SSCC: a computational framework for rapid and accurate clustering of large-scale single cell RNA-seq data

Xianwen Ren[1*], Liangtao Zheng[1] and Zemin Zhang[1*]
1, Biodynamic Optical Imaging Center, Beijing Advanced Innovation Center for Genomics, and School of Life Sciences, Peking University, Beijing 100871, China

*, to whom correspondence should be addressed

Emails:
XR: renxwise@pku.edu.cn
LZ: zhengliangtao2015@pku.edu.cn
ZZ: zemin@pku.edu.cn

Abstract

Clustering is a prevalent analytical means to analyze single cell RNA sequencing data but the rapidly expanding data volume can make this process computational challenging. Although *ad hoc* frameworks based on subsampling were proposed to handle large-scale datasets, they have not been well benchmarked or optimized. Here, we conducted a comparative study and found that subsampling compromised almost all the evaluated clustering algorithms within the existing computational frameworks. To surmount the difficulties, we introduced feature-engineering and projecting techniques in the machine learning field and proposed a new clustering framework, which greatly improves clustering accuracy, robustness and computational efficacy for various state-of-the-art algorithms benchmarked on multiple real datasets. On a dataset with 68,578 human blood cells, our method reached 20% improvements for clustering accuracy and 50-fold acceleration but only consumed 66% memory usage compared to the widely-used software package SC3. Compared to k-means, the accuracy improvement can reach 3-fold depending on the concrete dataset. An R implementation of the framework is available from https://github.com/Japrin/sscClust.

Keywords: single-cell RNA-seq, clustering, random projection, Spearman correlation, classification

Background

Single cell RNA sequencing (scRNA-seq) has revolutionized transcriptomic studies by revealing the heterogeneity of individual cells with high resolution [1-6]. Clustering has become a routine analytical means to identify cell types, depict their functional states and infer potential cellular dynamics [4-10]. Multiple clustering algorithms have been developed, including Seurat [11], SC3 [12], SIMLR [13], ZIFA [14] and CIDR [15]. These algorithms improve the clustering accuracy of scRNA-seq data greatly but often have high computational complexity, impeding the extension of these elegant algorithms to large-scale scRNA-seq datasets. With the rapid development of scRNA-seq technologies, the throughput has increased from initially hundreds of cells to tens of thousands of cells in one run [16]. Integrative analyses of scRNA-seq datasets from multiple runs or even across multiple studies further exacerbate the computational difficulties. Thus, algorithms that can cluster single cells both efficiently and accurately are needed.

To handle multiple large-scale scRNA-seq datasets, *ad hoc* computational strategies have been proposed by subsampling the large datasets to small ones [12, 17, 18]. Such strategies have variable levels of success but have not been well benchmarked. It is unknown whether such strategies can maintain high clustering accuracy in spite of the reduced computational cost. Further, there is a lack of methods for assigning large number of cells into clusters simultaneously with high efficiency and accuracy. To address this important problem, we benchmarked such computational strategies on multiple scRNA-seq datasets. We noticed that while the clustering accuracy on the subsampled cells is comparable to that upon the total cells, the assignment of unsampled cells to cellular clusters obtained based on subsampled cells is error-prone, thus compromising the total clustering accuracy and leading to biased results.

Considering the importance of efficient and accurate clustering tools for analyses of large-scale scRNA-seq data and the drawbacks of existing computational strategies, here we propose a new computational framework based on machine learning techniques including feature engineering and projecting to achieve improved clustering accuracy and efficacy. Benchmarking suggested that our new computational framework can reduce the computational complexity from $O(n^2)$ to $O(n)$ while maintaining the clustering accuracy based on as low as 10% cells. The clustering consistency between different subsampling runs is also significantly improved, compared to existing strategies. Flexibility of the new computational framework allows our methods to be further extended and adapted to a wide range of applications for scRNA-seq data analysis.

Results and Discussion

Overview of the new computational framework for clustering large-scale scRNA-seq data

The existing computational strategies for clustering large scRNA-seq data generally consist of four steps (Figure 1a). First, a gene expression matrix is constructed by data preprocessing techniques including gene and cell filtration and normalization. Second, cells are divided into two subsets for clustering and classification separately by subsampling. Third, the subseted cells for clustering are grouped into clusters by *k*-means [19], hierarchical clustering [20], density clustering [21] or algorithms developed specially for scRNA-seq. Fourth, supervised algorithms such as *k*-nearest neighbors [22], support vector machines [23] or random forests [24] are used to predict the labels of other cells based on the clustering results in the third step. For simplicity, we referred these existing strategies as SCC (**S**ubsampling-**C**lustering-**C**lassification).

Because clustering is time-consuming and memory-exhaustive, limiting this step to a small subset of cells through subsampling can greatly reduce the computational cost from $O(n^2)$ to $O(n)$ by leveraging the efficiency of supervised algorithms. However, classifiers built on the original gene expression data of a small subset of cells may be flawed and biased due to noise of the raw data and small number of cells, thus impairing the accuracy of label assignment for the total cells.

Here we proposed a new computational framework for clustering large scRNA-seq data by adding a feature engineering/projecting step (Figure 1b). Similar to SCC, a gene expression matrix is first constructed through gene and cell filtrations and normalization (Step 1 in Figure 1b), and is then split randomly into two subsets for clustering and classification separately (Step 2 in Figure 1b). Unlike SCC which directly use the rawdata of gene expression, our framework projects cells into a feature space (Step 3 in Figure 1b) for clustering (Step 4 in Figure 1b) and classification (Step 5 in Figure 1b). As the new framework is characterized by a Subsampling-Featuring-Clustering-Classification strategy, we named it as SFCC.

Specifically, we divide feature construction into two steps. First, feature extraction techniques are applied to cells subject to clustering. Then, according to the selection of feature extraction methods, cells for classification are projected into the built feature space. Many established techniques in the machine learning field can be exploited in these two steps. For example, principal component analysis (PCA) [25] can be used to first construct features for cells undergoing clustering while the resultant loading vectors can be used as linear transformations to project cells for classification into the feature space.

Selecting different algorithms in each step of the SFCC framework will thus form different pipelines for clustering large-scale scRNA-seq datasets. To reduce the total number of algorithmic combinations, here we focus on comparing the performance between various feature engineering algorithms. We hold algorithms for gene and cell filtration, normalization, subsampling and classification as the algorithms frequently used in practice (see Methods for details). For fair comparison, the existing SCC strategies are treated as a special case of SFCC in which the original data space is the feature space.

Feature engineering techniques involved in this study include distance-based methods (Euclidean and cosine), correlation-based methods (Pearson [26] and Spearman [27] correlations) and a neural network-based method (Autoencoder) [28]. For distance and correlation based methods, the distance/correlation matrix for cells subject to clustering was directly used as their features, and the distance/correlation matrix between cells subject to classification and clustering were used to construct features for cells undergoing classification. For Autoencoder, the gene expression data of cells for clustering were used to train a neural network model first and then all cells were projected into a feature space through the encoding function of the trained model. To obtain evaluation results independent of clustering algorithms, we used silhouette values [29] to examine the global performance of these feature engineering methods. Upon the global evaluation, we then select the most effective method to do further evaluations.

scRNA-seq datasets used to benchmark performance of different clustering frameworks

We selected five scRNA-seq datasets that have cell clusters determined by independent experimental methods to evaluate the effectiveness of the clustering frameworks [5, 8-10, 30]. Most of these scRNA-seq datasets have been widely used to benchmark clustering methods. As these datasets covered wide ranges of cell numbers, cluster numbers, sequencing depth, sequencing technologies, and biological objects, they provide adequate biological and technological variations to represent practical scenarios, thus allowing reliable assessment of the performance of different scRNA-seq clustering frameworks. Among these five datasets, the total cell numbers range from 249 to 3,005, the cell cluster numbers range from 3 to 11, and the sequencing technologies include Smart-seq2 [31] and Drop-seq [17]. The gene expression quantification methods include TPM (transcripts per million) and UMI (unique molecular identifier), and the sequencing depth increases from less than one million of reads to ten millions. The species cover human and mouse, and the cell types cover embryonic stem cells, skin cells, pluripotent stem cells, blood cells, neural cells, etc.

Subsampling compromised the overall clustering accuracy of SCC frameworks

First, we examined the impacts of subsampling on the clustering accuracy of SCC frameworks. On each benchmark dataset, we randomly selected a subset of cells for clustering, with subsampling rates ranging from 10% to 50%, and then classified the unselected cells to clusters obtained in the previous step by the classification algorithm *k*-nearest neighbors (*k*=1). We compared the predicted cluster labels of all the cells with reference through calculating normalized mutual information (NMI, ranging from 0 to 1) [32]. NMI is a frequently used consistency measure of clustering results, with 0 indicating discordance and 1 complete concordance. Differences of the NMI metrics at each subsampling rate and the total cells ($\Delta NMI = NMI_r - NMI_{100\%}$ where r=10%, 20%, 30%, 40% and 50%) were calculated to indicate the accuracy changes. We noticed a significant reduction of the clustering accuracy on all datasets, with none of the clustering algorithms immune to the impacts of subsampling (Figure 2).

We applied four clustering algorithms on each benchmark dataset to eliminate biases introduced by algorithmic selection. These algorithms included *k*-means [19], *k*-medoids [33], SC3 [12] and SIMLR [13]. *K*-means and *k*-medoids are clustering algorithms designed for general datasets while SC3 and SIMLR are clustering algorithms specially designed for scRNA-seq datasets. We observed that almost all ΔNMI values were negative, i.e., the clustering accuracy with subsampling is reduced compared to those based on the total cells, regardless of the selection of clustering algorithms. The few positive ΔNMI values were attributed to poor clustering accuracy with total cells, most of which were related to the *k*-medoids algorithm. To absolutely eliminate the influence of the selection of clustering algorithms, we added an oracle-clustering algorithm. That is, we did not conduct clustering on the selected cells but rather directly used the reference cell labels for classification. Consistently, the clustering accuracy of the total cells was significantly compromised by subsampling, confirming that the accuracy reduction was irrelevant of the selection of clustering algorithms.

We further examined whether classification algorithms affect the total clustering accuracy. By alternating the parameter *k* of the *k*-nearest neighbor algorithm from 1 to 3, 5 and 10, we observed the same phenomena on all the datasets and with all the clustering algorithms. Further, we switched the classification algorithm to random forest, which is quite robust to its parameter selection and fits many linearly or nonlinearly separated scenarios. Similar to cases using the *k*-nearest neighbor algorithm, subsampling impaired the total clustering accuracy significantly on almost all the benchmark datasets and for all clustering algorithms. Changing the classification algorithm to support vector machines revealed the same trend of accuracy reduction.

Feature construction can greatly improve the consistency of cell features and the reference cell labels

We further evaluated how SFCC affects clustering of scRNA-seq data. First, we calculated the silhouette values between cell features extracted by various methods and the reference labels. Silhouette values are frequently used to indicate whether a sample is properly clustered. But here we can use silhouette values to reversely indicate whether the extracted features are properly consistent with the reference cell labels. By comparing with silhouette values of the original scRNA-seq data, we observed that most of the evaluated feature-extracting methods can improve the silhouette values for many cells on multiple datasets (Figure 3). On the Kolodziejczyk [30] and Pollen [8] datasets, all the five feature-extraction methods improved the silhouette values compared with the original data. On the Usoskin [9] and Zeisel [10] datasets, all methods showed good performance except Euclidean and cosine. On the Zheng [5] dataset, most methods failed except the Spearman correlation method. The Spearman correlation-based method consistently improved the accordance between cell features and labels on all the five datasets. Considering the robustness of Spearman correlation-based method and the great improvement of silhouette values of single cells, we evaluated this **S**pearman correlation-based **S**ingle-**C**ell **C**lustering framework (*abbr.* SSCC) in the next sections.

Clustering accuracy of the total cells is enhanced in feature space when subsampling is applied

We observed that the improvements of silhouette scores by SSCC were robust to subsampling fluctuations (Figure 4). On all five evaluated datasets, the silhouette values of Spearman correlation-based features were almost unchanged with subsampling rates (Figure 4), implying that features constructed upon SSCC with low subsampling rates may contain information approximate to those upon total cells.

We further evaluated whether the improved silhouette values can be translated into clustering accuracy. By evaluating five clustering algorithms including *k*-means, *k*-medoids, affinity propagation, SC3 and SIMLR, we observed that SSCC can significantly improve the clustering accuracy for all the five clustering algorithms on all the benchmark datasets (Figure 5). The accuracy scores measured by NMI can be improved from 0.12 to 0.60 on the Kolodziejczyk dataset, 0.04 to 0.19 on the Pollen dataset, 0.14 to 0.37 on the Usoskin dataset, 0.02 to 0.28 on the Zeisel dataset, and 0.10 to 0.28 on the Zheng dataset, which depended on the specific selection of algorithms and subsampling rates. The improvements can even reach 1-3 folds. Other accuracy metrics including rand index, adjusted rand index and adjusted mutual information revealed the same trends, suggesting that SSCC can greatly enhance the clustering power of multiple algorithms when subsampling is used.

Clustering consistency between different subsampling runs is also greatly improved with SSCC

In practice, the reference cell labels are generally unknown. The confidence of clustering results is often evaluated with the consistency between different algorithms. Because of subsampling fluctuations, clustering results based on SCC are inconsistent among different subsamplings. However, in the new framework of SSCC, the consistency was much improved for all evaluated clustering algorithms on all datasets (Figure 6). On the Kolodziejczyk dataset, all the five clustering algorithms reached consistency much larger than 0.5 in SSCC while the corresponding consistency in SCC was smaller than 0.5. On the Pollen dataset, both SSCC and the original framework had high clustering consistency, but consistency in SSCC was still larger than that in SCC. Consistency on the Usoskin, Zeisel and Zheng datasets were similar to that on the Pollen dataset.

Application of SSCC to large scRNA-seq datasets with or without reference cell labels

Besides the former five scRNA-seq datasets, we further tested SSCC on two additional large scRNA-seq datasets. One is the PBMC 68k dataset [16], which contains 10X Genomics-based expression data for 68,578 blood cells from a healthy donor. The other is the Macoskco dataset [17] containing 49,300 mouse retina cells but without cell labels determined by experimental methods. The large cell numbers generally prohibit classic scRNA-seq clustering algorithms running on a desktop computer, and thus provide two realistic examples to demonstrate the performance of SCC and SSCC.

On the PBMC 68k dataset, we compared SSCC with SCC by using SC3 [12] as the clustering algorithm. The SC3 software package inherently applied an SCC strategy to handle large scRNA-seq datasets. By default, if a dataset has more than 5,000 cells, the SCC strategy will be triggered, with 5,000 cells randomly subsampled for SC3 clustering and the other cells for classification by SVM. We applied SC3 to the PBMC 68k dataset on a desktop computer with 8GB memory and 3GHz 4-core CPU and repeated ten times. The average clustering accuracy of SC3 was 0.48 measured by NMI, the calculation lasted 99 minutes on average, and the maximum memory usage exceeded 5.6 GB (Figure 7a). With our SSCC strategy, the average clustering accuracy reached 0.59, a ~21% increase over SC3 (default). The computation time was dramatically reduced to 2.2 minutes on average, while occupying only 2.2% of the computation time of SC3 (default). The maximum memory usage of SC3+SSCC was 3.7GB, saving >33% compared to that of SC3 (default).

On the Macoskco dataset, with *k*-means for clustering and *k*-nearest neighbors for classification, the SCC strategy resulted great average silhouette difference (-0.80 -

-0.51=0.29) between two subsampling schemes (upon 5% and 10% cells) while the difference with SSCC became negligible. The NMI value between the two clustering results with SCC was 0.60 while that with SSCC was 0.69. Pearson correlations of silhouette values between the two clustering schemes were increased from 0.47 to 0.58 while switching from SCC to SSCC (Figure 7b). All these metrics suggest that SSCC can not only greatly improve the clustering efficiency and accuracy for large-scale scRNA-seq datasets, but also can greatly improve the consistency.

Conclusions

The availability of large-scale scRNA-seq data raise urgent need for efficient and accurate clustering algorithms. Clustering upon subsampled cells and then classification of unsampled cells has been proposed to address this important issue. However, this computational framework suffers from biases introduced by subsampling, which greatly reduce the clustering accuracy and robustness of the total cells. Here we employ feature engineering and projecting techniques and propose a new computational framework. Specially, with Spearman correlations as the feature engineering and projecting methods, we formulated a framework named as SSCC which significantly improves clustering accuracy and consistency for many general and specially designed clustering algorithms. The superior performance of SSCC and its flexibility allows further unleashing the power of scRNA-seq technologies.

Material and methods

scRNA-seq datasets used in this study

We used seven scRNA-seq datasets to evaluate the effects of clustering in feature space. Detailed descriptions of all the seven datasets are listed below.

1) The Kolodziejczyk dataset [30]. This dataset contains 704 cells with three clusters that were obtained from mouse embryonic stem cells with di erent culture conditions. About 10,000 genes were profiled with high sequencing depth (average 9 million reads per cell) by using the Fluidigm C1 system and applying the SMARTer Kit to obtain cDNA and the Nextera XT Kit for Illumina library preparation.
2) The Pollen dataset [8]. This dataset contains 249 cells with eleven clusters, which contain skin cells, pluripotent stem cells, blood cells, neural cells, etc. Either low or high sequencing depth based on the C1 Single-Cell Auto Prep Integrated Fluidic Circuit, the SMARTer Ultra Low RNA Kit and the Nextera XT DNA Sample Preparation Kit was used to depict the gene expression profiles of individual cells.
3) The Usoskin dataset [9]. This dataset contains 622 mouse neuronal cells with four

clusters i.e., peptidergic nociceptors, nonpeptidergic nociceptors, neuroflament containing, and tyrosine hydroxylase containing. The neuronal cells were picked with a robotic cell-picking setup and positioned in wells of 96-well plates and then were subject to RNA-seq with 1.14 million reads per cell.

4) The Zeisel dataset [10]. This dataset contains 3,005 cells from the mouse brain with nine major subtypes. The gene expression levels were estimated by counting the number of unique molecular identifiers (UMIs) obtained by Drop-seq.
5) The Zheng dataset [5]. This dataset contains 5,063 T cells from five patients with hepatocellular carcinoma of which nine subtypes of samples were prepared according to the tissue types and cell types and then were subject to Smart-seq2 for profiling the gene expressions.
6) The PBMC 68k dataset [16]. Gene expression of 68,578 peripheral blood mononuclear cells (PBMC) of a healthy human were profiled by the 10X Genomics GemCode platform, with 3' UMI counts used to quantify gene expression levels by their customized computational pipeline. This cell population includes eleven major immune cell types.
7) The Macoskco dataset [17]. This dataset contains 49,300 mouse retina cells without known distinct clusters. The gene expression levels were estimated by counting the number of UMIs obtained by Drop-seq, which were further clustered into 39 cell subtypes by the authors based on the Seurat algorithm.

Data preprocessing

The first four datasets (i.e., the Kolodziejczyk, Pollen, Usoskin and Zeisel datasets) has been widely used for evaluating clustering algorithms, of which the preprocessed data have been included in the SIMLR software package for testing use. We downloaded these four datasets from the Matlab subdirectory of the SIMLR package, and then selected the top 5000 most informative genes (with both the average and the standard deviation of log2-transformed expressing values more than 1) for subsequent analysis. If the gene number in the data was smaller than 5000, then all the genes were retained for further analysis. For the Zheng dataset, one patient (P0508) was selected for comparison of different clustering algorithms, which had 1,020 T cells with eight subtypes defined by the tissue sources and the cell surface markers. The same criteria were used to filter genes and then the transcripts per million (TPM) values were used to do clustering evaluations. For the PBMC 68k dataset, the preprocessing pipeline of the original paper [16] was used to prepare data for clustering (https://github.com/10XGenomics/single-cell-3prime-paper). For the Macoskco dataset, the UMI counts were used for evaluations without gene filter.

Feature engineering methods evaluated in this study

Totally twelve different methods were used to project the original scRNA-seq data into feature space. These twelve methods include a neural network based denoising Autoencoder and eleven distance-based metrics. An autoencoder tries to map the original

data into a space with low-dimensionality through a layer of neurons[28], in which the original data can be reversely decoded from the low-dimensionality representation as precisely as possible. The representation of each sample in the learnt low-dimensionality space was what we called features after projection. We used the train autoencoder function implemented in Matlab R2016a with the default parameters for evaluations in this study.

The eleven distance-based metrics include Euclidean, squared Euclidean, standardized Euclidean, city block, Minkowski, Chebychev, 1-cosine, 1-Pearson's correlation, 1-Spearman's correlation, Hamming, and Jaccard distances, which are implemented in the pdist2 function of Matlab R2016a. Except Pearson's and Spearman's correlations, all other nine distance metrics were calculated by pdist2 while the Pearson's and Spearman's correlations were calculated by the corr function of Matlab R2016a. The distance or correlation matrixes were used as features constructed by the corresponding metrics in the subsequent analyses. To simplify the illustrations, the silhouette values of squared Euclidean, standardized Euclidean, city block, Minkowski, Chebychev, Haming and Jaccard, which always performed poorer than Spearman correlation, were not shown in Supplementary Figure 1.

Consistency between true labels and the original and projected data

The silhouette value [29] is used to measure the consistency between the true labels and the original and projected data. Given a dataset with n samples and a clustering scheme, a silhouette value is calculated for each sample. For a sample i, its silhouette value s_i is calculated according to the following formula:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

Where a_i is the average dissimilarity of i to samples in its own cluster and b_i is the lowest average dissimilarity of i to any other cluster of which i is not a member. s_i is between -1 and 1. Closeness to 1 means it is well matched to its cluster while closeness to -1 means that i would be more appropriate if it was clustered in its neighboring cluster. For each feature construction method, the median silhouette value of the whole cells after projection was used to evaluate its consistency with the true cluster labels. The fraction of cells that have silhouette values increased after projection compared to the original data (i.e., the fraction of cells above the diagonal in Supplementary Figure 1) was also used to evaluate the feature construction method.

Clustering accuracy/consistency evaluation

Normalized mutual information (NMI)[32] was used to evaluate the accuracy of various

clustering results. Given two clustering schemes $\mathbf{A}=\{A_1,\cdots,A_R\}$ and $\mathbf{B}=\{B_1,\cdots,B_S\}$, the overlap between $\mathbf{A}$ and $\mathbf{B}$ can be represented through the contingency table $\mathbf{C}$ (also named as confusion matrix) of size $R\times S$, where C_{ij} denotes the number of cells that clusters A_i and B_j share. The normalized mutual information $NMI_{(\mathbf{A},\mathbf{B})}$ of the two clustering schemes $\mathbf{A}$ and $\mathbf{B}$ is defined as follows.

$$NMI_{(\mathbf{A},\mathbf{B})}=\frac{-2\sum_{i=1}^{R}\sum_{j=1}^{S}C_{ij}\log(\frac{C_{ij}\times n}{C_{i-}\times C_{-j}})}{\sum_{i=1}^{R}C_{i-}\log(\frac{C_{i-}}{n})+\sum_{j=1}^{S}C_{-j}\log(\frac{C_{-j}}{n})}$$

Where *n* is the number of total cells, C_{i-} is the number of cells assigned to cluster i in the clustering scheme $\mathbf{A}$ and C_{-j} is the number of cells assigned to cluster j in the clustering scheme $\mathbf{B}$. If $\mathbf{A}$ is identical to $\mathbf{B}$, $NMI_{(\mathbf{A},\mathbf{B})}=1$. If $\mathbf{A}$ and $\mathbf{B}$ are completely different, $NMI_{(\mathbf{A},\mathbf{B})}=0$. When true cluster labels were available, the NMI values between true cluster labels and various clustering results were used to evaluate the clustering accuracy. When true cluster labels were not available, NMI was used to evaluate clustering consistency between different subsampling rates in this study. Besides NMI, we also used rand index and adjusted rand index to evaluate clustering accuracy and consistency and obtained similar observations.

Clustering algorithms

Many clustering algorithms are available. We selected five widely-used clustering algorithms in this study to evaluate the impacts of Spearman correlation-based feature construction method. These five algorithms include three general clustering algorithms which were designed initially not for scRNA-seq data (affinity propagation (AP)[34], *k*-means[19] and *k*-medoids[33]) and two algorithms that were specially designed for clustering of scRNA-seq data (SC3[12] and SIMLR[13]). *K*-means and *k*-medoids are pure algorithms that partition samples into groups while AP, SC3 and SIMLR inherently include feature construction techniques. All these clustering algorithms were evaluated on the five small-scale datasets while on the PBMC 68k dataset only SC3 was evaluated and on the Macoskco dataset only k-means was evaluated for simplicity. Parameters (ks = 10:12, gene_filter = FALSE, biology = FALSE, svm_max = 5000) were used for SC3 (default) while (ks = 11, gene_filter = FALSE, biology = FALSE, svm_max = 200) were used for SC3+SSCC. On the Macoskco dataset, ~5% and 10% cells were randomly picked out to do clustering. Evaluations suggest that the Spearman correlation-based feature

construction method can significantly improve the clustering performance for both general clustering algorithms and specially designed algorithms and for both pure clustering algorithms and algorithms that have already applied feature construction techniques.

Clustering with subsampling and classification

The cell numbers of large-scale scRNA-seq datasets could reach tens of thousands or even millions, raising great challenging to the computational efficiency of clustering algorithms. To circumvent this challenge, a clustering framework with subsampling and classification has been proposed[12, 17]. In this framework, a small fraction of cells is firstly subsampled randomly from the whole cell set and undergoes traditional clustering, and then the cluster labels serve as sample labels to train a classifier (e.g., *k*-nearest neighbors[22], support vector machines[23], or random forests[24]) with supervision. The cluster labels for the un-subsampled cells are finally obtained through prediction based on the trained classifier, which can be done cell by cell and thus significantly reduces the computational burden. Although this framework has been used to effectively cluster single cells, the performance has not been well evaluated and great improvement may be possible as evidenced in this study. Here we evaluated this clustering framework on the original data and in the feature space by using the *k*-nearest neighbors classifier, which is robust to parameter settings. In details, given a subsampling rate, e.g., 10%, the corresponding fraction of cells were sampled from the whole cell set and the five clustering algorithms referred above were applied to the subsampled cells upon the original data and the features constructed by Spearman correlations. Cluster labels of the un-sampled cells were then predicted by the knnclassify function of Matlab R2016a with default parameters (k=1 with the Euclidean distance and the “nearest” rule). Cluster labels of the whole cell set were then compared to the true cluster labels by calculating NMIs.

Abbreviations

NMI: Normalized Mutual Information; SCC: Subsampling-Clustering-Classification; SFCC: Subsampling-Feature engineering-Clustering-Classification; SSCC: Spearman correlation-based Single Cell Clustering; PCA: Principal Component Analysis.

Acknowledgments

This project was supported by grants from Beijing Advanced Innovation Center for Genomics at Peking University, Key Technologies R&D Program (2016YFC0900100) and the National Natural Science Foundation of China

(81573022 and 31530036).

Author contributions

X.R. and Z.Z. designed the study. X.R. and L.Z. collected the data, implemented the software and did the analysis. X.R. and Z.Z. wrote the manuscript.

Competing financial interests

The authors declare no competing financial interests.

Data and code availability

The five scRNA-seq datasets involved in this study and the Matlab codes are available from https://sites.google.com/site/renxwise/home/sccs. The newly proposed clustering framework is implemented as an R package available from https://github.com/Japrin/sscClust.

References

1. Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, Wang X, Bodeau J, Tuch BB, Siddiqui A, et al: **mRNA-Seq whole-transcriptome analysis of a single cell.** *Nature Methods* 2009, **6:**377.
2. Tang F, Barbacioru C, Nordman E, Li B, Xu N, Bashkirov VI, Lao K, Surani MA: **RNA-Seq analysis to capture the transcriptome landscape of a single cell.** *Nature Protocols* 2010, **5:**516.
3. Tang F, Lao K, Surani MA: **Development and applications of single-cell transcriptome analysis.** *Nature Methods* 2011, **8**.
4. Tirosh I, Izar B, Prakadan SM, Wadsworth MH, Treacy D, Trombetta JJ, Rotem A, Rodman C, Lian C, Murphy G, et al: **Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq.** *Science* 2016, **352:**189-196.
5. Zheng C, Zheng L, Yoo JK, Guo H, Zhang Y, Guo X, Kang B, Hu R, Huang JY, Zhang Q, et al: **Landscape of Infiltrating T Cells in Liver Cancer Revealed by Single-Cell**

Sequencing. *Cell* 2017, **169:**1342-1356.e1316.

6. Cao J, Packer JS, Ramani V, Cusanovich DA, Huynh C, Daza R, Qiu X, Lee C, Furlan SN, Steemers FJ, et al: **Comprehensive single-cell transcriptional profiling of a multicellular organism.** *Science* 2017, **357:**661-667.
7. Trapnell C: **Defining cell types and states with single-cell genomics.** *Genome Research* 2015, **25:**1491-1498.
8. Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, Li N, Szpankowski L, Fowler B, Chen P, et al: **Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex.** *Nat Biotechnol* 2014, **32:**1053-1058.
9. Usoskin D, Furlan A, Islam S, Abdo H, Lonnerberg P, Lou D, Hjerling-Leffler J: **Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing.** *Nat Neurosci* 2015, **18:**145-153.
10. Zeisel A, Munoz-Manchado AB, Codeluppi S, Lonnerberg P, La Manno G, Jureus A, Marques S, Munguba H, He L, Betsholtz C, et al: **Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq.** *Science* 2015, **347:**1138-1142.
11. Satija R, Farrell JA, Gennert D, Schier AF, Regev A: **Spatial reconstruction of single-cell gene expression data.** *Nat Biotechnol* 2015, **33:**495-502.
12. Kiselev VY, Kirschner K, Schaub MT, Andrews T: **SC3: consensus clustering of single-cell RNA-seq data.** *Nat Methods* 2017, **14:**483-486.
13. Wang B, Zhu J, Pierson E, Ramazzotti D: **Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning.** *Nat Methods* 2017, **14:**414-416.
14. Pierson E, Yau C: **ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis.** *Genome Biology* 2015, **16:**241.
15. Lin P, Troup M, Ho JWK: **CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data.** *Genome Biology* 2017, **18:**59.
16. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J, et al: **Massively parallel digital transcriptional profiling of single cells.** *Nature Communications* 2017, **8:**14049.
17. Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, Tirosh I, Bialas AR, Kamitaki N, Martersteck EM, et al: **Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets.** *Cell* 2015, **161:**1202-1214.
18. Sinha D, Kumar A, Kumar H, Bandyopadhyay S, Sengupta D: **dropClust: efficient clustering of ultra-large scRNA-seq data.** *Nucleic Acids Research* 2018**:**gky007-gky007.
19. Lloyd S: **Least squares quantization in PCM.** *IEEE Transactions on Information Theory* 1982, **28:**129-137.
20. Sibson R: **SLINK: An optimally efficient algorithm for the single-link cluster method.** *The Computer Journal* 1973, **16:**30-34.
21. Rodriguez A, Laio A: **Clustering by fast search and find of density peaks.** *Science* 2014, **344:**1492-1496.
22. Coomans D, Massart DL: **Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules.** *Analytica Chimica Acta* 1982, **136:**15-27.

23. Cortes C, Vapnik V: **Support-vector networks.** *Machine Learning* 1995, **20:**273-297.
24. Tin Kam H: **The random subspace method for constructing decision forests.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1998, **20:**832-844.
25. Pearson K: **On lines and planes of closest fit to systems of points in space.** *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 1901, **2:**559-572.
26. Pearson K: **Note on regression and inheritance in the case of two parents.** *Proceedings of the Royal Society of London* 1895, **58:**240-242.
27. Spearman C: **The proof and measurement of association between two things. By C. Spearman, 1904.** *Am J Psychol* 1987, **100:**441-471.
28. Hinton GE, Salakhutdinov RR: **Reducing the Dimensionality of Data with Neural Networks.** *Science* 2006, **313:**504.
29. Rousseeuw PJ: **Silhouettes: A graphical aid to the interpretation and validation of cluster analysis.** *Journal of Computational and Applied Mathematics* 1987, **20:**53-65.
30. Kolodziejczyk AA, Kim JK, Tsang JC, Ilicic T, Henriksson J, Natarajan KN, Tuck AC, Gao X, Buhler M, Liu P, et al: **Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation.** *Cell Stem Cell* 2015, **17:**471-485.
31. Picelli S, Faridani OR, Björklund ÅK, Winberg G, Sagasser S, Sandberg R: **Full-length RNA-seq from single cells using Smart-seq2.** *Nature Protocols* 2014, **9:**171.
32. Strehl A, Ghosh J: **Cluster ensembles --- a knowledge reuse framework for combining multiple partitions.** *J Mach Learn Res* 2003, **3:**583-617.
33. Park H-S, Jun C-H: **A simple and fast algorithm for K-medoids clustering.** *Expert Systems with Applications* 2009, **36:**3336-3341.
34. Frey BJ, Dueck D: **Clustering by passing messages between data points.** *Science* 2007, **315:**972-976.

Figures

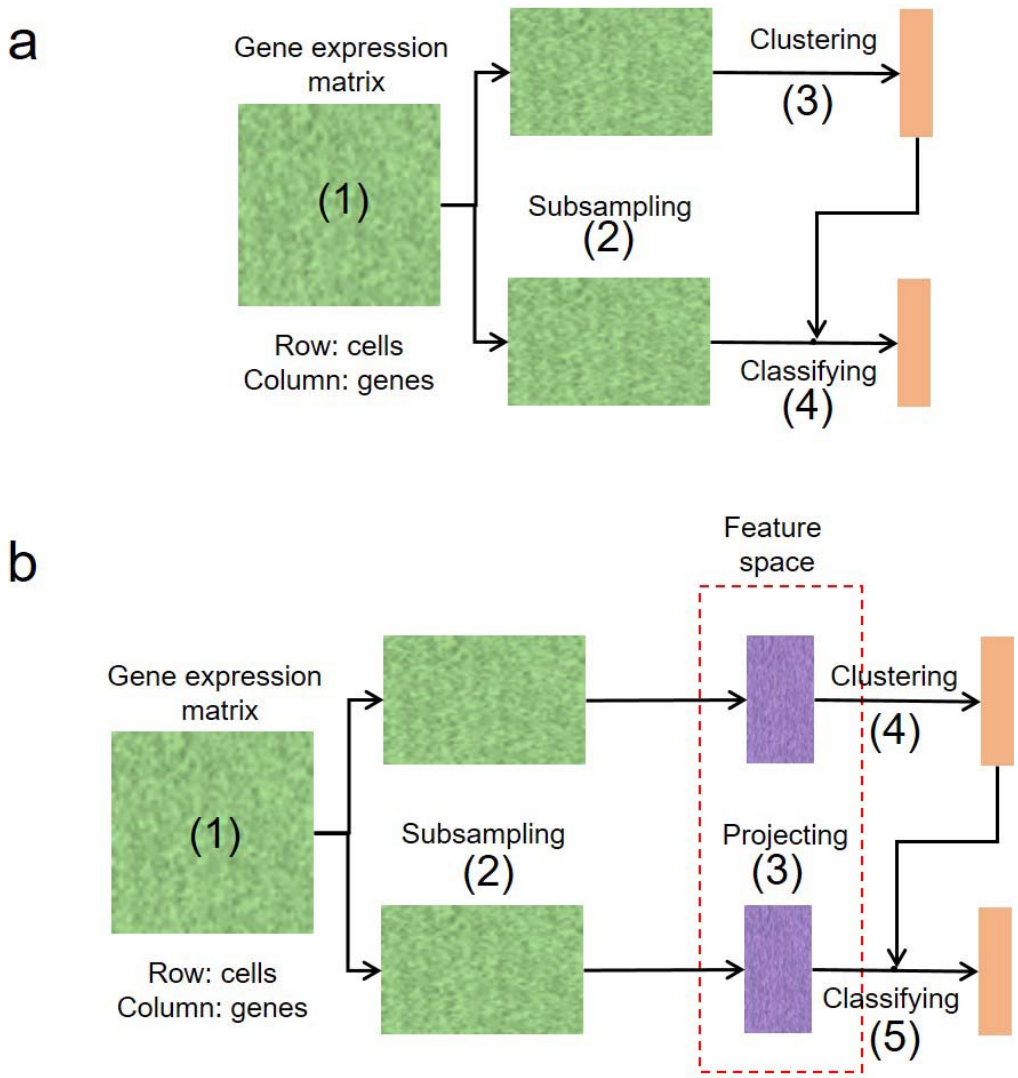


Fig. 1 Two computational frameworks for rapid clustering large-scale single-cell RNA-seq datasets. a, the original computational framework proposed in SC3 to handle large single-cell RNA-seq datasets, which consists of four main steps: (1) constructing the gene expression matrix; (2) dividing the matrix into two parts through cell subsampling; (3) clustering the subsampled cells; (4) classifying the unsampled cells into clusters formed by the subsampled cells. b, the new computational framework we proposed to handle large single-cell RNA-seq datasets. Different from the original framework, a feature construction step is added before clustering and classifying. Thus the whole framework is composed by five steps: (1) constructing the gene expression matrix; (2) dividing the matrix into two parts through cell subsampling; (3) projecting those subsampled/unsampled cells into feature space; (4) clustering the subsampled cells in the

feature space; (5) classifying the unsampled cells into clusters formed by the subsampled cells in the feature space.

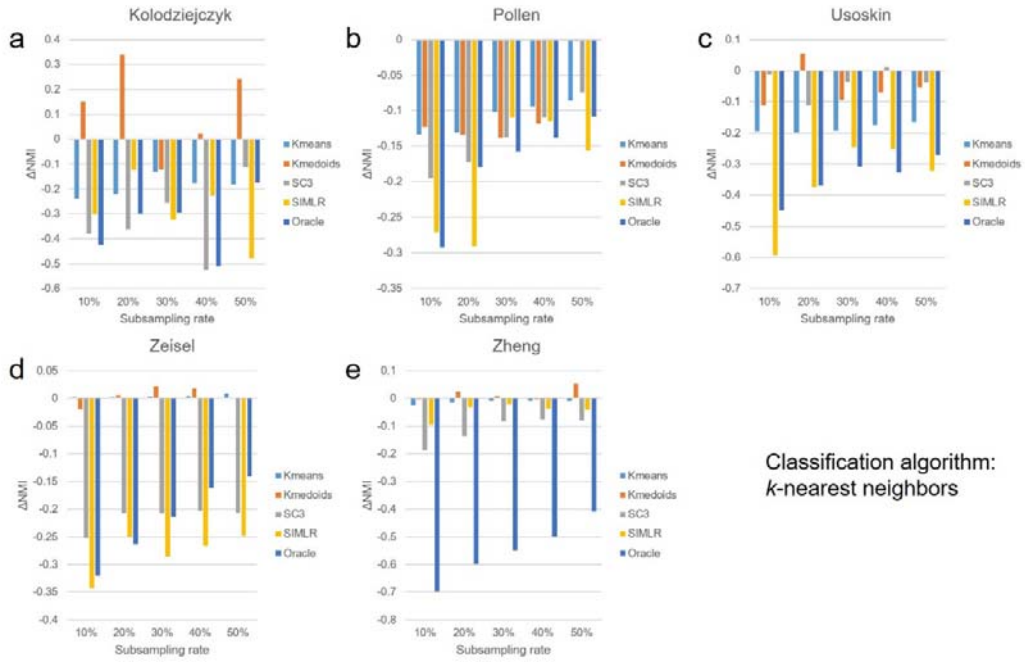


Fig. 2 The difference of clustering accuracy with/without subsampling measured by ΔNMI in the original framework ($\Delta NMI = NMI_{\text{with subsampling}} - NMI_{\text{without subsampling}}$). Evaluated on five single-cell RNA-seq datasets, the performance of almost all clustering algorithms were greatly reduced. The *k*-medoids algorithm is an exception because it generated poor performance when the total cells were used for clustering. When subsampling was applied, the k-nearest neighbors algorithm (*k*=1) was used to classify the unsampled cells. Changing k to 3, 5, 10 or switching the classification algorithm to random forest or support vector machines obtained similar results.

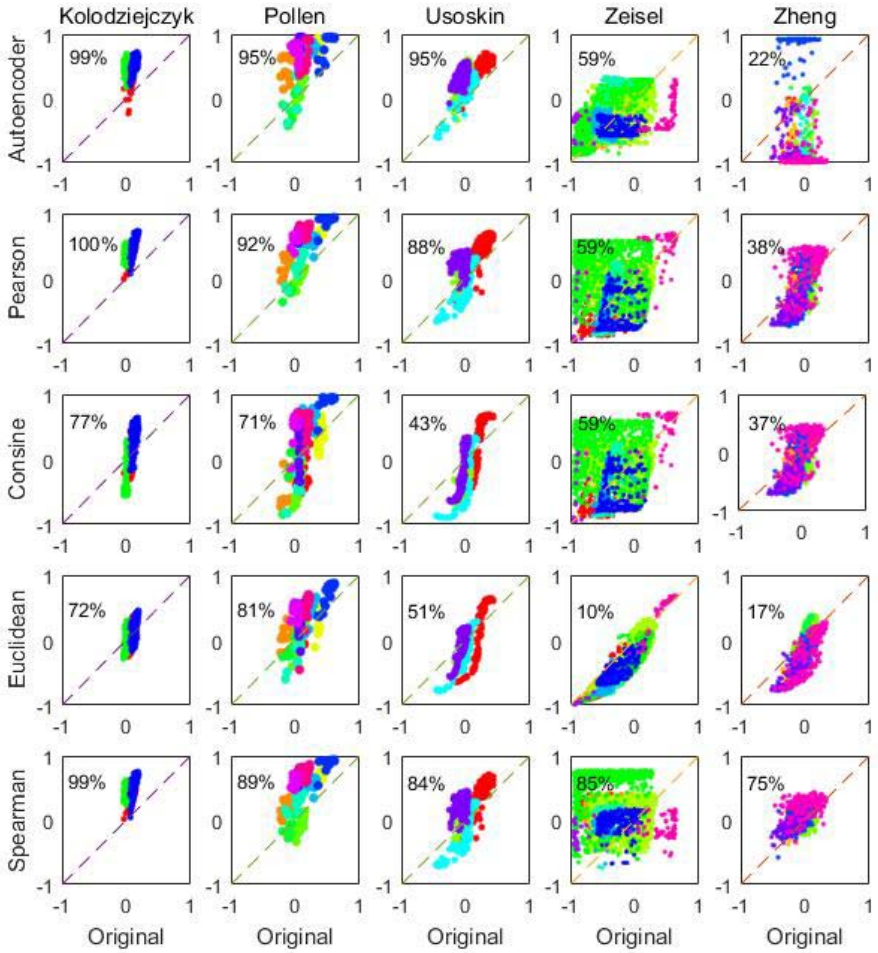


Fig. 3 Features constructed by Spearman correlations consistently have better consistency with true cluster labels than the original data on all the five datasets (Kolodziejczyk, Pollen, Usoskin, Zeisel and Zheng). X axis: consistency between true cluster labels and the original data; Y axis: consistency between true cluster labels and various features extracted by Autoencoder, Pearson correlation, consine, Euclidean distance and Spearman correlation. Silhouette values6 were used to measure the consistencies and calculated for each cell by using the silhouette function of Matlab R2016a, with 1 representing perfect match while -1 indicating that the cell might be mis-clustered. The digital numbers in each plot suggest the fraction of cells whose clustering are improved in the feature space compared to those based on the original data (i.e., above the diagonals).

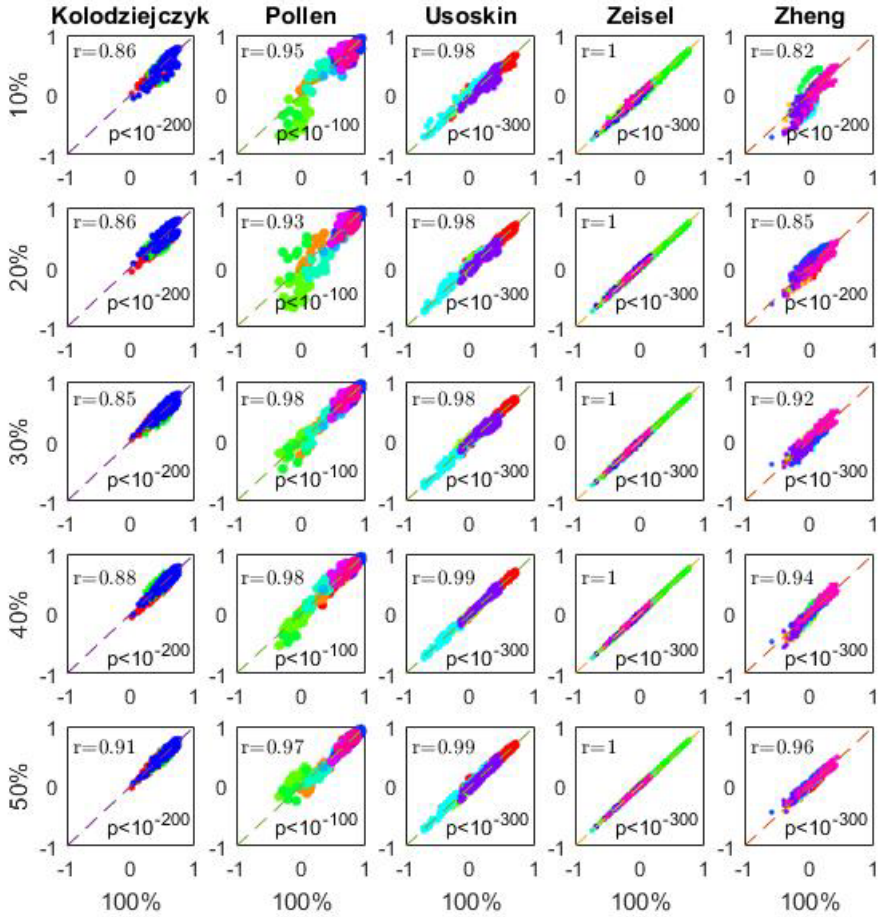


Fig. 4 Silhouette values between Spearman correlation features and true cluster labels were independent on subsampling rates. X axis: silhouette values of Spearman correlation features constructed with 100% cells. Y axes: silhouette values of Spearman correlation features constructed with10%, 20%, 30%, 40% and 50% cells in each dataset. Pearson correlation was calculated to evaluate the consistency between subsampled and unsubsampled features, of which the correlation coefficients (r) were plotted in the upper triangles while the corresponding p values were plotted in the lower triangles. It is evident that silhouette values of features constructed upon subsampled cells were almost the same as those constructed upon all the cells in the datasets. The independence of such Spearman correlation-based features on subsampling rates will greatly reduce the computational complexity of feature construction (from $O(n^2)$ to $O(n)$), enabling clustering of large scRNA-seq datasets be finished with high efficacy.

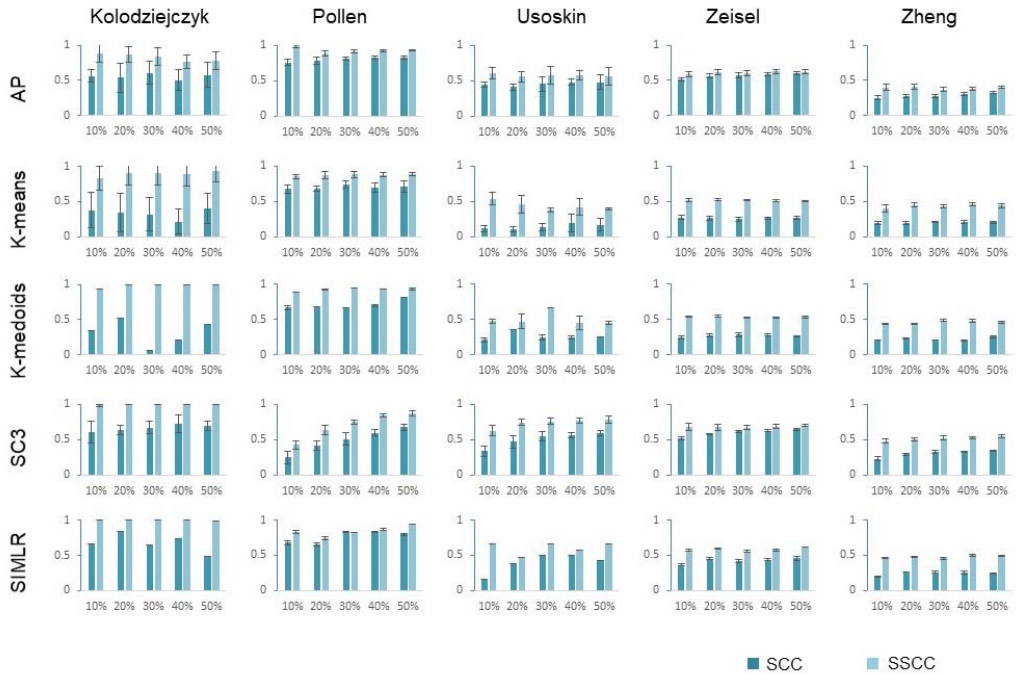


Fig. 5 Performance comparison between SCC and SSCC. For each plot, the X axis was the subsampling rate, i.e., the percentage of cells used in clustering, and the Y axis was the clustering accuracy measured by NMI. For each subsampling rate, calculations were repeated for ten times, based on which the average and the standard deviation of the clustering accuracy were calculated and plotted. It is evident that SSCC significantly improved clustering accuracy for almost all algorithms, all scRNA-seq datasets and all subsampling rates.

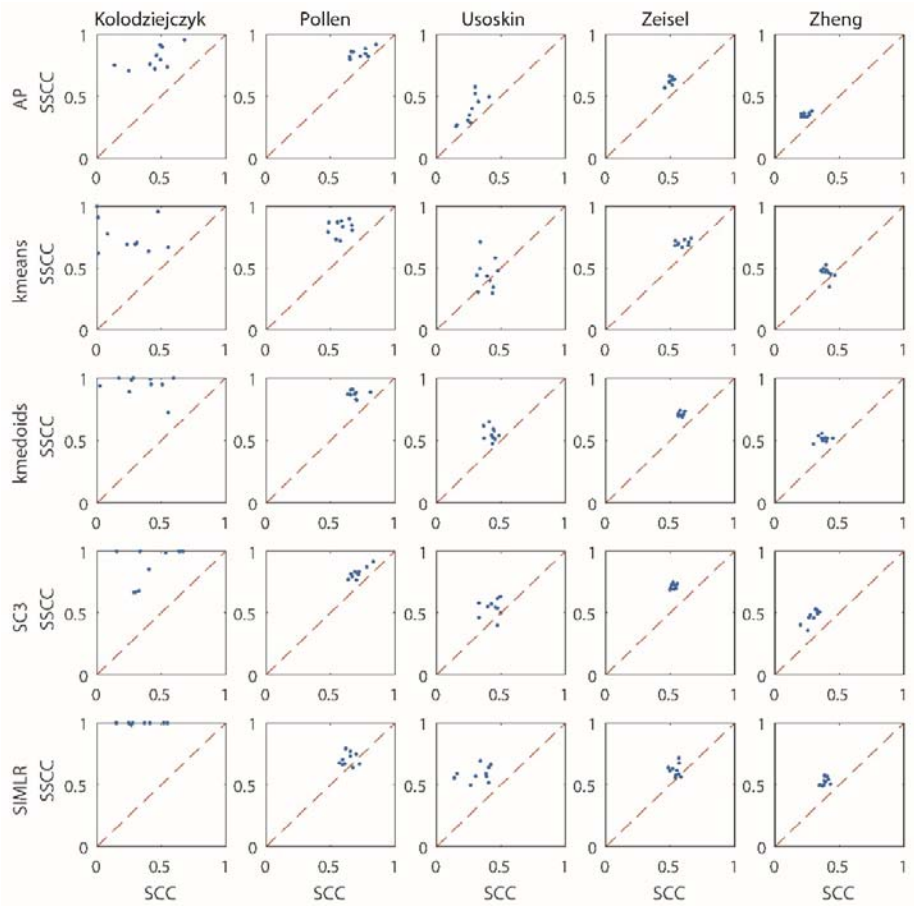


Fig. 6 Comparison of clustering consistency between SSCC and SCC. X axis: consistency (measured by NMI) of clustering upon 10% cells to that based on 50% cells with SCC. Y axis: consistency (measured by NMI) of clustering upon 10% cells to that based on 50% cells with SSCC. Subsamplings were repeated for ten times and each subsampling result was processed by five clustering algorithms (AP, k-means, k-medoids, SC3, and SIMLR).

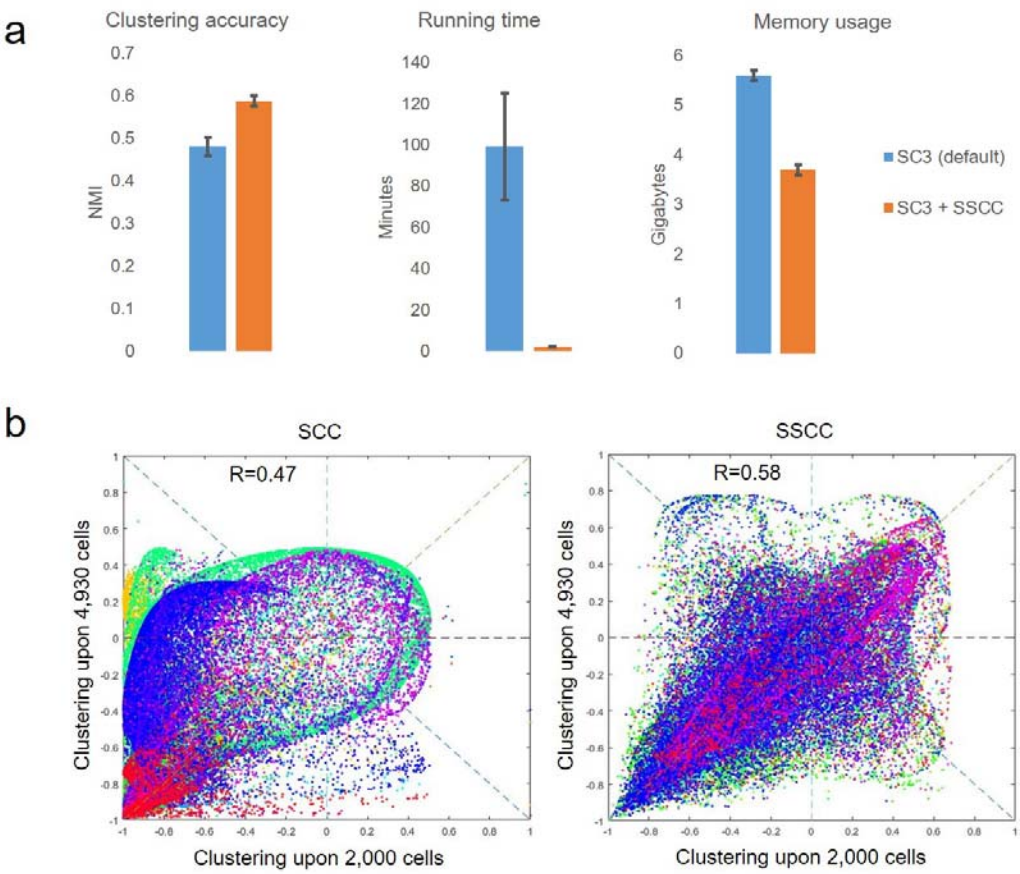
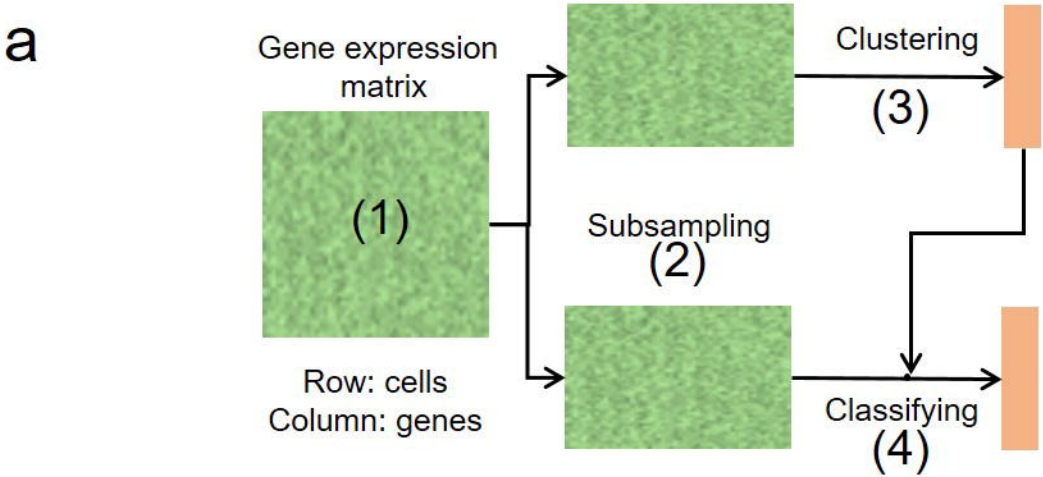
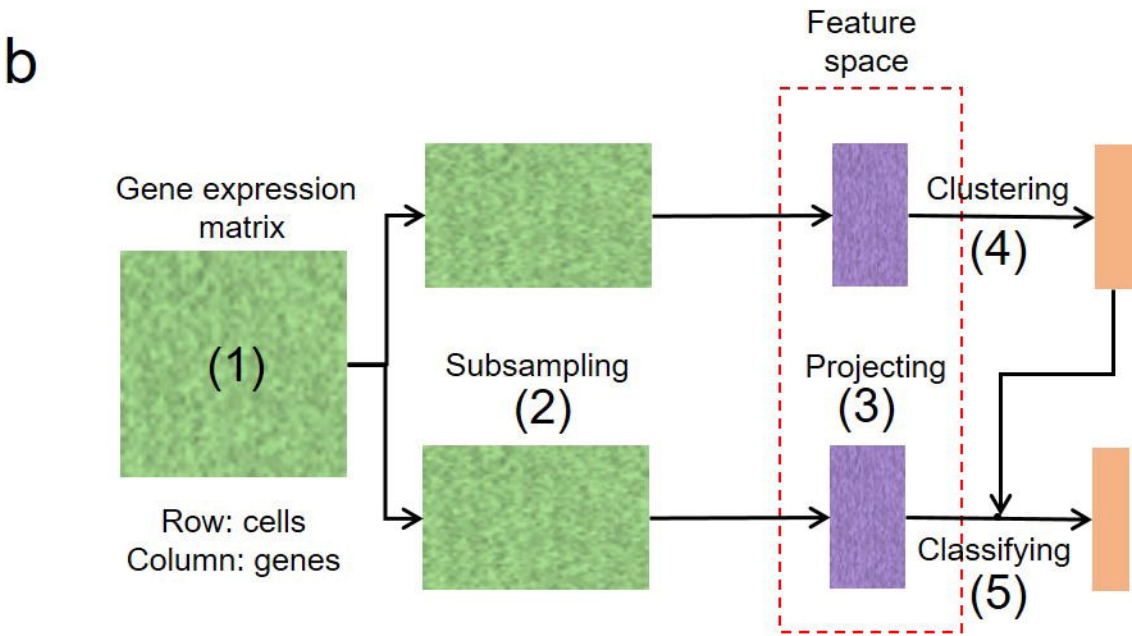
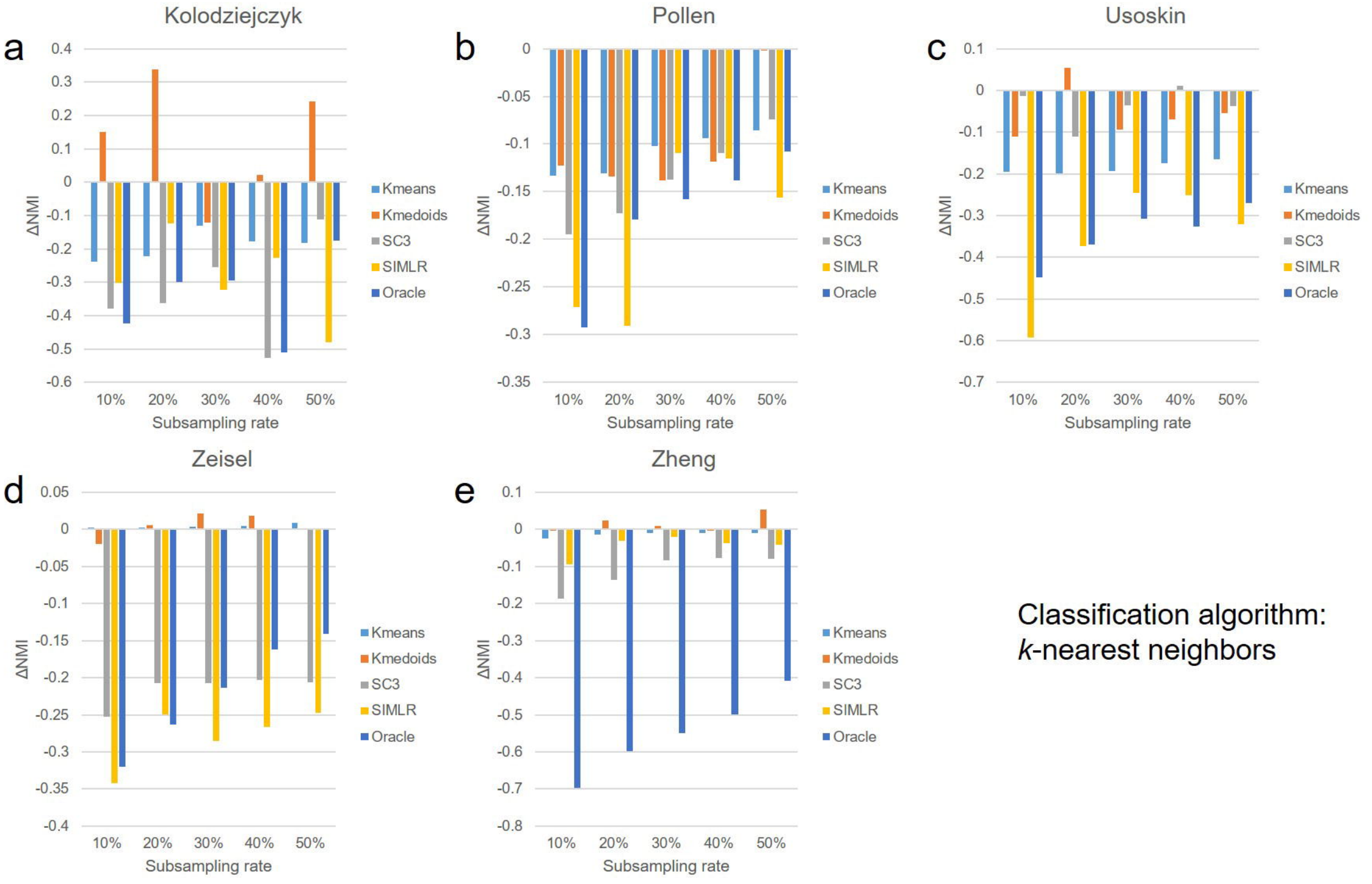


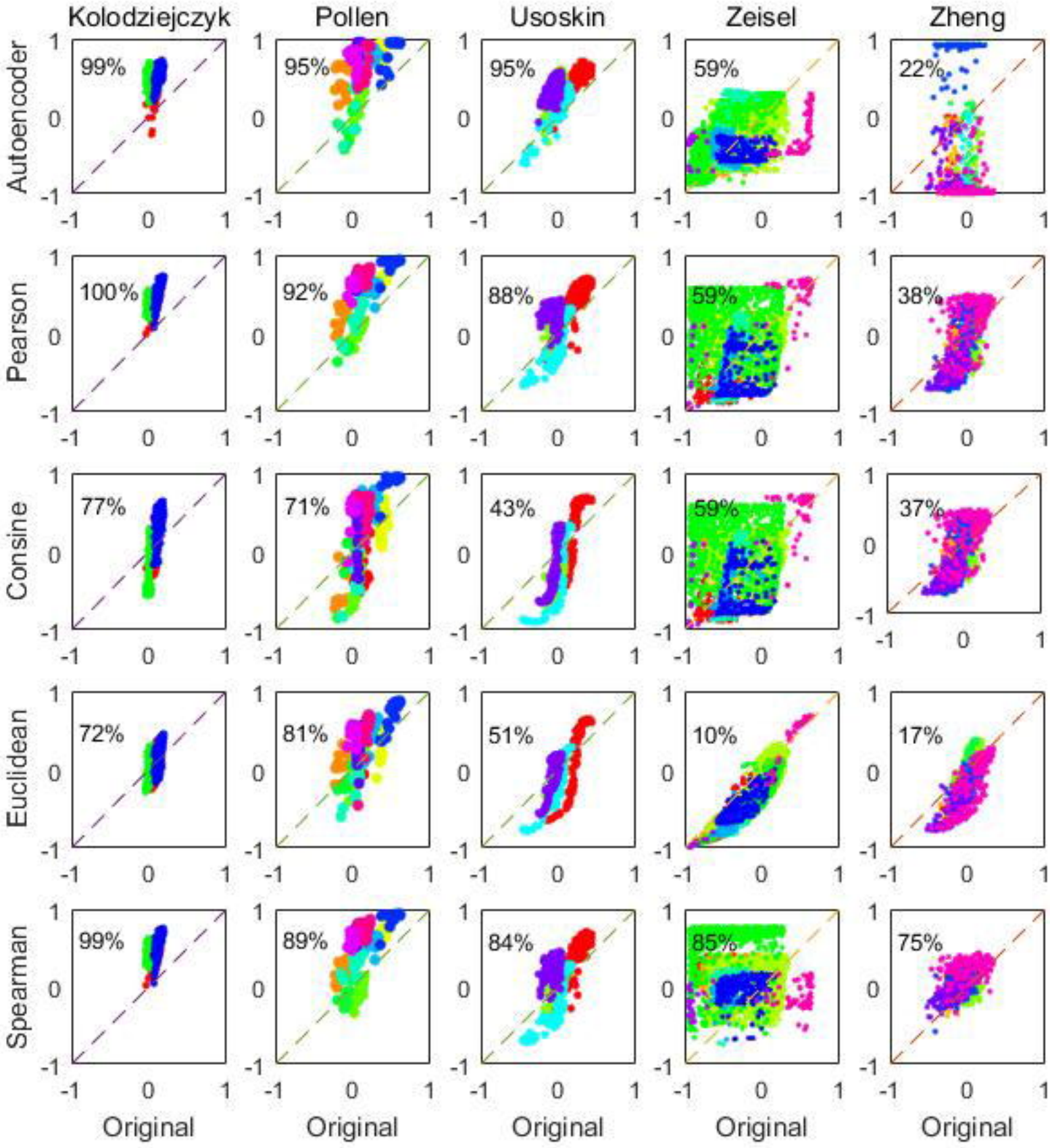
Fig. 7 Performance evaluation of SSCC on two extremely large scRNA-seq datasets. a, Performance comparison between SC3 (default) and SC3+SSCC on the PBMC 68k dataset. SC3 (default) inherently applied the SCC strategy to deal with large scRNA-seq datasets, of which 5,000 cells were subsampled by default for clustering and other cells were subject to classification based on SVM. For SC3+SSCC, an extremely low number of cells (n=200) were subsampled to construct cell features and then for clustering and classification. b, Consistency comparison between SSCC and SCC on a large scRNA-seq dataset with 49,300 mouse retina cells. The silhouette values of two clustering schemes (by 2000 and 4930 cells separately) with the SCC framework (the left panel) were small and relatively weak-correlated to each other (R=0.47, Pearson's correlation). The silhouette values of two clustering schemes (by 2000 and 4930 cells separately) with the SSCC framework were large and highly correlated to each other (R=0.58, Pearson's correlation). Cells were colored according to cluster labels based on ~10% cells and original expression data.

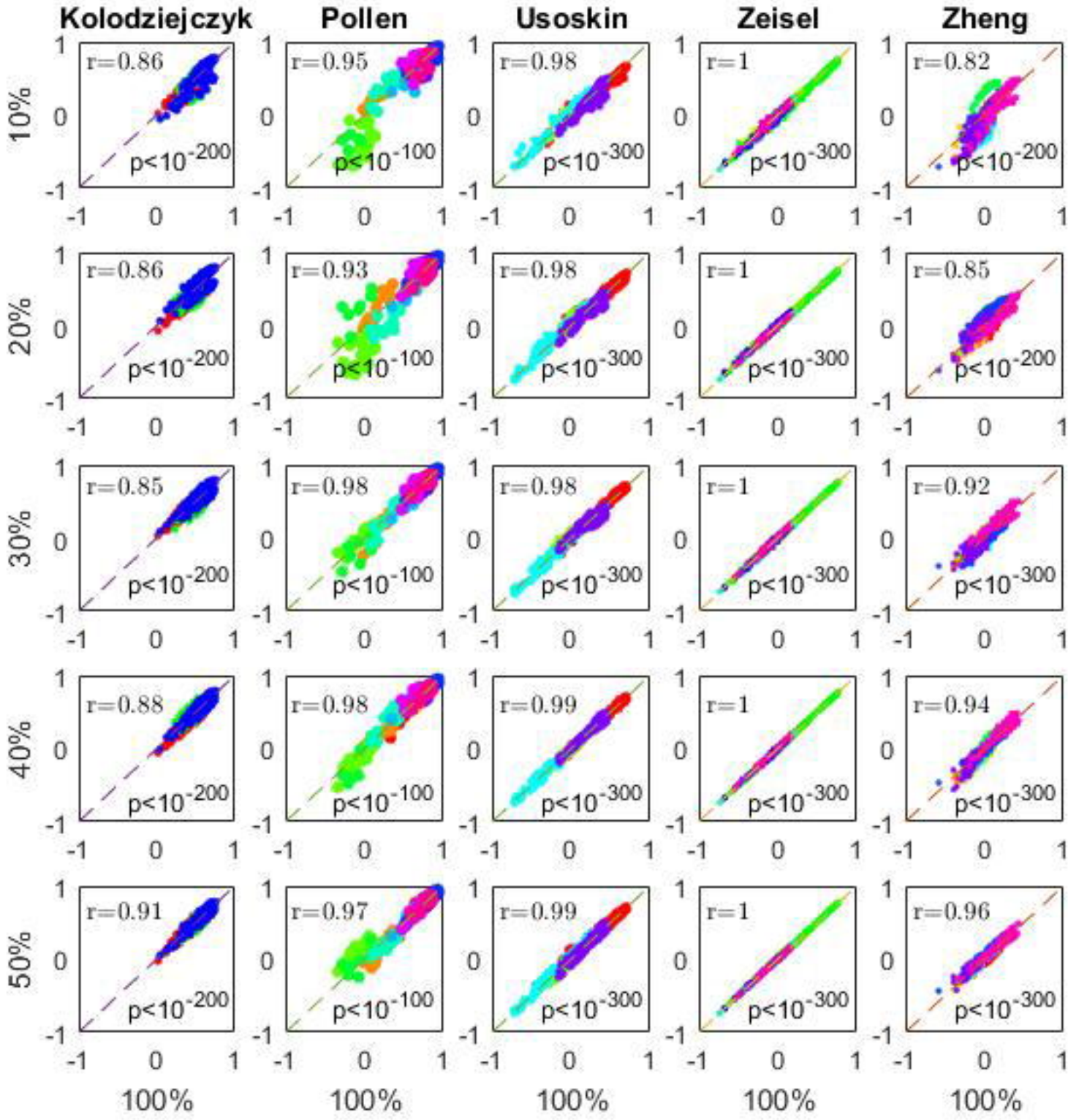

a
Gene expression matrix
(1)
Row: cells
Column: genes
Subsampling
(2)
Clustering
(3)
Classifying
(4)

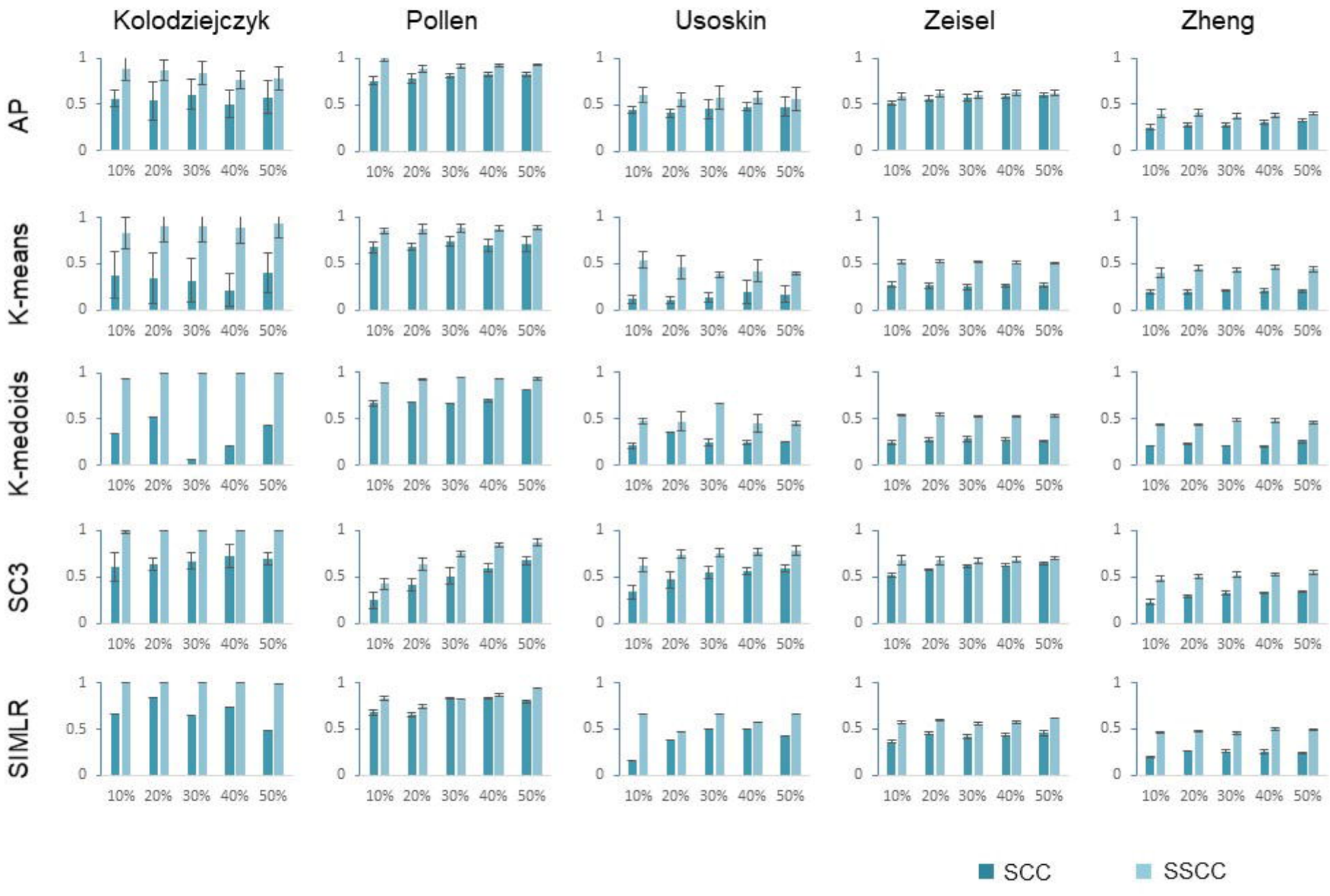

b
Gene expression matrix
(1)
Row: cells
Column: genes
Subsampling
(2)
Feature space
Projecting
(3)
Clustering
(4)
Classifying
(5)

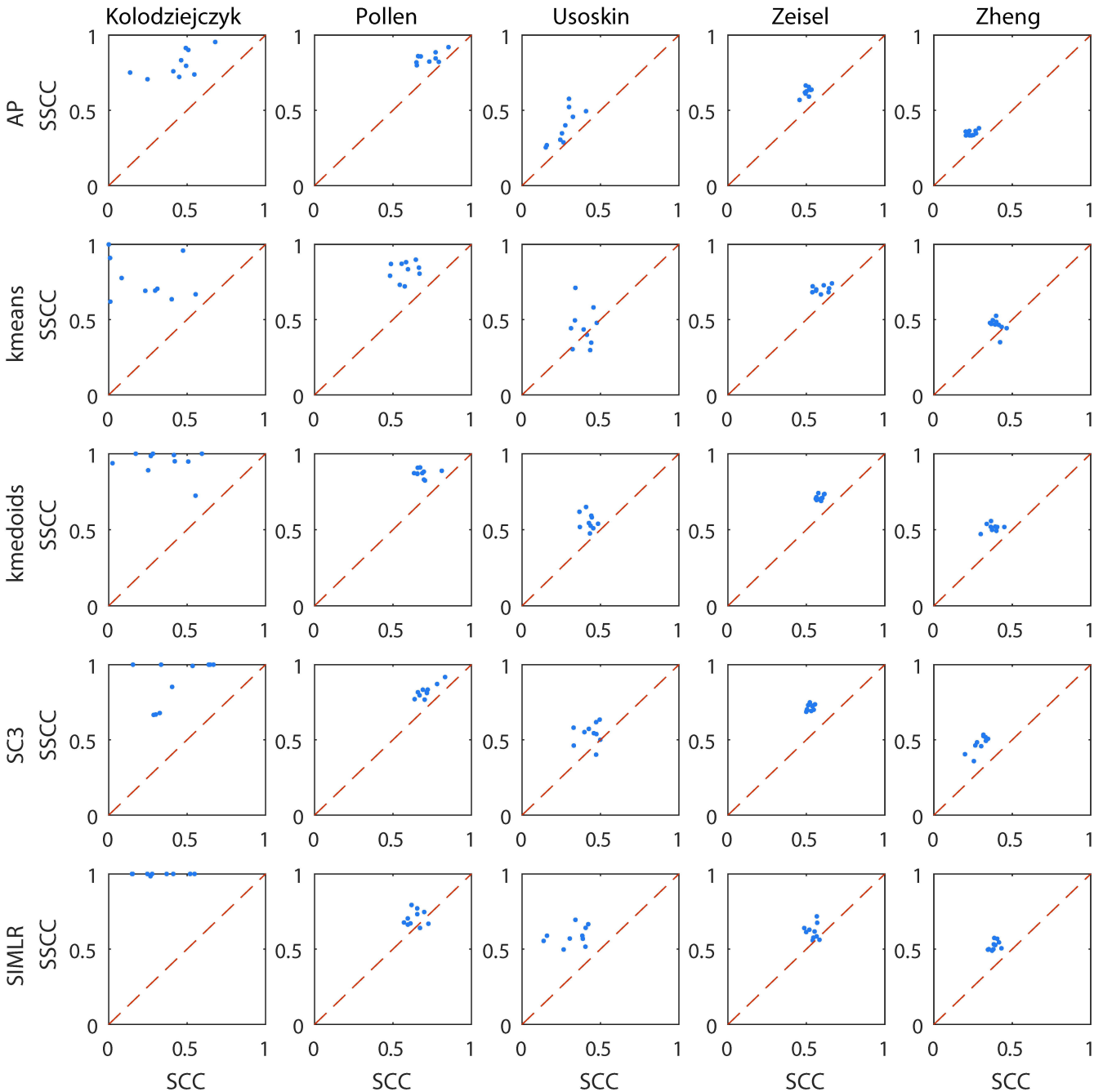


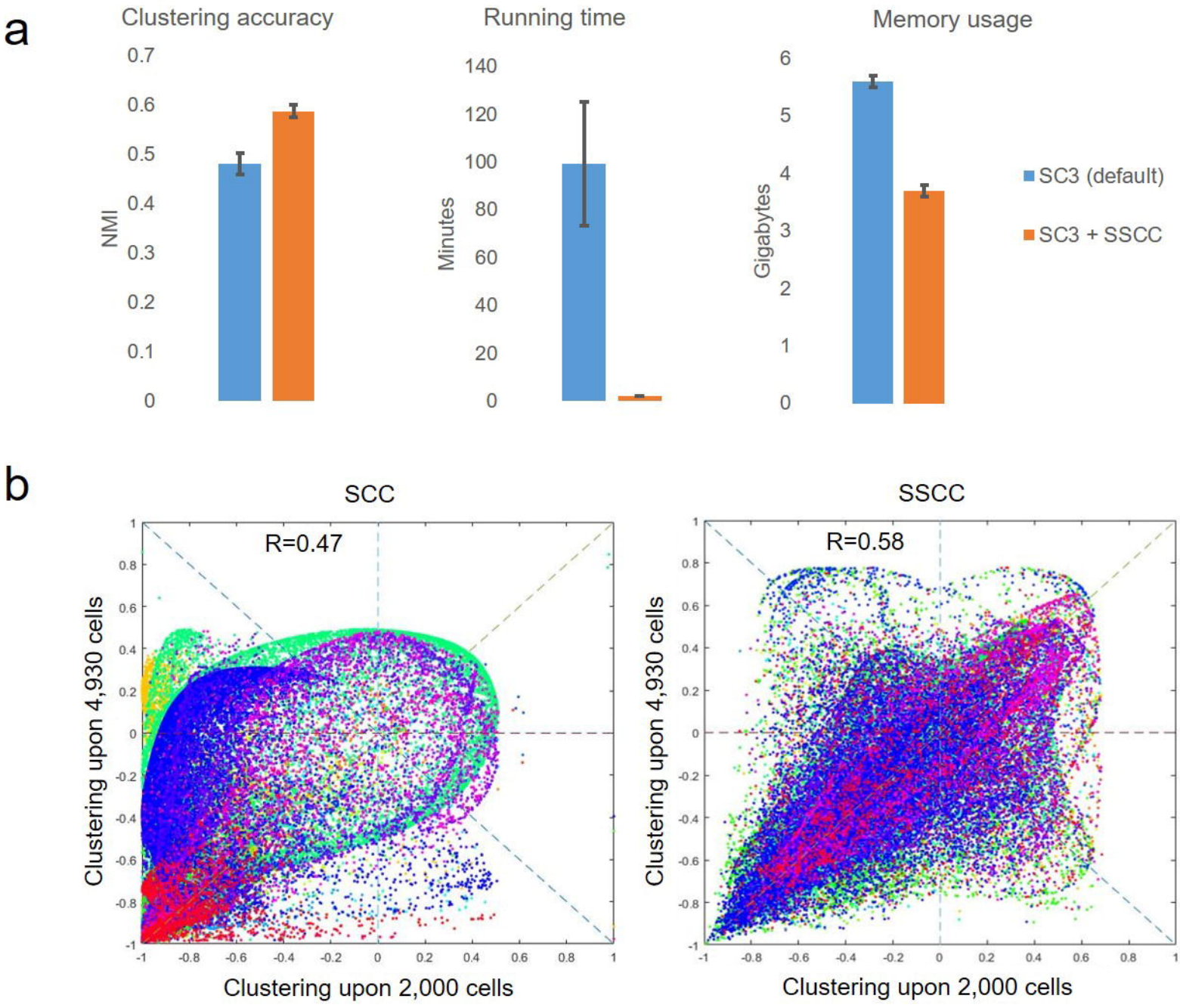
Classification algorithm:
k-nearest neighbors


Kolodziejczyk
Pollen
Usoskin
Zeisel
Zheng
Autoencoder
Pearson
Consine
Euclidean
Spearman
99%
95%
95%
59%
22%
100%
92%
88%
59%
38%
77%
71%
43%
59%
37%
72%
81%
51%
10%
17%
99%
89%
84%
85%
75%
1
0
-1
Original


Kolodziejczyk
Pollen
Usoskin
Zeisel
Zheng
10%
20%
30%
40%
50%
100%
r=0.86
p<10^-200
r=0.95
p<10^-100
r=0.98
p<10^-300
r=1
p<10^-300
r=0.82
p<10^-200
r=0.86
p<10^-200
r=0.93
p<10^-100
r=0.98
p<10^-300
r=1
p<10^-300
r=0.85
p<10^-200
r=0.85
p<10^-200
r=0.98
p<10^-100
r=0.98
p<10^-300
r=1
p<10^-300
r=0.92
p<10^-300
r=0.88
p<10^-200
r=0.98
p<10^-100
r=0.99
p<10^-300
r=1
p<10^-300
r=0.94
p<10^-300
r=0.91
p<10^-200
r=0.97
p<10^-100
r=0.99
p<10^-300
r=1
p<10^-300
r=0.96
p<10^-300


Kolodziejczyk
Pollen
Usoskin
Zeisel
Zheng
AP
K-means
K-medoids
SC3
SIMLR
0
0.5
1
10%
20%
30%
40%
50%
SCC
SSCC


Kolodziejczyk
Pollen
Usoskin
Zeisel
Zheng
AP
kmeans
kmedoids
SC3
SIMLR
SSCC
SCC
0
0.5
1


a
Clustering accuracy
NMI
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Running time
Minutes
140
120
100
80
60
40
20
0
Memory usage
Gigabytes
6
5
4
3
2
1
0
SC3 (default)
SC3 + SSCC
b
SCC
R=0.47
Clustering upon 4,930 cells
Clustering upon 2,000 cells
SSCC
R=0.58
Clustering upon 4,930 cells
Clustering upon 2,000 cells