Bioinformatics, 38(10), 2022, 2692–2699
https://doi.org/10.1093/bioinformatics/btac168
Advance Access Publication Date: 22 March 2022
Original Paper

Sequence analysis

EDClust: an EM–MM hybrid method for cell clustering in multiple-subject single-cell RNA sequencing

Xin Wei[1,†], Ziyi Li[2,†], Hongkai Ji[3] and Hao Wu [1,*]

[1]Department of Biostatistics and Bioinformatics, Emory University, Atlanta, GA 30322, USA, [2]Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA and [3]Department of Biostatistics, Johns Hopkins University, Baltimore, MD 21205, USA

*To whom correspondence should be addressed.
[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Christina Kendziorski

Received on August 30, 2021; revised on February 24, 2022; editorial decision on March 16, 2022; accepted on March 18, 2022

Abstract

Motivation: Single-cell RNA sequencing (scRNA-seq) has revolutionized biological research by enabling the measurement of transcriptomic profiles at the single-cell level. With the increasing application of scRNA-seq in larger-scale studies, the problem of appropriately clustering cells emerges when the scRNA-seq data are from multiple subjects. One challenge is the subject-specific variation; systematic heterogeneity from multiple subjects may have a significant impact on clustering accuracy. Existing methods seeking to address such effects suffer from several limitations.
Results: We develop a novel statistical method, EDClust, for multi-subject scRNA-seq cell clustering. EDClust models the sequence read counts by a mixture of Dirichlet-multinomial distributions and explicitly accounts for cell-type heterogeneity, subject heterogeneity and clustering uncertainty. An EM-MM hybrid algorithm is derived for maximizing the data likelihood and clustering the cells. We perform a series of simulation studies to evaluate the proposed method and demonstrate the outstanding performance of EDClust. Comprehensive benchmarking on four real scRNA-seq datasets with various tissue types and species demonstrates the substantial accuracy improvement of EDClust compared to existing methods.
Availability and implementation: The R package is freely available at https://github.com/weix21/EDClust.
Contact: hao.wu@emory.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) is a powerful technology for measuring gene expression at the single-cell level. It offers unprecedented opportunities to answer questions related to cell-specific changes in transcriptome, such as identification of rare cell types (Jindal *et al.*, 2018) and heterogeneity of cell responses (Buettner *et al.*, 2015). Several experimental protocols of scRNA-seq have been developed in the past few years, including SMART-seq2 (Picelli *et al.*, 2013), CEL-seq2 (Hashimshony *et al.*, 2016) and Drop-seq (Macosko *et al.*, 2015), providing additional choices to meet diverse research needs. Among all, droplet-based technologies encapsulate each individual cell in a nanoliter droplet together with a bead, substantially reducing the experimental cost (Macosko *et al.*, 2015). Moreover, droplet-based methods use unique molecular identifiers (UMIs) to eliminate the effects of PCR amplification bias (Kivioja *et al.*, 2011). The good scalability, high efficiency and low cost make droplet-based methods the top choice for scRNA-seq experiments in population-scale studies (Mazutis *et al.*, 2013). However, UMI data from droplet-based technology are generally harder to analyze due to their low signal-to-noise (Kiselev *et al.*, 2019).

The first step of scRNA-seq data analysis is usually cell clustering. The main purpose of clustering is to group cells by their transcriptomic similarity, and then annotate the groups by cell types based on existing biological knowledge. This is a fundamental step in scRNA-seq analysis, since many downstream analyses, including cellular composition estimation, cell type-specific differential expression and rare cell type discovery, are carried out based on the clustering results (Chen *et al.*, 2019). Though classic unsupervised clustering methods, such as *K*-means and hierarchical clustering can be applied, in view of the sparse, noisy and large-dimensional characteristics of scRNA-seq data (Qi *et al.*, 2020), many unsupervised methods customized for scRNA-seq data have been developed and widely used. For example, SC3 combines feature selection and dimension reduction in a consensus clustering framework, and it has been proven a highly robust clustering method (Kiselev *et al.*, 2017). Seurat is another popular method that adopts community detection to identify similar cells, and it shows good scalability for large

datasets (Satija *et al.*, 2015). TSCAN fits a mixture of multivariate normal distributions and uses hierarchical clustering to identify cell clusters (Ji and Ji, 2016). Lastly, observing the needs for clustering large-scale study with thousands to millions of cells, SHARP is developed for ultra-fast clustering through a divide-and-conquer strategy (Wan *et al.*, 2020).

All the aforementioned clustering methods have been developed without consideration of systematic biases in the data. They assume that the expressions of a gene from all cells in the same cell type are identically distributed. However, similar to many other high-throughput technologies, scRNA-seq data also suffer from a number of technical biases. One such bias in the population-level study is the subject-specific effect: there could be a systematic, subject-specific shift in the gene expression. Thus, the distributions of the gene expression can be different between subjects even within the same cell type. That shift can be induced by different characteristics of the subjects, such as demographics or clinical conditions. It can also be caused by the batch effect. The batch effect occurs when cells are cultured, captured and sequenced in different conditions (Hicks *et al.*, 2018), which could lead to inevitable technical variability (Tung *et al.*, 2017). It is worth mentioning that the batch effect can be severe in scRNA-seq, since it is exacerbated by the fact that most scRNA-seq protocols require fresh tissue for experiments (Wohnhaas *et al.*, 2019), making a standard balanced experimental design for removing batch effect impossible in many cases. Nevertheless, most existing clustering methods do not explicitly address the heterogeneity among multiple subjects. Rather than providing useful biological similarities that one may expect, direct application of those methods on data from population studies could lead to inaccurate clustering results due to correlated measurement errors (Tang, 2015).

One possible remedy for the problem is to consider the subject-specific effect as a batch effect, correcting for that before cell clustering. Several computational methods have been developed for batch effect correction that can be applied before clustering. For example, ComBat and ComBat-seq (Zhang *et al.*, 2020) were developed originally for bulk sequencing data, and they use linear models to remove batch effects. Mutual-nearest-neighbor (MNN) corrects batch effects by constructing a shared space between datasets (Haghverdi *et al.*, 2018). Harmony is another popular batch correction method that uses an iterative approach to eliminate batch effects for cells calculated in PCA space (Korsunsky *et al.*, 2019).

Though it is possible to cluster the cells after removing the subject-specific effects, this two-step approach has some drawbacks. First, the batch effect correction procedure often produces negative values for gene expressions, which will generate errors in many cell clustering tools. Second, such an approach is generally not efficient due to the transformation of data and alteration of data structure. For example, several clustering methods adopt distributional assumptions based on count data, while the transformed data after batch effect correction are not counts anymore. Such a discrepancy will lead to undesirable clustering performance.

In comparison, a more rigorous and potentially better approach is to design a clustering method that takes subject-specific effects into consideration. Both BAMM-SC (Sun *et al.*, 2019) and BUSseq (Song *et al.*, 2020) are tailored methods for addressing subject-specific effect during clustering. BAMM-SC implements a Bayesian mixture model, which uses information across genes and individuals to account for heterogeneity. BUSseq adopts a more complicated hierarchical model that strictly follows the data generation process of scRNA-seq experiments to correct batch effects and cluster cells. Both methods use Markov chain Monte Carlo (MCMC) to solve the model, which does not scale well for large datasets. In a comprehensive benchmarking study (Tran *et al.*, 2020), LIGER (Welch *et al.*, 2019) and Seurat v3 (Stuart *et al.*, 2019) show outstanding performance on addressing the problem of removing batch effects. LIGER takes multiple single-cell datasets as input, uses integrative non-negative matrix factorization to obtain a low-dimensional representation of the input data, and performs joint clustering based on it. The latest version of Seurat v4 (Hao *et al.*, 2021), inherited from Seurat v3, uses the strategy of combing MNN and canonical correlation analysis (Butler *et al.*, 2018) to address the batch effect. Some other methods based on deep-learning techniques have also been developed. DESC (Li *et al.*, 2020) iteratively optimizes a clustering objective function in an autoencoder to remove the batch effect and provide cluster assignments. CarDEC (Lakkis *et al.*, 2021) implicitly makes batch effects correction by jointly optimizing reconstruction loss and clustering loss through transfer learning.

To provide a complementary approach to address the cell clustering problem in population-scale scRNA-seq data, we designed EDClust, which is an Expectation–Maximization (EM) (Dempster *et al.*, 1977) and Minorization–Maximization (MM) (Hunter and Lange, 2004) hybrid method based on a Dirichlet-multinomial mixture model, for clustering. EDClust takes the raw count data from multiple subjects without transformation, avoiding the possible destruction of data structure and loss of information. This modeling strategy is especially suitable for analyzing sparse UMI data from droplet-based technology. Meanwhile, EDClust explicitly quantifies the effects of heterogeneity from different sources and provides posterior probabilities for cells being in each cluster. Through extensive simulation studies and four real datasets analyses, we show EDClust has better clustering accuracy compared with existing methods.

In the following sections, we first introduce the data model and derivation of the EM–MM method. The simulation design and results are presented in Section 3. Lastly, we showcase the performance and utility of EDClust using four real scRNA-seq datasets in Section 4.

2 Materials and methods

To cluster population-scale scRNA-seq data, we propose a Dirichlet-multinomial mixture model to capture the cell type-specific and subject-specific effects on gene expression. Dirichlet-multinomial is a commonly used model for counts data, and it has been applied to several sequencing data, such as ChIP-seq (Wu and Ji, 2014) and microbiome (Wadsworth *et al.*, 2017). Our data model has a similar structure to BAMM-SC; however, our estimation procedures are completely different. Aiming to cluster all the cells from multiple subjects simultaneously, we utilize several tools for selecting features, determining the baseline and initializing parameters, providing an EM and MM hybrid framework for parameter estimations. Overall, the complete EDClust algorithm is summarized in Fig. 1.

2.1 Data model

Let y_{lji} represent the sequence counts for gene j in cell i from subject l ($1 \le i \le I_l$, $1 \le j \le J$, $1 \le l \le L$), where I_l, J and L indicate the total number of cells (in subject l), genes and subjects, respectively. We assume all subjects in the data share the same cell types, the number of which is K. Note that K can be specified by investigators based on biological knowledge, or it can be determined by a number of software tools. Throughout this work, we assume K is known. Based on the assumption that $\boldsymbol{Y}_{li} = (Y_{l1i}, Y_{l2i}, \ldots, Y_{ljI})$ follows a Dirichlet-multinomial mixture distribution, $\boldsymbol{Y}_{li}$ can be viewed as generated in two steps. First, a cell type label $W_{li} \in \{1,2,\ldots,K\}$ is assigned to cell i in subject l with probability $\Pr(W_{li} = k) = \pi_{lk}$. Second, given the cell label (i.e. $W_{li} = k$), $\boldsymbol{Y}_{li}$ will be generated from a Multinomial distribution by $\boldsymbol{Y}_{li} \sim \text{Multinomial}(T_{li}, p_{li})$. Here, $T_{li} = \sum_j^J Y_{lji}$ indicates total read counts, and the proportion p_{li} represents the relative gene expressions. We further assume that p_{li} follows a cell type-specific prior distribution $\text{Dirichlet}(\alpha_{lk}) = \text{Dirichlet}(\alpha_{lk1}, \alpha_{lk2}, \ldots, \alpha_{lkJ})$. To simultaneously account for cell type-specific and subject-specific effects, we assume the overall effect α_{lk} can be expressed as the sum of cell type-specific effect α_{0kj} and subject-specific effect δ_{lkj}: $\alpha_{lkj} = \alpha_{0kj} + \delta_{lkj} > 0$. Finally, we assume that all cells in all L subjects are independent and treat cell type label $W_{li} = k$ as the missing data. Then, the observed and complete data log-likelihood can be written as:

$$l(\boldsymbol{T},\boldsymbol{\Theta};\boldsymbol{Y})=\sum_{l=1}^{L}\sum_{i=1}^{I_l}\log\left[\sum_{k=1}^{K}\pi_{lk}P_{lik}\right], \tag{1}$$

$$l_c(\boldsymbol{T},\boldsymbol{\Theta};\boldsymbol{Y},\boldsymbol{W})=\sum_{l=1}^{L}\sum_{i=1}^{I_l}\sum_{k=1}^{K}I(W_{li}=k)[\log\pi_{lk}+\log P_{lik}]. \tag{2}$$

Here, $\boldsymbol{W}=\{W_{li}:i=1,\ldots,I_l,l=1,\ldots,L\}$ includes the indicator of cell type labels, and $\boldsymbol{\Theta}=\{\pi_{lk},\alpha_{0k},\delta_{lk}:k=1,\ldots,K,l=1,\ldots,L\}$ contains all the model parameters. $P_{lik}=P(Y_{li}|T_{li},\alpha_{0k}+\delta_{lk})$ represents the Dirichlet-multinomial probability density, which is provided in Supplementary Materials.

2.2 The EM–MM hybrid algorithm for maximum likelihood

The introduction of the latent variable $\boldsymbol{W}$ facilitates the use of the EM algorithm to maximize the observed data likelihood and obtain posterior probabilities for cell type assignment (W_{li}). An EM algorithm iterates between two steps: an expectation step (E-step) and a maximization step (M-step) (Dempster *et al.*, 1977). Let $\boldsymbol{\Theta}^{(t)}$ be the parameter estimate in iteration t. In the E-step, we compute the conditional expectation of W_{li}:

$$\begin{aligned}\mu_{lik}^{(t)}&=E[I(W_{li}=k)|\boldsymbol{Y},\boldsymbol{\Theta}^{(t)}]\\&=P(W_{li}=k|\boldsymbol{Y},\boldsymbol{\Theta}^{(t)})\\&=\frac{\pi_{lk}^{(t)}P(Y_{li}|T_{li},\alpha_{0k}^{(t)}+\delta_{lk}^{(t)})}{\sum_{k'=1}^{K}\pi_{lk'}^{(t)}P(Y_{li}|T_{li},\alpha_{0k'}^{(t)}+\delta_{lk'}^{(t)})}.\end{aligned} \tag{3}$$

In the M-step, we maximize the 'Q function' (the expected complete data log-likelihood with respect to $\boldsymbol{\Theta}$) to obtain $\boldsymbol{\Theta}^{(t+1)}$. Then, π_{lk} can be updated by solving $\partial Q/\partial\pi_{lk}=0$.

$$\begin{aligned}Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})&=E[l(\boldsymbol{\Theta})|\boldsymbol{Y},\boldsymbol{\Theta}^{(t)}]\\&=\sum_{l=1}^{L}\sum_{i=1}^{I_l}\sum_{k=1}^{K}\mu_{lik}^{(t)}[\log\pi_{lk}+\log P_{lik}]\end{aligned} \tag{4}$$

$$\pi_{lk}^{(t+1)}=\frac{\sum_{i=1}^{I_l}\mu_{lik}^{(t)}}{I_l}. \tag{5}$$

The M-step derivation for α_{0k} and δ_{lk} is much more difficult, and there is not a closed form solution. For that, we design the following MM algorithm (Hunter and Lange, 2004) for updating α_{0k} and δ_{lk}.

Extending the work by Zhou and Lange (2010), we rewrite the log-likelihood function in Equation (2) as the following:

$$\begin{aligned}l(\boldsymbol{\Theta})=\sum_{l=1}^{L}\sum_{k=1}^{K}\Bigg[&\sum_{i=1}^{I_l}I(W_{li}=k)\log\pi_{lk}\\&-\textstyle\sum_{c_{1l}}r_{lkc}\log(\|\alpha_{0k}+\delta_{lk}\|_1+c_{1l})\\&+\sum_{j=1}^{J}\sum_{c_{2lj}}s_{lkjc}\log(\alpha_{0kj}+\delta_{lkj}+c_{2lj})\Bigg]+\text{const},\end{aligned} \tag{6}$$

where

$$r_{lkc}=\sum_{i=1}^{I_l}I(W_{li}=k)I(T_{li}\geq c_{1l}+1), \tag{7}$$

$$s_{lkjc}=\sum_{i=1}^{I_l}I(W_{li}=k)I(Y_{lji}\geq c_{2lj}+1), \tag{8}$$

and $\|\alpha_{0k}+\delta_{lk}\|_1=\sum_{j=1}^{J}|\alpha_{0kj}+\delta_{lkj}|=\sum_{j=1}^{J}(\alpha_{0kj}+\delta_{lkj})$. The index c_{1l} ranges from 0 to $\max_i(T_{li})-1$, and the index c_{2lj} runs from 0 to $\max_i(Y_{lij})-1$. In the MM algorithm, we design a surrogate function that minorizes the log-likelihood function. Assuming that $\alpha_{0kj}>0$ and $\delta_{lkj}\geq 0$, we can use the following inequalities:

$$\begin{aligned}-\log(c+\|\alpha_{lk}\|_1)\geq&-\frac{1}{\|\alpha_{0k}^{(n)}+\delta_{lk}^{(n)}\|_1+c}(\|\alpha_{0k}+\delta_{lk}\|_1)\\&+\text{const},\end{aligned} \tag{9}$$

$$\begin{aligned}\log(\alpha_{0kj}+\delta_{lkj}+c)\geq&\frac{\alpha_{0kj}^{(n)}}{\alpha_{0kj}^{(n)}+\delta_{lkj}^{(n)}+c}\log(\alpha_{0kj})\\&+\frac{\delta_{lkj}^{(n)}}{\alpha_{0kj}^{(n)}+\delta_{lkj}^{(n)}+c}\log(\delta_{lkj})+\text{const}.\end{aligned} \tag{10}$$

For these inequalities, the equality holds when $\alpha_{0kj}=\alpha_{0kj}^{(n)}$ and $\delta_{lkj}=\delta_{lkj}^{(n)}$. We construct the following surrogate function $g(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t,n)})$ as:

$$\begin{aligned}g(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t,n)})=\sum_{l=1}^{L}\sum_{k=1}^{K}\Bigg\{&\sum_{i=1}^{I_l}\mu_{lik}^{(t)}\log\pi_{lk}\\&-\textstyle\sum_{c_{1l}}r_{lkc}^{(t)}\frac{\|\alpha_{0k}+\delta_{lk}\|_1}{\|\alpha_{0k}^{(t,n)}+\delta_{lk}^{(t,n)}\|_1+c_{1l}}\\&+\sum_{j=1}^{J}\sum_{c_{2lj}}s_{lkjc}^{(t)}\left[\frac{\alpha_{0kj}^{(t,n)}\log(\alpha_{0kj})}{\alpha_{0kj}^{(t,n)}+\delta_{lkj}^{(t,n)}+c_{2lj}}\right.\\&\left.+\frac{\delta_{lkj}^{(t,n)}\log(\delta_{lkj})}{\alpha_{0kj}^{(t,n)}+\delta_{lkj}^{(t,n)}+c_{2lj}}\right]\Bigg\}+\text{const},\end{aligned} \tag{11}$$

where

$$r_{lkc}^{(t)}=\sum_{i=1}^{I_l}\mu_{lik}^{(t)}I(T_{li}\geq c_{1l}+1), \tag{12}$$

$$s_{lkjc}^{(t)}=\sum_{i=1}^{I_l}\mu_{lik}^{(t)}I(Y_{lji}\geq c_{2lj}+1). \tag{13}$$

By solving $\partial g(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t,n)})/\partial\delta_{lkj}=0$ and $\partial g(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t,n)})/\partial\alpha_{0kj}=0$, we obtain the MM updates for δ_{lkj} and α_{0kj} as:

$$\delta_{lkj}^{(t,n+1)}=\frac{\sum_{c_{2lj}}\frac{s_{lkjc}^{(t)}\delta_{lkj}^{(t,n)}}{\alpha_{0kj}^{(t,n)}+\delta_{lkj}^{(t,n)}+c_{2lj}}}{\sum_{c_{1l}}\frac{r_{lkc}^{(t)}}{\|\alpha_{0k}^{(t,n)}+\delta_{lk}^{(t,n)}\|_1+c_{1l}}}, \tag{14}$$

$$\alpha_{0kj}^{(t,n+1)}=\frac{\sum_{l=1}^{L}\sum_{c_{2lj}}\frac{s_{lkjc}^{(t)}\alpha_{0kj}^{(t,n)}}{\alpha_{0kj}^{(t,n)}+\delta_{lkj}^{(t,n)}+c_{2lj}}}{\sum_{l=1}^{L}\sum_{c_{1l}}\frac{r_{lkc}^{(t)}}{\|\alpha_{0k}^{(t,n)}+\delta_{lk}^{(t,n)}\|_1+c_{1l}}}. \tag{15}$$

Within the M-step in each EM iteration, EDClust runs multiple MM iterations to update α_0 and δ. To reduce the computational burden, we only run three MM iterations in each M-step. Real data analyses show that such a procedure provides comparable performance as running more (such as 20) iterations.

2.3 Feature selection

Feature selection is one of the key steps before clustering. We aim to select a subset of informative genes that can capture the data structure and thereby improve clustering performance. A recently developed feature selection tool tailored to scRNA-seq, FEAture SelecTion (FEAST) (Su *et al.*, 2021), shows great potential for improving clustering accuracy. FEAST computes the F-statistics for each feature based on embedded consensus clustering results and provides a ranked feature list by significance. By default, EDClust applies FEAST to obtain the top 500 features for clustering. In the

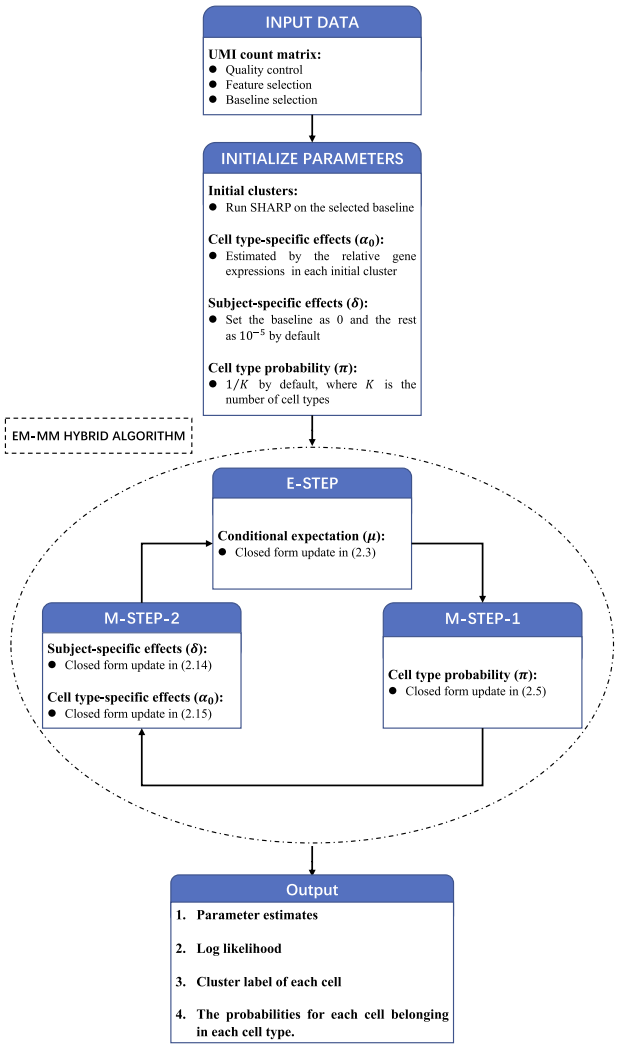


Fig. 1. A schematic plot to summarize the EDClust algorithm

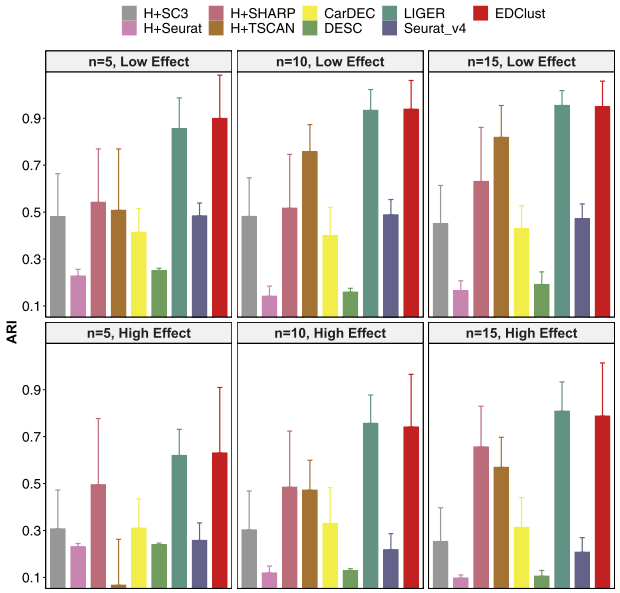


Fig. 2. Barplots of average ARIs for nine clustering methods across 100 simulations, where 'H +' indicates that the simulation data are processed by Harmony to remove the subject-specific effects. Each subplot presents the performance on simulation data with different subject-specific effects and subject numbers

software implementation, users have the option to specify the gene features.

2.4 Obtaining initial values

It is known that the EM algorithm often suffers from locally optimal solutions. Our method, due to the high dimensionality and complex nature of the data, is particularly prone to such challenges. Thus, it is crucial to provide good initial values for the parameter estimations, especially α_0 and δ. We design the following algorithm to obtain the initial values. We first choose a subject as the 'baseline', which is assumed to have no subject-specific effect ($\delta = 0$). We then run unsupervised clustering on the cells for the baseline subject using SHARP (Wan *et al.*, 2020). Based on the clusters in the baseline subject, we obtain naive estimates for $\hat{\alpha}_0$ according to the relative gene expression in the clusters and take them as the initial values. We set the selected baseline subject with a subject-specific effect of zero, and set initial values for the rest of δ's to be small positive numbers (10^{-5} by default).

In real data analyses, we notice that the choice of baseline subject occasionally would lead to bad results in some datasets (Supplementary Fig. S1). Careful investigation indicates that the bad results are caused by bad initial values. When the selected baseline subject has low signal-to-noise, SHAPR would provide bad clustering results, which subsequently leads to bad initial values and then EDClust is more likely to be stuck at local optimal solutions. To address this problem, we use the following algorithm to select the best possible subject as the baseline. We perform SHARP clustering on each subject. With the cell cluster assignment, we compute the F-test statistics for all genes and then compute their mean. The mean F-scores can roughly quantify how well the data are clustered. We then take the subject with the highest mean F-scores as the baseline subject and obtain initial value based on it.

3 Simulation studies

We design a series of simulation studies to comprehensively evaluate the performance of EDClust and compare it to a number of competing methods. We evaluate the methods when data have different levels of subject-specific effects (low and high), as well as for different sample size selections (5, 10, 15). Specifically, for gene j, cell i and subject l, we generate observed single-cell RNA-seq counts from a Dirichlet-multinomial distribution by $\boldsymbol{Y}_{li} \sim \text{multinomial}(T_{li}, \boldsymbol{p}_{li})$, where $\boldsymbol{p}_{li} \sim \text{Dirichlet}(\boldsymbol{\alpha}_{lk})$. The total read counts T_{li} are randomly selected from the observed total counts of the real data. We generate the prior parameter $\boldsymbol{\alpha}_{lk}$ by combining two parts $\boldsymbol{\alpha}_{lk} = \boldsymbol{\alpha}_{0k} + \boldsymbol{\delta}_{lk} \cdot \tau$. The cell type-specific effects $\boldsymbol{\alpha}_{0k}$ are drawn from $\boldsymbol{\alpha}_{0k} \sim \text{Lognormal}(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2) + \boldsymbol{A}_0$, and the subject-specific effects $\boldsymbol{\delta}_{lk}$ are obtained from $\boldsymbol{\delta}_{lk} \sim \text{Lognormal}(\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d^2) + \boldsymbol{D}_0$. The hyper-parameters $\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2, \boldsymbol{\mu}_d$ and $\boldsymbol{\sigma}_d^2$ are chosen so that the data demonstrate similar summary statistics as the real scRNA-seq data from the human skin study in Section 4.2. The $\boldsymbol{A}_0$ and $\boldsymbol{D}_0$ are parameters generated from Lognormal distributions so that the cell type-specific and subject-specific effects have some correlation among cell types as what we observed in real data. We vary the parameter τ in our simulation settings to control the magnitude of cross-subject heterogeneity. Larger heterogeneity indicates stronger subject-specific effects, and thus it is more difficult to cluster.

We compare EDClust with the other eight clustering methods (SC3, Seurat, SHARP, TSCAN, CarDEC, DESC, LIGER and Seurat v4). The batch correction method Harmony is applied to the raw data to remove the batch effect before clustering. We use the adjusted Rand index (ARI) (Rand, 1971) and NMI (Vinh *et al.*, 2010) as the evaluation criterion to benchmark the predicted cell type labels. We summarize the simulation results of over 100 Monte Carlo datasets.

As shown in Figure 2, across almost all scenarios with different levels of cross-subject heterogeneity, EDClust constantly achieves

the highest average ARI. The performance of EDclust remains stable, even as the subject-specific effects vary from low level to high level. And when the number of subjects increases, the level of heterogeneity also increases. EDClust still consistently outperforms most methods in terms of ARI. We also present the simulation results measured by NMI, which has similar performance (Supplementary Fig. S2). The simulation studies showcase outstanding performance of EDClust in clustering population-scale scRNA-seq data while accounting for subject-specific effects.

4 Real data analyses

We benchmark EDClust and other methods on four sets of real scRNA-seq with multiple subjects. Greater description of the datasets and data processing procedures is provided in each of the subsections below. In Table 1, we present the overall results for all four datasets, including the mean and standard deviation of ARIs from 50 runs in each dataset. In addition to the eight clustering methods compared in the simulation study, we also compare EDClust with BUSseq, DIMM-SC and BAMM-SC, where DIMM-SC and BAMM-SC have similar model assumptions. Since both TSCAN and Seurat are deterministic clustering methods, they do not have standard deviation in the results. The bar plots of average ARIs are also displayed in Supplementary Figure S3. These results show that for two out of the four datasets, EDClust has the best performance, and the performance improvement can be significant. For example, in the Mouse Retina data, EDClust has mean ARI 0.82, while the second best performer (Harmony+SC3) only has ARI 0.70. In the Baron Pancreas data, EDClust performs slightly worse than CarDEC and BUSseq, but there is no distinct difference. Additionally, we also compute NMI, AMI (Vinh *et al.*, 2010) and homogeneity (Rosenberg and Hirschberg, 2007) (Supplementary Fig. S3) and present the t-SNE plots with subject ID (Supplementary Fig. S4).

4.1 Mouse Retina dataset

We first evaluate the clustering performance of EDClust in mouse tissues through a Mouse Retina dataset, which is collected from 14-day-old mice in seven batches (Macosko *et al.*, 2015). Cells are first pooled together to filter out low-expression genes based on the dropout rate. We apply FEAST to generate a ranking list of features and select the top 500 genes based on this list. Five major cell types are retained, and the number of cells is 43 603.

As shown in Table 1 and Supplementary Figure S3, most of the methods struggle on this dataset with an exceedingly low average ARI. Though the performances of Harmony+SC3 and Harmony+SHARP are slightly better than any other methods except EDClust, their average ARIs are still below 0.70. EDClust achieves the highest ARI (0.8219), showing the superior performance of EDClust. To visualize the clustering results, we generate some t-SNE plots as shown in Figure 3a. The t-SNE plot generated based on the clustering result of EDClust is highly similar to the t-SNE plot with the true labels. We also show the t-SNE plot based on the clustering results from BAMM-SC and Harmony+SC3, where the circled regions highlight the incorrectly clustered cells. These plots provide clear visualization demonstrating the superior performance of EDClust over the other methods.

4.2 Human Skin dataset

To evaluate the performance of EDClust in human tissue, we evaluate the clustering performance of EDClust on a Human Skin dataset that includes skin samples collected from three healthy donors in a systemic sclerosis study (Sun *et al.*, 2019). Their study identified eight major types of cells. We use their results as the ground truth, but remove cells with an uncertain cell type. After quality control and feature selection, 3067 cells with 500 selected genes are used in the clustering analysis.

From Table 1 and Supplementary Figure S3, we can find that EDClust has the most outstanding performance (average ARI = 0.9497) of all the methods, while the average ARIs for most methods are close to 0.80. EDClust is more accurate in the clustering of several cell types. As shown in Figure 3b, compared with BAMM-SC and Harmony+SC3, basal keratinocytes, fibroblasts and macrophages/DC can all be classified well by EDClust, and each is assigned a specific cell type label, exhibiting the superior performance of EDClust on the Human Skin dataset.

4.3 Baron Pancreas dataset

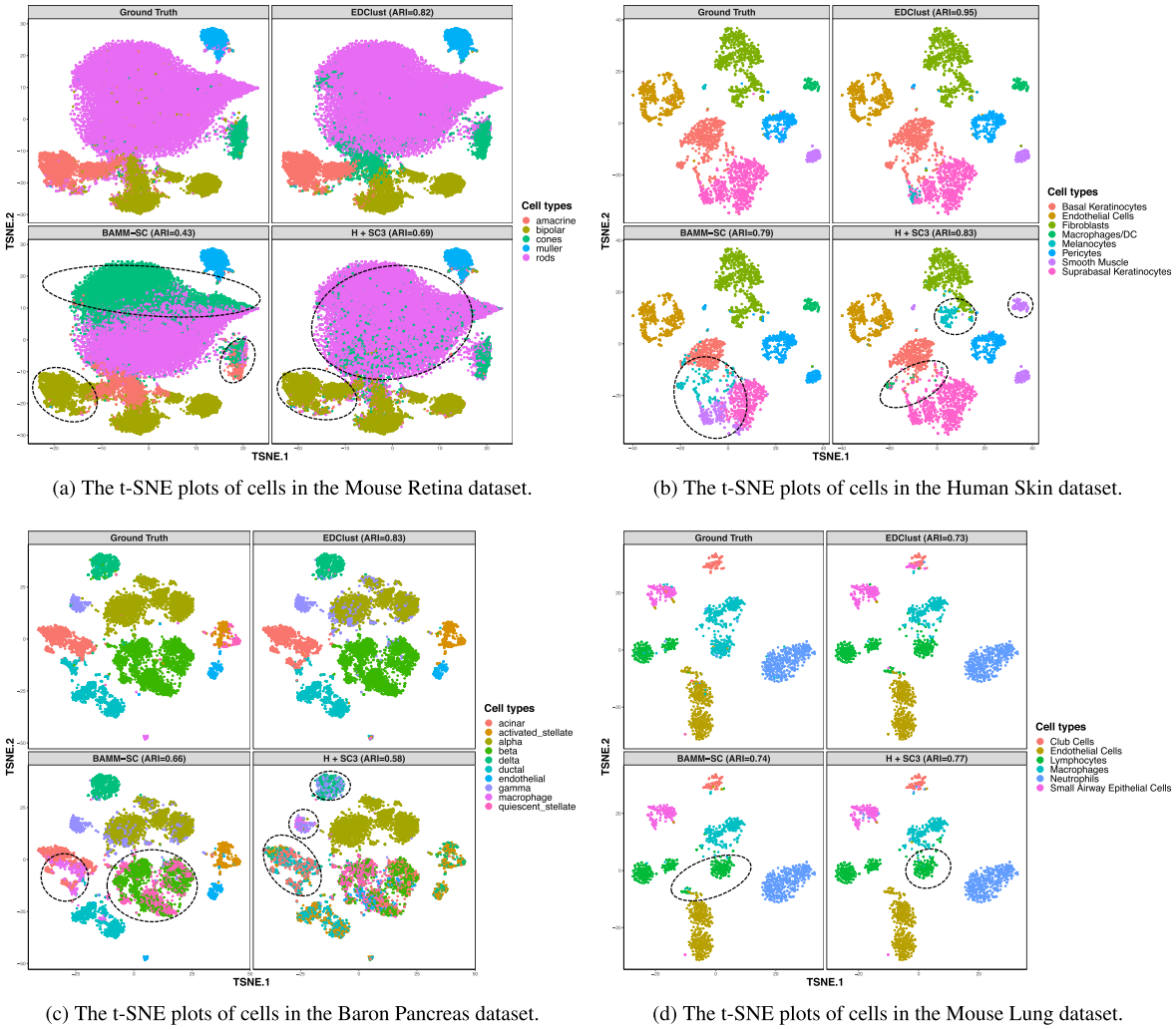
We further evaluate the clustering performance of EDClust on another human tissue type, a set of human pancreas data (named the 'Baron Pancreas' dataset). The original data include over 12 000 pancreatic cells from four human donors and two mouse strains (Baron *et al.*, 2016). We extract cells from the human donors and filter out the lowly expressed genes. The processed data contain 500 genes and a total of 8506 cells. Some extremely rare cell types with only a few cells (such as T cells) are removed, and 10 major cell types are kept for further analysis.

Both Table 1 and Supplementary Figure S3 show that the average ARI of EDClust is up to 0.8259, which could be ranked as top 3. And most methods fail to achieve the average ARI of 0.70. Figure 3c elucidates that Harmony+SC3 mixed massive cells. Compared to BAMM-SC, EDClust correctly identifies beta cells. Based on EDClust, for most of the cells, we are able to assign labels that are close to the approximated truth. These results showcase the outstanding performance of EDClust on the Baron Pancreas dataset.

Table 1. The ARI of 50 times clustering analyses for each method on four real datasets

Method	Mouse Retina		Human Skin		Baron Pancreas		Mouse Lung	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Harmony+SC3	0.6972	0.0687	0.8260	0.0249	0.5590	0.1035	**0.7652**	0.0026
Harmony+Seurat	0.1024	—	0.6520	—	0.5137	—	0.5307	—
Harmony+SHARP	0.6572	0.0095	0.8369	0.0533	0.3115	0.0236	0.7029	0.0271
Harmony+TSCAN	0.2905	—	0.6486	—	0.6392	—	0.6354	—
BAMM-SC	0.4273	0.0058	0.7732	0.0688	0.6411	0.0686	0.7354	0.0323
DIMM-SC	0.4221	0.0065	0.7975	0.0839	0.6968	0.0692	0.7003	0.0643
BUSseq	0.6499	0.2112	0.7560	0.1169	0.8372	0.0602	0.6775	0.0948
CarDEC	0.4462	0.0406	0.9281	0.0206	**0.8876**	0.0238	0.6169	0.1153
DESC	0.2512	0.0331	0.4078	0.0320	0.4909	0.0201	0.5008	0.0292
LIGER	0.4446	0.0036	0.6664	0.0518	0.6352	0.0704	0.6927	0.0742
Seurat_v4	0.2108	—	0.4859	—	0.5369	—	0.6156	—
EDClust	**0.8219**	0.0700	**0.9497**	0.0370	0.8259	0.0393	0.7445	0.0476

Note: The largest ARI in each dataset is highlighted in bold.



(a) The t-SNE plots of cells in the Mouse Retina dataset.

(b) The t-SNE plots of cells in the Human Skin dataset.

(c) The t-SNE plots of cells in the Baron Pancreas dataset.

(d) The t-SNE plots of cells in the Mouse Lung dataset.

Fig. 3. Comparison of the predicted results with the ground truth. Each plot is colored by the ground truth, labels inferred by EDClust, BAMM-SC and Harmony+SC3 (H+SC3), respectively. The circled regions highlight the incorrectly clustered cells from BAMM-SC and H+SC3. (**a**) The t-SNE plots of cells in the Mouse Retina dataset. (**b**) The t-SNE plots of cells in the Human Skin dataset. (**c**) The t-SNE plots of cells in the Baron Pancreas dataset. (**d**) The t-SNE plots of cells in the Mouse Lung dataset

4.4 Mouse Lung dataset

Finally, we evaluate the performance of EDClust in a real dataset with fewer cells. We mainly analyze a Mouse Lung dataset, which is obtained by collecting lung mononuclear cells from four mouse samples in the *Streptococcus pneumonia* infected group and control group (Sun *et al.*, 2019). After the data processing step, we obtain 500 top features provided by FEAST and a total of 1756 cells. Each cell is assigned a cell type label according to the previous study by Sun *et al.* (2019), and the expected number of clusters is set as six.

All methods have similar performances on the Mouse Lung dataset (results shown in Table 1 and Supplementary Fig. S3). The performance of EDClust (average ARI = 0.7445) is slightly worse than Harmony+SC3. Figure 3d presents a consistent pattern. In general, despite some mixed cell types, EDClust successfully characterizes endothelial cells and neutrophils with notable accuracy.

4.5 Computational performance

The EM algorithm usually converges slowly and has a heavy computational burden. Our proposed method embeds a few MM iterations within each EM iteration, bringing a higher computational cost. However, we implement the software in Julia and develop an R package with an interface to Julia based on JuliaCall (Li, 2019), achieving reasonable computational performance. We benchmark the computational performances of all the methods in a comparison shown in Supplementary Table S1. For our biggest dataset (the Mouse Retina dataset with seven batches and 43 603 cells), EDClust takes about 37 min on a normal computer with a single node. This is four times faster than BAMM-SC and BUSseq (∼180 min), which serve a similar purpose but are based on MCMC. Other methods either ignore the subject-specific effect or perform a two-step approach (i.e. batch effect removal and then cell clustering), making them not comparable in our setting. To further benchmark the computational performances in larger dataset, we apply EDClust, BAMM-SC and BUSseq on a brain dataset for autism study (Velmeshev *et al.*, 2019), which contains over 100 000 brain cells from 41 tissue samples from two brain regions of 15 patients with ASD and 16 control subjects. For this dataset, EDClust, BAMM-SC and BUSseq take 542, 389 and 477 min, respectively. Overall, the three methods (EDClust, BAMM-SC and BUSseq) have comparable computational performances. We acknowledge that EDClust is not very computationally efficient, especially in large dataset. Our plan in the near future is to implement parallel computing to improve EDClust's computational performance.

5 Discussion

In this work, we develop a novel statistical method for cell clustering in multi-subject scRNA-seq data. We model the read counts by a Dirichlet-multinomial mixture distribution, where the Dirichlet parameters contain subject-specific and cell type-specific effects. We

develop an EM–MM hybrid algorithm for fitting the mixture model and performing model-based clustering. Compared to existing clustering methods that ignore the subject-specific effects, EDClust has the following advantages: (i) it provides a tool to describe data heterogeneity among multiple subjects and more effectively identify subject-specific cell types; (ii) it utilizes the shared information among subjects, clustering all subject's cells at the same time, which improves the accuracy of cell clustering; (iii) it offers a one-step service that can be directly applied on raw count data, compared to other clustering methods that first require several preprocessing approaches (e.g. normalization and batch effect removal); and (iv) it quantifies cluster uncertainty with the probability that each cell belongs to a given cluster, contributing to further statistical inference and biological interpretation. Through a series of simulation experiments and real-world data applications, EDClust demonstrates considerable improvement in clustering accuracy over existing methods serving similar purposes.

EDClust is especially suitable for modeling droplet-based scRNA-seq data, such as that from Drop-seq or inDrop workflows (Klein and Macosko, 2017). Due to the differences in data characteristics, Dirichlet-multinomial distribution may not fit data from other platforms well. However, as the main focus of EDClust is on large-scale studies with more than one subject, droplet-based technology is usually chosen for such studies for high efficiency and low expense. Thus, our data assumption can be easily met in real practice.

EDClust uses 500 genes for clustering by default. We investigate the impact of gene number and find that the performance is similar as long as the selected gene number is reasonable (Supplementary Fig. S5). Nevertheless, users have the options to use different genes for analysis in our software. In addition, EDClust implicitly assumes that all subjects in the data have the same cell types. We investigate the case when there are missing cell types in the baseline subject, and find that the impact on the results is minimal when the missing cell type has a low abundance (Supplementary Fig. S6). When major cell types are missing, the performance drop can be significant. However, since the possibility of one subject missing a major cell type is very low (if not impossible), EDClust should work fine in the real data.

Additionally, since the initial values of the cell type-specific effects are set as the naive estimates based on the clustering results given by SHARP, we recommend running EDClust multiple times, each time using a different random seed, and selecting the one with the highest likelihood as the final result. Estimation of $\sum_{i=1}^{I} \alpha_{0kj}$ provided by Ronning (1989) or moment estimates proposed by Weir and Hill (2002) can also be appropriate choices for obtaining initial values. Moreover, the determination of the number of clusters is a crucial step. We suggest predefining it based on prior biological knowledge or model selection criteria, such as AIC (Akaike, 1974) and BIC (Schwarz *et al.*, 1978). Some other existing software, such as SC3 and Seurat, could also serve as tools for determining the number of clusters.

There are several limitations of EDClust. First, the main purpose of EDClust is cell clustering. Therefore, it may not be used to obtain batch-corrected profiles of the data, which is a limitation shared by most one-step clustering methods, such as BUSeq and BAMM-SC. Second, similar to many other unsupervised clustering methods, EDClust may not be scalable enough for analyzing very large datasets, e.g. millions of cells.

We anticipate a few natural extensions of EDClust. First, the Dirichlet-multinomial distribution can be replaced by other distributions to adjust for subject-specific effects in different experimental settings. For example, the negative binomial distribution is usually adopted for SMART-seq2 and CEL-seq2 platforms. Second, statistical testing and inference procedures can be developed to examine how the changes of sample phenotypes (e.g. their disease status) impact the estimated subject-specific effects and cell type-specific effects. Third, with the fast accumulation of single-cell data, we will further improve the computational performance of EDClust by implementing parallel computing.

Data availability

The study used various publicly available scRNA-seq datasets. The Mouse Retina dataset was downloaded from the Gene Expression Omnibus (GEO, accession number: GSE63473). Both the Human Skin dataset and the Mouse Lung dataset were obtained from GEO (accession number: GSE128066). The Baron Pancreas dataset was also downloaded from GEO (accession number: GSE84133). Published cell labels were obtained for each study and used as true cell type labels.

Funding

This work was partially supported by the National Institutes of Health [R01GM122083 to H.W., R01HG010889 to H.J.].

Conflict of Interest: none declared.

References

Akaike,H. (1974) A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, **19**, 716–723.

Baron,M. *et al.* (2016) A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Syst.*, **3**, 346–360.

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.

Butler,A. *et al.* (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**, 411–420.

Chen,G. *et al.* (2019) Single-cell RNA-seq technologies and related computational data analysis. *Front. Genet.*, **10**, 317.

Dempster,A.P. *et al.* (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B Methodol.*, **39**, 1–22.

Haghverdi,L. *et al.* (2018) Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, **36**, 421–427.

Hao,Y. *et al.* (2021) Integrated analysis of multimodal single-cell data. *Cell*, **184**, 3573–3587.e29.

Hashimshony,T. *et al.* (2016) CEL-Seq2: sensitive highly-multiplexed single--cell RNA-Seq. *Genome Biol.*, **17**, 1–7.

Hicks,S.C. *et al.* (2018) Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, **19**, 562–578.

Hunter,D.R. and Lange,K. (2004) A tutorial on mm algorithms. *Am. Stat.*, **58**, 30–37.

Ji,Z. and Ji,H. (2016) TSCAN: pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res.*, **44**, e117.

Jindal,A. *et al.* (2018) Discovery of rare cells from voluminous single cell expression data. *Nat. Commun.*, **9**, 1–9.

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.

Kivioja,T. *et al.* (2011) Counting absolute numbers of molecules using unique molecular identifiers. *Nat. Methods*, **9**, 72–74.

Klein,A.M. and Macosko,E. (2017) InDrops and Drop-seq technologies for single-cell sequencing. *Lab Chip*, **17**, 2540–2541.

Korsunsky,I. *et al.* (2019) Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods*, **16**, 1289–1296.

Lakkis,J. *et al.* (2021) A joint deep learning model enables simultaneous batch effect correction, denoising, and clustering in single-cell transcriptomics. *Genome Res.*, **31**, 1753–1766.

Li,C. (2019) JuliaCall: an R package for seamless integration between R and Julia. *J. Open Source Softw.*, **4**, 1284.

Li,X. *et al.* (2020) Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat. Commun.*, **11**, 1–14.

Macosko,E.Z. *et al.* (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**, 1202–1214.

Mazutis,L. *et al.* (2013) Single-cell analysis and sorting using droplet-based microfluidics. *Nat. Protoc.*, **8**, 870–891.

Picelli,S. *et al.* (2013) Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nat. Methods*, **10**, 1096–1098.

Qi,R. *et al.* (2020) Clustering and classification methods for single-cell RNA-sequencing data. *Brief. Bioinform.*, **21**, 1196–1208.

Rand,W.M. (1971) Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, **66**, 846–850.

Ronning,G. (1989) Maximum likelihood estimation of dirichlet distributions. *J. Stat. Comput. Simul.*, **32**, 215–221.

Rosenberg,A. and Hirschberg,J. (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic. pp. 410–420.

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

Schwarz,G. *et al.* (1978) Estimating the dimension of a model. *Ann. Stat.*, **6**, 461–464.

Song,F. *et al.* (2020) Flexible experimental designs for valid single-cell RNA-sequencing experiments allowing batch effects correction. *Nat. Commun.*, **11**, 1–15.

Stuart,T. *et al.* (2019) Comprehensive integration of single-cell data. *Cell*, **177**, 1888–1902.

Su,K. *et al.* (2021) Accurate feature selection improves single-cell RNA-seq cell clustering. *Brief. Bioinform.*, **22**, bbab034.

Sun,Z. *et al.* (2019) A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. *Nat. Commun.*, **10**, 1–10.

Tang,Y. (2015) *Cluster analysis with batch effect*. University of South Carolina.

Tran,H.T.N. *et al.* (2020) A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol.*, **21**, 1–32.

Tung,P.-Y. *et al.* (2017) Batch effects and the effective design of single-cell gene expression studies. *Sci. Rep.*, **7**, 39921.

Velmeshev,D. *et al.* (2019) Single-cell genomics identifies cell type–specific molecular changes in autism. *Science*, **364**, 685–689.

Vinh,N.X. *et al.* (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, **11**, 2837–2854.

Wadsworth,W.D. *et al.* (2017) An integrative Bayesian Dirichlet-multinomial regression model for the analysis of taxonomic abundances in microbiome data. *BMC Bioinformatics*, **18**, 1–12.

Wan,S. *et al.* (2020) SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection. *Genome Res.*, **30**, 205–213.

Weir,B.S. and Hill,W.G. (2002) Estimating f-statistics. *Annu. Rev. Genet.*, **36**, 721–750.

Welch,J.D. *et al.* (2019) Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, **177**, 1873–1887.

Wohnhaas,C.T. *et al.* (2019) DMSO cryopreservation is the method of choice to preserve cells for droplet-based single-cell RNA sequencing. *Sci. Rep.*, **9**, 1–14.

Wu,H. and Ji,H. (2014) PolyaPeak: detecting transcription factor binding sites from ChIP-seq using peak shape information. *PLoS One*, **9**, e89694.

Zhang,Y. *et al.* (2020) ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genom. Bioinform.*, **2**, lqaa078.

Zhou,H. and Lange,K. (2010) MM algorithms for some discrete multivariate distributions. *J. Comput. Graph. Stat.*, **19**, 645–665.