices. (**C**) A series of binary slice matrices are obtained after executing the Data Adaptive Slicing. The core function of module 1 is deriving more informative slice matrices from raw matrix, which mainly depends upon the design of increment index $SCRIstd$.

In this study, we present a novel ensemble classification framework called scASK based on adaptive data slicing, which is the first generic ensemble classification framework especially for classifying cell types based on scRNA-seq data with high dimensionality. Compared with traditional individual or ensemble classifiers, the most remarkable advantage of scASK is that it applies the known cell-type labels to new samples with high accuracy and high robustness in a near-real time manner.

MATERIAL AND METHODS

The scASK framework

One important target for scRNA-seq analysis is the identification of cell types from different samples, where each cell-type consists of cells with the similar gene expression patterns (36). In this paper, we defined the similarity of gene expression patterns as the differential expression of cells and its structural similarity at different levels of gene expression values. Computational techniques focus on extracting, measuring and evaluating these cell-to-cell similarities in scRNA-seq data to construct the core functions of scASK. The framework of scASK is shown in Figure 1, which contains the following three modules:

Module 1: Simplify the computational complexity of high-dimensional scRNA-seq data with novel adaptive slicing procedure for sufficiently extracting latent classification features.

Module 2: Select kNN with three kinds of efficient distance measures Correlation, Jaccard and Cosine to construct competitive meta classifiers for diversity and complementarity.

Module 3: Apply a new switching strategy for ensemble classification with dual operating mode to enhance the accuracy and the robustness of the final prediction.

Module 1: Data Adaptive Slicing (DAS)

To sufficiently extract latent classification features from scRNA-seq data, we execute the module DAS on raw gene expression matrix. Then we obtain a series of binary matrices called slice matrices that are selected for the subsequent ensemble learning. The core function and basic principle of module 1 are shown in Figure 2.

The slicing procedure is designed as the threshold operation:

$$S_k(s_{ij}) = \left[\begin{matrix} s_{ij} = 1, r_{ij} > z_k \\ s_{ij} = 0, r_{ij} \le z_k \end{matrix}\right], r_{ij} \in R, z_k \in \{z_1, z_2, \dots, z_q\} \quad (1)$$

where R is the raw matrix of gene expression counts that has been normalized and transformed to proper range (27, 37). S_k is the kth slice matrix of the raw scRNA-seq data, which keeps the structure of differential expression at value z_k with binary manner. $\{z_1, z_2, \dots, z_q\}$ is a set of slice points between the minimum and maximum values of R, which are in ascending order for slicing procedure. This slicing procedure is used to extract necessary classification features from scRNA-seq data for machine learning.

By analyzing the histogram distribution of raw gene expression matrix, we can obtain the following results: (i) Different slice matrices at different slice points are derived, while these matrices keep different information of differential expression for cells. (ii) Informative slice points are selected. To quantify the information of switching in expression states, we defined the row increment index named $SCRIstd$ as

$$SCRIstd_k = std\left(\left\{\sum_{j=1}^{n}\left|S_{k+1}(s_{ij}) - S_k(s_{ij})\right| : i = 1,2, \dots, m\right\}\right) \quad (2)$$

where m represents the maximum number of rows, and n represents the maximum number of columns. The term "std" is used here to refer to the process of calculating standard deviation of all row increments. If the calculated increment index $SCRIstd_{k+1} > SCRIstd_{k+2}$ from slice points z_k, z_{k+1}, z_{k+2}, which indicates that the slice matrix S_{k+1} contains more expression state switching increment information than S_{k+2}. We calculate all the increment index $SCRIstd$ between slice matrices and arrange them in descending order, then select the slice points corresponding to the first p increment indices to obtain the candidate set of slice points $\{z_{(1)}, z_{(2)}, \dots, z_{(p)}\}$. Note that in scASK, we added 0 as the first slice point for the technical requirements, and the actual set of candidate slice points is $\{0, z_{(1)}, z_{(2)}, \dots, z_{(p)}\}$.

Module 2: Meta Classifiers Selecting (MCS)

The kNN classifier is a commonly used supervised machine learning algorithm, it is particularly well suited for multi-modal classes as its classification decision is based on a small neighbourhood of similar objects. As such, even if the target class consists of cells whose types have different characteristics for gene expression patterns, it can still lead to good classification accuracy (38).

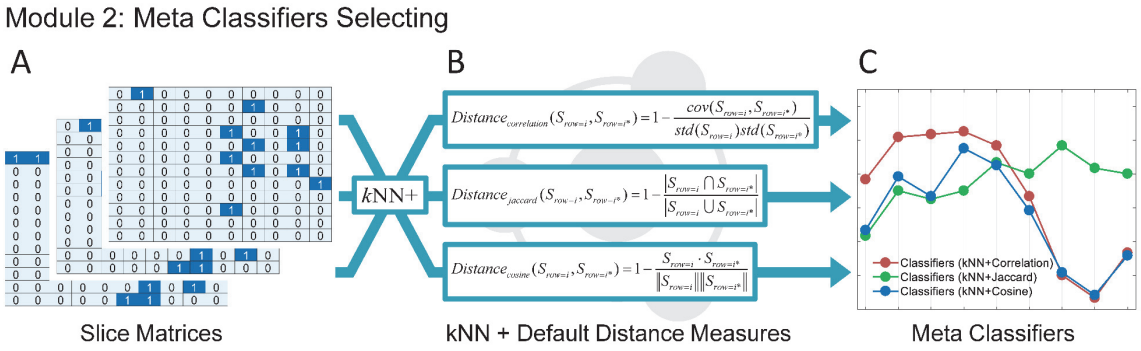


Figure 3. Module 2 of the scASK: Meta Classifiers Selecting. (**A**) The series of binary slice matrices. (**B**) scASK adopts Pearson's correlation coefficient, Jaccard similarity and Cosine similarity as the default distance measures, where these three approaches could keep both diversity and complementarity in extracting the classification features from scRNA-seq data. (**C**) Three types of meta classifiers could be obtained on every slice point (training on every slice matrix). The core function of module 2 is selecting more appropriate fundamental algorithm and distance measures for binary slice matrices, which guarantees that all trained meta classifiers are competitive enough for subsequent ensemble classification.

scASK adopts Pearson's correlation coefficient, Jaccard similarity and Cosine similarity as the default distance measures, where these three approaches could keep both diversity and complementarity in extracting the classification features from scRNA-seq data. Specifically, Pearson's correlation coefficient measures the increasing and decreasing trend of gene expression data between the cells, Jaccard similarity measures the degree of overlap between intercellular binary expression patterns, and Cosine similarity calculates the vector angle between the cells using gene expression data as features. The exact formula definition of these three distance measures for scRNA-seq data can be found in Supplementary Materials Formulas S1 to S3.

As we know, kNN with three distance measures can produce $3(p+1)$ classifiers on $(p+1)$ slice matrices. Not all of trained classifiers could carry out adequate classification, some of classifiers may be strong, others may be weak. A common practice is to calculate the accuracy on training data and testing data for every classifier, then compare and select the more competitive classifiers (called meta classifiers). The core function and basic principle of module 2 are shown in Figure 3.

Module 3: Ensemble Mode Switching (EMS)

In many metrics used for evaluating the performance of a classification model, one important metric is the accuracy on testing data. In real scenarios, the accuracy on testing data is often inconsistent with the accuracy on training data, the essential reason of this phenomenon in machine learning is a high bias problem or a high variance problem, i.e., an under-fitting problem or an over-fitting problem (39). Both of problems could cause the significant inconsistency. Therefore, in scASK, we propose a weighted accuracy named $SMARwit$ to evaluate and select out the meta classifiers with better performance, which is designed as follows:

$$SMARwit_k^{(s)} = \lambda A_k^{(s)}{}_{testing} + (1-\lambda) A_k^{(s)}{}_{training} + \rho\left(A_k^{(s)}{}_{penalty}\right) \quad (3)$$

where λ is a predefined weight coefficient, $A_k^{(s)}{}_{testing}$ represents the accuracy of the sth meta classifier on testing data from the kth slice matrix, and $A_k^{(s)}{}_{training}$ represents the accuracy of the sth meta classifier on training data from the kth slice matrix. The strength function ρ and the penalty term $A_k^{(s)}{}_{penalty}$ are designed as follows:

$$\begin{cases} \rho\left(A_k^{(s)}{}_{penalty}\right) = 0.01 \times \begin{cases} log_{10}\left(1 - A_k^{(s)}{}_{penalty}\right), 0 \le A_k^{(s)}{}_{penalty} \le 0.9 \\ -1, A_k^{(s)}{}_{penalty} > 0.9 \end{cases} \\ A_k^{(s)}{}_{penalty} = \dfrac{\left|A_k^{(s)}{}_{testing} - A_k^{(s)}{}_{training}\right|}{\sum_{t=1}^{3}\left|A_k^{(t)}{}_{testing} - A_k^{(t)}{}_{training}\right| + eps} \end{cases} \tag{4}$$

where $\rho\left(A_k^{(s)}{}_{penalty}\right)$ is a piecewise and decreasing function based on logarithm, which controls the strength of the penalty term affecting the weighted accuracy, and the penalty term $A_k^{(s)}{}_{penalty}$ is used for measuring the consistency of $A_k^{(s)}{}_{testing}$ and $A_k^{(s)}{}_{training}$, where the differences between testing accuracy and training accuracy for classifiers on each slice matrix are mapped to the proportional values in a same scale. It is important to note here that the term "eps" represents a floating-point number (is 2.2204e-16 in Matlab) small enough to keep the denominator away from zero. Generally, the value of $SMARwit$ is mainly adjusted by λ (set 0.8 as default), and is slightly adjusted by the $\rho\left(A_k^{(s)}{}_{penalty}\right)$ (range from -0.01 to 0). If the value of $SMARwit$ is close to 1, it means that the meta classifier is more robust.

The generalization ability, which characterizes how well the results learned from a given training dataset can be applied to a new dataset, is the most central concept in machine learning. Researchers have devoted tremendous efforts to the pursuit of techniques that could lead to a learning system with a strong generalization ability. One of the most successful paradigms is ensemble learning (40). In scASK, we developed a new switching strategy especially for slice matrices, which is distinct from the traditional strategies for ensemble classification (i.e., bagging, boosting and stacking).

Now we can associate the steps established previously within a complete process. Suppose that a set of $(p+1)$ candidate slice points was selected out by adaptive slice algorithm in module 1, then $(p+1)$ slice matrices would be obtained accordingly as the training dataset. Three types of meta classifiers were constructed in module 2, would totally produce $3(p+1)$ classifiers on all slice matrices. Finally, the last step of scASK is to decide how the candidate classifiers can be integrated in the ensemble.

In order to simplify expressions, we first define a special function named $sort$ which sorts the given accuracy set in descending order and selects out the element of specified order number as

$$sort(\{A_1, A_2, \dots, A_I\}, i) = A_i, 1 \le i \le I \tag{5}$$

Then, two operating modes have been developed respectively. One mode named $SMESrws$, which is designed as follows:

$$SMESrws = \left\{(s,k) | sort\left(\left\{SMARwit_k^{(s=1,2,3)}\right\}, i = 1,2\right) : k = 1,2,\dots,p+1\right\} \tag{6}$$

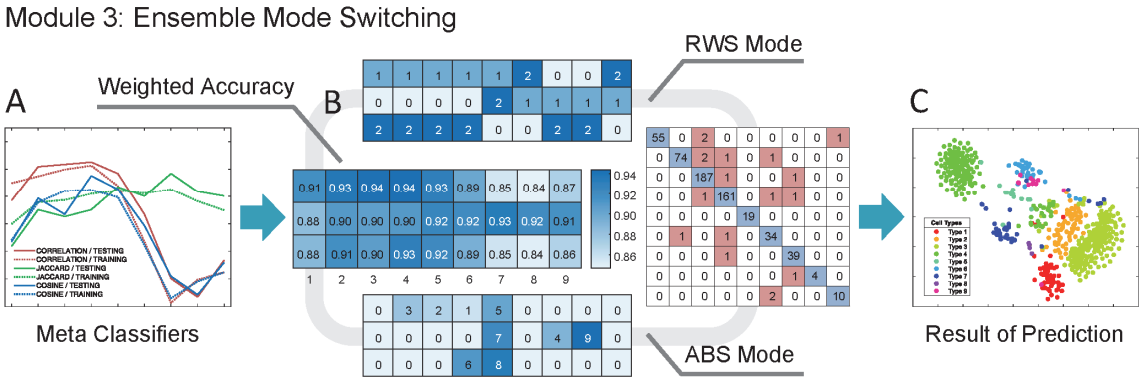


Figure 4. Module 3 of the scASK: Ensemble Mode Switching. (**A**) The accuracy of meta classifiers is calculated on every slice point. The solid line represents accuracy on testing data, and the dotted line represents accuracy on training data. Different colours represent different distance measures adopted by kNN. (**B**) scASK sorts all of meta classifiers with the weighted accuracy (implementing by the $SMARwit$), and then integrates them for final ensemble with two modes: the RWS mode and the ABS mode. We denote the dual operating mode on slice points for ensemble as "switching strategy", which enhances the accuracy and the robustness of the final ensemble classifier. (**C**) scASK running with switching strategy (implementing by the $SMESrws$ and $SMESabs$) can achieve better prediction for classifying cell types based on scRNA-seq data. The core function of module 3 is switching meta classifiers for ensemble according to the weighted accuracy.

where (s,k) represents the index location of classifier (i.e., the sth meta classifier from the kth slice matrix) that satisfies the given condition, $SMARwit_k^{(s=1,2,3)}$ represents the set of every weighted accuracy from the kth slice matrix. After that the index locations of optimal and suboptimal classifiers are derived by the aid of the $sort$ function. The procedure of $SMESrws$ arranges classifiers on each slice matrix, which generates the index table of meta classifiers for the final ensemble classifier.

Another mode named $SMESabs$, which is designed as follows:

$$SMESabs = \{(s,k)|sort(\{SMARwit_{k=1,2,\dots,p+1}{}^{(s=1,2,3)}\}, j = 1,2,\dots,p+1)\} \quad (7)$$

where (s,k) represents the index location of classifier (i.e., the sth meta classifier from the kth slice matrix) that satisfies the given condition, $SMARwit_{k=1,2,\dots,p+1}{}^{(s=1,2,3)}$ represents the set of every weighted accuracy from all slice matrices. After that the index locations of the top $(p+1)$ best classifiers are derived by the aid of the $sort$ function. The procedure of $SMESabs$ arranges classifiers on all slice matrices, which generates the index table of meta classifiers for the final ensemble classifier.

To summarize, $SMESrws$ selects out two meta classifiers from each slice matrix for the final ensemble with the simple voting scheme. The process of the selection performed by $SMESrws$ is called "$railway\ switching$" (RWS). $SMESabs$ selects out $(p+1)$ meta classifiers from all slice matrices for the final ensemble with the simple voting scheme. The process of the selection performed by $SMESabs$ is called "$absolute\ best\ switching$" (ABS). Briefly, the RWS mode is a local optimal solution, while the ABS mode is a global optimal solution. For better accuracy and robustness, scASK switches the ensemble strategy between two modes manually according to the practice and experience. The core function and basic principle of module 3 are shown in Figure 4.

DATASETS AND RUNNING PROCEDURES

To verify the classification performance of scASK on real single-cell datasets, we used five single-cell datasets from Wang et al., 2017 (27) and eight single-cell datasets from Park and Zhao, 2018 (36). All these datasets were preprocessed and transformed into appropriate ranges (e.g., $log_{10}(x+1)$). All cells have accurate and reliable cell-type labels with sample dimensions from 80 to 3005, the gene dimensions from 959 to 17772. The proportion of zeros is from 27.87% to 78.10%. These data have typical characteristics of high dimensionality and high sparsity. The summary of all datasets used in this paper is shown in Table 1.

Table 1. Summary of the thirteen real single-cell datasets

Dataset	Number of Cells	Number of Genes	Populations	Sparsity
Buettner	182	8989	3	37.94%
Kolod	704	10685	3	27.87%
Pollen	249	14805	11	51.00%
Usoskin	622	17772	4	78.10%
Zeisel	3005	4412	9	46.01%
Buettner*	182	8989	3	36.29%
Deng	135	12548	7	31.85%
Ginhoux	251	11834	3	66.54%
Pollen*	249	14805	11	50.79%
Tasic	1727	5832	48	32.71%
Ting	114	14405	5	52.17%
Treutlin	80	959	5	51.65%
Zeisel*	3005	4412	48	46.01%

Note: The superscript * means the dataset from the same origin literature with different data preprocessing. Datasets 1 to 5 are available from the Supplementary data of Wang et al., 2017, and datasets 6 to 13 are available from the Supplementary data of Park and Zhao, 2018, where the descriptions for datasets with more details can be found respectively.

In machine learning, we always build a model based on the training set (a subset of a raw dataset), and evaluate the model based on the test set (the remaining subset of the raw dataset). Because the test set is completely independent of the training process, so the prediction accuracy on the test set is regarded as the most significant evaluation metric of the model. The cross validation is a popular resampling procedure which is also used to evaluate the performance of the model on new data. This procedure has a single parameter called k-fold that refers to the number of pieces that a given dataset is to be split into. There is no hard and fast rule for the choice of k: the larger value of k causes the less bias and more variance, while the smaller value of k yields more bias and less variance (39). It is noteworthy that, the procedure of cross validation is also adopted in scASK for evaluating the performance of meta classifiers during training process.

Next we present the running procedures of scASK for cell type classification on single-cell datasets, and explain some key parameters and give empirical values. The running procedures include six steps. On each step, one or more computing tasks are executed by running specific functions sequentially. The complete codes and output results for each single-cell dataset can be found in Supplementary Materials.

Step 1. Input the dataset and run the **slicematrix** function to initially determine the starting point, end point and interval of the slice by plotting the raw data histogram. The slice interval needs to cover all occurrences of "peaks" and "valleys" in the histogram. In theory, the smaller the slice interval is, the easier it is to capture the classification features hidden in the slice matrices. In practice, this value is

determined by the equilibrium between storage cost and calculation duration. (the empirical value: q=30~60).

Step 2. Run the **slicediffer** function to calculate the increment index $SCRIstd$ between the slice matrices at all slicepoints, and select the first $(p+1)$ slice points to form the candidate set of slice points. (the empirical value: $(p+1)$=6, 9, 15, 21, 36).

Step 3. Run the **slicemethod** function to train the kNN classifiers with Pearson's correlation coefficient, Jaccard similarity and Cosine similarity on the $(p+1)$ slice matrices. The total number of $3(p+1)$ classifiers form the candidate set of meta classifiers. (the empirical value: k=1 or 5, distance weighted by "inverse").

Step 4. Run the **sliceweight** function to calculate the accuracy of the test set and the training set for each classifier, then calculate the corresponding weighted accuracy according to the $SMARwit$.

Step 5. Run the **sliceswitch** function in $SMESrws$ and $SMESabs$ modes respectively to implement the meta classifiers ensemble with the switching strategy. As a matter of experience, $SMESrws$ is more conservative than $SMESabs$ on the prediction.

Step 6. Run the **sliceprerws** function and the **slicepreabs** function respectively to verify the prediction accuracy of the test set of scASK under the two ensemble strategies. The classification results are given in the form of confusion matrix, and the cell-type label and support degree of each sample are given at the same time.

Finally, we provide the empirical parameters and the command-line format for scASK on Buettner dataset as a running example:

Demo: running procedures for scASK on Buettner dataset

```
[...] = slicematrix(in_X,true_labs,0:0.01:0.5);
[...] = slicediffer(in_X_SLC,slice_tik,9);
[...] = slicemethod(in_X_SLC,true_labs,slice_tik,slice_bst,5,'correlation','inverse');
[...] = slicemethod(in_X_SLC,true_labs,slice_tik,slice_bst,5,'jaccard','inverse');
[...] = slicemethod(in_X_SLC,true_labs,slice_tik,slice_bst,5,'cosine','inverse');
[...] = sliceweight(SLC_Model_DIS1,SLC_Model_DIS2,SLC_Model_DIS3);
[...] = sliceswitch(SLC_Model_All,SLC_Model_All_SMARwit,'rws',2);
[...] = sliceswitch(SLC_Model_All,SLC_Model_All_SMARwit,'abs',9);
[...] = sliceprerws(...,binary_mod,class_num,slice_vle,SLC_Model_RWS);
[...] = slicepreabs(...,binary_mod,class_num,slice_vle,SLC_Model_ABS);
```

Note: The statements are simplified for demonstration in compact form, and the full version can be found in Supplementary Materials Table S1.

RESULT

To evaluate the effectiveness and generality of our new ensemble framework for classifying cell types based on scRNA-seq data, we have implemented scASK on thirteen real single-cell datasets listed in Table 1. All of computational results and figures during the running process of scASK on every dataset have been collected in Supplementary Materials Tables S1 to S17 and Figures S1 to S187.

In this section, firstly we demonstrate the core functions of scASK on three real single-cell datasets: Buettner, Pollen and Zeisel. These datasets, from small to large scale, are typical kind of scRNA-seq data. Then we evaluate the performance of scASK by comparing with five baseline methods on all datasets listed in Table 1. The parameters for scASK on one specific dataset are dependent of data

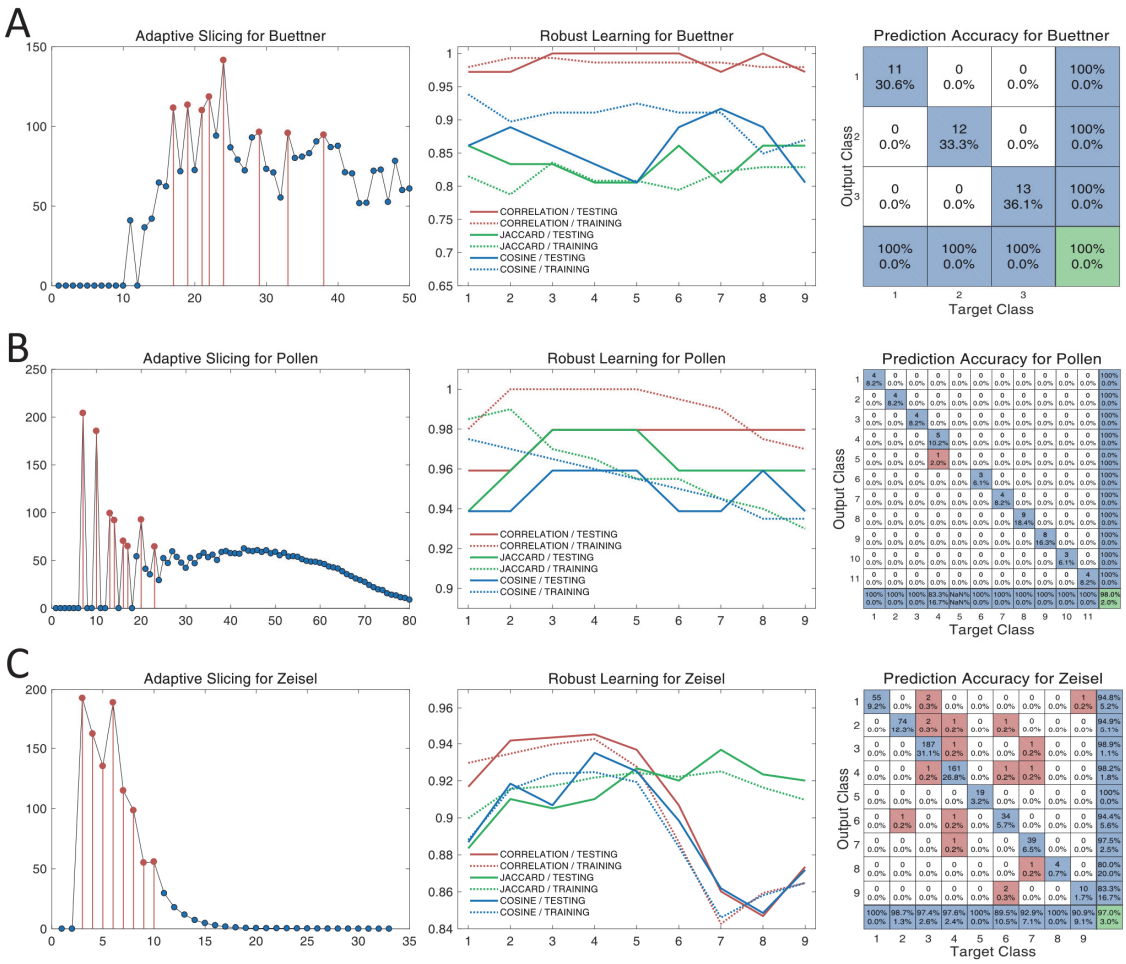


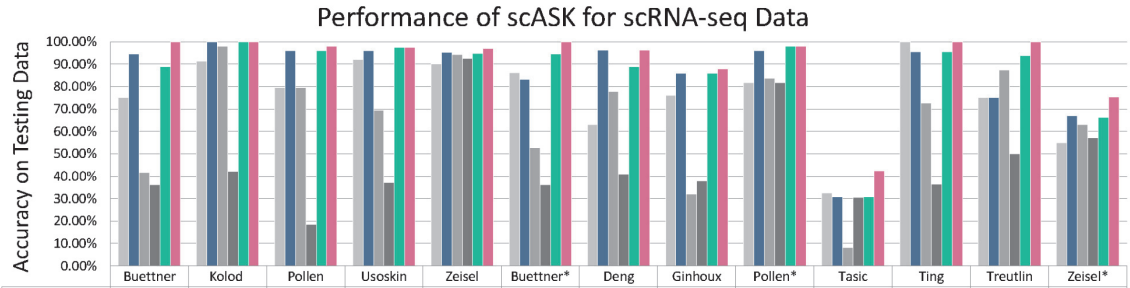
Figure 5. Performance of scASK on three real datasets Buettner, Pollen and Zeisel. (A) Result for Buettner dataset. 9 slice points (0.17,0.19,0.21,0.22,0.24,0.29,0.33, 0.38, and 0) were selected out with the aid of adaptive slicing procedure; 27 meta classifiers were constructed by kNN with 3 kinds of efficient distance measures (Correlation, Jaccard and Cosine); After ensemble of these meta classifiers, scASK (both of RWS mode and ABS mode) achieved the classification accuracy of 100% on the Buettner dataset (test set). (B) Result for Pollen dataset. 9 slice points (0.35,0.5,0.65,0.70,0.80,0.85,1,1.15, and 0) were selected out with the aid of adaptive slicing procedure; 27 meta classifiers were constructed by kNN with 3 kinds of efficient distance measures (Correlation, Jaccard and Cosine); After ensemble of these meta classifiers, scASK (both of RWS mode and ABS mode) achieved the classification accuracy of 98% on the Pollen dataset (test set). (C) Result for Zeisel dataset (9 classes). 9 slice points (0.9,1.2,1.5,1.8,2.1,2.4,2.7,3, and 0) were selected out with the aid of adaptive slicing procedure; 27 meta classifiers were constructed by kNN with 3 kinds of efficient distance measures (Correlation, Jaccard and Cosine); After ensemble of these meta classifiers, scASK (RWS mode) achieved the classification accuracy of 97% on the Zeisel dataset (test set).

experiments. Therefore, the given empirical parameters are only for result reproducibility, and may change in different scenarios.

Performance of scASK on three real datasets

To evaluate the performance of our proposed scASK, we first introduce three real single-cell datasets: Buettner, Pollen and Zeisel (36).

(1) Buettner: In this dataset, the transcriptional profiles of 182 embryonic stem cells (ESCs) had been staged for cell-cycle phases (G1, S, and G2M) based on sorting of the Hoechst 33342-stained



	Buettner	Kolod	Pollen	Usoskin	Zeisel	Buettner*	Deng	Ginhoux	Pollen*	Tasic	Ting	Treutlin	Zeisel*
Complex Tree	75.00%	91.43%	79.59%	91.94%	90.18%	86.11%	62.96%	76.00%	81.63%	32.46%	100.00%	75.00%	54.91%
Quadratic SVM	94.44%	100.00%	95.92%	95.97%	95.17%	83.33%	96.30%	86.00%	95.92%	30.72%	95.45%	75.00%	66.89%
Medium KNN	41.67%	97.86%	79.59%	69.35%	94.18%	52.78%	77.78%	32.00%	83.67%	8.12%	72.73%	87.50%	63.06%
Boosted Trees	36.11%	42.14%	18.37%	37.10%	92.68%	36.11%	40.74%	38.00%	81.63%	30.43%	36.36%	50.00%	57.24%
Bagged Trees	88.89%	100.00%	95.92%	97.58%	94.84%	94.44%	88.89%	86.00%	97.96%	30.72%	95.45%	93.75%	66.22%
scASK Best	100.00%	100.00%	97.96%	97.58%	97.00%	100.00%	96.30%	88.00%	97.96%	42.32%	100.00%	100.00%	75.21%

Figure 6. Comparisons of scASK in accuracy with five other baseline methods for scRNA-seq Data. The term "scASK Best" represents the optimal mode between $SMESrws$ and $SMESabs$. scASK keeps the highest prediction accuracy on thirteen real scRNA-seq datasets.

cell area of a flow cytometry (FACS) distribution (41). The cells were grouped into three stages of the cell cycle, and they were validated using the gold-standard Hoechst staining. scASK achieves the classification accuracy of 100% on Buettner dataset, and the key parameters and summary results are shown in Figure 5A.

(2) Pollen: it consists of the transcriptional profiles of 249 single cells from 11 populations using microuidics, including neural cells and blood cells (42). The 11 clusters in the dataset were from different sources (CRL-2338, CRL-2339, K562, BJ, HL60, hiPSC, Keratinocyte, Fetal cortex (GW21+3), Fetal cortex (GW21), Fetal cortex(GW16), and NPC) that are expected to show robust differences in gene expression. scASK achieves the classification accuracy of 98% on Pollen dataset, and the key parameters and summary results are shown in Figure 5B.

(3) Zeisel: it consists of the transcriptional profiles of 3005 cells from the mouse cortex and hippocampus. Zeisel et al., 2015 (30) found 9 or 48 molecularly distinct subclasses identified by hierarchical biclustering and validated by gene markers. scASK achieves the classification accuracy of 97% on Zeisel dataset (9 classes), and the key parameters and summary results are shown in Figure 5C.

Comparisons of scASK with five baseline methods for scRNA-seq Data

To further demonstrate the effectiveness of the proposed scASK, we compare scASK with five representative classification algorithms including Complex Tree, Quadratic SVM, Medium KNN, Bagged Tree and Boosted Tree in terms of accuracy and robustness. All approaches are run on a laptop computer with Intel Core i5-6200U @ 2.30GHz and 8 GB of RAMs and five other algorithms are implemented by using their corresponding functions provided by Matlab 2017a.

On the one hand, we compare the prediction accuracy of all methods. The prediction accuracy on test sets is not the only metric for evaluating the performance of a classification model, but it is most important for measuring the classification rate which ranges from 0 to 100%, and a high value means that the algorithm accurately discovers cell types from new samples (43). For classifying cell types based on scRNA-seq data, three competing algorithms (Bagged Tree, Quadratic SVM and scASK)

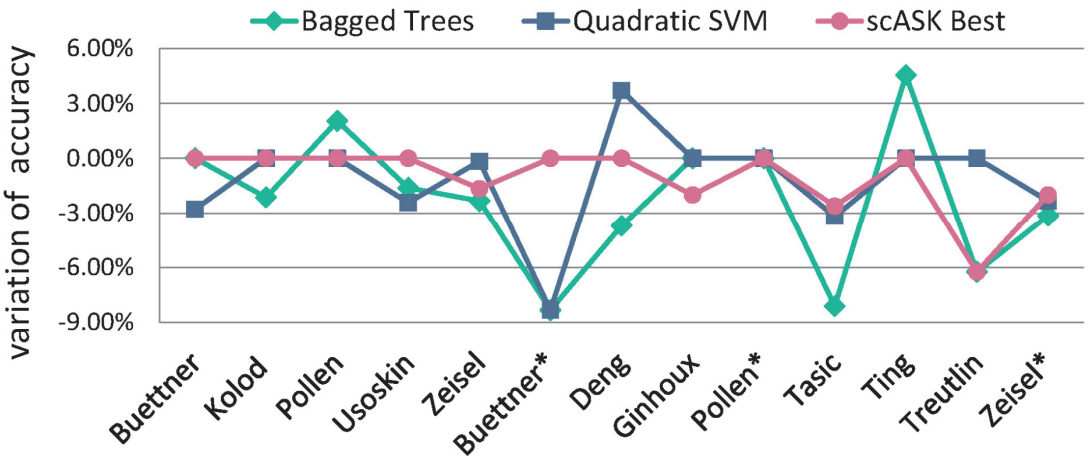


Figure 7. Comparisons of scASK in robustness with two other methods Bagged Trees and Quadratic SVM. The green, blue and red colours are used to represent Bagged Trees, Quadratic SVM and the scASK, respectively. The smaller variation of accuracy on datasets indicate that the method has stronger robustness.

have been selected out because of their outstanding performance. From Figure 6 we can observe that scASK achieves the best prediction accuracy on all datasets.

On the other hand, we evaluate the robustness of these algorithms on real single-cell datasets. One important characteristic of scRNA-seq data is the “dropout” phenomenon where a gene is observed in one cell but undetected in another cell (44). Based on this consideration, we simulate missing values by randomly replacing the non-zero elements of the original data with zeroes for a certain proportion (10%) on thirteen real scRNA-seq datasets. The variations of prediction accuracy before and after the simulations are used for measuring the robustness of the algorithms. As shown in Figure 7, scASK keeps the best robustness under the strong disturbances during the simulations across all datasets.

Software package

We develop a graphical user interface (GUI) software package for implementing scASK in Matlab 2019a (shown in Figure 8), which can significantly simplify the analysis and the process of parameter selection. The software is publicly available at https://github.com/liubo2358/scASKapp. The description documentation and quick tutorials of this software package are presented online in the same repository.

DISCUSSION

Generally, if we regard the Human Cell Atlas as the dictionary of information on all cell types in human body, scASK provides a fast, accurate and reliable method for querying this dictionary. The main goal of scASK is to address the challenges of high dimensionality and high variability in single-cell data. Although, there is no universally best classification algorithm for all datasets, we emphasize that scASK has the potential to be extended to the most matching classification algorithm for classifying cell types based on scRNA-seq data, and even for classifying cancer types based on gene expression data or DNA methylation data. In this section, we discuss some key details and further functions of scASK.

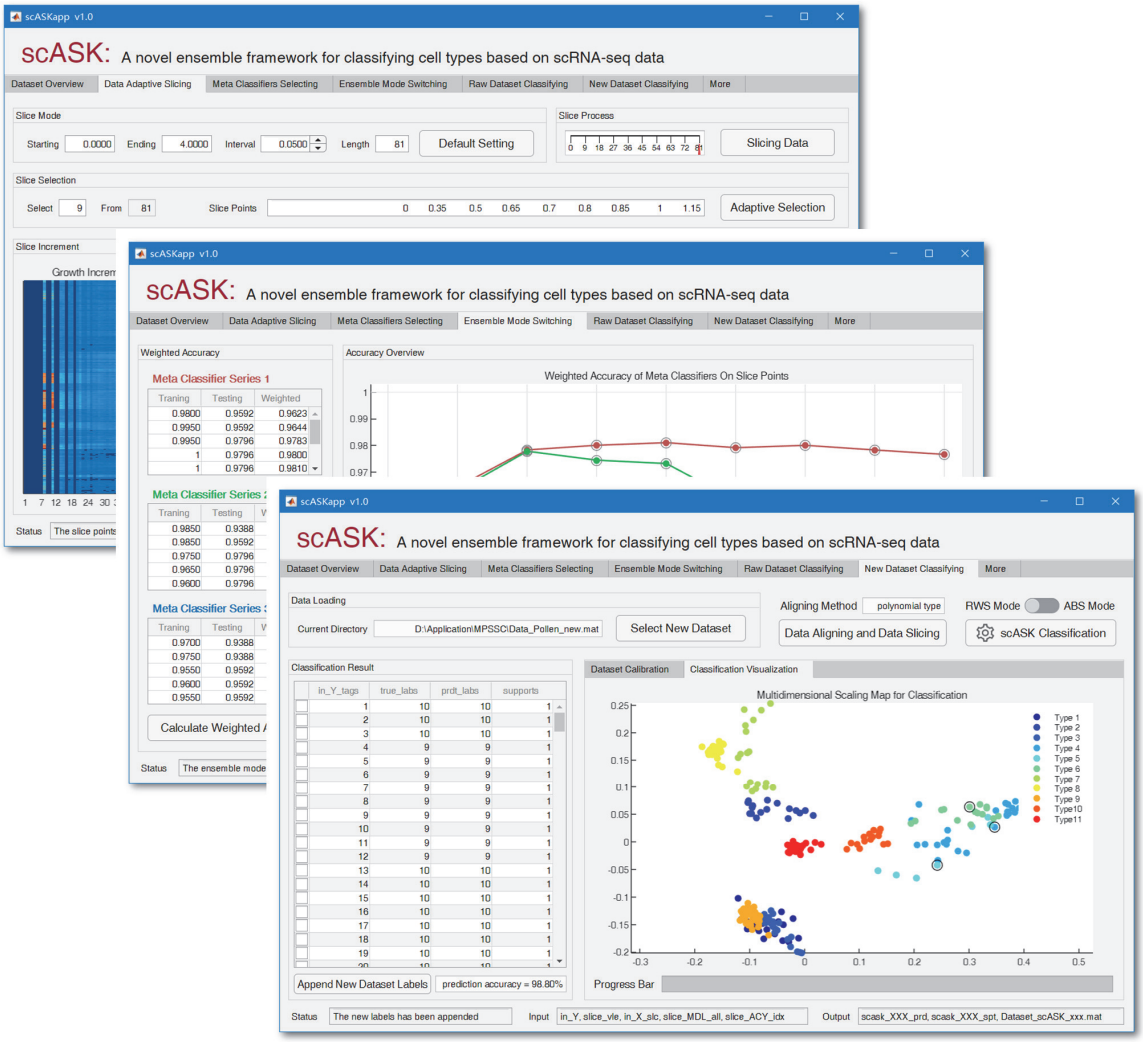


Figure 8. Screenshots of scASKapp. Shown here are the Data Adaptive Slicing tab, Ensemble Mode Switching tab and New Dataset Classifying tab.

First of all, the slicing procedure used by scASK is not intended to replace PCA (or t-SNE), but rather provides a choice that is parallel to existing mainstream dimension reduction methods for high dimensional single-cell data. On the one hand, the exact SVD of very large matrix is still a difficult computational problem, essentially because matrix decomposition is a series of related operations which are hard to parallelize. Since the slicing procedure is a linearization process, the classifier training based on each slice matrix is completely independent, so it can be performed without interference at the same time. This feature makes scASK more easily for parallelization. On the other hand, the matrix decomposition (such as PCA) of the existing data during the data preprocessing step may cause some troubles for classification on future data. It is unrealistic to require subsequent samples to participate in the matrix decomposition of initial samples in the training phase, because the matrix decomposition is equivalent to the feature reconstruction. The lack of the process could finally lead to the failure of classification of cell types across datasets. For example, the Pollen* dataset cannot be directly imported into the classifiers trained from the Pollen dataset with PCA for classification, although they are essentially the same data with different preprocessing. If we align the Pollen* dataset with the Pollen

dataset after appropriate scale-transformation, the classifiers trained from the Pollen dataset by scASK can directly identify cell types for Pollen* dataset. Actually, the data alignment across datasets is an important topic of our future work (for some basic implementation, see the tutorials of scASKapp).

Secondly, scASK does not exclude PCA, in fact, they can cooperate very well. Let's take the Macosk dataset as an example, the dimension of raw data is very large (6418×12822). In the process of implementing scASK, the slicing procedure takes up so much memory that exceeds the capacity of our laptop. Back to square one, once we reduce the dimension to 6148×100 by the aid of PCA, and append a non-negative processing of matrix elements, then scASK can also achieve a good enough classification accuracy of 90.34% on test set (for running codes and output results, see Supplementary Materials Table S11 and Figure S111 to S121). Next, it is worth mentioning that Pearson's correlation coefficient, Jaccard similarity and Cosine similarity as the default distance measures of scASK is recommended but not necessary. In fact, we find that the accuracy of scASK using only the Pearson's correlation coefficient is obviously higher in some datasets (such as Usoskin data and Tasic data, see Supplementary Materials Table S5 and Figure S45 to S55, Table S14 and Figure S144 to S154). It is indicated that Pearson's correlation coefficient as distance measure always performs well in scASK for classifying cell types based on scRNA-seq data.

Finally, the significant advantages of scASK in implementing cell type classification for single-cell data is its high accuracy and high robustness. The adaptive slicing and switching strategy all play crucial role in scASK, the former can be regarded as a new structural reduction procedure which is distinct from the popular procedure for high-dimensional data (e.g., PCA, t-SNE and NMF), and the latter can be regarded as a new ensemble strategy which is distinct from the traditional strategies for classification (i.e., bagging, boosting and stacking). scASK runs on slice matrices with the switching strategy, which not only helps confirm the results, but also enhances the reliability of the classification. Both of technologies above can be extended to a wider range beyond this work, including medical diagnosis, data analysis, machine learning and so on.

ACKNOWLEDGEMENTS

We thank xxx and xxx for helpful discussions of this manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (Nos. 11831015 and 61672388), the National Key Research and Development Program of China (No. 2018YFC1314600) and the Natural Science Foundation of Hubei Province No. 2019CFA007.

Conflict of Interest: none declared.

REFERENCES

1. Regev,A., Teichmann,S.A., Lander,E.S., Amit,I., Benoist,C., Birney,E., Bodenmiller,B., Campbell,P., Carninci,P., Clatworthy,M., *et al.* (2017) The Human Cell Atlas. *eLife*, **6**, e27041.

2. Haque,A., Engel,J., Teichmann,S.A. and Lönnberg,T. (2017) A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine*, **9**.

3. Hon,C.-C., Shin,J.W., Carninci,P. and Stubbington,M.J.T. (2018) The Human Cell Atlas: Technical approaches and challenges. *Briefings in Functional Genomics*, **17**, 283–294.

4. Angerer,P., Simon,L., Tritschler,S., Wolf,F.A., Fischer,D. and Theis,F.J. (2017) Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, **4**, 85–91.

5. Svensson,V., Vento-Tormo,R. and Teichmann,S.A. (2018) Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, **13**, 599–604.

6. Poirion,O.B., Zhu,X., Ching,T. and Garmire,L. (2016) Single-Cell Transcriptomics Bioinformatics and Computational Challenges. *Frontiers in Genetics*, **7**.

7. Rostom,R., Svensson,V., Teichmann,S.A. and Kar,G. (2017) Computational approaches for interpreting scRNA-seq data. *FEBS Letters*, **591**, 2213–2225.

8. Zappia,L., Phipson,B. and Oshlack,A. (2018) Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLOS Computational Biology*, **14**, e1006245.

9. Stegle,O., Teichmann,S.A. and Marioni,J.C. (2015) Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, **16**, 133–145.

10. Camara,P.G. (2018) Methods and challenges in the analysis of single-cell RNA-sequencing data. *Current Opinion in Systems Biology*, **7**, 47–53.

11. Grün,D., Kester,L. and van Oudenaarden,A. (2014) Validation of noise models for single-cell transcriptomics. *Nature Methods*, **11**, 637–640.

12. Mukherjee,S., Zhang,Y., Fan,J., Seelig,G. and Kannan,S. (2018) Scalable preprocessing for sparse scRNA-seq data exploiting prior knowledge. *Bioinformatics*, **34**, i124–i132.

13. David Bau,I. and Trefethen,L.N. (1997) Numerical Linear Algebra Springer, New York.

14. Vishwas,B.C., Gadia,A. and Chaudhuri,M. (2009) Implementing a parallel matrix factorization library on the cell broadband engine. *ieee international conference on high performance computing data and analytics*, **17**, 3–29.

15. Der Maaten,L.V. and Hinton,G.E. (2008) Visualizing Data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

16. Wattenberg,M., Viégas,F. and Johnson,I. (2016) How to Use t-SNE Effectively. *Distill*, 10.23915/distill.00002.

17. Carmona-Saez,P., Pascual-Marqui,R.D., Tirado,F., Carazo,J.M. and Pascual-Montano,A. (2006) Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*.

18. Devarajan,K. (2008) Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. *PLoS Computational Biology*, **4**, e1000029.

19. Nik-Zainal,S., Alexandrov,L.B., Wedge,D.C., Van Loo,P., Greenman,C.D., Raine,K., Jones,D., Hinton,J., Marshall,J., Stebbings,L.A., *et al.* (2012) Mutational Processes Molding the Genomes of 21 Breast Cancers. *Cell*, **149**, 979–993.

20. Lin,C.-J. (2007) Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, **19**, 2756–2779.

21. Taslaman,L. and Nilsson,B. (2012) A Framework for Regularized Non-Negative Matrix Factorization, with Application to the Analysis of Gene Expression Data. *PLoS ONE*, **7**, e46331.

22. Wang,Y.-X. and Zhang,Y.-J. (2013) Nonnegative Matrix Factorization: A Comprehensive Review. *IEEE Transactions on Knowledge and Data Engineering*, **25**, 1336–1353.

23. Xu,C. and Su,Z. (2015) Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, **31**, 1974–1980.

24. Grün,D., Lyubimova,A., Kester,L., Wiebrands,K., Basak,O., Sasaki,N., Clevers,H. and van Oudenaarden,A. (2015) Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, **525**, 251–255.

25. žurauskienė,J. and Yau,C. (2016) pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, **17**.

26. Kiselev,V.Y., Kirschner,K., Schaub,M.T., Andrews,T., Yiu,A., Chandra,T., Natarajan,K.N., Reik,W., Barahona,M., Green,A.R., *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods*, **14**, 483–486.

27. Wang,B., Zhu,J., Pierson,E., Ramazzotti,D. and Batzoglou,S. (2017) Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature Methods*, **14**, 414–416.

28. Hastie,T., Tibshirani,R. and Friedman,J. (2009) Unsupervised learning. In *The elements of statistical learning*. Springer, pp. 485–585.

29. Xu,R. and WunschII,D. (2005) Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, **16**, 645–678.

30. Zeisel,A., Munozmanchado,A.B., Codeluppi,S., Lonnerberg,P., La Manno,G., Jureus,A., Marques,S., Munguba,H., He,L., Betsholtz,C., *et al.* (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.

31. Kharchenko,P.V., Silberstein,L. and Scadden,D.T. (2014) Bayesian approach to single-cell differential expression analysis. *Nature Methods*, **11**, 740–742.

32. Ilicic,T., Kim,J.K., Kolodziejczyk,A.A., Bagger,F.O., McCarthy,D.J., Marioni,J.C. and Teichmann,S.A. (2016) Classification of low quality cells from single-cell RNA-seq data. *Genome Biology*, **17**.

33. Pouyan,M.B. and Nourani,M. (2017) Clustering Single-Cell Expression Data Using Random Forest Graphs. *IEEE Journal of Biomedical and Health Informatics*, **21**, 1172–1181.

34. Kiselev,V.Y., Yiu,A. and Hemberg,M. (2018) scmap: projection of single-cell RNA-seq data across data sets. *Nature Methods*, **15**, 359–362.

35. Crow,M., Paul,A., Ballouz,S., Huang,Z.J. and Gillis,J. (2018) Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor. *Nature Communications*, **9**.

36. Park,S. and Zhao,H. (2018) Spectral clustering based on learning similarity matrix. *Bioinformatics*, **34**, 2069–2076.

37. Zheng,G.X.Y., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J., *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, **8**, 14049.

38. Kuramochi,M. and Karypis,G. (2001) Gene Classification using Expression Profiles: A Feasibility Study.

39. Gutierrez,D.D. (2015) Machine learning and data science: an introduction to statistical learning methods with R Technics Publications, Basking Ridge.

40. Zhou,Z.H. (2012) Ensemble Methods: Foundations and Algorithms Taylor & Francis, New York.

41. Buettner,F., Natarajan,K.N., Casale,F.P., Proserpio,V., Scialdone,A., Theis,F.J., Teichmann,S.A., Marioni,J.C. and Stegle,O. (2015) Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, **33**, 155–160.

42. Pollen,A.A., Nowakowski,T.J., Shuga,J., Wang,X., Leyrat,A.A., Lui,J.H., Li,N., Szpankowski,L., Fowler,B., Chen,P., *et al.* (2014) Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, **32**, 1053–1058.

43. Barron,M., Zhang,S. and Li,J. (2018) A sparse differential clustering algorithm for tracing cell type changes via single-cell RNA-sequencing data. *Nucleic Acids Research*, **46**, e14–e14.

44. Li,W.V. and Li,J.J. (2018) An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature Communications*, **9**.