Bioinformatics, 37(19), 2021, 3235–3242
doi: 10.1093/bioinformatics/btab276
Advance Access Publication Date: 7 May 2021
Original Paper

Gene expression

Clustering single-cell RNA-seq data by rank constrained similarity learning

Qinglin Mei[1,2], Guojun Li [1,2,3,*] and Zhengchang Su [4,*]

[1]Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Jinan 250100, China, [2]School of Mathematics, Shandong University, Jinan 250100, China, [3]School of Mathematical Science, Liaocheng University, Liaocheng 252000, China and [4]Department of Bioinformatics and Genomics, The University of North Carolina at Charlotte, Charlotte, NC 28223, USA

*To whom correspondence should be addressed.

Associate Editor: Anthony Mathelier

Received on August 5, 2020; revised on April 19, 2021; editorial decision on April 20, 2021; accepted on April 22, 2021

Abstract

Motivation: Recent breakthroughs of single-cell RNA sequencing (scRNA-seq) technologies offer an exciting opportunity to identify heterogeneous cell types in complex tissues. However, the unavoidable biological noise and technical artifacts in scRNA-seq data as well as the high dimensionality of expression vectors make the problem highly challenging. Consequently, although numerous tools have been developed, their accuracy remains to be improved.

Results: Here, we introduce a novel clustering algorithm and tool RCSL (Rank Constrained Similarity Learning) to accurately identify various cell types using scRNA-seq data from a complex tissue. RCSL considers both local similarity and global similarity among the cells to discern the subtle differences among cells of the same type as well as larger differences among cells of different types. RCSL uses Spearman's rank correlations of a cell's expression vector with those of other cells to measure its global similarity, and adaptively learns neighbor representation of a cell as its local similarity. The overall similarity of a cell to other cells is a linear combination of its global similarity and local similarity. RCSL automatically estimates the number of cell types defined in the similarity matrix, and identifies them by constructing a block-diagonal matrix, such that its distance to the similarity matrix is minimized. Each block-diagonal submatrix is a cell cluster/type, corresponding to a connected component in the cognate similarity graph. When tested on 16 benchmark scRNA-seq datasets in which the cell types are well-annotated, RCSL substantially outperformed six state-of-the-art methods in accuracy and robustness as measured by three metrics.

Availability and implementation: The RCSL algorithm is implemented in R and can be freely downloaded at https://cran.r-project.org/web/packages/RCSL/index.html.

Contact: guojunsdu@gmail.com or zcsu@uncc.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Recent advances in single-cell RNA sequencing (scRNA-seq) technologies have revolutionized the study of many important biological processes, such as embryogenesis and tumorigenesis, in which an understanding of the functions and composition of heterogeneous cell types in the tissues is critical. As the transcriptome of a cell largely determines its molecular makeup, and thus its functions and cellular type, unsupervised clustering of individual cells based on their transcriptomes can be a powerful approach to identifying all the cell types including rare ones in complex tissues in an unbiased manner (Buettner *et al.*, 2015; Jiang *et al.*, 2016; Xu and Su, 2015). Despite great progress made in the last few years, the task remains highly challenging owing to the unavoidable biological noise and technical artifacts in scRNA-seq data as well as the high dimensionality of expression vectors.

The biological noise is related to the inherently stochastic nature of gene transcription in individual cells of the same type, due to the small copy number of molecules involved, unsynchronized cell cycles and uneven cell divisions (Becskei *et al.*, 2005; Kaern *et al.*, 2005; Paulsson, 2004; Raj and van Oudenaarden, 2008). As a result, even different cells of the same type display a broad range of variation in RNA levels (Bar-Even *et al.*, 2006; Newman *et al.*, 2006; Raj *et al.*, 2008; Taniguchi *et al.*, 2010; Xie *et al.*, 2008; Young *et al.*, 2012; Zenklusen *et al.*, 2008). Technical artifacts result from dropout events and batch factors in data generation, which are different from the conventional bulk RNA-seq data (Kiselev *et al.*, 2019). Intense efforts have been made to address these challenges over the past few years. For example, we previously proposed a quasi-clique-based algorithm with shared nearest neighbor (SNN), SNN-cliq, to identify groups of highly similar cells (Xu and Su, 2015). Guo *et al.* (2015) designed a top-to-toe pipeline (SINCERA) to distinguish major cell types. Seurat, which has become one of the most popular choices for scRNA-seq data analysis, combines SNN graphs with Louvain community detection to group cells iteratively (Satija *et al.*, 2015). SC3 integrates the results of multiple clustering methods to obtain a consensus result (Kiselev *et al.*, 2017). Additionally, dimensionality reduction has also been integrated into clustering methods, such as pcaReduce (Žurauskienė and Yau, 2016) and ZIFA (Pierson and Yau, 2015), to reduce the computational complexity. Meanwhile, some approaches like CIDR have been proposed to mitigate the impact of dropout events by improving the dissimilarity matrix (Lin *et al.*, 2017).

Another challenge in accurately clustering cells using scRNA-seq data are related to their high dimensionality. Although a cell may express tens of thousands of genes, only few of them determine its type (Graf and Enver, 2009), and we usually have no prior knowledge of which specific genes determine cell types. To address this, many new vector similarity metrics have been proposed such as SIMLR (Bo *et al.*, 2017) and MPSSC (Park *et al.*, 2018). SIMLR obtains a similarity matrix and identifies clusters via multikernel learning. MPSSC clusters a learned multiple doubly stochastic similarity matrix using sparse spectral clustering. In both the algorithms, a similarity matrix was learned from the data to better capture global structural relationships between cells. Nevertheless, these similarity metrics do not take into account local structures in quantifying cell similarities, which can be critical to discern subtle differences between cells of the same type and cells of different types, as indicated by our earlier proposed SNN metric (Xu and Su, 2015). In addition, most graph-based methods use the *k*-means algorithm to find clusters in a post-process step after constructing the similarity graph. Such two-stage algorithms inevitably lose some information from the original data in the post-process step.

In this work, we first propose a new metric that considers both global and local similarities between the expression vectors. Specifically, we use Spearman's rank correlation (SRC) to measure the global similarity and neighbor representation (NR) to capture the local similarity between the cells. NR has been successfully used in many dimensionality reduction algorithms, such as locally linear embedding (Roweis and Saul, 2000), to capture local structures. We adaptively learn NR for each cell using an optimization procedure. Our metric is an optimized linear combination of Spearman's correlation and the learned NR. Once the similarity matrix between cells is computed using this metric, the task of clustering the cells according to their types is to identify all block-diagonal submatrices in the similarity matrix or to partition the cognate similarity graph into several connected components, where each block-diagonal submatrix or connected component corresponds to a cell type. It has been proved that the number of block-diagonal submatrices in the similarity matrix is equal to the number of zero eigenvalues of its Laplacian matrix (Luxburg, 2007; Mohar, 1991; Nie *et al.*, 2016). Therefore, we posit that if we can reliably estimate the number of clusters or cell types C based on the similarity matrix, then we can transform it into a new matrix containing C block-diagonal submatrices, or partition the cognate similarity graph into C connected components, by constraining the rank of its Laplacian matrix. More specifically, if we can construct a matrix such that the number of zero eigenvalues of its Laplacian matrix is exactly equal to the estimated number of cell types, then we can divide cells into the same number of groups/types.

Based on this idea, we have developed a novel clustering algorithm RCSL (Rank Constrained Similarity Learning) by constructing a block-diagonal matrix such that the number of zero eigenvalues of its Laplacian matrix is equal to the estimated number of cell types in the dataset and its distance to the similarity matrix is minimized. When tested on 16 public scRNA-seq datasets with verified cell types, RCSL generally outperforms the six state-of-the-art methods compared.

2 Materials and methods

2.1 The RCSL algorithm

RCSL takes a single-cell gene expression matrix $\boldsymbol{M}_{G'\times N}$ as the input, where G' denotes the number of genes, N denotes the number of cells and an element m_{ij} in $\boldsymbol{M}$ represents the expression value of gene i in cell j. RCSL consists of three steps (Fig. 1) detailed as follows.

Step 1. Data preprocessing. To control the quality of scRNA-seq data, we filter out rarely expressed genes and ubiquitously expressed genes, which contribute little to clustering. Specifically, we discard genes expressed in <2.5% of cells as well as genes expressed in >97.5% of cells with a variance <90% of the mean variance of these selected ubiquitously expressed genes. These parameters were chosen as we found that they could slightly improve the accuracy of clustering results on most of the datasets (Supplementary Fig. S1). Let the resulting matrix be $\boldsymbol{M}_{G\times N}$.

Step 2. Construction of similarity matrix. For each pair of cells i and j in $\boldsymbol{M}_{G\times N}$, we calculate the SRC between their expression vectors, defined as:

$$S_S(i,j) = \frac{\sum_{g=1}^{G}(R_i(g) - -R_i)(R_j(g) - -R_j\big)}{\sqrt{\sum_{g=1}^{G}(R_i(g) - -R_i)^2 \sum_{g=1}^{G}(R_j(g) - -R_j)^2}}, \quad (1)$$

where $R_i(g)$ and $R_j(g)$ are the ranks of the expression level of gene g in cells i and j, respectively, and $-R_i$ and $-R_j$ are the ranks in i and j of the gene in the middle ordered by gene IDs, respectively. Let the resulting matrix be $\boldsymbol{S}_S$ (Fig. 1). As SRC is based on the ranks of values, it is insensitive to the difference of cell sizes. We perform principal component analysis (PCA) on $\boldsymbol{M}_{G\times N}$ (Wold *et al.*, 1987), and keep the top Q PCs that explain 95% of the variance. Let the resulting matrix be $\boldsymbol{X}_{Q\times N}$ (Fig. 1). We found that such dimensionality reduction has little influence on the accuracy of clustering (Supplementary Fig. S1) but speeds up the algorithm somewhat

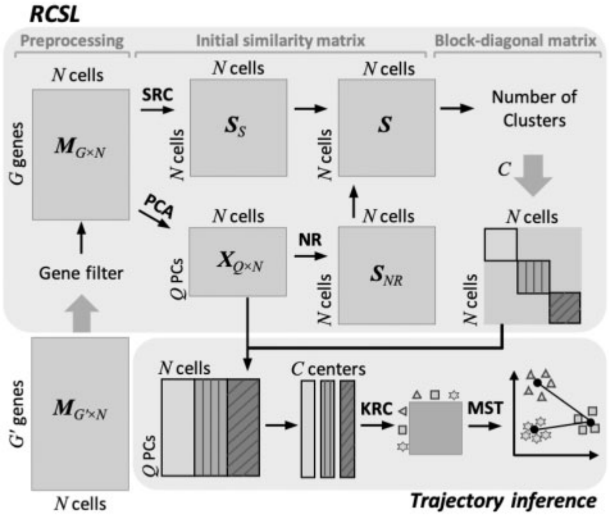

Fig. 1. Overview of RCSL. First, given a gene expression matrix $\boldsymbol{M}_{G'\times N}$ as the input, RCSL filters out non-informative genes, resulting in matrix $\boldsymbol{M}_{G\times N}$. Second, based on $\boldsymbol{M}_{G\times N}$, RCSL computes SRC matrix $\boldsymbol{S}_S$ between the cells, performs PCA on the genes, and preserves the top Q-PCs as matrix $\boldsymbol{X}$. Third, RCSL learns the NR matrix S_{NR} based on $\boldsymbol{X}$. Fourth, RCSL computes the similarity matrix $\boldsymbol{S}$ using a linear combination $\boldsymbol{S}_S$ and $\boldsymbol{S}_{\mathrm{NR}}$. Finally, RCSL estimates the number of clusters C in $\boldsymbol{S}$, and learns the block-diagonal matrix $\boldsymbol{B}$ from $\boldsymbol{S}$, with the constraint that $\boldsymbol{B}$ has C zero eigenvalues. Each block-diagonal submatrix in $\boldsymbol{B}$ is a cluster of cells. RCSL also infers trajectories of the identified cell clusters/types

(Supplementary Fig. S2). Based on $\boldsymbol{X}_{Q\times N}$, we compute the NR for each pair of cells as their local similarity as follows.

For each cell i, we find k-nearest-neighbors (k-NNs) using the Euclidean or cosine angle distance between their feature vectors in $\boldsymbol{X}_{Q\times N}$. We use the Euclidean distance, as it generally performs better than the cosine angle distance in these study (see below). By default, k is set to $0.65\,N$, which is a relatively large value to reduce the risk of information loss. Since cells of the same type are not necessarily close neighbors in Euclidean space as only few genes determine a cell's type, in order to identify the close neighbors of the cells, we model each cell's feature vector as a linear combination of the vectors of its neighbors; then those with higher weights are its closer neighbors. To find the weights, we solve a least-squares optimization problem:

$$\min_{s_{ij}^{NR}} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \sum_{j\in KNN(i)} s_{ij}^{\mathrm{NR}} \boldsymbol{x}_j||^2 \mathrm{s.t.} \sum_{j=1}^{N} s_{ij}^{\mathrm{NR}} = 1, \tag{2}$$

where $\boldsymbol{x}_i$ is the feature vector of cell i in $\boldsymbol{X}_{Q\times N}$, $KNN(i)$ the k-NNs of i in Euclidean space, and $\mathrm{s}_{\mathrm{ij}}^{\mathrm{NR}}$ the weight of cell j on i [$\mathrm{s}_{\mathrm{ij}}^{\mathrm{NR}} = 0$ if $j \notin KNN(i)$]. Intuitively, the greater the $\mathrm{s}_{\mathrm{ij}}^{\mathrm{NR}}$ value, the more similar j is to i. Therefore, we call the learned weight vector $\mathrm{s}_{\mathrm{i.}}^{\mathrm{NR}}$the NR of i. Let the resulting matrix for all the cells be $\boldsymbol{S}_{\mathrm{NR}} = \{\mathrm{s}_1^{\mathrm{NR}},\mathrm{s}_2^{\mathrm{NR}},\ldots,\mathrm{s}_{\mathrm{N}}^{\mathrm{NR}}\}$. Then, we define the similarity matrix between the cells as a linear combination of $\boldsymbol{S}_S$ and $\boldsymbol{S}_{\mathrm{NR}}$ (Fig. 1)

$$\mathrm{S} = \gamma \mathrm{S_s} + \lambda \mathrm{S_{NR}} \mathrm{s.t.} \gamma + \lambda = 1, \tag{3}$$

where $\gamma \geq 0$ and $\lambda \geq 0$ are scalar parameters that balance the contribution of $\boldsymbol{S}_S$ and $\boldsymbol{S}_{\mathrm{NR}}$ in $\boldsymbol{S}$. By default, we set $\gamma = 0.8$ and $\lambda = 0.2$, as they generally perform best among other choices tested (see below).

Step 3. Calculation of block-diagonal matrix. We first estimate the number of clusters C by hierarchically clustering the cells based on $\boldsymbol{S}$, and find C that yields the largest Krzanowski–Lai index (Krzanowski and Lai, 1988) value from a range of C (by default, 4–12). However, for small datasets ($N < 3000$), we use a two-step strategy to more accurately estimate C. Specifically, we choose three C values with the largest Krzanowski–Lai indexes, and pick the one with the largest sum of intra-class similarities based on $\boldsymbol{S}$ among the three clustering results of RCSL. The hierarchical clustering is performed using the R package *NbClust* with default settings.

To construct the block-diagonal similarity matrix $\boldsymbol{B}_{N\times N}$ between the cells, we adopt the Constrained Laplacian Rank (CLR) procedure (Nie *et al.*, 2010, 2016). Briefly, CLR defines a diagonal matrix $\boldsymbol{D}_B = \mathrm{diag}\ (d_{11},\ d_{22},\ \ldots,\ d_{NN})$, where $\mathrm{d_{ii}} = \sum_{j=1}^{N} \frac{\mathrm{b_{ij}+b_{ji}}}{2}$, and b_{ij} is the similarity between cell i and cell j in $\boldsymbol{B}$. The Laplacian matrix of $\boldsymbol{B}$ is defined as $\boldsymbol{L}_B = \boldsymbol{D}_B - \frac{\mathrm{B^T+B}}{2}$. An important property of the Laplacian matrix is that the number of its zero eigenvalues equals the number of connected components in the graph defined by $\boldsymbol{B}$ (Chung, 1997; Mohar, 1991). Therefore, if a similarity matrix $\boldsymbol{B}$ can be found, such that the rank of its Laplacian matrix is exactly $N - \mathrm{C}$, then $\boldsymbol{B}$ will have approximately C block-diagonal submatrix with proper permutations, and the corresponding similarity graph will contain C connected components. Each block-diagonal submatrix and corresponding connected component form a cluster of cells. Ideally, $\boldsymbol{B}$ should be highly similar to $\boldsymbol{S}$, and the rank of $\boldsymbol{L}_B$ is exactly $N - \mathrm{C}$. CLR therefore finds $\boldsymbol{B}$ by minimizing the difference between $\boldsymbol{B}$ and $\boldsymbol{S}$, with the constraint that the rank of $\boldsymbol{L}_B$ is $N - \mathrm{C}$;

$$\min_{\boldsymbol{B}} ||\boldsymbol{B} - \boldsymbol{S}||_1 \mathrm{s.t.} \sum_{j=1}^{N} b_{ij} = 1,\ b_{ij} \geq 0, \mathrm{rank}(\boldsymbol{L}_B) = N - C, \tag{4}$$

where the sum of each row in $\boldsymbol{B}$ is constrained to 1 to avoid rows of all zeros in $\boldsymbol{B}$ (see details in Supplementary Note).

2.2 Time complexity of the algorithm

Given the expression matrix $\boldsymbol{M}_{G'\times N}$, with N cells and G' genes, Step 1 needs $N{*}G'$ calculations, so it runs at $\mathrm{O}(N)$. In Step 2, we sequentially compute the SRC matrix $\boldsymbol{S}_S$ and PCA matrix $\boldsymbol{X}_{Q\times N}$, find k-NNs of N cells, and learn the $\boldsymbol{S}_{\mathrm{NR}}$ matrix, each of these procedures has a time complexity of $\mathrm{O}(N^2)$. In Step 3, we perform T iterations (by default, $T = 30$) on the $N\times N$ matrix $\boldsymbol{S}$ to estimate diagonal matrix $\boldsymbol{B}$, which needs $\mathrm{O}(TN^2)$ calculations. As T is a small constant, Step 3 runs at $\mathrm{O}(N^2)$. Therefore, the time complexity of the RSCL algorithm is $\mathrm{O}(N) + 5\mathrm{O}(N^2) = \mathrm{O}(N^2)$ (Supplementary Fig. S3).

2.3 Inference of trajectory and pseudo-time

Based on the clustering results of RCSL, we infer the developmental trajectories and pseudo-temporal ordering of cells for time-series scRNA-seq data. For each identified cluster of cells, we compute its center as the mean of the feature vectors from one cluster in $\boldsymbol{X}_{Q\times N}$ (Fig. 1), and the Kendall rank correlation (KRC) among all the centers. We construct a weighted similarity graph $\boldsymbol{G}$, where the vertices represent the centers and the edges represent their KRC values. We find the minimum spanning tree (MST) using the Prims' algorithm (Prim, 1957). The MST that represents the shortest path connecting all the centers without any circles is the most parsimonious explanation of the relationships among the cell types during cell differentiation, and thus likely reflects their developmental trajectory. We determine the pseudo-temporal ordering between the cell types by using the distance from a cell type to the predefined starting cell type. The distance is defined as the reciprocal of the average similarity between the two types of cells in the similarity matrix $\boldsymbol{S}$.

2.4 ScRNA-seq datasets

We collected 16 publicly available scRNA-seq datasets (Table 1), in which cell types were determined by the original authors using various methods, including microscopic inspections for the embryonic cells (oocyte, zygote, 2-cell stage, 4-cell stage, 8-cell stage, … and blast cells), time of post-inductions for artificially induced differentiated cells (day 0, day 1, day 2, …) as well as molecular markers and cell purification using Fluorescence-Activated Cell Sorting for the other datasets (for details see Supplementary Table S1). To further ensure the accuracy of cell type annotations, we excluded cells with ambiguous labels including 'dropped' cells in the Wang dataset, 'contaminated' cells in the Xin dataset and 'unclear' cells in the Muraro dataset. We normalize raw read counts of genes using counts per million followed by adding a pseudocount of 1 and log (base 2) transformation.

2.5 Simulation datasets

We generated 10 simulated datasets containing 300–3000 cells belonging to 4–7 cell types (Supplementary Table S3) using the *Splatter* Bioconductor package (Zappia *et al.*, 2017). Each cell expresses 10 000–15 000 genes, whose levels are determined by the cell's type. The script for constructing simulation datasets is available at https://github.com/QinglinMei/RCSL/tree/master/R.

2.6 Evaluation metrics

To measure the consistency between identified clusters and known cell types, we adopt three metrics: Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), Normalized Mutual Information (NMI) (Strehl and Ghosh, 2002) and Fowlkes–Mallows index (FM) (Fowlkes and Mallows, 1983). We represent the known cell types as R and the identified clusters as E. Let a be the number of pairs of cells that are clustered in the same group in both R and E; b the number of pairs of cells that are clustered in the same group in R but in different groups in E; c the number of pairs of cells that are clustered in different groups in R but in the same group in E and d the number of pairs of cells that are clustered in different groups both in both R and E (Kim *et al.*, 2019). Then, ARI, FM and NMI are defined as,

$$\mathrm{ARI}(R,\ E) = \frac{2(\mathrm{ad} - \mathrm{bc})}{(a+b)(b+d) + (a+c)(c+d)}, \tag{6}$$

$$\mathrm{FM}(R,\ E) = \sqrt{\left(\frac{a}{a+b}\right)\left(\frac{a}{\mathrm{a}+c}\right)} \text{and} \tag{7}$$

Table 1. Summary of the 16 scRNA-seq datasets used in this study to assess the performance of the methods for clustering cells

Accession ID	Dataset	Tissue	# Cells	# Genes	# Cell types	Protocol	Reference
GSE57249	Biase	Mouse Embryos	56	25 734	4	SMARTer	Biase *et al.* (2014)
GSE52583	Treutlein	Mouse Tissues	80	23 271	5	SMARTer	Treutlein *et al.* (2014)
GSE36552	Yan	Human Embryos	90	20 214	6	Tang	Yan *et al.* (2013)
GSE51372	Ting	Mouse Pancreas	114	14 450	5	Tang	Ting *et al.* (2014)
E-MTAB-3321	Goolam	Mouse Embryos	124	41 480	5	Smart-Seq2	Mubeen *et al.* (2016)
GSE45719	Deng	Mouse Embryos	268	22 431	6	Smart-Seq2	Deng *et al.* (2014)
GSE98664	Hayashi	Mouse Embryos	414	23 658	5	RamDA-seq	Hayashi *et al.* (2018)
GSE83139	Wang	Human Pancreas	457	19 950	7	SMARTer	Wang *et al.* (2016)
GSE67835	Darmanis	Human Brain	466	20 214	9	SMARTer	Darmanis *et al.* (2015)
E-MTAB-3929	Petropoulos	Human Preimplantation Embryos	1289	8772	5	Smart-Seq2	Petropoulos *et al.* (2016)
GSE81608	Xin	Human Pancreas	1492	39 851	8	SMARTer	Xin *et al.* (2016)
GSE85241	Muraro	Human Pancreas	2122	19 140	10	CEL-Seq2	Muraro *et al.* (2016)
GSE65525	Klein	Mouse Embryo Stem Cells	2717	24 175	4	inDrop	Klein *et al.* (2015)
GSE74672	Romanov	Mouse Brain	2881	24 341	7	Drop-seq	Romanov *et al.* (2017)
GSE60361	Zeisel	Mouse Brain	3005	19 972	9	STRT-seq UMI	Zeisel *et al.* (2015)
SRP073767	PBMC4K	Human	4292	58 302	11	10xGenomics Chromium	Zheng *et al.* (2017)

$$\mathrm{NMI}(R, E) = \frac{\mathrm{MI}(R, E)}{[H(R) + H(E)]/2}, \tag{8}$$

where MI(R, E) is the mutual information of R and E, and H an entropy function of R and E.

3 Results

3.1 Combination of global and local similarities improves the accuracy of RCSL

To find the optimal values of weights γ and λ of global similarity and local similarity, respectively, in the similarity metric S [formula (3)], and to see how they affect the accuracy of RCSL, we ran RCSL on the 16 datasets with varying vales of γ (0.0, 0.1, …, 1.0) and $\lambda = 1 - \gamma$ (1.0, 0.9, …, 0.0). As shown in Figure 2, with the increase in γ, that is, decrease in λ, the accuracy of RCSL generally increases, and reaches the highest level at $\gamma = 0.8$, $\lambda = 0.2$, then decreases. Thus, it appears that using only global similarity ($\gamma = 1.0$, $\lambda = 0.0$) as seen in most existing methods cannot guarantee the best performance on most datasets (Fig. 2). On the other hand, using local similarity alone ($\gamma = 0.0$, $\lambda = 1.0$) generally underperforms using global similarity alone (Fig. 2), due probably to information loss in the former. In this sense, it is not surprising that global similarity contributes more (80%) to the similarity score S than local similarity (20%) for the best performance. Nonetheless, this contribution of local similarity is necessary to the best accuracy in most datasets (Fig. 2). In addition, we evaluate the performance of Euclidean distance and cosine angle distance for defining k-NNs of the cells, and an approximate method LSH (Andoni *et al.*, 2017) for finding the k-NNs, and find that Euclidean distance in combination with our method for finding k-NNs (Section 2) outperforms all other combination on most of the datasets (Supplementary Fig. S4).

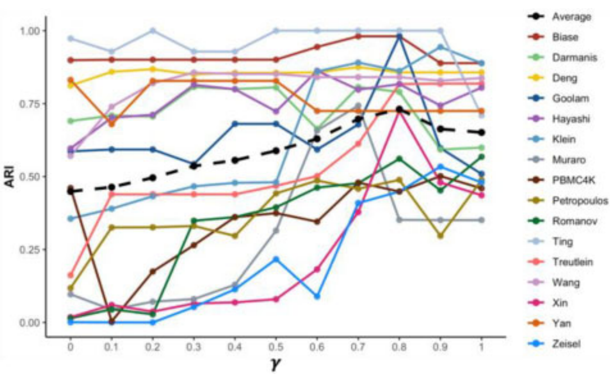

Fig. 2. Effects of the values of γ and $\lambda = 1 - \gamma$ on the accuracy of RCSL

3.2 RCSL outperforms existing methods in clustering cells

We next compare the performance RCSL on 16 scRNA-seq datasets with that of five state-of-the-art tools including SC3 (Kiselev *et al.*, 2017), SIMLR (Bo *et al.*, 2017), pcaReduce (Žurauskienė and Yau, 2016), CIDR (Lin *et al.*, 2017) and Seurat (Satija *et al.*, 2015) using three metrics (Section 2). As mentioned before, SC3 is a popular method based on the consensus result of multiple methods; SIMLR is a similarity learning algorithm based on multikernel; pcaReduce is an agglomerative clustering method based on statistical modeling; CIDR is an ultrafast algorithm that imputes dropouts; and Seurat is widely applied to large datasets. Notably, both SIMLR and RCSL identify clusters by learning a block-diagonal similarity matrix from similarity matrices defined differently, therefore, we can compare their learned block-diagonal matrix. Moreover, to show the contribution of local similarity metric NR to the performance, we also implement a variant (RCSL2) of RCSL that does not use NR, thus is much faster than RCSL, though also runs at O(N^2) (Supplementary Fig. S3). More specifically, versions of the packages we compare with our algorithm are as follows: SC3 (package version 1.14.0 from Bioconductor); SIMLR (package version 1.12.0 from GitHub) (github.com/BatzoglouLabSU/SIMLR); pcaReduce (package version 1.0 from GitHub) (github.com/JustinaZ/pcaReduce); *k*-means (*kmeans* function built-in R version 3.6.0); CIDR (package version 0.1.5 from GitHub) (github.com/VCCRI/CIDR); and Seurat (package version 3.1.5 from CRAN). For PCA-Kmeans, we use our $\boldsymbol{X}_{Q\times N}$ as the input matrix. Since PCA-Kmeans is a stochastic algorithm, we run it 100 times and present the average of the results. In addition, since Kmeans cannot determine the number of clusters, we estimate the number of clusters by *NbClust* for Kmeans. For the other methods, we follow its corresponding instructions and tutorials provided by the authors and use its default parameters.

As shown in Figure 3, both RCSL and RCSL2 outperforms the six existing methods on 11 of the 16 datasets (Table 1) based on ARI values (average 0.73 versus 0.64). Specifically, on the Biase, Darmanis, Goolam, Treutlein Xin and Ting datasets, RCSL achieves significantly higher ARI than all the other algorithms. On the Deng and Romanov datasets, both RCSL and RCSL2 perform much better than the other algorithms. On the Wang and Hayashi datasets, RCSL also gains the highest ARI. Only on the Zeisel, Petropoulos, PBMC4K, Muraro and Yan datasets, SIMLR, Seurat or pcaRduce outperform RCSL (Fig. 3). However, on average, both RCSL and RCSL2 substantially outperform the six other methods (Fig. 3). Similar results are seen using NMI and FM (Supplementary Figs S5 and S6). These results indicate that both the optimized similarity metric (Fig. 2) and the clustering algorithm contribute to the outstanding performance of RCSL.

3.3 RCSL outperforms existing methods in learning the similarity among cells

We further seek to see whether the block-diagonal similarity matrices $\boldsymbol{B}$ learned by RCSL have the intended block-diagonal structures. To this end, we compare the SRC matrix $\boldsymbol{S}_S$, the similarity matrix $\boldsymbol{S}$ in RCSL as well as block-diagonal matrices learned by RCSL, RCSL2 and SIMLR. In the ideal case, if all cells are correctly clustered, then when cells are sorted by their types, the resulting matrix should exhibit clear-cut block-diagonal submatrices, in which similarities between cells of different types are zero while those between cells of the same types are non-zero. Figure 4 shows the matrices learned by the four methods on eight datasets, in which cells are ordered according to their annotated types, and the results of the other eight datasets are shown in Supplementary Figure S7. Clearly, for most datasets except for the Ting, Klein, Wang, Zeisel and Yan datasets, there are no obvious block-diagonal submatrices for cells in $\boldsymbol{S}_S$ or $\boldsymbol{S}$ (Fig. 4 and Supplementary Fig. S7). In contrast, cells in the similarity matrices $\boldsymbol{B}$ learned by RCSL, RCSL2 and SIMLR possess clear-cut block-diagonal structures in all the datasets (Fig. 4 and Supplementary Fig. S7). However, there are subtle differences among the results of the three algorithms. For the Treutlein and Klein datasets, both RCSL and RCSL2 correctly cluster the cells according to their annotated types, whereas SIMLR incorrectly groups multiple annotated cell types into one cluster. For the Biase, Goolam and Darmanis datasets, RCSL and RCSL2 correctly clustered the cells according to their annotated types, while SIMLR divides one type into multiple clusters. For the Goolam, Ting and Xin datasets, RCSL2 divides one cell type into multiple clusters, while clusters identified by RCSL are in better agreement with their annotated types, indicating the importance of including NR in the similarity metric. In addition, clusters found by RCSL are cleaner than those identified by RCSL2 and SIMLR in the off-diagonal blocks. As expected, the quality of the learned block-diagonal similarity matrices is consistent with clustering results measured by ARI, NMI and FM (Fig. 3 and Supplementary Figs S5 and S6). Taken together, these results demonstrate that RCSL can better learn the block-diagonal similarity matrix of different cell types than the two other methods.

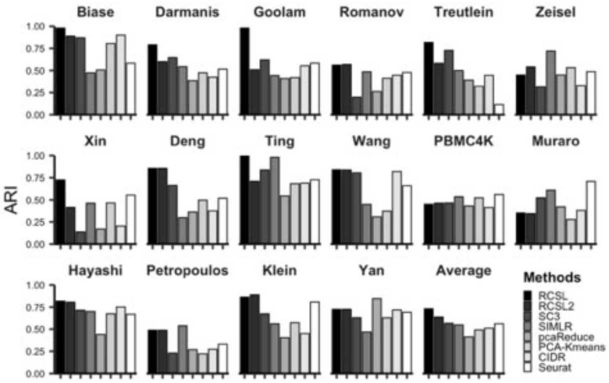

Fig. 3. Performance of the algorithms (RCSL, RCSL2, SC3, SIMLR, pcaReduce, PCA-Kmeans, CIDR, Seurat) on the datasets measured by ARI. The last panel shows the average ARI value for each algorithm over the 16 datasets

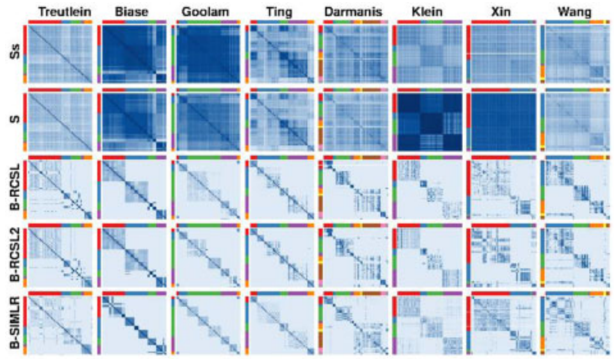

Fig. 4. Heatmap of the SRC matrix $\boldsymbol{S}_S$, similarity matrix $\boldsymbol{S}$ in RCSL and the block-diagonal similarity matrices $\boldsymbol{B}$ learned by RCSL, RCSL2, SIMLR in the indicted eight datasets. Cells are arranged according to their annotated types indicated by the differently colored bar at the top and left of the matrices

3.4 RCSL learns block-diagonal structures of cell–cell similarities in a step-wise manner

To see how RCSL gradually learns the block-diagonal structures of cell–cell similarities starting from a data matrix, thereby clustering cells, we visualize the data matrix $\boldsymbol{M}_{G'\times N}$, global similarity matrix $\boldsymbol{S}_S$, similarity matrix $\boldsymbol{S}$ and block-diagonal matrix $\boldsymbol{B}$ of the 16 datasets using three visualization tools [PCA (Žurauskienė and Yau, 2016), *t*-SNE (Maaten and Hinton, 2008) and UMAP (McInnes

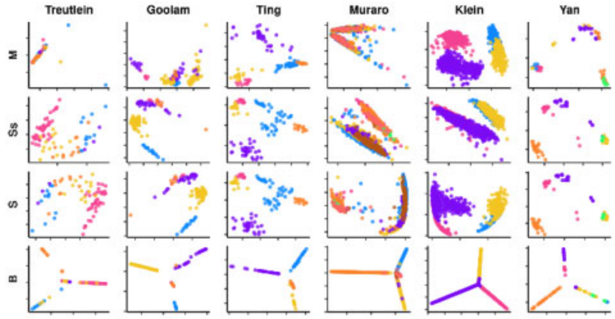

Fig. 5. 2D PCA display of the expression data matrices and matrices produced by RCSL in the indicated datasets. The rows, respectively, correspond to $\boldsymbol{M}_{G'\times N}$, Spearman's correlation $\boldsymbol{S}_S$, similarity matrix $\boldsymbol{S}$ and block-diagonal matrix $\boldsymbol{B}$

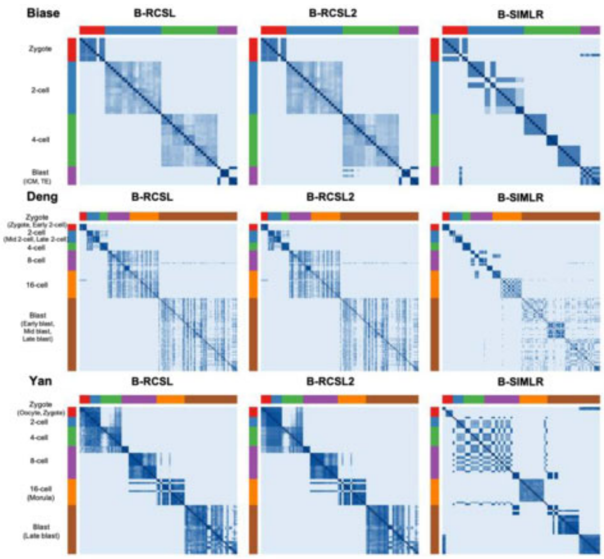

Fig. 6. Heatmap of block-diagonal matrixes constructed by RCSL, RCSL2 and SIMLR for the Bias, Deng and Yang dataset where sub-cell types are recorded. Cells are arranged according to their annotated types and subtypes indicated by differently colored bars at the top and left of the matrices

et al., 2018)]. Figure 5 shows the PCA 2D plots of the results from six datasets. The PCA plots for the remaining datasets as well as *t*-SNE and UMAP plots for all datasets are shown in Supplementary Figures S8–S10. Interestingly, in some datasets such as the Goolam, Ting, Klein and Yan datasets, most cells can be well separated according to their types simply by the data matrices $\boldsymbol{M}_{G' \times N}$ and $\boldsymbol{S}_S$ in the 2D plots, owing to the markedly distinct gene expression patterns of different cell types. In other datasets such as the Treutlein and Muraro datasets, cells cannot be simply separated by the data matrices $\boldsymbol{M}_{G' \times N}$, $\boldsymbol{S}_S$ and $\boldsymbol{S}$. Particularly, for the Muraro and Klein datasets, cells of the same type are dispersed, while cells of different types are mixed in $\boldsymbol{M}_{G' \times N}$, $\boldsymbol{S}_S$ and $\boldsymbol{S}$. In contrast, different cell types in all the 16 datasets display clear-cut clusters in the block-diagonal matrix $\boldsymbol{B}$ (Fig. 5 and Supplementary Figs S8–S10). Therefore, each step in the RCSL algorithm contributes to the identification of cell clusters/types in a dataset.

3.5 RCSL outperforms SIMLR in identifying sub-cell types

Notably, like SIMLR, RCSL also tend to divide one cell type into multiple sub-clusters in some datasets, particularly, developmentally related ones (Biase, Deng, Goolam, Klein, Hayashi, Petropoulos and Yan) (Fig. 4 and Supplementary Fig. S7). This might reflect the hierarchical lineage relationships of cell types produced in cell differentiation processes. Three (Biase, Deng and Yan) of these datasets record subtypes produced during embryogenesis (Supplementary Table S2). To see if RCSL is able to identify sub-cell types, we take a close look at the results from the three datasets (Fig. 6). For the Biase dataset containing four cell types, of which the Blast type is divided into the Inner cell mass (ICM) and trophectoderm (TE) subtypes (Supplementary Table S2), RCSL clusters the cells in five groups (Zygote, 2-cell, 4-cell, ICM and TE), and correctly splitting the Blast in the ICM and TE subtypes (Fig. 6). In contrast, SIMLR divides the cells into at least eight groups, with the 2-cell type incorrectly split in four clusters, the 4-cell type incorrectly split into three clusters (Fig. 6). Moreover, SIMLR fails to divide the Blast type into the ICM and TE subtypes (Fig. 6). For the Deng dataset containing six cell types, of which the Zygote type is further classified into Zygote and Early 2-cell types, the 2-cell type into Middle 2-cell and Late 2-cell types, and the Blast type into Early, Middle and Late types (Supplementary Table S2), RCSL clusters the cells into five clusters, identifying the Middle and Late 2-cell types, but failing identify subtypes of the Zygote and the Blast type (Fig. 6). However, there might be no clear-cut difference among these subtypes based on their stages (Early, Mid and Late). In contrast, SIMLR divides the cells into at least 13 clusters, with the 8-cell type incorrectly split into four clusters, and the 16-cell type incorrectly split into at least two clusters, though it also correctly clusters the two subtypes of the 2-cell type (Fig. 6). For the Yan dataset containing six cell types, of which the Zygote type is divided into Oocyte and Zygote subtypes, RCSL clusters the cells in five groups, correctly identifying the 8-cell, 16-cell and Blast types, but failing to identify subtypes of the Zygote type. Although SIMLR also is able to correctly divide the Zygote type into two clusters, it splits the 16-cell type into two clusters, the Blast type into three clusters, and the 8-cell type into at least five clusters.

On the other hand, it is difficult to justify the subtypes identified by RCSL and SIMLR in the other datasets, as subtype information is unavailable. However, based on the results from the three datasets where some subtypes are classified, it appears that RCSL is more accurate in identifying sub-cell types than SIMLR that tends to over-cluster the cells.

3.6 RCSL achieves high clustering accuracy on simulated datasets

We have thus far demonstrated the high accuracy of RCSL for classifying cell types using the 16 datasets with well-annotated cell types. However, accurate cell type determination is still a challenging task, particularly, in larger datasets, it is difficult to guarantee 100% accuracy. To further evaluate the accuracy of RCSL, we ran it on 10 simulated datasets, in which the cell types are well defined (Supplementary Table S3 and Section 2). Although RCSL slightly outperforms RCSL2 on the simulated datasets, both achieve a very higher average ARI of 0.95 and 0.94 (Supplementary Fig. S11), respectively, suggesting that both are able to identify well-defined cell clusters. However, this almost same performance of RCSL and RCSL2 is in stark contrast to the result on the 16 real datasets, where RCSL substantially outperforms RCSL with an average ARI of 0.73 and 0.64, respectively (Fig. 3). Nevertheless, this is not surprising, as the clusters in the simulated datasets are clear-cut though the information is rather weak (Supplementary Fig. S12), while the clusters in the real datasets are often vague with high background noise (Fig. 4 and Supplementary Fig. S7). These results indicate that considering both local and global similarity is more critical to identify cell types when the data are highly noisy.

3.7 RCSL correctly infers trajectories and pseudo-time orders

Based on the clustering results, RCSL infers the developmental trajectories and pseudo-temporal orders of the identified cell types in a dataset, particularly when it is time-series-related. Figure 7 shows UMAP displays of the trajectories and pseudo-temporal orders of the identified cell types in four mouse embryo datasets (Goolam, Hayashi, Yan and Deng) and a human preimplantation embryos dataset (Petropoulos), in which both the cell types and developmental order are known. Remarkably, in each dataset, the inferred

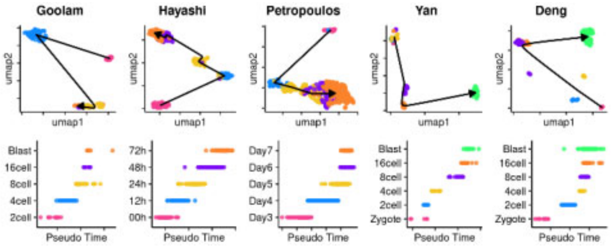

Fig. 7. Inference of developmental trajectories and pseudo-temporal orders of the identified cell types in four mouse embryo datasets (Goolam, Deng, Yan and Hayashi) and a human preimplantation embryos dataset (Petropoulos). The developmental trajectories (top) are visualized by UMAP. In the pseudo-temporal ordering of cells (bottom), the horizontal axis represents the estimated time of each cell type starting from the known initial cell type, and the vertical axis indicates the real cell stages/types

trajectory is the same as the real developmental trajectory, and the inferred pseudo-temporal ordering is also consistent with the true developmental stages. The results for the other datasets are shown in Supplementary Figure S13, where only an MST is shown if the starting cell type is unknown in a dataset, or it the dataset is not time series in nature.

4 Discussion

One of the major challenges for identifying heterogeneous cell types using scRNA-seq data lies in how to define the similarity among cells owing to inherent biological noise and unavoidable technical artifacts (Kiselev *et al.*, 2019). Moreover, only a few genes with similar expression patterns play a key role in defining a cell's type (Graf and Enver, 2009). To tackle these problems, many similarity metrics have been developed (Kim *et al.*, 2019); however, they only consider global similarities (Bo *et al.*, 2017; Park *et al.*, 2018) even though local similarities can be crucial to differentiate the subtle difference between cells of the same type and cells of different types (Xu and Su, 2015). In this study, we propose a new metric that considers both the global similarity and local similarity between the cells. Specifically, we quantify a cell's similarity to the other cells as an optimal linear combination of its global similarity and local similarity to other cells. For the global similarity of a cell, we utilize the SRC between the expression vector of the cell and those of all other cells in the data. For the local similarity of a cell, we adopt the NR that represents the cell's feature vector (principal components) as the optimal linear combination of the feature vectors of the cell's k-NNs in the Euclidean or cosine angle distance space. Thus NR in effect adaptively adjusts the weights on the edges between the cell and its k-NNs in the corresponding similarity graph. The importance of incorporating NR into the similarity metric is demonstrated by the better performance of RCSL in almost all the datasets compared to RCSL2, which does not use NR, particularly, when the data are very noisy.

Another major challenge for identifying cell types using scRNA data is how to cluster cells by their types based on the similarity matrix (Kiselev *et al.*, 2019). Although many clustering algorithms have been proposed to identify cell types, the results are far from satisfactory due to the often complex structures of similarity matrices (Kiselev *et al.*, 2019). On the other hand, clustering cells in groups by their types is equivalent to converting the similarity matrix into a block-diagonal matrix by permutation and minimal adjustment of the similarity values. In the resulting block-diagonal matrix, each block-diagonal submatrix corresponds to a connected component in the corresponding similarity graph, that is, a cluster or a cell type. We therefore adopt a method to compute such a block-diagonal matrix based on the similarity matrix. We first estimate the number of clusters defined in the similarity matrix and then iteratively find the block-diagonal matrix with the rank of its Laplacian matrix constrained.

We develop RCSL by combining the new similarity metric and the method for constructing the block-diagonal matrix, aiming to more accurately identify cells type in an often noisy scRNA-seq dataset. The results on the 16 diverse datasets show that RCSL substantially outperforms RCSL2 (a variant of RCSL that does not use local similarity), and RCSL2 outperforms the six other tools on many datasets, indicating that both the metric and block-diagonal matrix finding method contribute to the outstanding performance of RCSL. Although highly accurate, RCSL is limited by its $O(N^2)$ time complexity for computing S_S, S_{NR} and $\boldsymbol{B}$. We are currently developing a strategy to reduce the time complexity of RCSL to $N\log N$, so that it can be applied to very large datasets generated from millions of cells.

Acknowledgements

We would like to thank the three reviewers whose comments and suggestions improve the paper.

Author contributions

Q.M. conceived the algorithms and designed the software. Q.M, Z.S. and G.L wrote the paper. Z.S and G.L. oversaw the project.

Funding

This work was supported by the National Science Foundation of China [11931008, 61771009, 2020YFA0712400 to G.L.] and the US National Science Foundation [DBI-1661332 to Z.S.]. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Conflict of Interest: none declared.

References

Becskei,A. *et al.* (2005) Contributions of low molecule number and chromosomal positioning to stochastic gene expression. *Nat. Genet.*, **37**, 937–944.

Biase,F.H. *et al.* (2014) Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing. *Genome Res.*, **24**, 1787–1796.

Bo,W. *et al.* (2017) Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Methods*, **14**, 414.

Buettner,F. *et al.* (2015) Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.*, **33**, 155–160.

Chung, F., R.K. (1997) *Spectral Graph Theory*. American Mathematical Society, Providence, RI.

Darmanis,S. *et al.* (2015) A survey of human brain transcriptome diversity at the single cell level. *Proc. Natl. Acad. Sci. USA*, **112**, 7285–7290.

Deng,Q. *et al.* (2014) Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, **343**, 193–196.

Fowlkes,E.B. and Mallows,C.L. (1983) A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.*, **78**, 553–569.

Graf,T. and Enver,T. (2009) Forcing cells to change lineages. *Nature*, **462**, 587–594.

Guo,M. *et al.* (2015) SINCERA: a pipeline for single-cell RNA-seq profiling analysis. *PLoS Comput. Biol.*, **11**, e1004575.

Hayashi,T. *et al.* (2018) Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. *Nat. Commun.*, **9**, 1–16.

Hubert,L. and Arabie,P. (1985) Comparing partitions. *J. Classif.*, **2**, 193–218.

Jiang,L. *et al.* (2016) GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biol.*, **17**, 1–13.

Kaern,M. *et al.* (2005) Stochasticity in gene expression: from theories to phenotypes. *Nat. Rev. Genet.*, **6**, 451–464.

Kim,T. *et al.* (2019) Impact of similarity metrics on single-cell RNA-seq data clustering. *Brief. Bioinform.*, **20**, 2316–2326.

Kiselev,V.Y. *et al.* (2017) SC3—consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.

Kiselev,V.Y. *et al.* (2019) Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.*, **20**, 273–282.

Klein,A.M. *et al.* (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.

Krzanowski,W.J. and Lai,Y.T. (1988) A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, **44**, 23–34.

Lin,P. *et al.* (2017) CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.*, **18**, 59.

Luxburg,U.V. (2007) A tutorial on spectral clustering. *Stat. Comput.*, **17**, 395–416.

Maaten,L.V.D. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

McInnes,L. *et al.* (2018) UMAP: uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mohar,B. (1991) The Laplacian spectrum of graphs. *Graph Theory Comb. Appl.*, **18**, 871–898.

Mubeen,G. *et al.* (2016) Heterogeneity in Oct4 and Sox2 targets biases cell fate in 4-cell mouse embryos. *Cell*, **165**, 61–74.

Muraro,M.J. *et al.* (2016) A single-cell transcriptome atlas of the human pancreas. *Cell Syst.*, **3**, 385–394.e383.

Newman,J.R. *et al.* (2006) Single-cell proteomic analysis of S. cerevisiae reveals the architecture of biological noise. *Nature*, **441**, 840–846.

Nie,F. *et al.* (2010) Efficient and robust feature selection via joint ℓ2, 1-norms minimization. In: *Advances in Neural Information Processing Systems*, pp. 1813–1821.

Nie,F. *et al.* (2016) The Constrained Laplacian Rank algorithm for graph--based clustering. In: *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1969–1976.

Park,S. *et al.* (2018) Spectral clustering based on learning similarity matrix. *Bioinformatics*, **34**, 2069–2076.

Paulsson,J. (2004) Summing up the noise in gene networks. *Nature*, **427**, 415–418.

Petropoulos,S. *et al.* (2016) Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, **165**, 1012–1026.

Pierson,E. and Yau,C. (2015) ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, **16**, 241.

Prim,R.C. (1957) Shortest connection networks and some generalizations. *Bell Syst. Tech. J.*, **36**, 1389–1401.

Raj,A. *et al.* (2008) Imaging individual mRNA molecules using multiple singly labeled probes. *Nat. Methods*, **5**, 877–879.

Raj,A. and van Oudenaarden,A. (2008) Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell*, **135**, 216–226.

Romanov,R.A. *et al.* (2017) Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nat. Neurosci.*, **20**, 176–188.

Roweis,S.T. and Saul,L.K. (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**, 2323–2326.

Satija,R. *et al.* (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, **33**, 495–502.

Strehl,A. and Ghosh,J. (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, **3**, 583–617.[CVOCROSSCVO]

Taniguchi,Y. *et al.* (2010) Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, **329**, 533–538.

Ting,D.T. *et al.* (2014) Single-cell RNA sequencing identifies extracellular matrix gene expression by pancreatic circulating tumor cells. *Cell Rep.*, **8**, 1905–1918.

Treutlein,B. *et al.* (2014) Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature*, **509**, 371–375.

Wang,Y.J. *et al.* (2016) Single-cell transcriptomics of the human endocrine pancreas. *Diabetes*, **65**, 3028–3038.

Wold,S. *et al.* (1987) Principal component analysis. *Chemom. Intell. Lab. Syst.*, **2**, 37–52.

Xie,X.S. *et al.* (2008) Single-molecule approach to molecular biology in living bacterial cells. *Annu. Rev. Biophys.*, **37**, 417–444.

Xin,Y. *et al.* (2016) RNA sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metab.*, **24**, 608–615.

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

Yan,L. *et al.* (2013) Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. Struct. Mol. Biol.*, **20**, 1131–1139.

Young,J.W. *et al.* (2012) Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. *Nat. Protoc.*, **7**, 80–88.

Zappia,L. *et al.* (2017) Splatter: simulation of single-cell RNA sequencing data. Genome Biol., **18**, 1–15.

Zeisel,A. *et al.* (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.

Zenklusen,D. *et al.* (2008) Single-RNA counting reveals alternative modes of gene expression in yeast. *Nat. Struct. Mol. Biol.*, **15**, 1263–1271.

Zheng,G.X. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 1–12.

Žurauskienė,J. and Yau,C. (2016) pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, **17**, 140.