Briefings in Bioinformatics, 00(00), 2020, 1–11

doi: 10.1093/bib/bbaa216
Problem Solving Protocol

A spectral clustering with self-weighted multiple kernel learning method for single-cell RNA-seq data

Ren Qi†, Jin Wu†, Fei Guo, Lei Xu and Quan Zou

Corresponding authors: Quan Zou, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China. E-mail: zouquan@nclab.net; Lei Xu, School of Electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen, China. E-mail: csleixu@szpt.edu.cn
†These authors contributed equally to this study.

Abstract

Single-cell RNA-sequencing (scRNA-seq) data widely exist in bioinformatics. It is crucial to devise a distance metric for scRNA-seq data. Almost all existing clustering methods based on spectral clustering algorithms work in three separate steps: similarity graph construction; continuous labels learning; discretization of the learned labels by k-means clustering. However, this common practice has potential flaws that may lead to severe information loss and degradation of performance. Furthermore, the performance of a kernel method is largely determined by the selected kernel; a self-weighted multiple kernel learning model can help choose the most suitable kernel for scRNA-seq data. To this end, we propose to automatically learn similarity information from data. We present a new clustering method in the form of a multiple kernel combination that can directly discover groupings in scRNA-seq data. The main proposition is that automatically learned similarity information from scRNA-seq data is used to transform the candidate solution into a new solution that better approximates the discrete one. The proposed model can be efficiently solved by the standard support vector machine (SVM) solvers. Experiments on benchmark scRNA-Seq data validate the superior performance of the proposed model. Spectral clustering with multiple kernels is implemented in Matlab, licensed under Massachusetts Institute of Technology (MIT) and freely available from the Github website, https://github.com/Cuteu/SMSC/.

Key words: scRNA-Seq; spectral clustering; multiple kernel learning; self-weighted; cell clustering

Introduction

Single-cell RNA sequencing (scRNA-Seq) uses optimized next-generation sequencing technologies to analyze individual cells, leading to a better understanding of cell function at the genetic and cellular levels [1]. It allows researchers to analyze cellular heterogeneity and transcriptome heterogeneity at the single-cell level [2]. ScRNA-Seq can provide information on individual cell transcriptomes, and this information can be used to identify cell subsets and to determine the time stage of cell differentiation and the progression of single cells [3]. Therefore, devising a distance metric for scRNA-Seq data analysis is a crucial problem.

In virtually all areas of science, discovering natural groupings in data is a fundamental issue. Clustering methods boost the performance by pushing the samples between classes away

Ren Qi is a doctoral student in the College of Intelligence and Computing, Tianjin University. Her research interests include machine learning, metric learning and bioinformatics.
Jin Wu is a lecture at the School of Management, Shenzhen Polytechnic. Her research interests include microbiome and bioinformatics.
Fei Guo is an associate professor at the College of Intelligence and Computing, Tianjin University. Her research interests include bioinformatics, algorithms and machine learning.
Lei Xu is an associate professor at the School of Electronic and Communication Engineering, Shenzhen Polytechnic. Her research interests include machine learning and bioinformatics.
Quan Zou is a professor at the University of Electronic Science and Technology of China. He is a senior member of the Institute of Electrical and Electronics Engineers and Association for Computing Machinery. He won the Clarivate Analytics Highly Cited Researchers in 2018 and 2019. He majors in bioinformatics, machine learning and algorithms. His email is zouquan@nclab.net.
Submitted: 5 June 2020; **Received (in revised form):** 14 August 2020

while pulling the samples in the same classes closer. ScRNA-Seq data brings some challenges, such as dropout events [4, 5] and the curse of dimension [6]. These challenges bring many difficulties to the downstream analysis of scRNA-Seq data. The desired clustering methods for different applications vary widely due to differences in the underlying structure and distribution of the data and the specificity of the learning tasks. Task-specific clustering metrics have been devised for cancer, metagenomics, regulatory networks [7–10], and other areas of study.

The performance of model-based methods heavily depends on whether the data fit the model. SEURAT used an analytical strategy for integrating scRNA-seq data sets based on common sources of variation, enabling the identification of shared populations across data sets and downstream comparative analysis [11, 12]. It can be applied to infer the spatial location of a complete transcriptome and correctly located unusual subpopulations and aligned scRNA-Seq data sets of peripheral blood mononuclear cells. SNN-Cliq combined with an SNN similarity metric can automatically determine the number of clusters, especially in high-dimensional single-cell data, which is a great advantage [13]. Analysis of mouse lung cells using the SINCERA pipeline distinguished the main cell types of fetal lung [14]. Unfortunately, in most cases, we do not know the distribution of data in advance. To some extent, this problem is alleviated by multiple kernel learning.

According to whether or not the number of clusters in the data is known, unsupervised clustering learning methods can be divided into three categories. Some can automatically determine the true number of clusters with reasonable accuracy [15], some require the entry of the minimum number of clusters [16], and most require the number of clusters as a priori knowledge. The most widely used clustering techniques include center-based (e.g. k-means [17]), hierarchical clustering [18] and methods that view clustering as a graph partitioning problem (e.g. spectral clustering [SC] [19]). Unlike shallow clustering learning, deep learning-based methods learn the feature and the metric jointly and achieve superior performance [20].

Due to its advantages of simplicity and effectiveness, the SC algorithm is often adopted in various real-world problems. Almost all existing clustering methods based on SC focus on similarity graph construction, continuous labels learning and discretization of the learned labels by k-means clustering [17]. However, predefined similarity graphs might not allow subsequent clustering. It is known that the clustering results are largely determined by the similarity graph. Although this approach has been widely used in practice, it may exhibit poor performance since the k-means method is well-known as sensitive to the initialization of cluster centers.

The unique features of such scRNA-Seq data are high dimensionality and many expression values of zero [21]. Traditional bulk RNA-Seq data are obtained from a series of cells, and the level of expression of each gene is the average or sum of its expression in all of the analyzed cells [22]; therefore, only a few values of zero appear in bulk RNA-Seq data. However, if a gene is not expressed in a cell, the corresponding feature of the gene will be zero in scRNA-Seq [23]. Moreover, some undetected genes with relatively low expression levels were also labeled as zero [24]. This is the main reason for the poor analysis of scRNA-Seq data.

To this end, we propose a new clustering method in the form of a multiple kernel combination that can directly discover groupings in scRNA-Seq data. The proposed method automatically learns a similarity metric from the single-cell data and simultaneously takes into account the constraint that the similarity matrix has s connected components if it has s clusters into account. Besides, we transform the candidate solution into a new solution that better approximates the discrete one. It is well-accepted that the choice of kernels greatly affects the performance of a kernel method. To find the most suitable kernel, we extend the model to incorporate multiple kernel learning ability. The proposed SC with Multiple kernel learning for Single-Cell RNA-Seq data (SMSC) framework is shown in Figure 1. The contributions of this paper are summarized as follows:

1. A unified clustering learning framework is proposed for use with scRNA-Seq data. Instead of using predefined similarity metrics, we combine similarity learning with subsequent clustering into a unified framework.
2. The proposed method is formulated as a multiple kernel model. Sparse single-cell data can be mapped by a kernel function to obtain a non-sparse kernel matrix.
3. Experiments on gold-standard and silver-standard scRNA-Seq datasets show that our method achieves superior performance compared to state-of-the-art methods.

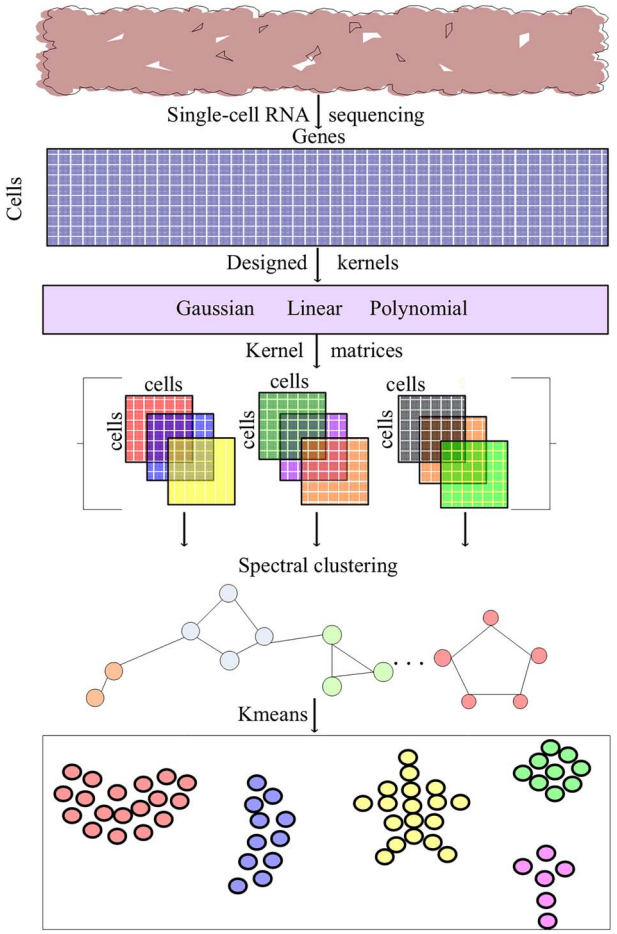

Figure 1. The SMSC framework for the clustering of scRNA-seq data. The SMSC framework is composed of four stages. The first stage obtains dataset expression matrices. The Cells-Genes matrix was raw data. The second stage is to calculate the kernel matrices. They were obtained by processing the original data with effective fusion multiple kernel functions. The kernel matrices can be used for spectral clustering in the third stage, and they can be obtained by any reasonable kernel functions. Finally, cells are clustering by K-means algorithm.

Table 1. Summary of 11 single-cell RNA-seq datasets

Dataset		genes	cells	Clusters	Cell Resource	Download
1	Yan's	20 214	124	7	Human preimplantation	GSE36552
2	Goolam's	41 480	124	5	4-Cell Mouse Embryos	E-MTAB-3321
3	Deng's	22 457	268	10	Mouse preimplantation embryos	GSE45719
4	Pollen's	23 730	301	11	Human	SRP041736
5	Treutlein's	23 271	80	5	Human lung epithelium	GSE52583
6	Ting's	29 018	149	7	Human pancreatic circulating tumor cells	GSE51372, GSE60407 and GSE51827.
7	Patel's	5948	430	5	Human glioblastomas	GSE57872
8	Usoskin's	25 334	622	11	Human neuron	GSE59739
9	Klein's	24 175	2717	4	Human Embryonic Stem Cells	GSE65525
10	Zeisel's	19 972	3005	9	Mouse cortex	GSE60361
11	Chen's	28 234	14 437	47	Mouse Brain	GSE87544

Materials and methods

Overview of SMSC pipeline

The SMSC framework is composed of four stages. The first stage input scRNA-Seq expression matrices data. The cells–genes matrices are raw scRNA-Seq expression data. The second stage calculates multiple kernel matrices, which can be obtained by any reasonable kernel functions. Here, we used Gaussian kernels and polynomial kernels with 11 different combined parameters. Then, we used fusion multiple kernel functions to combine kernel matrices and input them to SC in the third stage. Finally, cells are clustering by K-means algorithm (Figure 1).

Datasets

We collected the data from 11 publicly available scRNA-Seq datasets. The number of cells among the above datasets varies from 80 to 14 437. We downloaded most datasets from Gene Expression Omnibus (GEO) website (https://www.ncbi.nlm.nih.gov/geo/) except Goolam by project E-MTAB-3321 (https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-3321/). Detailed information regarding these datasets and GEO accessions were shown in Table 1.

The first four datasets were selected on the basis that one can be highly confident in the cell labels as they represent cells from different stages, conditions or lines, and thus we consider them as 'gold standard' [25]. Yan's dataset contains 124 individual cells obtained from human preimplantation embryos and human embryonic stem cells and consists of 20 214 genes [26]. Goolam's dataset characterizes the transcriptomes of all individual cells throughout mouse preimplantation development; it consists of 41 480 genes from 124 single cells [27]. Deng's dataset presents an analysis of allelic expression across individual cells of mouse preimplantation embryos and consists of 22 958 genes from 268 single cells [28]. Pollen's dataset captured 301 single cells from 11 populations using microfluidics; it consists of 23 730 genes [29].

We further tested the SMSC pipeline on seven other published datasets, where the cell labels can only be considered 'silver standard' since they were assigned using computational methods and the authors' knowledge of the underlying biology [25]. Treutlein et al. used microfluidic scRNA-Seq to detect six different stages of distal mouse lung epithelium cells to form the dataset of 23 271 genes in 80 cells [30]. Ting et al. isolated individual circulating tumor cells using epitope-independent microfluidic capture, followed by scRNA-Seq [31]. They published Ting's dataset, which contains 149 cells and 29 018 genes. Patel et al. used scRNA-Seq to profile 430 cells from five primary glioblastomas [32]; the resulting dataset contains 5948 genes. Usoskin et al. used a comprehensive transcriptome analysis of 622 single mouse neurons to form a dataset of 25 334 genes [33]. Zeisel's dataset contains 19 972 genes from 3005 cells and was used to study specialized cell types in the mouse cortex and hippocampus [34]. Klein's dataset is sequenced messenger RNA of thousands of mouse embryonic stem and differentiating cells, which contains 24 175 genes and 2727 cells [35]. Chen's dataset contains 14 437 cells and 28 234 genes. The Hypothalamus of mice was dissected for single-cell RNA-seq through the Drop-seq technique.

Preliminary knowledge

Notations

Given a data set $[x_1, x_2, \cdots, x_n]$, we denote $X \in \mathcal{R}^{n\times d}$ with d features and n samples. Then the $i-th$ sample and $(i,j)-th$ element of matrix X are denoted by $x_i \in \mathrm{R}^{n\times 1}$ and x_{ij}, respectively. The ℓ_2- norm of a vector $\S$ is defined as $\|\S\|^2 = \S^T$, where T means transpose. The squared Frobenius norm is denoted by $||X||_F^2 = \sum_{ij} x_{ij}^2$. The ℓ_1- norm of matrix X is defined as the absolute summation of its entries. I denotes the identity matrix. $Tr(\bullet)$ is the trace operator. $M \geq 0$ indicates that all the elements of M are nonnegative.

Sparse representation

Recently, sparse representation, which assumes that each data point can be reconstructed as a linear combination of the other data points, has shown its power in many tasks [36]. Sparse representation assumes that each data point can be reconstructed as a linear combination of other data points. It often solves the following problem:

$$\min_{M} ||X - XM||_F^2 + \alpha||M||_1, \text{s.t.} M \geq 0, \text{diag}(M) = 0, \quad (1)$$

where $\alpha > 0$ is a balancing parameter. Equation (1) simultaneously determines both the neighboring samples of a data point and the corresponding weights by the sparse reconstruction from the remaining samples. In principle, more similar points should receive bigger weights and the weights should be

smaller for less similar points. Thus M is also called similarity graph matrix. In addition, sparse representation enjoys some nice properties, e.g. the robustness to noise and datum-adaptive ability [37]. On the other hand, model (1) has a drawback, i.e. it does not consider nonlinear data sets where data points reside in a union of manifolds [38].

Spectral clustering

The SC requires Laplacian matrix $L \in \mathcal{R}^{n\times n}$ as input, which is computed as $L = D - \frac{M^T+M}{2}$, where $D \in \mathcal{R}^{n\times n}$ is a diagonal matrix with the $i-th$ diagonal element $\sum_j \frac{M_{ij}+M_{ij}}{2}$. In traditional SC methods, similarity graph $M \in \mathcal{R}^{n\times n}$ is often constructed in one of the three ways aforementioned. Supposing there are s clusters in the data$X$, SC solves the following problem:

$$_E{}^{\min}\ Tr\left(E^TLE\right), s.t. E \in Idx, \tag{2}$$

where $E = [e_1, e_2, \cdots, e_n]^T \in R^{n\times s}$ is the cluster indicator matrix, and $E \in Idx$ represents the clustering label vector of each point $e_i \in \{0,1\}^{s\times 1}$ contains one and only one element '1' to indicate the group membership of x_i. Due to the discrete constraint on E, problem (2) is NP-hard. In practice, E is relaxed to allow continuous values and solve

$$_C{}^{\min}\ Tr\left(C^TLC\right), s.t. C^TC = I, \tag{3}$$

where $C \in \mathcal{R}^{n\times s}$ is the relaxed continuous clustering label matrix, and the orthogonal constraint is adopted to avoid trivial solutions. The optimal solution is obtained from the s eigenvectors of L corresponding to s smallest eigenvalues. After obtaining E, traditional clustering method, e.g. k-means, is implemented to obtain discrete cluster labels.

Methods

SC with single kernel

One drawback of Equation (1) is that it assumes that all the points lie in a union of independent or disjoint subspaces and are noiseless. In the presence of nonlinear manifolds or dependent subspaces, the algorithm may choose points from different structures, making the representation less informative [39]. It is recognized that nonlinear data may represent linearity when mapped to a higher-dimensional space via kernel function. To fully exploit data information, we formulate Equation (1) in a general manner with a kernelization framework.

Let ϕ: $\mathcal{R}^D \to \mathcal{H}$ be a kernel mapping the data samples from the input space to a reproducing kernel Hilbert space $\mathcal{R}$. Then X is transformed to $\phi(X) = [\phi(x_1), \cdots, \phi(x_n)]$. The kernel similarity between data samples x_i and x_j is defined through a predefined kernel as $K_{x_i,x_j} =< \phi(x_i), \phi(x_i) >$. By applying this kernel trick, we do not need to know the transformation ϕ. In the new space, Equation (1) becomes [40].

$$\begin{aligned} &\min_M \left\|\phi(X) - \phi(X)M\right\|_F^2 + \alpha\|M\|_1, \\ \Longleftrightarrow &\min_M Tr\begin{pmatrix} \phi(X)^T\phi(X) - \phi(X)^T\phi(X)M \\ -M^T\phi(X)^T\phi(X) + M^T\phi(X)^T\phi(X)M \end{pmatrix} + \alpha\|M\|_1 \\ \Longleftrightarrow &\min_M Tr\left(K - 2KM + M^TKM\right) + \alpha\|M\|_1 \\ &s.t. M \geq 0, \quad diag(M) = 0 \end{aligned} \tag{4}$$

This model recovers the linear relations among the data in the new space, and thus the nonlinear relations in the original representation. The kernelized Equation (4) is more general than Equation (1) and is supposed to learn arbitrarily shaped data structure. Moreover, Equation (4) goes back to Equation (1) when a linear kernel is applied.

To achieve the clustering task, we combine SC with single kernel method as following:

$$\begin{aligned} &\min_{M,E,C,Q} \underbrace{Tr\left(K - 2KM + M^TKM\right) + \alpha\|M\|_1}_{\text{similarity learning}} \\ &+\beta \underbrace{Tr\left(C^TLC\right)}_{\text{continuous label learning}} + \gamma \underbrace{\|E - CQ\|_F^2}_{\text{discrete label learning}} \\ &s.t. \quad M \geq 0, \quad diag\left(M\right) = 0, \\ &C^TC = I, \\ &Q^TQ = I, \\ &E \in Idx, \end{aligned} \tag{5}$$

where Q is a rotation matrix, and α, β and γ are penalty parameters. Due to the spectral solution invariance property [41], for any solution C, CQ is another solution. The purpose of the last term is to find a proper orthonormal Q such that the resulting CQ is close to the real discrete clustering labels. In Equation (5), the similarity graph and the final discrete clustering labels are automatically learned from data. Ideally, whenever data points i and j belong to different clusters, we must have $‡_{ij} = 0$and it is also true vice versa. That is to say, we have $‡_{ij} \neq 0$if and only if data points i and j are in the same cluster, or, equivalently $⌉_i = ⌉_j$. Therefore, Equation (5) can exploit the correlation between the similarity matrix and the labels.

Equation (5) learns a similarity graph with an optimal structure for clustering. Ideally, M should have exactly s connected components if there are s clusters in the data set. This is to say that the Laplacian matrix L has s zero eigenvalues [42]. To ensure the optimality of the similarity graph, we can minimize $\sum_{i=1}^s \sigma_i(L)$. According to Ky Fan's theorem [43], $\sum_{i=1}^s \sigma_i(L) = \min_{C^TC=I} Tr(C^TLC)$. Therefore, the SC term will ensure learned M is optimal for clustering.

Optimization

To efficiently solve Equation (5), we design an alternated iterative method.

Computation of M: With E, C, Q fixed, the problem is reduced to

$$\begin{aligned} &\min_M Tr\left(K - 2KM + M^TKM\right) + \alpha\|M\|_1 + \beta\ Tr\left(C^TLC\right), \\ &s.t. M \geq 0, \quad diag(M) = 0. \end{aligned} \tag{6}$$

We introduce an auxiliary variable V to make the above objective function separable and solve the following equivalent problem:

$$\begin{aligned} &\min_M Tr\left(K - 2KM + M^TKM\right) + \alpha\|V\|_1 + \beta\ Tr\left(C^TLC\right), \\ &s.t. M \geq 0, \quad diag(M) = 0, \quad V = C. \end{aligned} \tag{7}$$

This can be solved by using the augmented Lagrange multiplier (ALM) type of method. We turn to minimize the following augmented Lagrangian function:

$$\mathcal{L}(V, M, Y) = Tr\left(K - 2KM + M^T KM\right) + \alpha V_1 \tag{8}$$

$$+\beta\ Tr\left(C^T LC\right) + \frac{\mu}{2}\left\| V - M + \frac{Y}{\mu}\right\|_F^2,$$

where $\mu > 0$ is the penalty parameter and Y is the Lagrange multiplier. This problem can be minimized with respect to V, M and Y alternatively, by fixing the other variables.

For S, by letting $N = M - \frac{Y}{\mu}$, it can be updated element-wisely as below

$$S_{ij} = \max\left(\left|N_{ij}\right| - {}^{\alpha}\!/_{\mu}, 0\right) \bullet \operatorname{sign}\left(N_{ij}\right). \tag{9}$$

For M, by letting $F = V + \frac{Y}{\mu}$, it can be updated column wisely as:

$$\min_{M_i} M_i^T\left(\frac{\mu}{2} I + K\right) M_i + \left(\frac{\beta}{2} d_i^T - \mu F_i^T - 2K_{i,:}\right) Z_i, \tag{10}$$

where$d_i \in \mathcal{R}^{n\times 1}$ is a vector with the $j-th$ element d_{ij} being $d_{ij} = \|C_{i,:} - C_{j,:}\|^2$. It is easy to obtain M_i by setting the derivative of Equation (10) w.r.t. M_i to be zero.

Computation of C: With E, M, Q fixed, it is equivalent to solving.

$$\min_C\ \beta\ Tr\left(C^T LC\right) + \gamma\left\| E - CQ\right\|_F^2 \text{ s.t.} C^T C = I. \tag{11}$$

The above problem with an orthogonal constraint can be efficiently solved by the algorithm proposed by Wen and Yin [44].

Computation of Q: With E, M, C fixed, we have

$$\min_Q\ \left\| E - CQ\right\|_F^2 \text{ s.t.} Q^T Q = I. \tag{12}$$

It is the orthogonal Procrustes problem [45], which admits a closed-form solution. The solution is

$$Q = UV^T, \tag{13}$$

where U and V are left and right parts of the SVD decomposition of $E^T C$.

Computation of E: With M, C, Q fixed, the problem becomes

$$\min_E\ \left\| E - CQ\right\|_F^2, \text{s.t.} E \in Idx. \tag{14}$$

Note that $Tr(E^T E) = n$, the above subproblem can be rewritten as below:

$$\max_E Tr\left(E^T CQ\right) \text{ s.t.} E \in Idx. \tag{15}$$

The optimal solution can be easily obtained as follows:

$$E_{i,j} = \begin{cases} 1, & j = argmax_k (PQ)_{ik} \\ 0, & otherwise \end{cases} \tag{16}$$

The updates of M, C, E and Q are coupled with each other, so we could reach an overall optimal solution.

Complexity analysis

With our optimization strategy, the updating of V requires $\mathcal{O}(n^2)$ complexity. The solution of Q involves SVD and its complexity is $\mathcal{O}(ns^2 + s^3)$. To update C, we need $\mathcal{O}(ns^2 + s^3)$. The complexity of E is $\mathcal{O}(ns^2)$. The number of s is often small, so the main computational load arises from solving M. Fortunately, M is solved in parallel.

SC with multiple kernel learning

Although the model in Equation (5) can automatically learn the similarity graph matrix and discrete cluster labels, its performance will strongly depend on the choice of kernels. It is often impractical to exhaustively search for the most suitable kernel. Moreover, real-world datasets are often generated from different sources along with heterogeneous features. The single kernel method may not be able to fully utilize such information. Multiple kernel learning has the ability to integrate complementary information and identify a suitable kernel for a given task. To this end, we present a way to learn an appropriate consensus kernel from a convex combination of a number of predefined kernel functions.

Suppose there are t different kernel functions $\{K^i\}_{i=1}^t$. A Hilbert space can be constructed by the mapping of $\overline{\phi}(x) = \left[\sqrt{\omega_1}\phi_1(x), \sqrt{\omega_2}\phi_2(x), \cdots, \sqrt{\omega_t}\phi_t(x)\right]^T$, with different weights $\omega_i \geq 0$. The combined kernel K_ω can then be represented as [46]

$$K_\omega(x, y) = <\phi_\omega(x), \phi_\omega(y)> = \sum_{i=1}^t \omega_i K^i(x, y). \tag{17}$$

Note that the convex combination of the positive semidefinite kernel matrices $\{K^i\}_{i=1}^r$ is still a positive semidefinite kernel matrix. Thus the combined kernel still satisfies Mercer's condition. Then our proposed method of SC with multiple kernels (SMSC) can be formulated as

$$\min_{M,E,C,Q,\omega} Tr\left(K_\omega - 2K_\omega M + M^T K_\omega M\right) + \alpha M_1$$

$$+\beta\ Tr\left(C^T LC\right) + \gamma\left\| E - CQ\right\|_F^2 \tag{18}$$

$$\text{s.t.} M \geq 0, \operatorname{diag}(M) = 0,$$

$$C^T C = I, Q^T Q = I, E \in Idx,$$

$$K_\omega = \sum_{i=1}^t \omega_i K^i, \sum_{i=1}^t \sqrt{\omega_i} = 1, \omega_i \geq 0.$$

Now above model will learn the similarity graph, discrete clustering labels, and kernel weights by itself. By iteratively updating M, E and ω, each of them will be iteratively refined according to the results of the others.

Optimization

In this part, we show how to solve Equation (18).

***ω* is fixed:** Updated other variables when ω is fixed: We can directly calculate K_ω, and the optimization problem is exactly Equation (5). Thus we just need to use the process of SC with a single kernel with K_ω as the input kernel matrix.

Update *ω*: Optimize with respect to ω when other variables are fixed: solving Equation (18) for ω can be rewritten as [47]

$$\min_\omega \sum_{i=1}^t \omega_i h_i$$

Algorithm 1. The algorithm of SMSK.

Input: A set of kernel matrix $\{K^i\}_{i=1}^t$, parameters $\alpha > 0, \beta > 0, \gamma > 0$, $\mu > 0$.
Initialize: Random matrices M, C and Q. $Y = 0$ and $E = 0$. $\omega_i = 1/r$.

REPEAT
1. Calculate K_ω by Equation (17).
2. Update V according to Equation (9).
3. $V = V - \mathrm{diag}(\mathrm{diag}(V))$ and $V = \max(V, 0)$.
4. Update M according to Equation (10).
5. $Y = Y + \mu(V - Z)$.
6. Update C by solving the problem of Equation (11).
7. Update Q according to Equation (11).
8. Update E according to Equation (16).
9. Calculate h by Equation (20)
10. Calculate ω by Equation (22)
UNTIL stopping criterion is met.

$$\text{s.t.} \sum_{i=1}^{t} \sqrt{\omega_i} = 1, \omega_i \geq 0. \tag{19}$$

where

$$h_i = Tr\left(K^i - 2K^iM + M^TK^iM\right) \tag{20}$$

The Lagrange function of Equation (19) is

$$\mathcal{J}(\omega) = \omega^T h + \gamma\left(1 - \sum_{i=1}^{t} \sqrt{\omega_i}\right). \tag{21}$$

By utilizing the Karush–Kuhn–Tucker (KKT) condition with $\frac{\partial \mathcal{J}(\omega)}{\partial \omega_i} = 0$ and the constraint $\sum_{i=1}^{t} \sqrt{\omega_i} = 1$, we obtain the solution of ω:

$$\omega_i = \left(h_i \sum_{j=1}^{t} \frac{1}{h_j}\right)^{-2}. \tag{22}$$

We can see that ω is closely related to M. Therefore, we could obtain both optimal similarity matrix M and kernel weight ω. We summarize the optimization process of Equation (18) in Algorithm 1.

Evaluation metrics

To evaluate the performance of each clustering method, we selected four clustering criteria including Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Accuracy (ACC), and Purity. ARI is used to compute similarities by considering all pairs of the samples that are assigned in clusters in the current and previous clustering adjusted by random permutation. ACC gives the accuracy of clustering no matter what the actual labeling of any cluster is, as long as the members of one cluster are together. NMI can be information theoretically interpreted. Purity is a simple and transparent evaluation measure. The criteria were calculated for each clustering method based on the prediction labels obtained by SMSC and the real labels provided by the dataset. All criteria scores range from 0 to 1, and the higher values the better performance.

Experiments and results

Kernel design

We designed 11 kernels in multiple kernel learning using common kernels, including Gaussian kernels and polynomial kernels. The form of the Gaussian kernel is $K(x_1, x_2) = \exp(-\frac{x_1 - x_2^2}{(td_{\max}^2)})$; t varies over the set {0.01, 0.5, 1, 10, 50, 100, 1000}. and $d_{\max}$ is the max distance between samples. The definition of the polynomial kernel is $K(x_1, x_2) = (a + x_1^T x_2)^b$, where a ∈ {0, 1} and b ∈ {2, 4}. After generating all of the kernels, we rescaled them to [0,1].

Benchmarking

To benchmark SMSC, we considered seven methods and tools. Four were machine learning methods and three were tools for analyzing scRNA-Seq data. Benchmark methods are implemented by MATLAB, Python or R, which are all publicly available online. We provided the comparison benchmark methods information and download links in Table 2.

The four baselines are in three variants of clustering algorithms that cover the entire representative clustering approaches: (i) Algorithms that require the number of clusters as input, such as K-means [17] and SC [19]. K-means is the most common center-based clustering method. A cluster is a set of objects such that an object in a cluster is closer (more similar) to the 'center' of the cluster than to the center of any other cluster, and each point is assigned to the cluster with the closest centroid. SC is based on graph theory, and it introduces the concept of degree and then uses K-means to cluster after steps of eigenvalue decomposition. (ii) Algorithms such as Greedy [48] that do not require the number of clusters, however, hyperparameter settings should be given. Greedy is a hierarchical clustering algorithm that is adapted to large networks; it can detect high modularity partitions without a limit to the number of nodes. (iii) Algorithms that estimate the number of clusters automatically, i.e. FINCH [49]. FINCH is a fully parameter-free clustering algorithm; it defines a clustering equation to find the nearest neighbor and compute the adjacency matrix. We used the raw scRNA-Seq expression data in SMSC without any preprocessing. For fairness, the same approach is adopted in four machine learning clustering methods.

Three tools for analyzing scRNA-Seq data are SEURAT [11, 12], SINCERA [14] and SNN-Cliq [13]. For SEURAT we used the new Seurat R package version (3.1.5). We performed the same initial normalization. Gene expression values for each cell were divided by the total number of transcripts and multiplied by 10 000. After normalization, we calculated z-scores for downstream dimensional reduction [12]. SINCERA provides both gene-level and cell-level normalizations. For gene-level normalization, per-sample z-score transformation is applied to each expression profile as SINCERA pipeline did [14]. For SNN-Cliq, the package includes Matlab and Python functions. Firstly, SNN-Cliq calculated primary similarity and list k-nearest-neighbors, and then construct similarity graph and find quasi-cliques in SNN graph. Finally, it iteratively merges quasi-cliques and cluster cells [13]. The default parameters are applied in all test tools.

SMSC can accurately predict cell clusters compared with four clustering methods

We began by making clustering comparisons between 11 scRNA-Seq datasets predicted by SMSC and four clustering machine

Table 2. Summary of scRNA-Seq data analysis tools

Tools		Language	Download	Cite
1	Kmeans	Matlab	Call directly in Matlab	10.2307/2346830
2	Spectral	Matlab	https://www.mathworks.com/matlabcentral/fileexchange/26354-spectral-clustering-algorithms	N/A
3	Greedy	Python	http://perso.crans.org/aynaud/communities/	abs/0803.0476
4	FINCH	Matlab/Python	https://github.com/ssarfraz/FINCH-Clustering	10.1109/CVPR.2019.00914
5	SNN-Cliq	Matlab/Python	http://bioinfo.uncc.edu/SNNCliq	10.1093/bioinformatics/btv088
6	SINCERA	R	https://github.com/xu-lab/SINCERA	10.1371/journal.pcbi.1004575;10.1007/978-1-4939-7710-9_15
7	SEURAT	R	https://github.com/satijalab/seurat	10.1038/nbt.4096, 10.1016/j.cell.2019.05.031

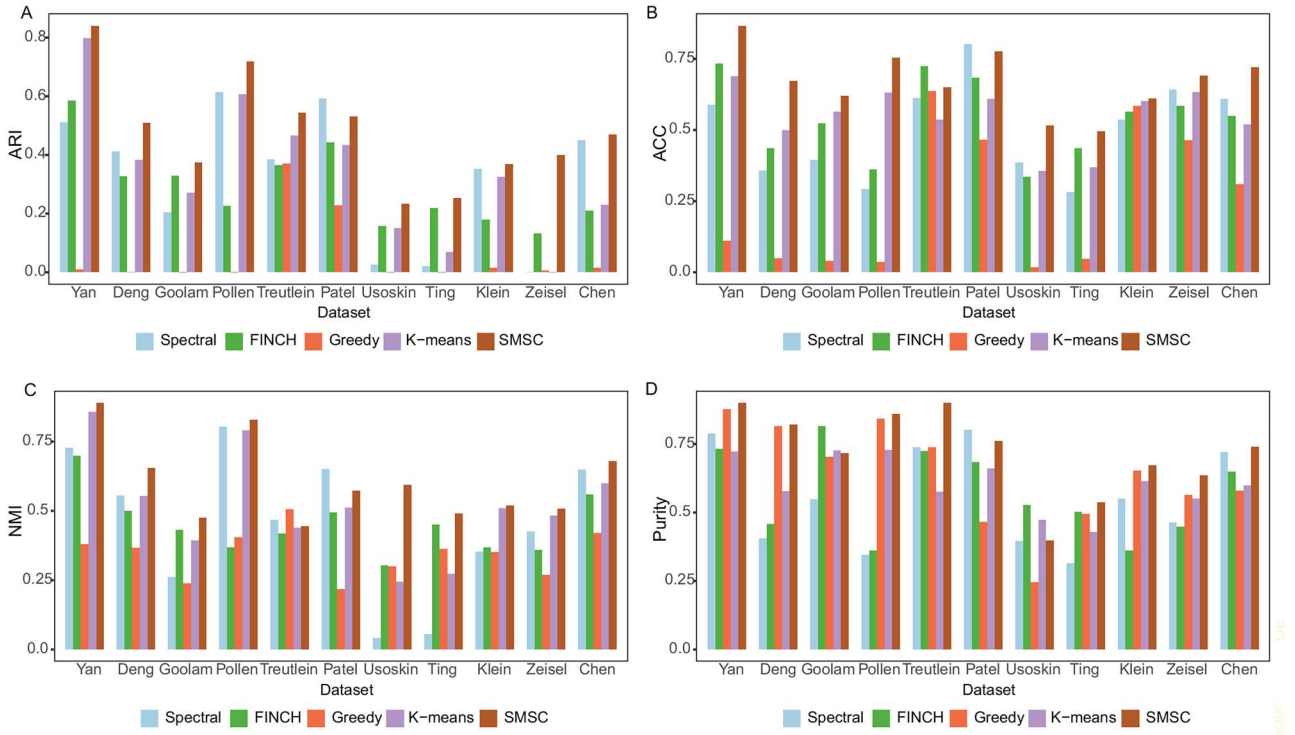

Figure 2. Benchmarking of SMSC against four clustering methods. (A) Comparison of ARI among SMSC and four clustering methods using 11 datasets. (B) Comparison of ACC among SMSC and four clustering methods using 11 datasets. (C) Comparison of NMI among SMSC and four clustering methods using 11 datasets. (D) Comparison of Purity among SMSC and four clustering methods using 11 datasets. Color boxes represent different methods, which were noted below the figure. And y-axis means clustering performances of criteria. The larger the better.

learning methods. The results showed that SMSC improved the clustering performance, covered by the above 11 scRNA-Seq datasets.

First, to assess the clustering performance, we used four criteria (i.e. ARI, ACC, NMI and Purity) by all the five methods in comparison. SMSC showed significant enhancement in cell clustering compared to the clustering methods when using raw data (Figure 2). In general, although the four machine methods showed differences in the results for the 11 datasets, the clustering of expression data showed accuracies that could reach over 80%. Overall, SMSC was the most stable achieving good classification accuracy on all 11 data sets, which may be explained by its multiple kernel fusion mechanisms. For the four machine learning methods, SC was better than the other three methods expect SMSC on five datasets according to the performance of ARI (Figure 2A). The big difference between SC and SMSC results confirms that the multiple kernel learning has a huge influence on the performance of spectral methods. This motivates our multiple kernel fusion model. Although the best results of the three separate steps approach sometimes close to our proposed unified method, their average values are often lower than our method. Besides, among all datasets, Usoskin's and Ting's datasets showed worse results on most methods, possibly because these two datasets contain too much noise, which affected the ability of the algorithms to accurately classify the expression data. Greedy showed the worst result because Greedy is unable to recognize expression data when there is a lot of noise (Figure 2B). On top of that, to address the significance of using the multiple kernel fusion and SC in SMSC, we compare the NMI results on the Klein dataset (Figure 2C). Another test using Purity also showed good performance in the results of SMSC (Figure 2D).

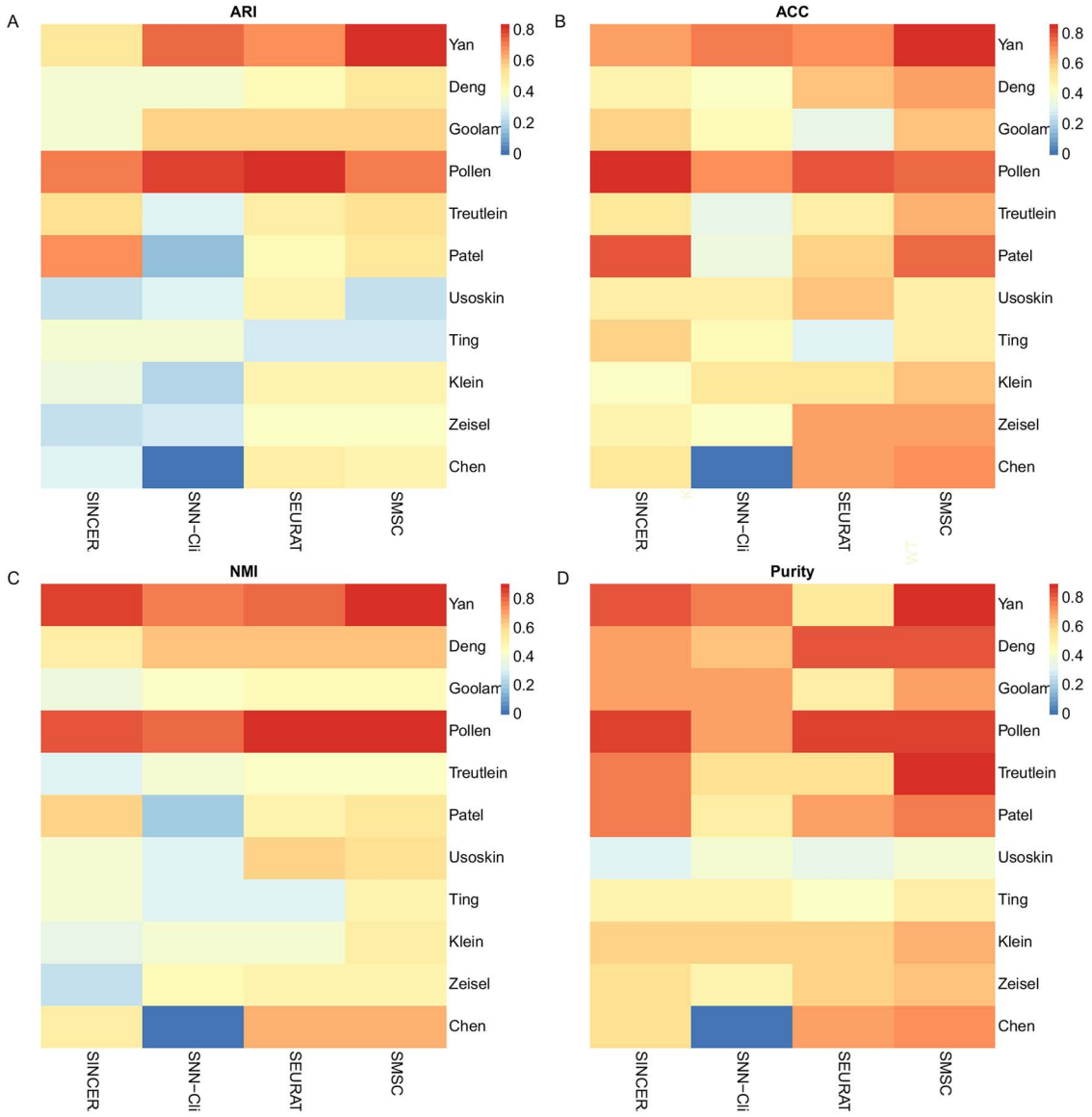

Figure 3. Cell clustering performances of four scRNA-Seq analyzing tools. (A) Comparison of ARI among SMSC and three tools using 11 datasets. (B) Comparison of ACC among SMSC and three tools using 11 datasets. (C) Comparison of NMI among SMSC and three tools using 11 datasets. (D) Comparison of Purity among SMSC and tools using 11 datasets. Additionally, 11 datasets were distributed on the y-axis, and four methods were distributed on the x-axis. The color legend in the upper right of each image represents the clustering performances of the criteria. The redder the color, the better the clustering performance.

SMSC performs better than three analysis tools on scRNA-Seq datasets

Besides the machine learning benchmarks, we continued to evaluate the clustering performance of SMSC and the three tools on the same 11 datasets. The predicted cell labels were systematically evaluated using 4 criteria (Figure 3).

SMSC achieves promising performance in cell cluster prediction on 11 scRNA-Seq datasets, compared to three existing clustering tools. The results of SNN-Cliq is not obtained on the Chen dataset because of the time complexity. SMSC performed better than the benchmark tools. Additionally, SMSC performed better than SNN-Cliq and SEURAT on all but the Ting and Pollen dataset according to the ARI scores (Figure 3A). The results showcase that SMSC clusters cells with greater accuracy and precision (Figure 3B). The NMI of SMSC was greater than the other three clustering methods for all datasets (Figure 3C). Moreover, Purity evaluation, SMSC performs well (Figure 3D). Compared to other clustering models, multiple kernel fusion SC takes advantage of the effective fusion of multiple features to more accurately predict data labels. SMSC could reach an overall optimal solution, and more adeptly reveals the internal nature and regularity of scRNA-Seq data and cluster cells than previous methodologies.

Performance on Yan dataset show that SMSC is insensitive to parameters

To investigate SMSC's parameter sensitivity in scRNA-Seq data, we also extensively studied the parameter selection. We first consider selecting a dataset that can be highly confident in the cell labels. Yan dataset was used as an example to demonstrate the sensitivity of the parameters in our model (Figure 4). We used the quality criteria ARI to address the robustness of SMSC to

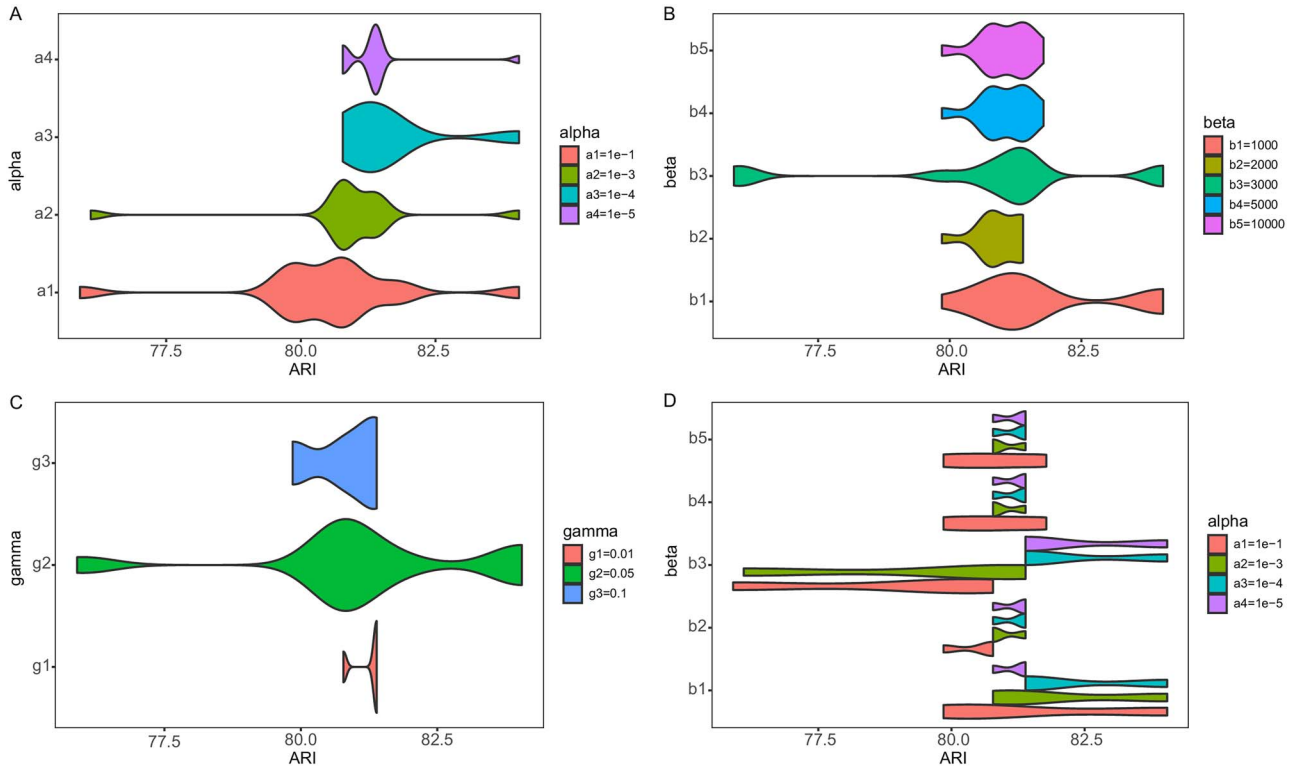

Figure 4. Results of SMSC parameter sensitivity on Yan. (A) Performance of ARI about the parameter alpha on Yan dataset. (B) Performance of ARI about the parameter beta on Yan dataset. (C) Performance of ARI about the parameter gamma on Yan dataset. (D) Performance of ARI among all parameters on Yan dataset. The x-axis represents ARI scores in each figure, and y-axis represents parameters. Different violins represent different parameter values, and the wider the violin, the more parameters correspond to the same ARI.

hyper-parameter choices for a different α, β and γ in the stage of fusing multiple kernels.

We calculated the criterion by running SMSC 10 times and showed averages. Figure 4 demonstrates the clustering results on the Yan dataset. Alpha is selected as 10^{-1}, 10^{-3}, 10^{-4} and 10^{-5}. Gamma is selected as 0.01, 0.05 and 0.1. In each alpha and gamma, the beta was selected as the gradient of 1×10^3, 2×10^3, 3×10^3, 5×10^3, and 10^4.

SMSC is insensitive to alpha over the observed ranges of the parameter values (Figure 4A). Most SMSC values are concentrated between 80 and 82.5. We also tested the effectiveness of involving beta and gamma in the multiple kernel fusion (Figure 4B and C). Using the parameters set in Figure 4 shows tests in the different constraints of SMSC. Smaller alpha usually leads to better results if the choice of beta is not too large (Figure 4D). These results demonstrate that SMSC is robust in selecting hyperparameters and that it does not significantly affect clustering quality. Results in Figure 4 proves the effectiveness and stability of the SMSC algorithm. Additionally, the results presented in the paper were obtained using fixed parameter values as $\alpha = 10^{-5}$, $\beta = 1\times10^3$ and $\gamma = 0.05$.

Conclusion

Several fundamental problems are existing in most classical SC algorithms to explore cell clusters in noisy scRNA-Seq data. The key innovations of SMAC are incorporating SC, K-means, together with integrating multiple kernels learning in an iterative process to directly discover groupings in scRNA-Seq data analysis. The benefit of multiple kernel fusion is its intrinsic learnable properties of fusing attributes to capture relationships across the whole cell–cell relationship. Hence, the learned represented kernel can be treated as the high-order representations of cell–cell relationships in scRNA-Seq data.

Unlike the previous kernel learning and SC applications in scRNA-Seq data analysis, SMSC automatically learns similarity information from scRNA-Seq data and transforms the candidate solution into a new one that better approximates the discrete solution. Besides, we discuss the necessity of combining multiple kernel learning. Processes in SMSC can help identify biologically meaningful cell–cell relationships as they apply to our framework and eventually, they are proven capable of enhancing performance. Extensive experiments on 11 real datasets demonstrated the promising performance of our methods compared to existing clustering approaches.

Some limitations can still be found in SMSC. (i) It is prone to achieve better results with small datasets, compared to relatively large datasets,as it is designed to learn better representations with many genes and not so many cells from scRNA-Seq data, as shown in the benchmark results and (ii) For large-scale datasets (e.g. Chen,more than 14 000 cells), compared with statistics model-based methods, the computing and fusing of multiple kernel matrices framework need more computational resources, which is more time-consuming. In the future, we will investigate creating a more efficient SMSC model with a lighter and more compressed architecture.

In the future, we will continue to enhance SMSC by incorporate deep neural networks to obtain more useful features. It may improve both the accuracy and robustness of predictions regarding cell–cell interactions. We plan to develop a more user-friendly software system from our SMSC model, together with interactive visualizations.

Key Points

- A unified clustering learning framework is proposed for use with scRNA-seq data. Rather than using predefined similarity metrics, the similarity graph is adaptively learned from the data in kernel space. We combine similarity learning with subsequent clustering into a unified framework, we can ensure the optimality of the learned similarity graph.
- The proposed method is formulated as a multiple kernel model. Sparse single-cell data can be mapped by a kernel function to obtain a non-sparse kernel matrix. Unlike existing spectral clustering methods that work in three separate steps, we simultaneously three subtasks, which were showed in the paper.
- Experiments on gold-standard and silver-standard scRNA-seq datasets show that our method achieves superior performance compared with state-of-the-art methods.

Authors' contributions

R.Q., L.X., F.G. and Q.Z. designed this study and developed the algorithm. R.Q. and J.W. made a detailed implementation and performed the data analysis. R.Q. and J.W. wrote this manuscript. Both authors read and approved the final manuscript.

Conflict of Interest

The authors declare that they have no conflict of interest with any organization in the subject matter or materials discussed in this manuscript.

Funding

The National Natural Science Foundation of China (Grant Nos. 61922020, 61771331 and 61902259); the Natural Science Foundation of Guangdong Province (No. 2018A0303130084).

References

1. Shalek AK, Satija R, Adiconis X, *et al*. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 2013;**498**(7453):236–40.
2. Petegrosso R, Li Z, Kuang R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Briefings in bioinformatics*, 2020;**21**(4):1209–23.
3. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications*, 2018;**9**(1):1–9.
4. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome biology*, 2016;**17**(1):63.
5. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 2015;**16**(3):133–45.
6. Li X, Chen W, Chen Y, *et al*. Network embedding-based representation learning for single cell RNA-seq data. *Nucleic acids research*, 2017;**45**(19):e166–6.
7. Xu Y, Zhou X. Applications of single-cell sequencing for Multiomics. *Methods Mol Biol* 2018;**1754**:327–74.
8. Yang J, Gruenewald S, Wan X-F. Quartet-net: a quartet-based method to reconstruct phylogenetic networks. *Mol Biol Evol* 2013;**30**(5):1206–17.
9. Yang JL, Grünewald S, Xu Y, *et al*. Quartet-based methods to reconstruct phylogenetic networks. *BMC Syst Biol* 2014;**8**(1):21.
10. Wang Y, Zhang X-S, Chen LN. Systems biology intertwines with single cell and AI. *BioMed Central* 2019:1–3.
11. Satija R, Farrell JA, Gennert D, *et al*. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 2015;**33**(5):495–502.
12. Butler A, Hoffman P, Smibert P, *et al*. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 2018;**36**(5):411–20.
13. Xu C, Su ZJB. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 2015;**31**(12):1974–80.
14. Guo M, Wang H, Potter S,*et al*. SINCERA: a pipeline for single-cell RNA-Seq profiling analysis. *PLoS computational biology*, 2015;**11**(11):e1004575.
15. Sarfraz MS, Sharma V, Stiefelhagen R. *Efficient Parameter-free Clustering Using First Neighbor Relations*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, 8934–43.
16. Ishioka, T. Extended K-means with an Efficient Estimation of the Number of Clusters. In: *Seventeenth International Conference on Machine Learning*. 2000.
17. Hartigan JA, Wong MA. Algorithm AS 136: a K-means clustering algorithm. *J R Stat Soc* 1979;**28**(1):100–8.
18. Yau C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics* 2016;**17**(1):140.
19. Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. *Adv Neural Information Processing Sys* 2002.
20. Yang, B, Fu, X, Sidiropoulos, ND, *et al*. Towards k-means-friendly spaces: simultaneous deep learning and clustering. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017. JMLR.org.
21. Li G, Ma Q, Tang H, *et al*. QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, 2009;**37**(15):e101–1.
22. Ma A, Sun M, McDermaid A, *et al*. MetaQUBIC: a computational pipeline for gene-level functional profiling of metagenome and metatranscriptome. *Bioinformatics*, 2019;**35**(21):4474–7.
23. Xie J, Ma A, Zhang Y, *et al*. QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data. *Bioinformatics*, 2020;**36**(4):1143–9.
24. Jiang H, Sohn LL, Huang H, *et al*. Single cell clustering based on cell-pair differentiability correlation and variance analysis. *Bioinformatics* 2018;**34**(21):3684–94.
25. Kiselev VY, Kirschner K, Schaub MT, *et al*. SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*, 2017;**14**(5):483–6.
26. Yan L, Yang M, Guo H, *et al*. Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat Struct Mol Biol* 2013;**20**(9):1131.
27. Goolam M, Scialdone A, Graham SJL, *et al*. Heterogeneity in Oct4 and Sox2 targets biases cell fate in 4-cell mouse embryos. *Cell* 2016;**165**(1):61–74.
28. Deng Q, Ramsköld D, Reinius B, *et al*. Single-cell RNA-Seq reveals dynamic, random Monoallelic gene expression in mammalian cells. *Science* 2014;**343**(6167):193–6.

29. Pollen AA, Nowakowski TJ, Shuga J, *et al*. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat Biotechnol* 2014;**32**(10):1053.
30. Treutlein B, Brownfield DG, Wu AR, *et al*. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature* 2014;**509**(7500):371–75.
31. Ting DT, Wittner BS, Ligorio M, *et al*. Single-cell RNA sequencing identifies extracellular matrix gene expression by pancreatic circulating tumor cells. *Cell Rep* 2014;**8**(6): 1905–18.
32. Patel AP, Tirosh I, Trombetta JJ, *et al*. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* 2014;**344**(6190):1396–401.
33. Usoskin D, Furlan A, Islam S, *et al*. Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat Neurosci* 2015;**18**(1):145–53.
34. Zeisel A, Muñoz-Manchado AB, Codeluppi S, *et al*. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 2015;**347**(6226):1138–42.
35. Klein AM, Mazutis L, Akartuna I, *et al*. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 2015;**161**(5):1187–201.
36. Cheng B, Yang J, Yan S, *et al*. Learning with ℓ^1-graph for image analysis. *IEEE transactions on image processing*, 2009;**19**(4):858–66.
37. Huang J, Nie F, Huang H. A new simplex sparse learning model to measure data similarity for clusterin. In: *Twenty-Fourth International Joint Conference ong. Artificial Intelligence*. 2015.
38. Kang Z, Peng C, Cheng Q. Kernel-driven similarity learning. *Neurocomputing*, 2017;**267**:210–9.
39. Elhamifar, E. and R. Vidal. Sparse subspace clustering. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009:2790–7.
40. Zhang C, Nie F, Xiang S. A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomputation* 2010;**74**(4–6):959–67.
41. Stella XY, Shi J. *Multiclass Spectral Clustering. In null*. IEEE, 2003.
42. Mohar B, Alavi Y, Chartrand G, *et al*. The Laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 1991;**2**(871–898):12.
43. Fan K. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences of the United States of America*, 1949;**35**(11):652.
44. Wen Z, Yin W. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 2013;**142**(1–2):397–434.
45. Schönemann PH. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 1966;**31**(1):1–10.
46. Zeng H, Cheung Y. Feature selection and kernel learning for local learning-based clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2010;**33**(8):1532–47.
47. Cai, X, Nie, F, Cai, W, *et al*. Heterogeneous image features integration via multi-modal semi-supervised learning model. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013:1737–44.
48. Blondel VD, Guillaume JL, Lambiotte R, *et al*. Fast unfolding of community hierarchies in large networks. *J Stat Mech* 2008(10):P10008.
49. Sarfraz, S, Sharma, V, Stiefelhagen, R. Efficient parameter-free clustering using first neighbor relations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019:8934–43.