STATISTICAL METHODS FOR DECODING GENE REGULATION IN SINGLE CELLS

by
Zhicheng Ji

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
April, 2020

Abstract

Single-cell sequencing is rapidly transforming biomedical research. With the ability to measure omics information in individual cells, it provides unprecedented resolution to study heterogeneous biological and clinical samples, enabling scientists to discover and characterize previously unknown biological signals and processes carried by novel or rare cell subpopulations. The new data structure and high level of noise in the single-cell genomic data pose significant analytical challenges. To address these challenges, we developed new statistical and computational methods for analyzing single-cell transcriptome and regulome data. First, to infer cells' underlying developmental trajectories, we developed TSCAN that performs "pseudotime" analysis with a cluster-based minimum spanning tree approach. TSCAN facilitates accurate construction of pseudotemporal trajectories by regularizing the complexity of spanning trees. By improving the bias-variance tradeoff of the spanning tree estimation, TSCAN substantially improved the accuracy and robustness of the pseudotime analysis. Second, we developed RAISIN to support regression and differential analysis in single-cell RNA-seq datasets with multiple samples. Compared to classical linear mixed effects model, RAISIN improves variance estimate and statistical power for datasets with small sample size or cell number, and improves scalability for datasets with large sample size and millions of cells. Third, we developed SCATE to extract and enhance signals from the highly noisy and sparse single-cell ATAC-seq data. SCATE accurately infers genome-wide activities of each individual cis-regulatory element by adaptively integrating information from co-activated cis-regulatory elements, similar cells, and massive amounts of publicly available regulome data. The enhanced signal improves the performance of downstream analyses such as peak calling and prediction of transcription factor binding sites. These methods have been applied in numerous collaborative projects and helped decipher gene regulatory programs in T cell exhaustion process and identify molecular signatures in neoadjuvant immunotherapy.

Primary Reader and Advisor: Hongkai Ji

Secondary Reader: Jiou Wang, Stephanie Hicks, Kellie Smith

This thesis is dedicated to my wife, my parents, my thesis advisor, and my collaborators who made everything possible.

Acknowledgements

I would like to express the deepest appreciation to my thesis advisor Dr. Hongkai Ji for the invaluable support of my Ph.D. study. Without his mentorship, I will not be able to get prepared as an independent researcher in the field of statistical genomics and computational biology. I have learned from him not only the necessary knowledge and skills but also the way of independent and critical thinking as well as how to efficiently present and communicate the results to the broad scientific community. I am extremely lucky to have him as my Ph.D. thesis advisor.

I would like to thank all members of my thesis committee for their insightful comments and suggestions.

I would like to thank my current and former lab colleagues: Dr. Weiqiang Zhou, Dr. Wenpin Hou, Weixiang Fang, Boyang Zhang, Dr. Fang Du, Dr. Ben Sherwood, and many others for helpful discussions and generous support in various research projects.

I would like to thank all my collaborators: Dr. Drew M. Pardoll, Dr. Kellie N. Smith, Dr. Jiajia Zhang, Dr. Justina X. Caushi, Dr. Sneha Berry, and Dr. Janis M. Taube in the Bloomberg Kimmel Institute for Cancer Immunotherapy at Johns Hopkins School of Medicine; Dr. Andrew P. Feinberg, Dr. Michael A. Koldobskiy, and Dr. Varenka R. DiBlasi in the Epigenetics Center at Johns Hopkins School of Medicine; Dr. Steven A. Vokes, Dr. Kristin N. Falkenstein, and Rachel K. Lex at the University of Texas at Austin; Dr. E. John Wherry and Zeyu Chen at University of

Pennsylvania; Dr. Fang Han at University of Washington; and Dr. Stephanie Hicks, Dr. Ni Zhao, Dr. Xiaobing Wang, Dr. Xiumei Hong, and Dr. Guoying Wang at Johns Hopkins Bloomberg School of Public Health, and many others. I am honored to have the privilege of contributing to the scientific discoveries through these collaboration projects, and these collaborations have inspired many of my methodology work.

I would like to thank the Department of Biostatistics at Johns Hopkins Bloomberg School of Public Health for providing the extraordinary resource of education and research.

Finally, I would like to thank my wife and my parents for their huge support during the whole course of my graduate study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Gene expression is an essential process in all known life forms. Gene expression is the process where the information of a gene is used to synthesize functional gene products such as ribonucleic acid (RNA) and proteins. Gene expression can be regulated by a wide range of sophisticated mechanisms in almost all steps of the gene expression process including transcription [1], RNA splicing [2], and translation and post-translational modification of a protein [3]. Gene regulation controls the timing, location, and amount of gene products present in a cell, and it has a profound impact on the functions of cells. The regulation of gene expression is the basis of many biological processes such as cell differentiation and development. Misregulation of gene expression can cause many diseases including cancer, autoimmune disease, developmental disorder, diabetes, cardiovascular disease and others [4]. Thus, decoding the dynamic gene expression process and elucidating how gene expression is regulated are essential for understanding the mechanisms of these complex diseases and developing better intervention and treatment strategies.

Next-generation sequencing (NGS) [5] is a powerful tool to study and understand the gene expression process and how genes are regulated. The classical NGS technologies measure different perspectives of the gene expression process for a cell population at a given time point. For example, RNA sequencing (RNA-seq) [6] measures the transcriptome, which is the complete set of transcripts and their quantity in a sample. DNase I hypersensitive sites sequencing (DNase-seq) [7], assay for transposase-accessible chromatin sequencing (ATAC-seq) [8], and formaldehyde-assisted isolation of regulatory elements sequencing (FAIRE-seq) [9] measure genome-wide chromatin accessibility, which correlates with the degree to which nuclear macromolecules are able to physically contact chromatinized DNA [10]. Chromatin immunoprecipitation sequencing (ChIP-seq) [11] identifies the binding sites of DNA-associated proteins. Whole genome bisulfite sequencing (WGBS) [12] determines the DNA methylation status of single cytosines. NGS technologies have been widely applied in biomedical research. For example, the ENCODE project [13] has generated data using different types of NGS technologies for more than 16,000 samples by far, covering different species, cell types, tissues, developmental stage, and chemical treatments. This provides a rich resource for studying gene expression and gene regulation in a wide variety of biological contexts.

One limitation of the conventional NGS technology, or bulk sequencing technology, is that it only measures the averaged genomic information of a cell population. For example, DNase-seq usually requires one million cells [14] and ATAC-seq requires hundreds to thousands of cells [15]. They measure these cells' average behavior. However, in many situations, the genomic information for each cell is desired. For a heterogeneous cell population such as cancer, measuring the genomic information for each cell may reveal a clue for rare but critical cell subpopulation such as cancer stem cell [16]. This information is masked by bulk sequencing. Identifying such cell subpopulation may provide key insights into the failure of anticancer therapies [16]. Even for a homogeneous cell population, gene expression can also be heterogeneous due to stochastic variations [17]. The genomic information for each cell may help reveal how the stochastic events link to cell fate decisions [18]. Finally, it is very difficult, if not impossible, for bulk sequencing to study a cell population with a very small number of cells. For example, there are only a few precursor cells available of specific lineages from very early embryos [19]. Thus, it becomes crucial to develop new sequencing technologies that can measure the genomic information in a single cell. Such single-cell technology can help researchers to gain deeper understandings of basic biology and clinical practice which cannot be learned using bulk sequencing alone.

Thanks to the rapid development of the technology and experimental protocol, different types of single-cell sequencing technology have been developed in the past decade. The first generation of single-cell RNA-seq technologies was developed around ten years ago [20–22]. These technologies were able to measure the transcriptome in one or a few cells. After that, the number of single cells measured in a study increases exponentially [23]. Single-cell RNA-seq has now become a widely accessible tool to measure the transcriptome of hundreds of thousands of cells [24–26]. Single-cell RNA-seq has been widely applied in numerous fields of biomedical research, such as studying the dynamic changes of transcriptome in a developmental process [27], the immune cells in breast cancer tumor microenvironment [28], the developmental lineage of a whole animal [29], the spatial transcriptome of the mouse brain [30], and identifying new cell types in mouse kidney [31]. Meanwhile, single-cell sequencing technology to characterize the activities of all genomic regulatory elements, or regulome, starts to emerge around five years ago. Single-cell ATAC-seq (scATAC-seq [32, 33]) and single-cell DNase-seq (scDNase-seq [14]) are two technologies for measuring chromatin accessibility in single cells. Single-cell ChIP-seq (scChIP-seq [34]) measures histone modifications in single cells. These technologies have been used to uncover the composition of different cell types in a cell population, as well as to find a link between chromatin organization and cell-to-cell variation [32]. There are also technologies developed for mapping different -omics modalities simultaneously in single cells. For example, sci-CAR can jointly profile chromatin accessibility and mRNA in thousands of single cells [35]. Other types of single-cell sequencing technologies include single-cell genome sequencing [36], single-cell DNA methylation sequencing [37], and paired single-cell T cell receptor sequencing (TCR-seq) and RNA-seq [38].

However, analyzing data generated from single-cell sequencing technologies is an enormous challenge. Analytical methods developed for conventional bulk sequencing data usually are incapable of dealing with the unique challenges in analyzing single-cell data. First, new types of analytical tasks may emerge when analyzing data from single-cell sequencing. For example, to study a continuous biological process such as cell differentiation, one can collect time-course single-cell RNA-seq datasets from different experimental time points. However, cells may differentiate at different speeds, and the experimental time points may not represent the cells' true states in the differentiation trajectory. How to computationally order the cells to reflect the underlying biological process is a new challenge that is not seen in analyzing bulk sequencing data. Second, data from single-cell technology can be highly sparse and noisy. For example, data from 10x single-cell RNA-seq [24] can have as high as 90-95% of all gene expression measurements to be zero. Many of these zeroes are the so-called dropout events, where the observed zero read count does not represent the actual medium or high expression of the gene due to some technical bias. Data from single-cell ATAC-seq are even more sparse and are nearly binary. This happens because the diploid genome only has two copies of DNA and single-cell ATAC-seq only has a chance of one or two to capture the open chromatin regions. However, molecular events such as transcription factor binding to DNA is a temporal stochastic event. Thus, the highly sparse single-cell ATAC-seq signal cannot accurately describe the probability of the occurrence of molecular events, which is a continuous measure. Thus, it is essential to develop analytical methods to recover the true signal from the highly sparse and noisy single-cell sequencing data. Third, data from single-cell technology can be highly complex. Consider a gene expression dataset with multiple samples. For each sample, traditional bulk technology will generate a vector of gene expression values across all genes, while single-cell technology will generate a matrix of gene expression values across all genes and cells. Thus, while bulk data contains the variability across samples, single-cell data add an extra layer of variability across cells. More sophisticated analytical methods are needed to account for the extra complexity of the data. There are many other challenges as well, including how to develop scalable methods to analyze big single-cell datasets and how to integrate information from different single-cell data modalities.

To tackle these challenges, many novel statistical and computational methods have been developed. For example, pseudotime or trajectory analysis methods [39] have been proposed to tackle the new challenge of ordering cells computationally to infer their underlying biology. Imputation methods have been developed to tackle the high sparsity issue of the single-cell RNA-seq data [40]. New methods [41, 42] have also been created to handle the complex structure of the single-cell datasets. However, there are still many unsolved challenges. For example, although there are methods to impute single-cell RNA-seq data, there is no existing method to accurately recover the activities of genome-wide cis-regulatory elements (CREs).

This thesis introduces three methods we developed to tackle several open challenges in single-cell data analyses. TSCAN [43] is a pseudotime analysis method that computationally orders the cells and infers the underlying biological process. TSCAN reaches a better performance of pseudotime analysis by improving the bias-variance tradeoff in spanning tree inference. SCATE enhances highly sparse single-cell ATAC-seq signal and recovers activities of genome-wide CREs by adaptively integrating information from co-activated CREs, similar cells, and massive amounts of publicly available regulome data. RAISIN is a regression and differential analysis method that adequately accounts for the multi-level variance structure in single-cell RNA-seq data with multiple samples. We demonstrate that these methods can tackle the unique challenges in analyzing single-cell sequencing data with systematic benchmarking studies. We are applying these methods in several collaborative projects. Examples include studying the mechanisms of the T cell exhaustion process [44] and deciphering gene expression and gene regulation in various biological systems.

Chapter 2

TSCAN: Pseudo-time Reconstruction and Evaluation in Single-cell RNA-seq Analysis

2.1 Introduction

Single-cell RNA-seq is a transformative technology that allows researchers to measure transcriptomes of individual cells [20, 45]. Unlike single-cell RNA-seq, conventional RNA-seq (also referred to as "bulk RNA-seq") [6, 46] or microarray [47, 48] experiments are used to measure average gene expression of a cell population. In many applications, the cell population is heterogeneous and contains multiple cell types. As a result, the average transcriptome of the population may fail to capture important transcriptional signals in individual cells. Sometimes, using the population average to study cell type specific behavior can also be misleading due to Simpson's paradox [27, 49]. With the ability to measure the transcriptome of each individual cell, single-cell RNA-seq is capable of generating a higher resolution view of the gene expression landscape in a heterogeneous cell population [21, 22, 50]. This can lead to a more accurate molecular characterization of a complex biological phenomenon [51].

As demonstrated by [27], one useful way to gain biological insights from single-cell RNA-seq data is to computationally order cells according to the gradual transition of their transcriptomes. For example, in a cell differentiation process, cells can evolve at different speeds. A sample of cells collected at a particular time point during differentiation can actually contain cells representing different differentiation stages. Using single-cell RNA-seq data, one may construct an ordered sequence of cells to describe the gradual transition of the single-cell transcriptome. If this *in silico* order is consistent with cells' true differentiation stages, then by analyzing how gene expression changes along this ordered sequence of cells, one will be able to obtain insights on the transcriptome dynamics during the differentiation process. The process of ordering cells *in silico* is called pseudo-time reconstruction because it mimics a procedure that places cells on a time axis. Despite the use of the term "time", "pseudo-time reconstruction" can more generally refer to any cell ordering procedure regardless of whether the ordering has a time interpretation (e.g., the ordering of cells may reflect cells' spatial order rather than their temporal order).

Several computational methods have been proposed to analyze single-cell genomic data such as single-cell mass cytometry data [52–54] and single-cell gene expression data [27, 55–58]. However, for pseudo-time reconstruction in single-cell RNA-seq data, there are only a limited number of methods that have been systematically tested and have easily accessible software tools. In [27], an unsupervised approach Monocle was proposed to solve this problem. Monocle uses a minimum spanning tree (MST) to describe the transition structure among cells. The backbone of the tree is extracted to serve as the pseudo-time axis to place cells in order. A similar unsupervised spanning-tree approach has also been used previously for analyzing flow cytometry data [54]. As an unsupervised approach, pseudo-time reconstruction based on spanning trees does not require any prior information on cell ordering. When temporal order information is available, an alternative approach to analyzing single-cell gene expression dynamics is to use such information to supervise the analysis. An example of this supervised approach is SCUBA [55]. SCUBA uses bifurcation analysis to recover biological lineages from single-cell gene expression data collected from multiple time points. Here the multiple time points in a time course experiment are used to supervise the cell ordering and analyses of gene expression dynamics in cell differentiation processes. By using the available time information, supervised methods can be more accurate than unsupervised methods. However, in applications where time information is not available (e.g., if one needs to analyze a heterogeneous cell population from a single disease sample rather than from a time course experiment), the supervised approach is not applicable and one has to rely on unsupervised methods. For these reasons, both supervised and unsupervised methods are useful. The primary focus of this article is the unsupervised approach.

One potential limitation of Monocle is that its tree is constructed to connect individual cells. Since the cell number is large, the tree space is highly complex. Tree inference in such a complex space is associated with high variability and can be highly unstable. As a result, the optimal tree found by the algorithm may not represent cells' true biological order. This can be illustrated using a toy example in Figure 2-1A-C. Here dots represent cells placed in a two dimensional space (e.g., the space corresponding to the top two principal components of the gene expression profiles), and the true biological time runs top-down vertically. The MST solution is not unique. Figure 2-1A and Figure 2-1B show two possible solutions. When a slight measurement noise pushes the cell labeled by '*' away from other cells, the tree in Figure 2-1A can easily become a better solution based on the MST algorithm. However, this solution places cells in an order different from their true biological order. One approach that may alleviate this problem is to reduce the complexity of the tree space. This is analogous to the bias-variance tradeoff in the statistics and machine learning literature. For instance, if one clusters similar cells together as in Figure 2-1C and then constructs a tree to connect the cluster centers, recovering the true time-axis becomes easier. In this article, we exploit this idea to develop TSCAN, a new tool for pseudo-time reconstruction. One additional advantage offered by clustering cells is that users can more easily adjust the order of tree nodes (i.e., cell clusters) manually if they want to do so, since the number of clusters usually is not big. By contrast, manually specifying the order of hundreds of cells is much more difficult.

Another limitation of existing tools is that they are mostly command-line driven and do not allow users to interactively adjust or fine-tune the analysis. For example, users often want to use their existing knowledge such as marker genes to filter out contamination cells, determine the time origin, or manually change the order of certain tree nodes. However, these operations are not convenient for a command-line driven software tool such as Monocle. TSCAN addresses this limitation by providing a graphical user interface (GUI) (Figure 2-2). Using the GUI, users can interactively and conveniently incorporate prior biological information into the pseudo-time reconstruction analysis.

Last but not least, when several different pseudo-time reconstruction methods are available, being able to evaluate and compare them to identify the best solution is important. However, how to evaluate different pseudo-time reconstruction methods is also an open problem. Objective measures for comparing different methods are still lacking. This article introduces several quantitative measures for evaluating different cell ordering methods. Using these objective measures, we show that TSCAN is capable of providing more reliable unsupervised pseudo-time reconstruction results compared to alternative methods.

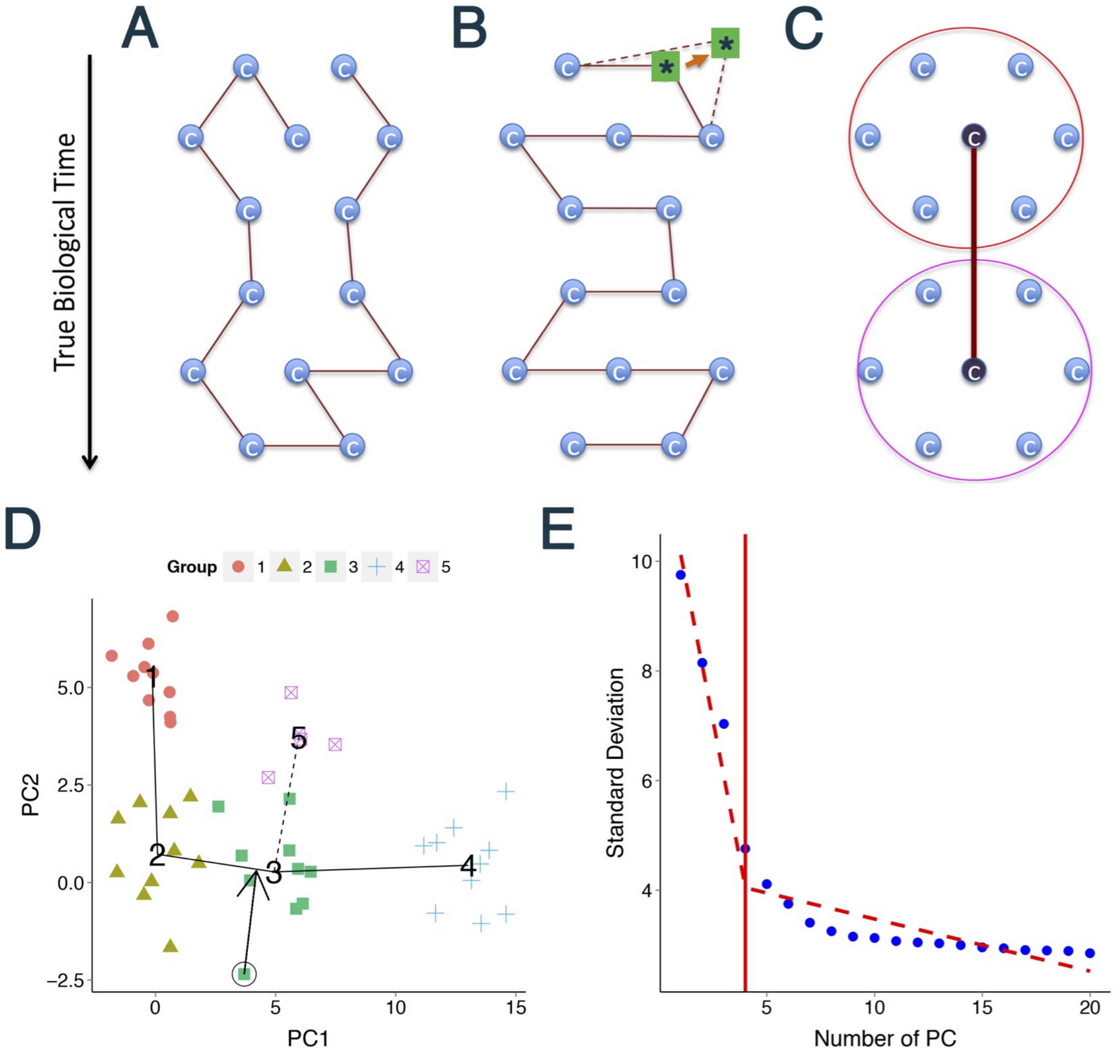
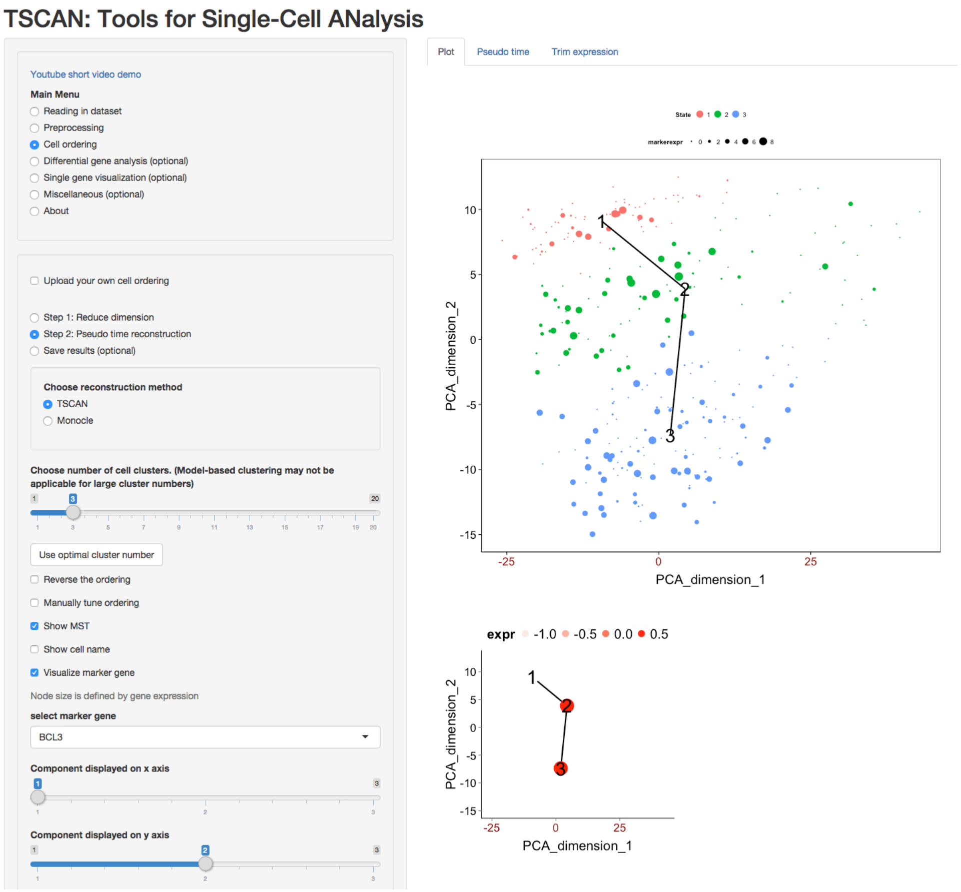


Figure 2-1. TSCAN Overview. (A-B) A toy example illustrating a limitation of cell-based MST. Here cells (blue circles) are placed in a two dimensional space, and the true biological time runs top-down. An MST that connects cells is not unique. Both (A) and (B) are possible solutions. (B) is more consistent with the truth. However, in reality, random measurement noise may shift the cell labeled by '*' away from other cells as indicated by the arrow and dashed lines. As a result, (B) is no longer an MST. The MST in (A) on the other hand does not reflect the true order of cells. (C) The true time-axis can be found if one first groups similar cells into clusters and then constructs an MST to connect cluster centers. (D) TSCAN first constructs cluster-based MST (five clusters of cells encoded by different colors are shown as an example; numbers indicate cluster centers). The tree can have multiple paths (e.g., 1-2-3-4 or 1-2-3-5). TSCAN orders cells along each path by projecting each cell onto the tree edge. (E) The number of principal components to retain is determined by finding the best piecewise linear fit consisting of two lines (dashed).



TSCAN: Tools for Single-Cell ANalysis

Youtube short video demo

Main Menu

○ Reading in dataset
○ Preprocessing
● Cell ordering
○ Differential gene analysis (optional)
○ Single gene visualization (optional)
○ Miscellaneous (optional)
○ About

☐ Upload your own cell ordering

○ Step 1: Reduce dimension
● Step 2: Pseudo time reconstruction
○ Save results (optional)

Choose reconstruction method
● TSCAN
○ Monocle

Choose number of cell clusters. (Model-based clustering may not be applicable for large cluster numbers): 3

☐ Reverse the ordering
☐ Manually tune ordering
☑ Show MST
☐ Show cell name
☑ Visualize marker gene

Node size is defined by gene expression

select marker gene: BCL3

Component displayed on x axis: 1

Component displayed on y axis: 2

Figure 2-2. TSCAN graphical user interface. Left panel contains function menus and tools for setting parameters. Right panel displays data and results. The top scatter plot shows the MST constructed for the LPS data (see Results). Cells (dots) are displayed based on their first two principal components. Clusters of cells are indicated by different colors. Numbers are cluster centers. Expression level of a marker gene BCL3 is shown for each cell. Larger marker size means higher expression. The bottom plot shows the average BCL3 expression for each tree node, standardized across all nodes to have zero mean and unit standard deviation.

2.2 Methods

2.2.1 Problem formulation

Consider a representative sample of N cells drawn from a heterogeneous cell population. Suppose the transcriptome $\mathbf{Y}_i$ of each cell $i \in \{1, 2, \ldots, N\}$ has been profiled using single-cell RNA-seq. Here $\mathbf{Y}_i$ is a G dimensional vector consisting of gene expression measurements for G genes. Assume that $\mathbf{Y}_i$ is appropriately transformed (e.g., by taking logarithm) and normalized across cells. The single cell ordering problem, also called pseudo-time reconstruction, is to place cells in an order based on the gradual transition of $\mathbf{Y}_i$.

TSCAN orders cells in three steps. First, cells with similar gene expression profiles are grouped into clusters. Second, a minimum spanning tree (MST) is constructed to connect all cluster centers. Finally, cells are projected to the tree backbone to determine their pseudo-time and order (Figure 2-1D). Once cells are ordered, users may use the ordered sequence to study cell state transition and gene expression dynamics in the underlying biological process from which the cells are sampled.

2.2.2 Preprocessing

Before pseudo-time reconstruction, the raw gene expression data are processed as follows. First, genes with zero read count in all samples are excluded. Second, in order to alleviate the effect of drop-out events [59] on the subsequent analyses, genes with similar expression patterns are grouped into clusters by hierarchical clustering (using Euclidean distance and complete linkage). The number of clusters is set to be 5% of the total number of genes with non-zero expression. For each cluster and each cell, the expression measurements of all genes in the cluster are averaged to produce a cluster-level expression which will be used for subsequent MST construction. The drop-out event refers to the phenomenon that expressed genes, some of which are highly expressed, may have zero read count in some cells as their molecules may not be captured and amplified by chance. This is a common phenomenon in single-cell RNA-seq data. By averaging across many genes, the cluster-level expression is more stable and has smaller estimation variance compared to the measurements of individual genes. This can help to dilute the impact of drop-out events.

After gene clustering, single-cell transcriptome for cell i becomes a H dimensional vector $\mathbf{E}_i$. Here H is the number of gene clusters. $\mathbf{E}_i$ still has high dimension, and many components in this vector are still correlated. The dimensionality makes visualization and statistical modeling difficult. For this reason, TSCAN further reduces the dimension of $\mathbf{E}_i$ using principal component analysis (PCA). Briefly, $\mathbf{E}_i$ from all cells are organized into a $H \times N$ matrix $\mathbf{E}$. Each row corresponds to a gene cluster. The matrix is standardized such that expression values within each row have zero mean and unit standard deviation. Then PCA is run on the standardized matrix, and the top K principal components (PCs) are retained. After PCA, the H dimensional vector $\mathbf{E}_i$ is mapped to a lower dimensional space and becomes a K dimensional vector $\tilde{\mathbf{E}}_i$. Here K is much smaller than H.

In order to determine K (i.e., how many PCs to retain), TSCAN uses the following criterion. First, let λ_i be the data variance explained by the i^{th} PC. Define $v_i \equiv \sqrt{\lambda_i}$. v_i is a non-increasing function of i. This function can be approximated using a continuous piecewise linear model $v_i = f(i) + \epsilon$ where ϵ represents noise and $f(i)$ consists of two regression lines (Figure 2-1E):

$$\begin{aligned} f(i) &= \begin{cases} \alpha_0 + \alpha_1 * i & \text{if } i \leq k \\ \beta_0 + \beta_1 * i & \text{if } i > k \end{cases} \\ s.t.\ \alpha_0 &+ \alpha_1 * k = \beta_0 + \beta_1 * k \end{aligned} \tag{2.1}$$

TSCAN computes the least squares fit of this model using the first 20 PCs. The fitted model varies when one changes k. TSCAN tries different $k \in [2, 19]$ and finds the k that produces the smallest squared error, $\sum_{i=1}^{20}[v_i - f(i)]^2$. This k will be used as the number of PCs to retain.

2.2.3 Cell clustering

After dimension reduction, cells with similar expression profiles are grouped into clusters using the model-based clustering approach described in [60]. The clustering is performed using the *mclust* [61] package in R which fits a mixture of multivariate normal distributions to the data $\tilde{\mathbf{E}}_i$. The variance-covariance matrix for each normal component in this mixture is designated as "ellipsoidal, varying volume, shape, and orientation". The number of clusters is chosen by *mclust* using the Bayesian Information Criterion (BIC). After model fitting, the posterior probability that each cell belongs to each cluster can be computed. Cells are assigned to clusters based on the largest posterior probability. For each cluster, the cluster mean of $\tilde{\mathbf{E}}_i$ is treated as the cluster center. Instead of using the cluster number determined by *mclust* based on BIC, users also have the option to specify their own cluster number.

2.2.4 Ordering cell clusters by MST

Next, TSCAN constructs a minimum spanning tree to connect all cluster centers. In a connected and undirected graph, a spanning tree is a subgraph that is a tree and connects all the vertices (or "nodes"). Suppose each edge in the graph has a length equal to the Euclidean distance between the two nodes (i.e., cluster centers) connected by the edge. A minimum spanning tree (MST) is a spanning tree with the smallest total edge length among all possible spanning trees. Unlike the MST approach used by Monocle where the tree is constructed to connect individual cells, the MST in TSCAN is constructed to connect clusters of cells. Clustering cells reduces the variability and complexity of the tree space. The cluster level MST therefore may yield better and more stable estimates of the tree backbone which largely determines the cell ordering. Another advantage of clustering is that it dramatically reduces the number of tree nodes, so that it becomes easier for users to interactively fine-tune the analysis later (e.g., manually adjust the order of tree nodes).

A tree may have multiple branches. By default, we define the main path of the tree (solid lines in Figure 2-1D) as the path with the largest number of clusters. If more than one path has the same largest number of clusters, the path with the largest number of cells becomes the main path. The main path has two ends. Without other information, one end will be randomly picked up as the origin of the path.

Alternatively, users can specify one end as the origin themselves using information such as marker gene expression. After the main path and its origin are determined, TSCAN will enumerate all branching paths starting from the origin. For instance, assume cluster 1 in Figure 2-1D is chosen as the origin, then TSCAN will report a main path 1-2-3-4 and a branching path 1-2-3-5. If the cluster order generated by the algorithm is not satisfactory to users, they have options to manually specify the paths and the order of clusters along each path.

2.2.5 Cell ordering and pseudo-time calculation

Once the cluster-level ordering is determined, individual cells are projected onto tree edges to create cell-level ordering along the main path and each branching path. For each path, all clusters on the path are collected. All cells in these clusters will be ordered along the path as follows. Let C_i $(i = 1, 2, ..., M)$ indicate the ordered clusters, where M is the number of clusters on the ordered path. Suppose $\tilde{\mathbf{E}}^{(i)}$ and $\tilde{\mathbf{E}}^{(j)}$ are the cluster centers for two neighboring clusters C_i and C_j in the path, and suppose C_i precedes C_j in the ordering. The edge that connects the two clusters is determined by $\mathbf{v}_{ij} = \tilde{\mathbf{E}}^{(j)} - \tilde{\mathbf{E}}^{(i)}$, and the projection of cell k to the edge is determined by the inner product $\mathbf{v}_{ij}^T \tilde{\mathbf{E}}_k / ||\mathbf{v}_{ij}||$ where $||.||$ is the l^2-norm of a vector. Cells in cluster C_1 are all projected onto the edge that connects C_1 and C_2. Cells in cluster C_M are all projected onto the edge that connects C_{M-1} and C_M. Cells from an intermediate cluster $C_m (1 < m < M)$ are divided into two groups according to whether they are closer to the center of cluster C_{m-1} or to the center of cluster C_{m+1} in terms of Euclidean distances. Cells closer to the center of cluster C_{m-1} are projected onto the edge that connects clusters C_{m-1} and C_m, while cells closer to the center of cluster C_{m+1} are mapped to the edge connecting clusters C_m and C_{m+1}.

Cell orderings are determined in three steps. First, for cells which are in the same cluster and are projected onto the same edge, their order is determined by the projected values on the edge. Second, within each cluster, the order of cells projected onto different edges is determined by the order of edges, which is given by the cluster-level ordering. Third, the order of cells in different clusters is determined by the order of clusters. In this way, all cells can be placed in order.

Once cells are ordered, pseudo-time is computed for each ordered path. For a given path, the order of a cell on the path is set to be its pseudo-time. For instance, the pseudo-time for the k^{th} cell on a path is set to k. The pseudo-time is constructed separately for the main path and each branching path.

2.2.6 Detecting differentially expressed genes

After cells are ordered, one can detect differentially expressed genes following the approach in Monocle [27]. A generalized additive model (GAM, effective degrees of freedom = 3) [62] is fitted for each gene to describe the functional relationship between its expression and pseudo-time. The GAM is fitted using the *mgcv* [62] package in R. The model is then compared to a null model that assumes constant expression along the pseudo-temporal path. The p-value is computed using a likelihood ratio test and then converted to false discovery rate (FDR) using the method in [63]. By default, genes with FDR < 0.05 are reported as differential. As in Monocle, the p-value and FDR are computed based on assuming that cell ordering is given. They do not consider uncertainties in cell ordering and that, instead of being determined by experiment design, cell ordering is derived from the same data used for analyzing differential expression. We note that how to evaluate statistical significance that further accounts for these additional uncertainties remains an open problem. It requires development of more sophisticated methods and a systematic investigation of how these additional uncertainties affect different methods (e.g., how p-values change when one treats cell ordering as an unknown parameter inferred from the data). These investigations are beyond the scope of the current study as the main focus of this article is how to improve and evaluate cell ordering.

2.2.7 Method evaluation

We use three methods to evaluate cell ordering performance. The first approach evaluates cell ordering accuracy based on the ordering expected by independent sources of information. It is assumed that external information not used in pseudo-time reconstruction is available to evaluate the pairwise order of cells. Formally, let π denote an ordered path of N_π cells produced by a particular pseudo-time reconstruction method. Let $g(\pi, i, j)$ be a score that characterizes how well the order of the i^{th} and j^{th} cells in the ordered path π matches their expected order based on the external information. We define Pseudo-temporal Ordering Score (POS) for cell ordering π as the sum of $g(\pi, i, j)$ for all pairs of cells:

$$POS_\pi = \sum_{i=1}^{N_\pi - 1} \sum_{j:j>i} g(\pi, i, j) \tag{2.2}$$

Cell orderings π produced by different pseudo-time reconstruction methods can then be compared based on the POS score.

As a concrete example, suppose one has single-cell RNA-seq data collected from a time course experiment. In such an experiment, the data collection time is known. For the purpose of evaluating unsupervised pseudo-time reconstruction methods, one can pool cells from all time points together, pretend that the data collection time for each cell is unknown, and apply different methods to reconstruct pseudo-time. Different methods will then be evaluated by comparing their cell ordering results to the order of cells based on the true data collection time. For instance, if one has N cells collected at V time points during a differentiation process. Among the N cells, N_v cells are from time T_v $(T_1 < T_2 < \cdots < T_V)$. Consider the i^{th} cell and the j^{th} cell in the ordered path π where i precedes j (i.e., $i < j$). One can define the pairwise score $g(\pi, i, j)$ as follows:

1. If the two cells are originally collected at the same time point (e.g., they are both from T_v), then $g(\pi, i, j) = 0$.

2. Otherwise, if the i^{th} cell is collected from time point T_v and the j^{th} cell is collected from time point T_u, then $g(\pi, i, j) = (u - v)/D_\pi$. The value $u - v$ is positive if v represents an earlier time point, or negative if v represents a time later than u.

The denominator D_π above is chosen to normalize POS so that $POS_\pi \in [-1, 1]$ (i.e., the maximal and minimal POS among all possible orderings of cells within each path π is 1 and -1 respectively). Based on this definition, a cell ordering more consistent with the known data collection time will have higher POS score. $POS_\pi = 1$ indicates that the order of cells produced by pseudo-time reconstruction perfectly matches the order determined by the data collection time. $POS_\pi = -1$ indicates that the order of cells produced by pseudo-time reconstruction is in the opposite direction compared to the order determined by the data collection time. Using POS to evaluate cell ordering is based on assuming that the external information (i.e., the true data collection time in this example) can roughly reflect the true biological order of cells (e.g., the differentiation stage of cells). In reality, since cells collected at each time point are heterogeneous, it is possible that some cells collected at an earlier (less differentiated) time point in the differentiation time course are actually more differentiated than certain cells collected at a later time point. Despite this, it is often reasonable to expect that cells collected at the earlier time point "on average" should be less differentiated than cells collected at the later time point. Therefore, the external information (i.e., the data collection time) used here can still roughly reflect the true biological order of cells and can be used as a surrogate to evaluate the cell ordering performance.

The second approach evaluates robustness of cell ordering by perturbing the original single-cell RNA-seq dataset (see below). Each cell ordering method is applied to both the original dataset and the perturbed data. Cell orderings produced by the original and perturbed data are then compared. To quantify the similarity between cell orderings in two pseudo-temporal paths π_1 and π_2, let A be the union of cells in π_1 and π_2, let $|A|$ be the cardinality of A (i.e., the number of distinct cells in π_1 and π_2), and define the similarity score between π_1 and π_2 as:

$$s_{\pi_1,\pi_2} = \frac{2}{|A|(|A|-1)} \sum_{i,j \in A; i \neq j} h(\pi_1, \pi_2, i, j) \tag{2.3}$$

Here $h(\pi_1, \pi_2, i, j) = 1$ if the order of two cells i and j remains the same in π_1 and π_2 (i.e., i appears before or after j in both orderings), and $h(\pi_1, \pi_2, i, j) = 0$ otherwise. If either i or j occurs only in one path (e.g., i is in π_1 but not π_2), the orderings between i and j in π_1 and π_2 are viewed as inconsistent, and $h(\pi_1, \pi_2, i, j)$ is also set to zero. A higher similarity score indicates that the two orderings π_1 and π_2 are more similar to each other, whereas a lower score indicates a larger deviation between the two orderings.

In this article, two different approaches were used to perturb data: cell-level perturbation and expression-level perturbation. For cell-level perturbation, x percent (x = 95%, 90% or 75%) of cells were randomly sampled from the original dataset to serve as the perturbed data. The gene expression profile of each cell remained unchanged. For expression-level perturbation, we retained all cells in the original dataset but added simulated noise to their gene expression profiles (i.e., $\mathbf{Y}$). To generate noise, the average expression value of each gene across all cells was computed and then subtracted from the gene's expression value in each cell. Residuals obtained in this way were scaled by multiplying with a scaling factor κ (κ = 5%, 10% or 25%). The scaled residuals were then permuted and added back to the original expression values of the gene. For each perturbation method and parameter value (x or κ), the original data were independently perturbed 100 times to generate 100 perturbed datasets. For each perturbed dataset, similarity score between the original and perturbed orderings was computed. Finally, the average similarity score from the 100 perturbations was calculated to measure the robustness of each pseudo-time reconstruction method.

The third approach evaluates the ability of a cell ordering method to detect known differentially expressed genes along the ordered cell path. Given a test dataset, one can collect genes known to be differentially expressed along the biologically ordered sequence of cells and treat them as the gold standard. One can then detect differential genes along the pseudo-time axis and compare different methods based on how they rank gold standard genes.

2.2.8 TSCAN package and GUI

TSCAN is implemented as a Bioconductor package using the statistical programming language R. It can be run both in a command-line mode and through a graphical user interface (GUI). The GUI is developed using the shiny package in R. It allows users to conveniently construct, visualize and tune cell ordering. For example, one can use the GUI to interactively trim unwanted cells based on expression levels of user-specified marker genes. One can also change the cluster-level ordering and then recompute the pseudo-time. TSCAN is open source, and it is freely available at https://github.com/zji90/TSCAN. Its bioconductor package can be downloaded from http://www.bioconductor.org/packages/release/bioc/html/ TSCAN.html.

2.2.9 Datasets

Three datasets were compiled from the literature to evaluate TSCAN. The first dataset consists of single-cell RNA-seq samples from differentiating human skeletal muscle myoblasts (HSMM) [27]. It contains 271 cells collected at 0, 24, 48 and 72 hours (hrs) after switching human myoblasts to low serum. The second dataset consists of single-cell RNA-seq samples collected after stimulating bone-marrow-derived dendritic cells by lipopolysaccharide (LPS) [64]. 306 cells collected at 1, 2, 4 and 6 hrs after the stimulation were used for our analysis. The third dataset consists of single-cell RNA-seq samples from hippocampal quiescent neural stem cells (qNSC) [65]. It contains 172 cells collected from the same cell population. For all datasets, the normalized gene expression values (fragments per kilo base pairs per million total reads for HSMM and transcripts per million total reads for LPS and qNSC) were log2 transformed after adding a pseudo-count of 1. After the raw data $\mathbf{Y}_i$ were processed to $\mathbf{E}_i$, $\mathbf{E}_i$ was used as input for different methods (i.e., TSCAN, Monocle, Waterfall, SCUBA and Wanderlust below) to construct pseudo-time. The normalized data for $\mathbf{Y}_i$ and $\mathbf{E}_i$ are available at the TSCAN GitHub website (https://github.com/zji90/TSCANdata).

2.2.10 Comparisons with other methods

Table 2-I compares TSCAN with a number of other single cell data analysis methods. Among these methods, MARS-seq [56] and SINCE-PCR [58] do not have associated software for others to use. SPADE [54] and viSNE [52] are developed for analyzing mass cytometry or flow cytometry data, and they do not provide a cell ordering function. Diffusion map [66] is a dimension reduction technique used to define differentiation trajectories. It cannot perform cell ordering itself. The scLVM method [57] primarily focuses on identifying cell subpopulations. Again, it cannot order cells. For the above reasons, these methods are not compared with TSCAN in our subsequent data analyses.

Among the remaining methods, Monocle is designed to handle unsupervised cell ordering of single-cell RNA-seq and has a software package. Wanderlust [53] is originally developed for mass or flow cytometry data. It uses a graph-based trajectory detection algorithm to order cells under the assumption that there is no branch. We tailored its MATLAB code to allow it to take single-cell RNA-seq data as input. SCUBA [55], as discussed before, is a supervised approach. However, the SCUBA package also provides an option for unsupervised cell ordering which is based on fitting a principal curve to the data and then mapping cells onto the curve. Waterfall is a data

Method	Data	Unsupervised	Cell Collection Time Information	Pseudo-time Reconstruction	Allow Branching Structure	Ready-to-use Software Package
TSCAN	Single-cell RNA-seq	YES	Not Required	YES	YES	YES
Monocle	Single-cell RNA-seq	YES	Not Required	YES	YES	YES
Waterfall	Single-cell RNA-seq	YES	Not Required	YES	YES	NO
Difussion map	Single-cell RNA-seq	YES	Not Required	NO	NA	YES
SCUBA (bifurcation)	Single-cell RNA-seq	NO	Required	YES	YES	YES
SCUBA (principal curve)	Single-cell RNA-seq	YES	Not Required	YES	NO	YES
MARS-seq	Single-cell RNA-seq	YES	Not Required	NO	NA	NO
scLVM	Single-cell RNA-seq	YES	Not Required	NO	NA	YES
SINCE-PCR	Single-cell PCR	YES	Not Required	NO	NA	NO
SPADE	Mass Cytometry and Flow Cytometry	YES	Not Required	NO	YES	YES
Wanderlust	Mass Cytometry and Flow Cytometry	YES	Not Required	YES	NO	YES
viSNE	Mass Cytometry and Flow Cytometry	YES	Not Required	NO	NA	YES

Table 2-I. Comparison of TSCAN and other single cell data analysis methods

analysis pipeline used by [65] to construct pseudo-time for their qNSC data. Similar to TSCAN, Waterfall first groups cells using k-means clustering before pseudo-time reconstruction. However, as an in-house data analysis pipeline, Waterfall does not have an associated software tool, and the pipeline cannot be directly used to analyze other datasets without manually editing the code. Also, an objective evaluation of the effects of cell clustering on cell ordering was not provided in [65]. A systematic comparison among different pseudo-time reconstruction methods discussed above is still lacking. In order to benchmark the unsupervised cell ordering performance of TSCAN, we compared it with Monocle, Wanderlust, unsupervised SCUBA and Waterfall in our subsequent data analyses.

2.3 Results

We evaluated TSCAN using the three datasets, HSMM, LPS and qNSC, described above. HSMM and LPS datasets contain cells collected from multiple time points in time course experiments. The actual data collection time provides important external information for evaluating cell orderings produced by unsupervised pseudo-time reconstruction methods. In our evaluation, cells from different time points were pooled together. We pretended that their data collection time were unknown. We applied different pseudo-time reconstruction methods to order these cells. Methods were then compared in terms of their accuracy, robustness and ability to detect known differentially expressed genes. Accuracy was characterized by the POS score computed using cells' actual data collection time. Robustness was characterized by the cell ordering similarity between the original and perturbed data. In the qNSC dataset, all cells were collected from the same cell population. Since there was no external information such as multiple time points to calculate the POS score, we only evaluated robustness and the ability to detect known differentially expressed genes in this dataset.

2.3.1 HSMM analysis using *a priori* chosen genes for pseudo-time reconstruction

We first evaluated the performance of TSCAN using the HSMM dataset, originally analyzed by [27] using Monocle. In the original Monocle analysis conducted by [27], the pseudo-time was constructed using 518 genes chosen *a priori* before ordering the single-cell RNA-seq data. These genes were derived by comparing different differentiation time points and therefore are known to be associated with myoblast differentiation. They represent a strong piece of prior knowledge for pseudo-time reconstruction. In real applications, if one has strong prior information such as these 518 genes, one can use them as the input (to replace $\mathbf{E}_i$) for TSCAN and Monocle to construct MST. We first performed analyses in this way by using the same 518 genes for pseudo-time reconstruction. Figure 2-3A and Figure 2-3B show the cluster-level MST constructed by TSCAN. Consistent with the original Monocle results reported in [27], TSCAN also detected two branches of biological process: the default main path 1-3-5-2 and a branching path 1-3-5-4. For the main path 1-3-5-2, neither Monocle nor TSCAN can determine whether node 1 or 2 should be the starting time point without other information. Therefore, the path has two possible directions. By default, TSCAN randomly picks one direction. However, if users have marker genes to inform the direction of the pseudo-temporal path, they can use this information in TSCAN. To illustrate, ENO3 is a marker gene for myoblast differentiation. Its expression is expected to increase as the differentiation progresses. After providing ENO3 as a marker gene, TSCAN displays its expression in each tree node. In this way, one can see that cluster 1 has low ENO3 expression while cluster 2 has high ENO3 expression (Figure 2-3C). Thus, the starting time point should be in cluster 1. As reported in [27], the branching path in the MST constructed by Monocle was driven by contaminating interstitial mesenchymal cells, and SPHK1 is a marker gene for these contaminating cells. Consistent with this, displaying SPHK1 expression in the

TSCAN tree nodes shows that cluster 4 in the branching path 1-3-5-4 had high SPHK1 expression (Figure 2-3D), indicating that this branch was driven by contaminating cells. Thus, the branching path 1-3-5-4 was not further analyzed.

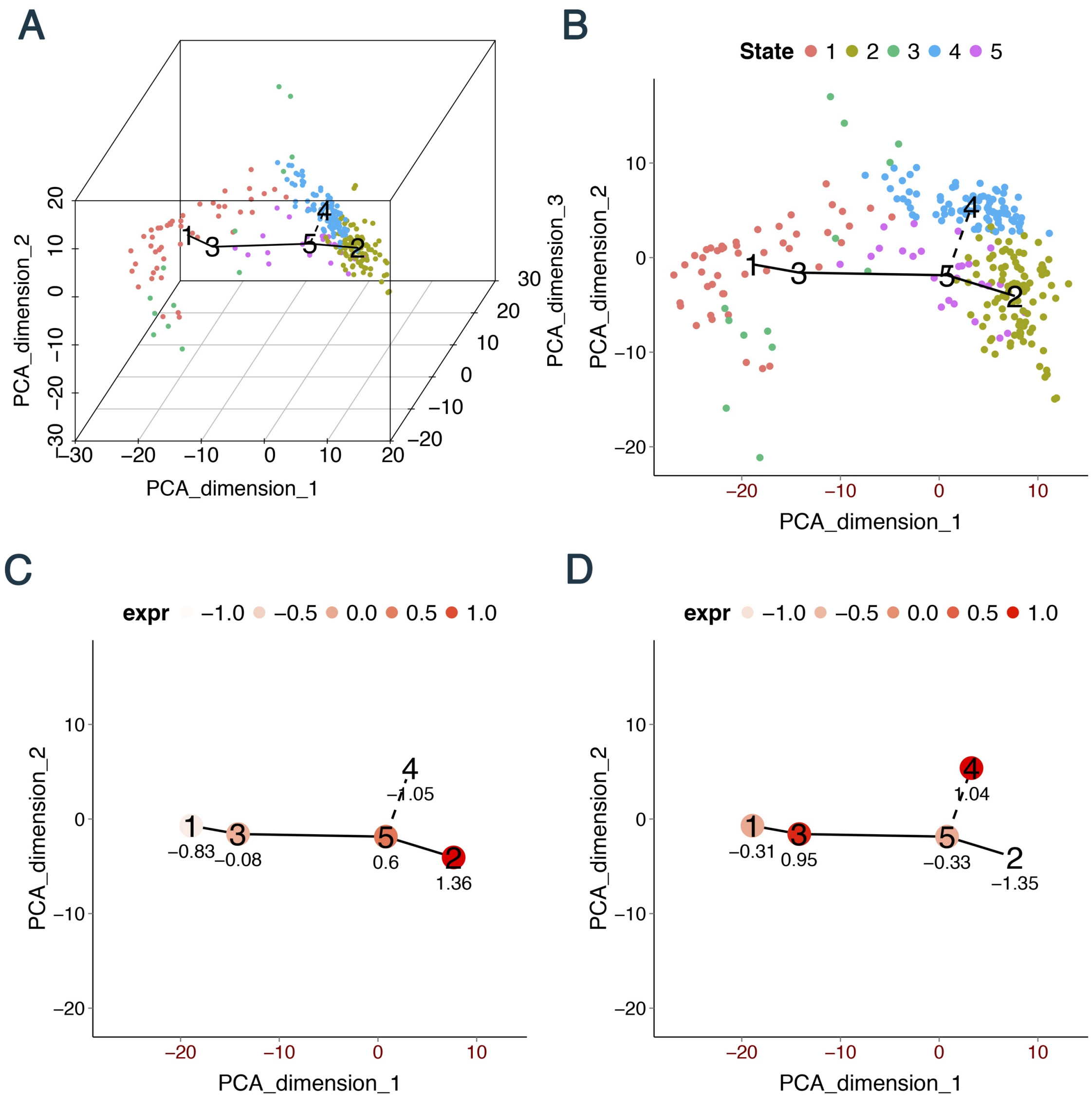


Figure 2-3. TSCAN analysis in HSMM dataset using 518 *a priori* chosen genes for pseudo-time reconstruction. (A) MST reported by TSCAN is shown in the 3 dimensional space spanned by the first three principal components (PCs) of $\mathbf{E}$. (B) Users can display cells and MST in chosen PCs (e.g., PC1 and PC2). (C) Mean expression level of ENO3 in each cluster. (D) Mean expression level of SPHK1 in each cluster. Values in (C) and (D) are both standardized across all clusters to have zero mean and unit SD.

For both Monocle and TSCAN, we calculated the POS score along their reported main path. According to [27], the main path produced by Monocle in this analysis corresponds to myoblast differentiation which is the biological process of interest. Figure 2-4A shows the POS scores. TSCAN outperformed Monocle in terms of the POS.

In order to understand how cell clustering affects the cell ordering performance, we tested a modified TSCAN (nocluTSCAN) in which the cell clustering step was skipped and MST was constructed directly to connect individual cells based on $\tilde{\mathbf{E}}_i$. The analyzed path and direction were then determined as above by using SPHK1 to exclude the contamination path and using ENO3 to determine the time origin. The comparison between TSCAN and nocluTSCAN was well-controlled since everything was the same for these two algorithms except for the use of cell clustering by TSCAN. By contrast, the performance difference between Monocle and TSCAN represents a combined effect of many factors since many of their implementation details are different. Many of these differences are difficult to control for as they are hidden in the computer code.

We also tested a marker-gene-only approach (marker) in which cells are directly ordered using the expression level of a marker gene (ENO3). Here, in order to conduct a relatively fair comparison with TSCAN, the marker-gene-only approach was only applied to cells from the analyzed TSCAN path (i.e., 1-3-5-2), and cells from the contaminated TSCAN branch (i.e. the branch with cluster 4) were excluded from this analysis. The comparison between the marker-gene-only approach and TSCAN can reveal whether the other genes used for pseudo-time reconstruction contribute additional information not provided by the marker gene (i.e., ENO3 in this example) for ordering cells.

As shown by Figure 2-4A, TSCAN had the best performance based on POS. It not only performed better than Monocle, but also outperformed nocluTSCAN and the marker-only approach, indicating that cell clustering and using multiple genes for ordering cells were both helpful for improving the pseudo-time reconstruction.

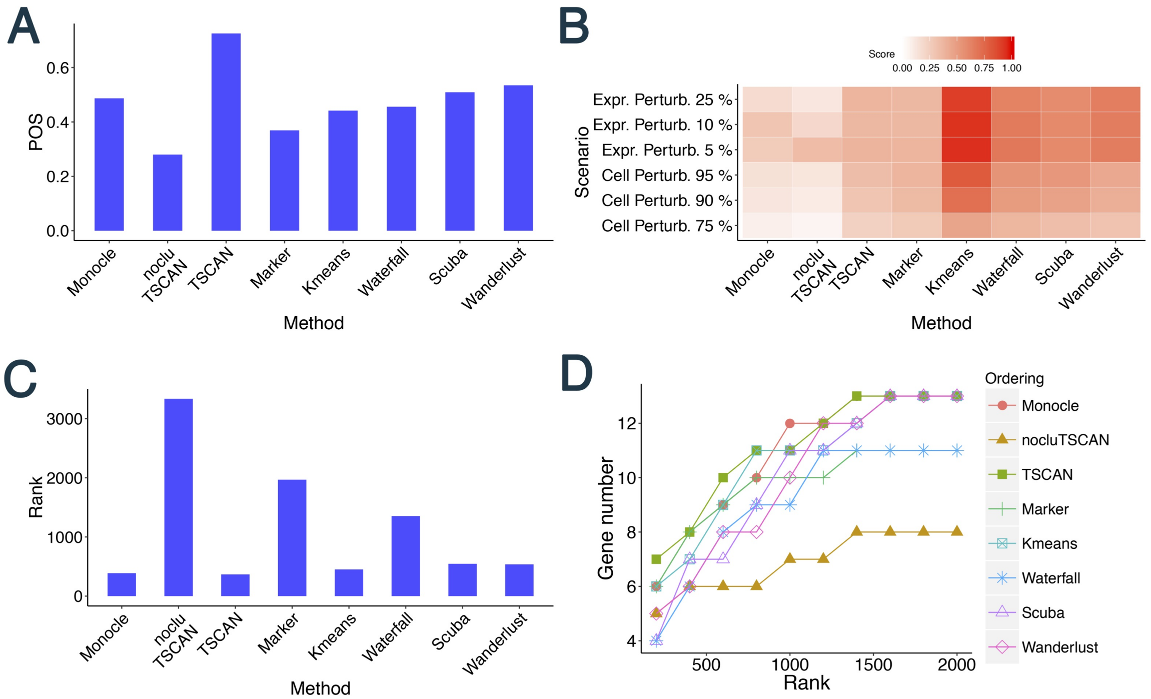


Figure 2-4. Evaluation results for different methods in HSMM dataset where pseudo-time was constructed based on 518 *a priori* chosen genes. (A) POS score. (B) Robustness measured by the average similarity score from 100 independent perturbations. The heat map shows robustness of each method in each perturbation scheme. Cell Perturb: cell-level perturbation. Expr Perturb: expression-level pertubation. (C) Mean rank of gold standard genes. (D) Number of detected gold standard genes among top differential genes.

Next, we compared robustness of different methods based on cell ordering similarity between the original and perturbed data. Figure 2-4B shows the similarity scores when the perturbed data were generated by randomly subsampling 75%, 90% or 95% of cells from the original dataset (cell-level perturbation) or by adding 5%, 10% or 25% random noise to the original gene expression values (expression-level perturbation). For each perturbed dataset, the same protocol and marker genes as described above were used to determine the path direction and eliminate contaminating branch. Compared to Monocle and nocluTSCAN, TSCAN consistently produced higher similarity scores in all perturbation schemes (Figure 2-4B). This shows that cell clustering increased the stability (or equivalently, reduced the variability) of cell ordering when data were perturbed. The marker-gene-only approach was also more robust than Monocle and nocluTSCAN, and it showed similar level of robustness compared to TSCAN (Figure 2-4B). The robustness of the marker gene approach was not unexpected. For cell-level perturbation, genes' expression values in each cell did not change. Consequently, the order of any pair of cells based on a marker gene's expression remained the same. The difference between the pseudo-temporal path in the original data and the path in the perturbed data in the marker gene approach mainly reflects the fact that these two paths did not contain the same set of cells. Note that not all cells in the original data were retained in the perturbed dataset. Also, contaminating branches of MST constructed by TSCAN were excluded from our marker-gene-only analyses, and the contaminating branches in the original and perturbed data could contain different sets of cells. For expression-level perturbation, noises added to gene expression values represented 5-25% of the cross-cell variation of the true biological signal. Consequently, the pairwise order of many cells was still driven by the biological variation and hence remained unchanged in the marker-gene-based ordering.

It is important to point out that robustness alone is not sufficient to indicate good cell ordering performance. For instance, suppose each cell has an arbitrary name. If cells are ordered based on cell name rather than gene expression profile, the order of any pair of cells will remain the same regardless of how gene expression values are perturbed. As a result, the cell ordering is robust, but it does not have any biological meaning since the cell names are arbitrary. This is similar to the well-known variance-bias tradeoff in statistics: an estimator with zero variance may have huge bias. For this reason, robustness of a pseudo-time reconstruction method needs to be interpreted in the context of whether it leads to improved cell ordering accuracy (e.g., increased POS score). Although the marker-gene-only approach was more robust than Monocle and nocluTSCAN (Figure 2-4B), its cell ordering accuracy was lower than Monocle and TSCAN (Figure 2-4A), indicating that its bias-variance tradeoff is not optimal. By contrast, TSCAN was not only more robust (Figure 2-4B) but also ordered cells more accurately (Figure 2-4A) than Monocle and nocluTSCAN.

For each method, we next detected differentially expressed genes along the ordered main path of cells. We ranked genes based on FDR, and then different methods were compared based on their ability to find genes known to be involved in the biological process in question. For the HSMM dataset, we compiled 13 genes (ENO3 excluded) known to be involved in myoblast differentiation according to [27]. Figure 2-4C shows the mean rank of these gold standard genes in the differential gene analysis. A smaller mean rank indicates better performance (i.e., gold standard genes are more likely to be ranked on top). Figure 2-4D shows the number of gold standard genes found in the top $200, 400, \ldots, 2000$ genes ranked by each method. Monocle and TSCAN had very similar results in this analysis, and both methods outperformed nocluTSCAN and the marker gene approach.

Besides TSCAN, we investigated two other ways to perform cell-clustering-based pseudo-time reconstruction. First, we replaced mclust by k-means clustering in the cell clustering step of TSCAN while keeping all other procedures the same (k-means TSCAN). Unlike mclust which allows ellipsoidal shape of clusters, k-means clustering only allows clusters with circle shape. In order to determine the cluster number of k-means, we used an approach similar to Figure 2-1E, with its y-axis changed to the proportion of total data variance unexplained by the cluster structure. Second, we tested the Waterfall algorithm [65] which also uses k-means to cluster cells before cell ordering. Waterfall does not provide a way to choose cluster number based on the data. Its cluster number was fixed to 10 which is the default value in Waterfall codes. Both the k-means TSCAN and Waterfall produced more robust cell ordering than Monocle and nocluTSCAN (Figure 2-4B). However, their cell ordering accuracy did not outperform Monocle and was clearly worse than TSCAN, as indicated by the

POS score (Figure 4A) and differential gene detection performance (Figure 2-4C,D). This suggests that although k-means TSCAN and Waterfall reduced the cell ordering variability, their bias-variance tradeoff was not optimal for improving the cell ordering accuracy.

We also tested unsupervised SCUBA (i.e., the principal-curve-based SCUBA) and Wanderlust. For SCUBA, low expression of the marker gene ENO3 was used to determine the path origin. Wanderlust was run by using the cell with the highest ENO3 gene expression as the path origin (because the lowest ENO3 expression was zero, and zero occurred in many cells, making the choice of path origin not unique). The cell ordering reported by Wanderlust was then reversed so that the reversed path had low ENO3 expression at the beginning and high ENO3 expression at the end. The same approach was also used in other test datasets below to run the Wanderlust analyses. For both methods, after cells were ordered, GAM was used to detect differentially expressed genes as in TSCAN. Both Wanderlust and SCUBA were more robust than Monocle and nocluTSCAN (Figure 2-4B). However, they both had lower cell ordering accuracy compared to TSCAN (Figure 2-4A,C,D). In fact, TSCAN produced the highest POS score (Figure 2-4A) and best differential gene detection performance (Figure 2-4C,D).

As demonstrated in [27], cell orderings based on pseudo-time may reveal gene expression patterns that cannot be discovered by bulk gene expression data. MEF2C and MYH2 are two genes involved in the HSMM differentiation. It is known that these two genes should have increasing expression during the differentiation, and the expression of MEF2C should start increasing earlier than the increase of MYH2 [27]. Based on the average bulk gene expression at different time points, it was not clear that MEF2C had a monotone increasing pattern, nor was it clear which gene started to increase first (Figure 2-5). By contrast, all single-cell analysis methods tested here were able to recover the overall increasing pattern of MEF2C and MYH2 along their analyzed pseudo-time axes, although in Monocle, k-means TSCAN, Waterfall, SCUBA and Wanderlust, MEF2C decreased a little before increasing (Figure 2-6). Compared to the other methods, the temporal expression curves fitted by TSCAN and nocluTSCAN more clearly showed that MEF2C increased earlier than the increase of MYH2 (Figure 2-6).

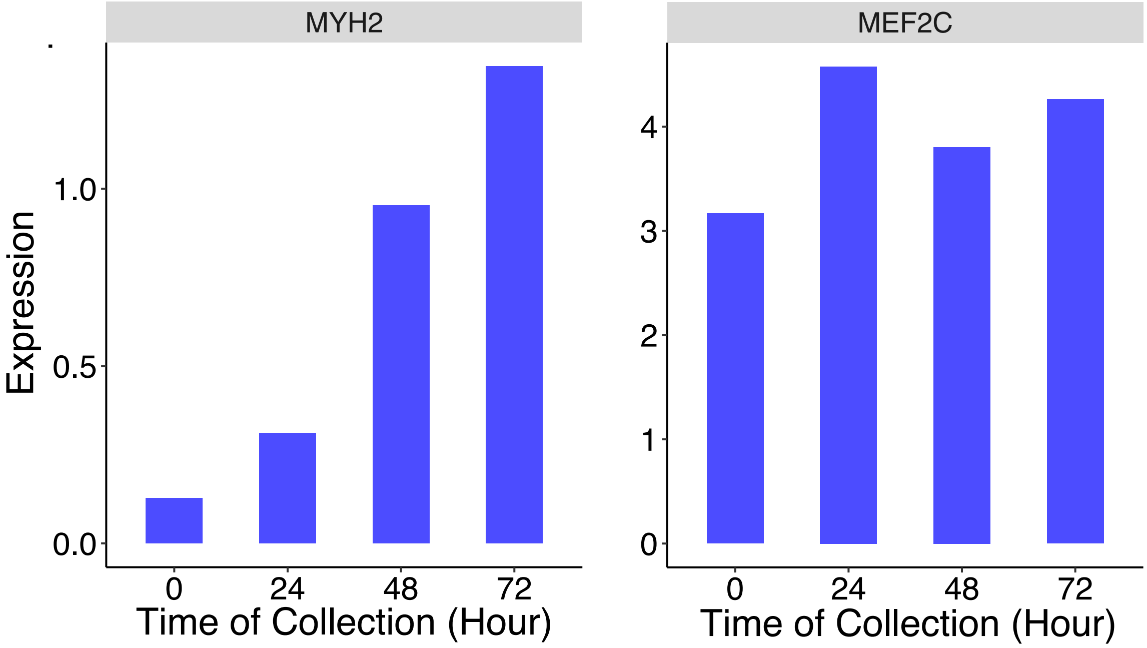


Figure 2-5. Averaged bulk gene expression level for MEF2C and MYH2 in HSMM data.

Based on all the analyses above, TSCAN was the method that provided the best overall performance. It offered the best cell ordering accuracy among all tested methods and improved cell ordering robustness compared to methods without using cell clustering (i.e., Monocle and nocluTSCAN).

2.3.2 HSMM analysis without using *a priori* chosen genes for pseudo-time reconstruction

In real applications, the prior information for pseudo-time reconstruction such as the 518 genes used above is not always available. When no such prior information is available, pseudo-time reconstruction has to rely on all genes in the RNA-seq data.

To evaluate the performance of TSCAN in such a scenario, we repeated the previous analysis but constructed pseudo-time without using the 518 *a priori* chosen genes. Instead, the $\mathbf{E}_i$ used for TSCAN was derived from all genes in the single-cell RNA-seq data using the protocol described in Methods. We also used $\mathbf{E}_i$ instead of $\mathbf{Y}_i$ as the input for Monocle, Waterfall, SCUBA and Wanderlust in order to make the method comparison relatively fair. Of note, the dimensionality of $\mathbf{Y}_i$ was also beyond the capacity that the Monocle software was able to handle.

The default main path given by TSCAN (Figure 2-7A, path 3-1-2) contained a cluster of cells with high expression in SPHK1 (Figure 2-7D), indicating that the main path was contaminated by interstitial mesenchymal cells and may not reflect myoblast differentiation. In such a scenario, TSCAN allows users to manually tune the analysis. For instance, with the GUI, one can conveniently visualize the expression of marker genes (Figure 2-7B) such as SPHK1 (Figure 2-7D, marker for contamination) and ENO3 (Figure 2-7E, marker for myoblast differentiation). Since SPHK1 is highly expressed in cluster 3, we chose to study path 2-1-4 which represents the myoblast differentiation. According to the increasing ENO3 pattern, one can specify that cluster 2 should be the path origin. Alternatively, one can also manually define a path by specifying the clusters and their order in the path (Figure 2-7C). In this example, both ways yielded the same path 2-1-4. Similar to TSCAN, the main path in Monocle was also contaminated by cells with high SPHK1 expression. However, Monocle does not provide an interface to help users conveniently incorporate such marker gene information and tune ordering. Users would need to be experienced in programming in order to adjust the analysis. In comparison, the TSCAN GUI allows users unfamiliar with programming to visualize and tune the ordering. Therefore, it lowers the bar for users to customize the pseudo-time analyses and can save them time and effort.

After using high expression of SPHK1 to exclude the contaminating branch and using low expression of ENO3 to determine the origin of the pseudo-temporal path for each method, different methods were then compared.

In terms of cell ordering accuracy, TSCAN had the highest POS score (Figure 2-8A) and the best mean rank of gold standard genes (Figure 2-8C) among all methods. It also had the highest power for detecting the gold standard differential genes (Figure 2-8D). In terms of robustness, methods based on cell clustering (TSCAN, k-means TSCAN, Waterfall) were more robust than methods that did not use cell clustering (Monocle, nocluTSCAN), as shown by the increased similarity scores between the original and perturbed data (Figure 2-8B).

Besides comparing cell orderings from the original and perturbed data, we also compared cell orderings constructed using and not using the 518 prior genes. To do so, similarity score between the cell ordering reported in this section and the ordering reported in the previous section was computed for each method. Figure 2-9A shows that TSCAN and the marker gene approach produced higher similarity scores than other methods, suggesting that they produced the most consistent cell ordering results. For each method, we also compared the consistency of differentially expressed genes detected by using and not using the 518 prior genes for pseudo-time reconstruction. For each analysis (i.e., using or not using the 518 prior genes), we obtained the top R ranked differential genes. The number of common genes between these two analyses was then counted and plotted as a function of R in Figure 2-9B. Figure 2-9C shows a similar analysis with a more stringent definition of common genes. Here, any gene that did not change in the same direction along the two pseudo-temporal paths (i.e., the fitted GAM functions from the two analyses have negative correlation) was not counted as a common gene even if the gene was identified by both analyses among their top R genes. After excluding these inconsistent genes from the common gene list, the number of genes remained in the common gene list was then shown as a function of R. In both Supplementary Figures 3B and 3C, TSCAN and the marker gene approach showed higher consistency than the other methods. Compared to the marker gene approach, TSCAN cell ordering was more accurate according to the POS score and differential gene detection performance (Figure 2-8A,C,D). Thus, our results show that TSCAN can make the ordering results less dependent on the availability of prior genes and at the same time provide the best accuracy compared to the other methods.

When comparing the expression patterns of MEF2C and MYH2 along the pseudo-time axis, Monocle and Wanderlust failed to reveal the temporal order of MEF2C and MYH2, and the increasing pattern of these genes also became less clear (Figure 2-10). In Waterfall, MEF2C first decreased and then increased, and the temporal order of MEF2C and MYH2 was not very clear. By contrast, the other methods successfully revealed the increasing pattern of MEF2C and MYH2 in this analysis. Their results also more clearly show that MEF2C increased before the increase of MYH2 (Figure 2-10).

Overall, our analyses again show that TSCAN produced the most accurate cell ordering results, and it was more robust than methods without cell clustering.

2.3.3 LPS analysis

For the LPS data, we reconstructed pseudo-time without using strong prior knowledge such as the 518 *a priori* chosen genes in the HSMM analysis. The analyses were run based on $\mathbf{E}_i$ which was computed using all genes following the protocol described in Methods. All methods only found one main path without branching paths. To determine the direction of the path, we used BCL3 as a marker gene. BCL3 is known to be involved in the response to viral and bacterial stimulus, and its expression level is expected to increase after LPS stimulation. Figure 2-2 shows the expression of this marker gene in the TSCAN GUI. Accordingly, cluster 1 was determined as the origin of the pseudo-time axis. Comparing different methods based on POS score again shows that TSCAN had the best accuracy Figure 2-11A, BCL3 was used as the marker gene for the marker-gene-only approach). Methods based on cell clustering (TSCAN, k-means TSCAN, Waterfall) were more robust than those not using cell clustering (Monocle and nocluTSCAN) (Figure 2-11B). To evaluate different methods based on differentially expressed genes, we compiled 125 known marker genes (BCL3 excluded) from [64]. Figure 2-11C and Figure 2-11D show the mean rank of these gold standard genes and the number of gold standard genes found in the top ranked genes reported by each method respectively. Again, TSCAN outperformed all other methods.

As a specific example, Figure 2-12 shows the expression level of a gold standard gene STAT2 for the LPS data [64]. STAT2 expression is expected to increase after LPS stimulation. One can see that the TSCAN result was most consistent with the known increasing pattern of STAT2. By contrast, the increasing pattern of STAT2 was much less clear in cell orderings produced by all the other approaches. In Monocle, nocluTSCAN, k-means TSCAN, Waterfall, SCUBA and Wanderlust, STAT2 first increased and then decreased. In the marker gene approach, the increasing pattern was weak compared to the high variability of cells around the fitted curve.

2.3.4 qNSC analysis

Lastly, we compared different methods using the qNSC dataset. This dataset does not have multiple time points or experimental conditions. A prior gene set for cell ordering was also not available. We therefore run the analyses based on $\mathbf{E}_i$ computed using all genes as described in Methods. All methods produced one single path without branches. To determine the path direction, we used FOXG1 as a marker gene. FOXG1 is known to be critically involved in proliferative adult NPCs. Low expression of FOXG1 was used to indicate the origin of the path.

In the qNSC analysis, the POS score cannot be calculated because external information such as data collection time is not available. Therefore, we only evaluated each method's robustness and its ability to detect known differential genes. For the differential gene analysis, 1999 known marker genes (excluding FOXG1) were compiled from [65] to serve as the gold standard. Once again, methods using cell clustering (TSCAN, k-means TSCAN, Waterfall) improved robustness of cell ordering compared to those without using cell clustering (Monocle, nocluTSCAN) (Figure 2-13A). TSCAN offered the best mean rank of gold standard genes among all methods (Figure 2-13B), and it also had the highest power for detecting the gold standard differential genes (Figure 2-13C). Figure 2-14 shows the expression level of a gold standard gene SOX9. As a down-regulated transcription factor, SOX9 expression is expected to decrease along the pseudo-time [65]. TSCAN and Waterfall results were consistent with this known decreasing pattern of SOX9, and the decreasing pattern was most evident in TSCAN. By contrast, SOX9 expression first increased and then decreased in Monocle, nocluTSCAN and SCUBA. For k-means TSCAN, SOX9 expression first decreased and then increased. For the marker-gene-only approach and Wanderlust, SOX9 expression slightly increased. Overall, TSCAN performed the best among all methods.

2.3.5 The graphical user interface

TSCAN has a GUI. As discussed above, the GUI in TSCAN allows users to visualize marker genes and tune main paths and cluster-level orderings. Besides these functions, the GUI also provides multiple trimming criteria for users to efficiently trim unwanted cells. For example, to exclude cells with high expression in two genes PDGFRA and SPHK1 in HSMM dataset, one can set up two trimming criteria such as PDGFRA > 1 and SPHK1 > 1 (Figure 2-15A) and TSCAN will exclude cells meeting both criteria (Figure 2-15B). Finally, the GUI can be used to visualize expression of user-specified genes along pseudo-time as heat maps. For example, Figure 2-15C visualizes the expression of two genes CCNA2 and CCNB2 after obtaining the pseudo-time ordering in HSMM data. Together, these functions make the pseudo-time analyses of single-cell

RNA-seq data more convenient and user-friendly.

2.4 Discussion

In summary, TSCAN offers a new tool to support pseudo-time analysis of single-cell RNA-seq data. As demonstrated by our results, this approach robustly provides competitive performance based on different criteria. By comparing methods using and not using cell clustering, we have shown that cell clustering is a useful technique for reducing the variability and improving the accuracy of the MST-based pseudo-time analysis. Although the cell clustering idea has also been used previously in Waterfall, a systematic evaluation of the impact of cell clustering on cell ordering was not provided in the Waterfall study [65]. Besides the development and systematic evaluation of the TSCAN algorithm, we also developed a GUI for TSCAN. The GUI of TSCAN provides users with the flexibility to interactively explore and adjust the analysis results.

In order to evaluate TSCAN and other unsupervised pseudo-time reconstruction methods, we used two time course datasets with multiple time points, HSMM and LPS, and intentionally avoided using any information on data collection time in our pseudo-time analyses. In this way, the data collection time can provide an independent source of information for evaluating the accuracy of cell ordering via POS score. Such an evaluation cannot be done if the test dataset has only one time point. This explains why we used HSMM and LPS for evaluation even though in principle such data could be analyzed in other ways. For instance, one could perform supervised rather than unsupervised analysis to order cells. Alternatively, one could perform an initial analysis to identify differentially expressed genes between different data collection time points and then use them as prior genes (similar to the 518 prior genes for HSMM) to order cells. Unlike the HSMM and LPS data, the qNSC dataset represents a different situation faced by many investigators. Here, single-cell RNA-seq data are collected from only one biological condition rather than from multiple time points or conditions. In such a scenario, supervised methods that use data collection time information to order cells cannot be applied, and one cannot compare different time points or conditions to find differential genes and use them as prior genes for cell ordering. It is therefore important to be able to perform unsupervised pseudo-time analysis such as TSCAN.

Besides TSCAN, this chapter also introduced several methods to quantitatively evaluate cell ordering performance. We expect that these evaluation methods will continue to be useful in the future for evaluating other pseudo-time reconstruction algorithms. Although TSCAN was tested using RNA-seq, in principle it should not be difficult to tailor this approach to other data types should single-cell data for those data types become available in the future.

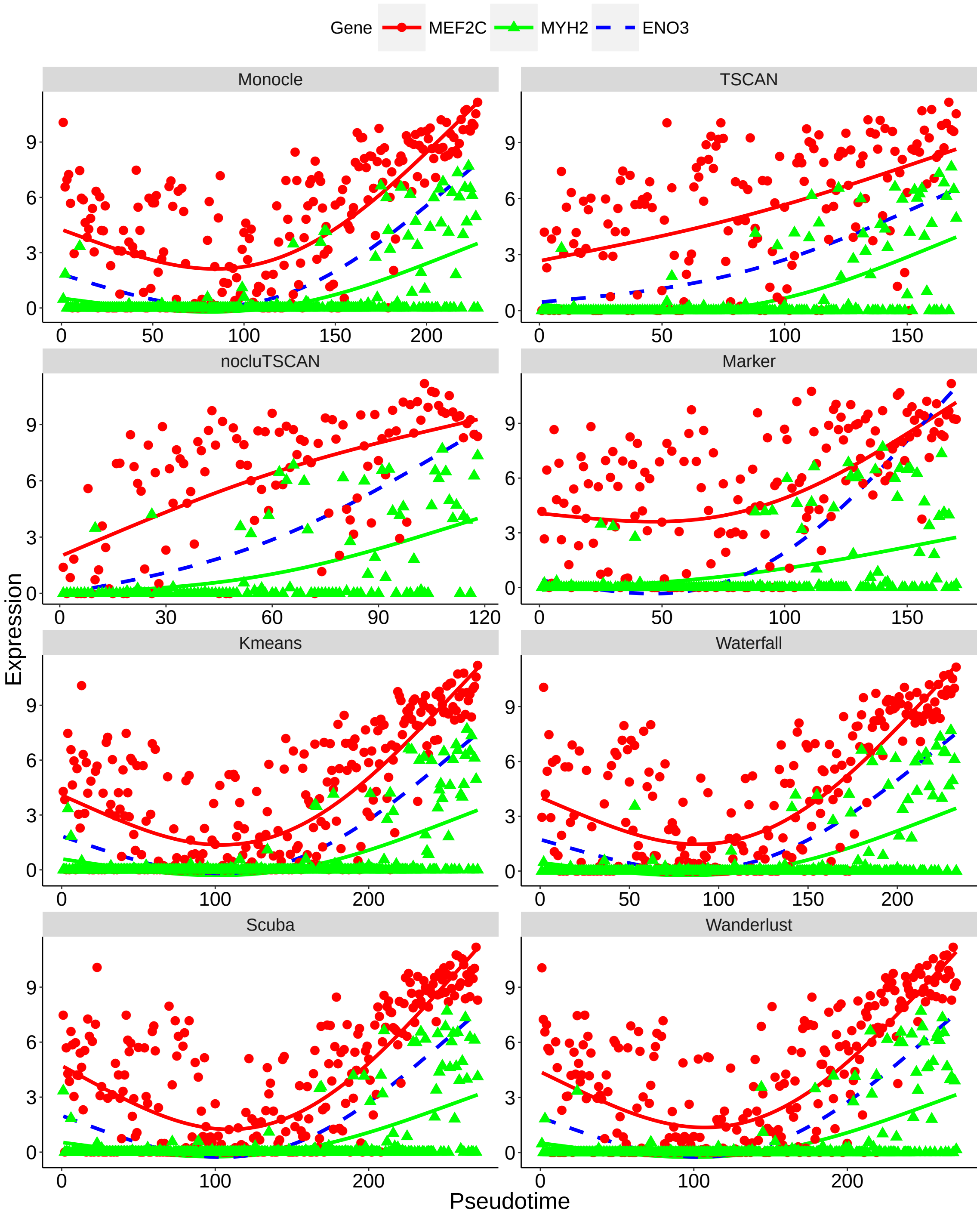
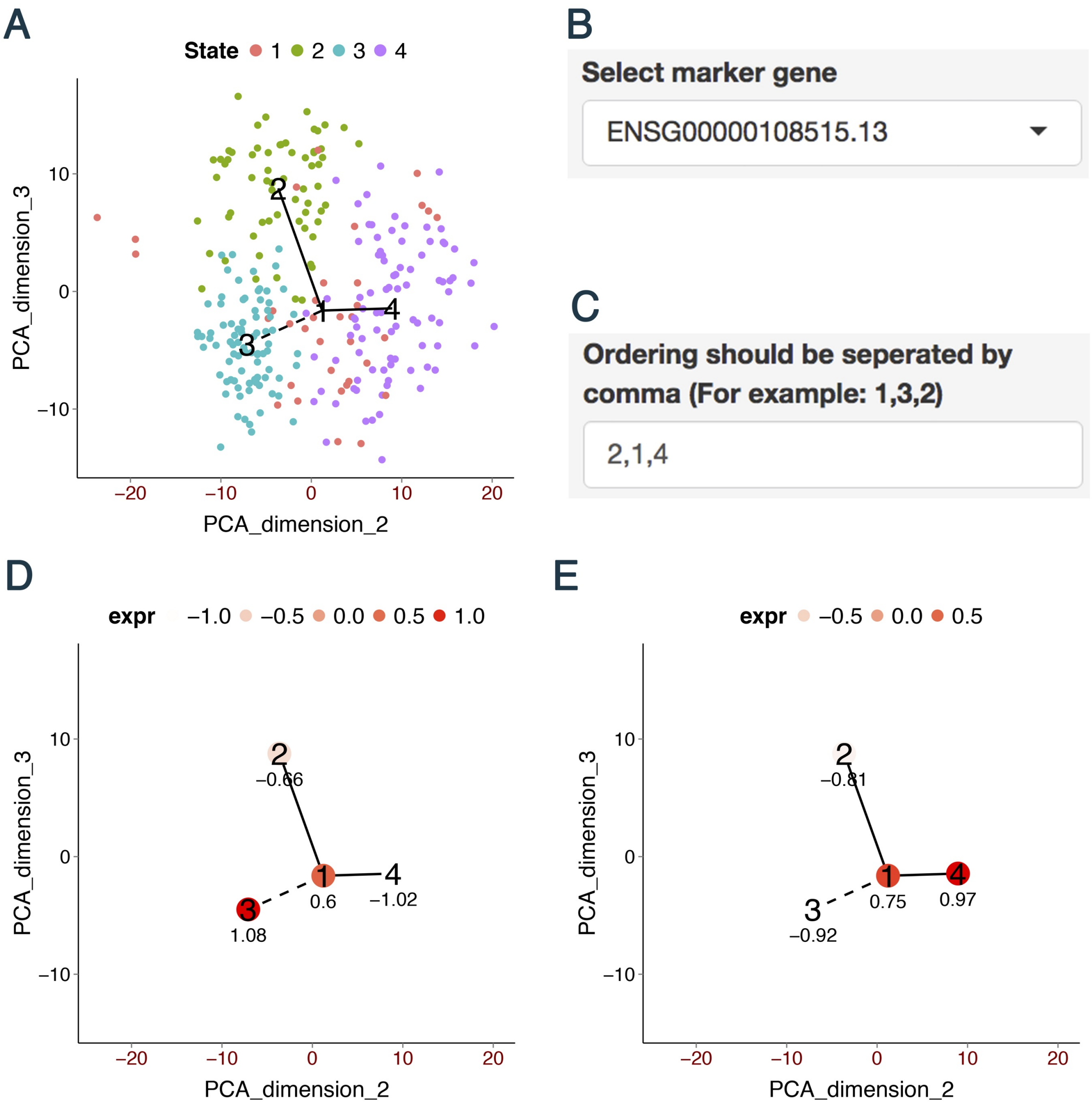


Figure 2-6. MEF2C and MYH2 expression patterns in HSMM dataset where pseudo-time was constructed based on 518 *a priori* chosen genes. MEF2C and MYH2 expression in each cell is plotted as a function of cell order on the analyzed pseudo-time axis. The curves are the fitted GAM function. The dashed curve is the GAM fit for ENO3, the marker used to determine the path direction.



Select marker gene

ENSG00000108515.13

Ordering should be seperated by comma (For example: 1,3,2)

2,1,4

Figure 2-7. Demonstration of GUI and TSCAN analysis of HSMM data using all genes for pseudo-time reconstruction. (A) MST constructed by TSCAN using all genes. (B) Users can choose a marker gene in GUI to visualize its expression. (C) Users can define a path by specifying the clusters to include and their ordering. (D) The average expression of SPHK1 in each cluster. (E) The average expression of ENO3 in each cluster.

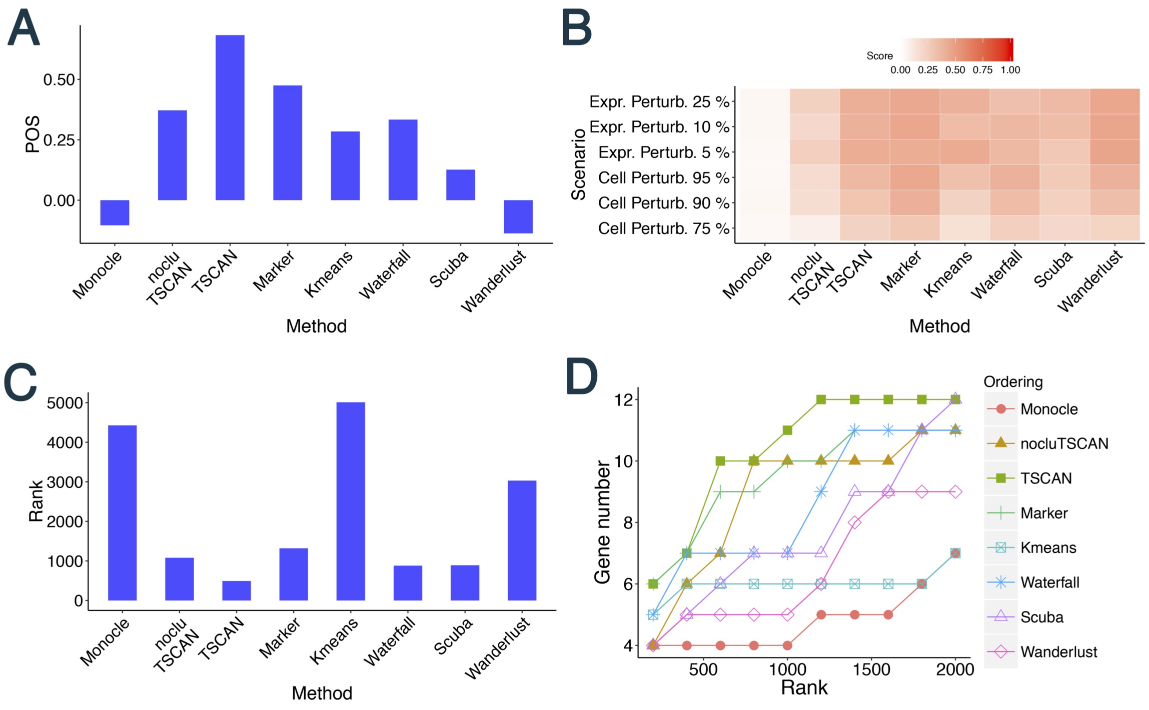


Figure 2-8. Evaluation results for different methods in HSMM data where pseudo-time was constructed using all genes. (A) POS score. (B) Robustness measured by the average similarity score from 100 independent perturbations. (C) Mean rank of gold standard genes. (D) Number of detected gold standard genes among top differential genes.

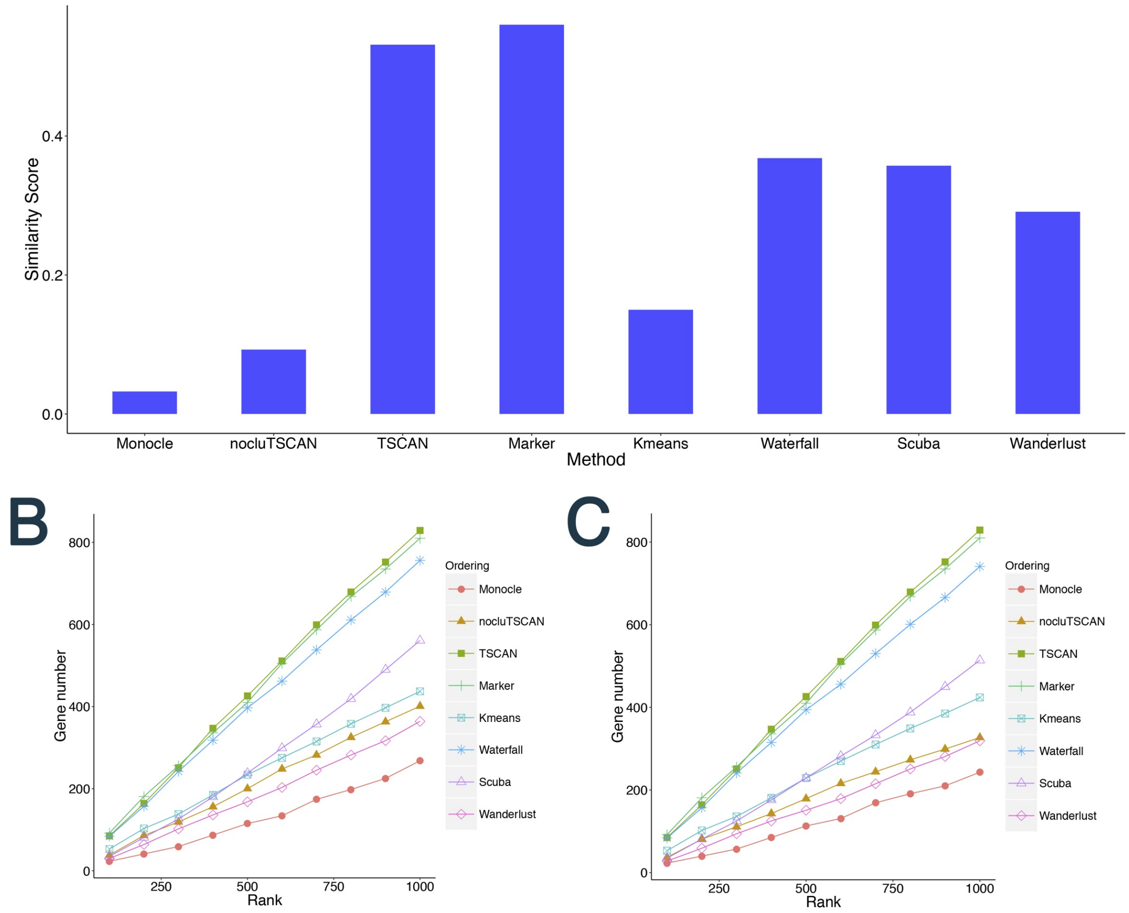


Figure 2-9. Comparing the cell ordering constructed using 518 prior genes and the cell ordering obtained without using these genes in the HSMM dataset. (A) Similarity score between the two orderings for each method. (B) The number of common genes among the top R differentially expressed genes detected by the two cell orderings is plotted as a function of R. (C) The number of common genes with consistent change directions among the top R differentially expressed genes detected by the two cell orderings is plotted as a function of R. In order to determine if a gene has consistent change direction in the two cell orderings, the fitted GAM functions of the gene from the two cell orderings are compared as follows. First, the pseudo-time axes for both cell orderings are linearly scaled to interval [0,1], and the GAM functions are scaled accordingly. Next, values of the GAM functions are extracted at 100 evenly spaced pseudo-time points (i.e., 0.01, 0.02, ..., 1), and then the Pearson's correlation between the two extracted vectors (representing the two GAM functions) is computed. Genes with negative correlation are viewed as inconsistent between the two cell orderings.

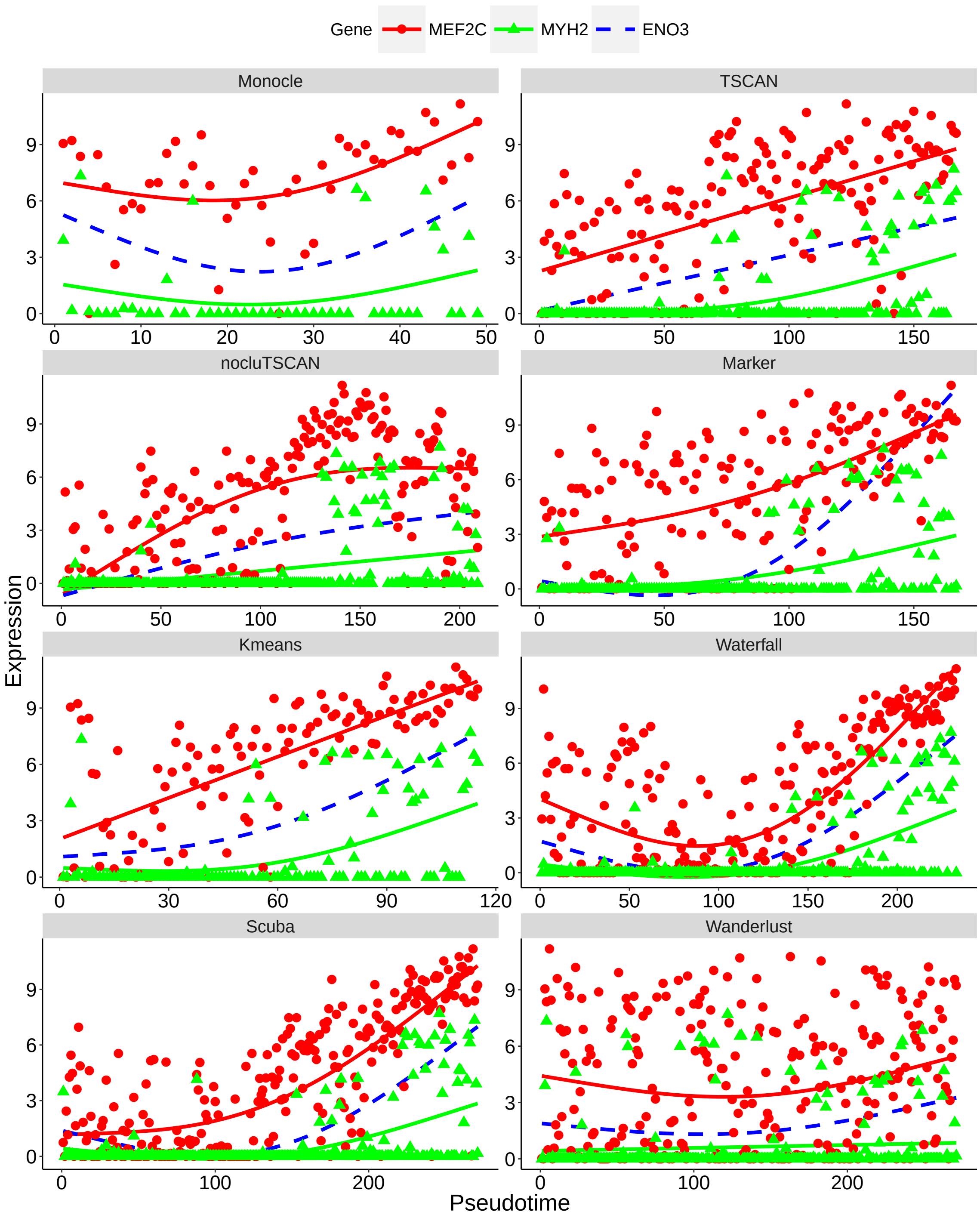


Figure 2-10. MEF2C and MYH2 expression patterns in HSMM dataset where pseudotime was constructed using all genes. The expression of each gene in each cell is plotted as a function of cell order on the pseudo-time axis. The solid curves are the fitted GAM function. The dashed curve is the GAM fit for ENO3, the marker gene used to determine the path direction.

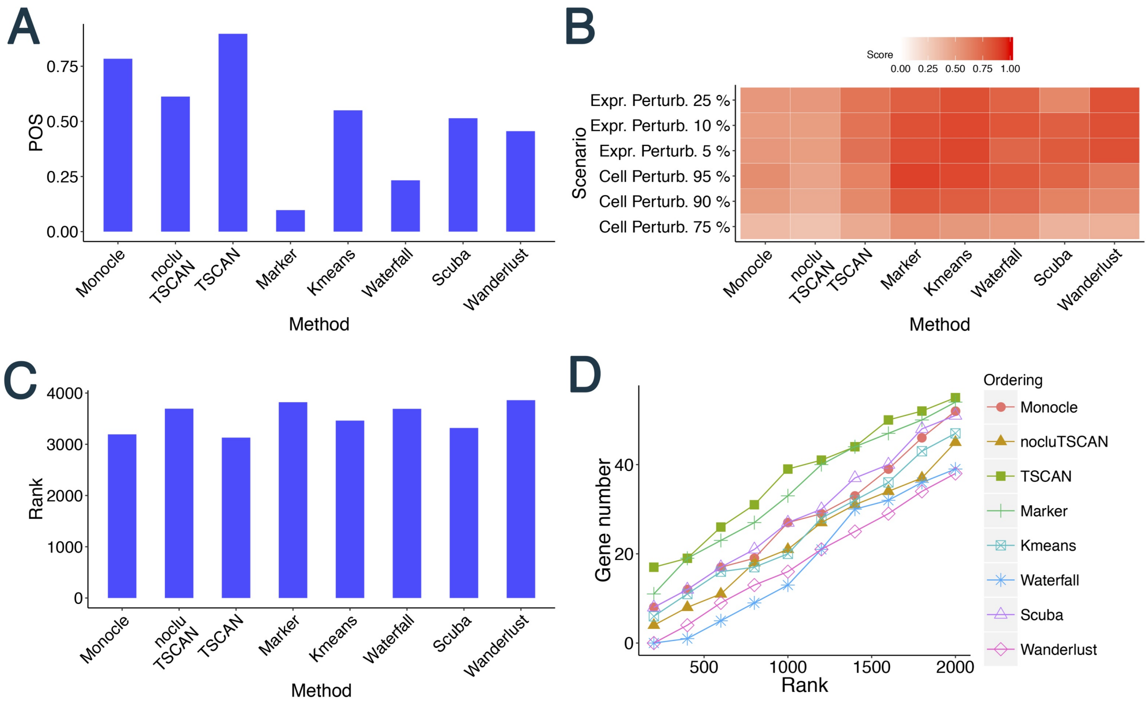


Figure 2-11. Evaluation results for different methods in LPS dataset. (A) POS score. (B) Robustness measured by the average similarity score from 100 independent perturbations. (C) Mean rank of gold standard genes. (D) Number of detected gold standard genes among top differential genes.

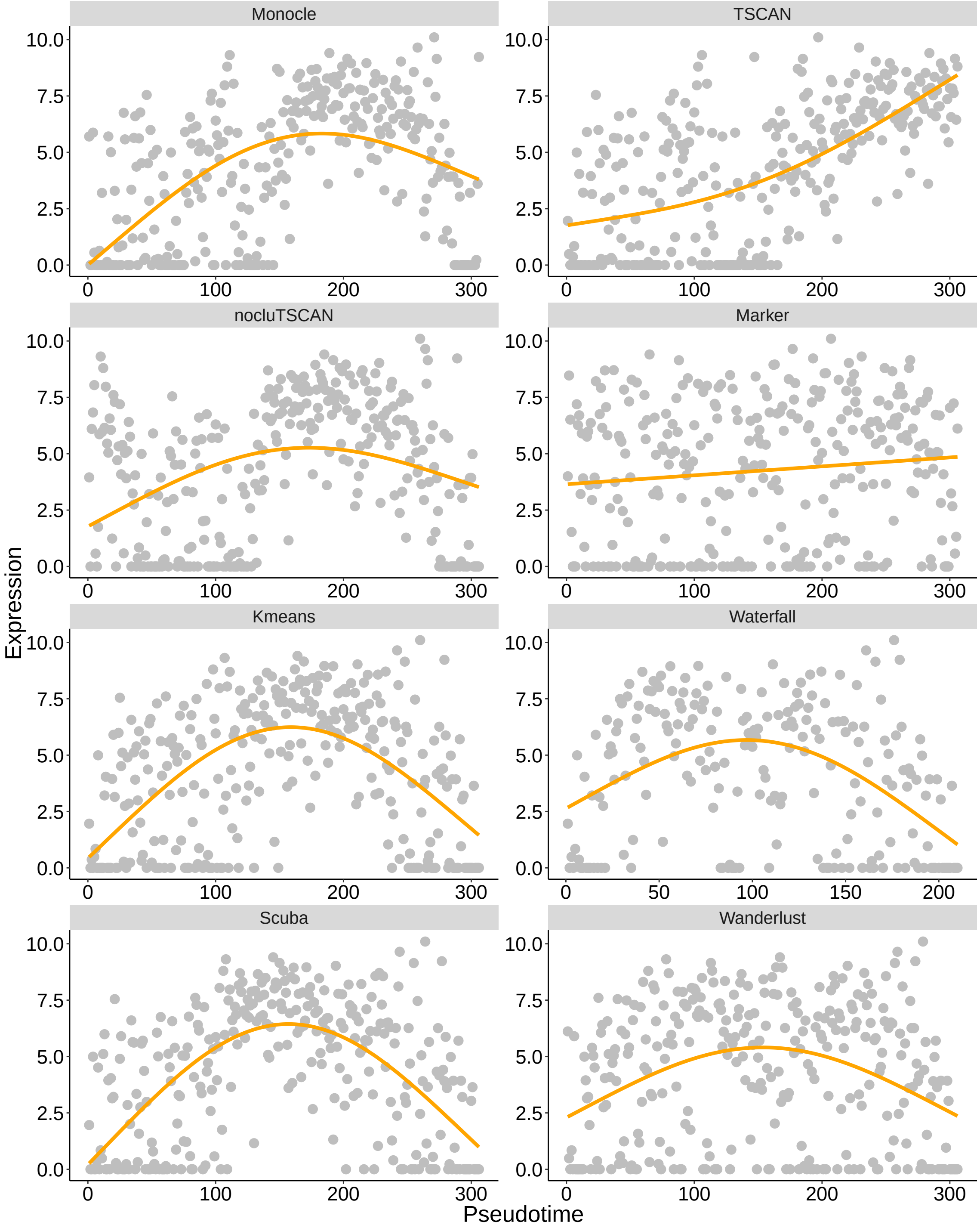


Figure 2-12. STAT2 expression patterns in LPS dataset. STAT2 expression in each cell is plotted as a function of cell order on the pseudo-time axis. The orange curve is the fitted GAM function.

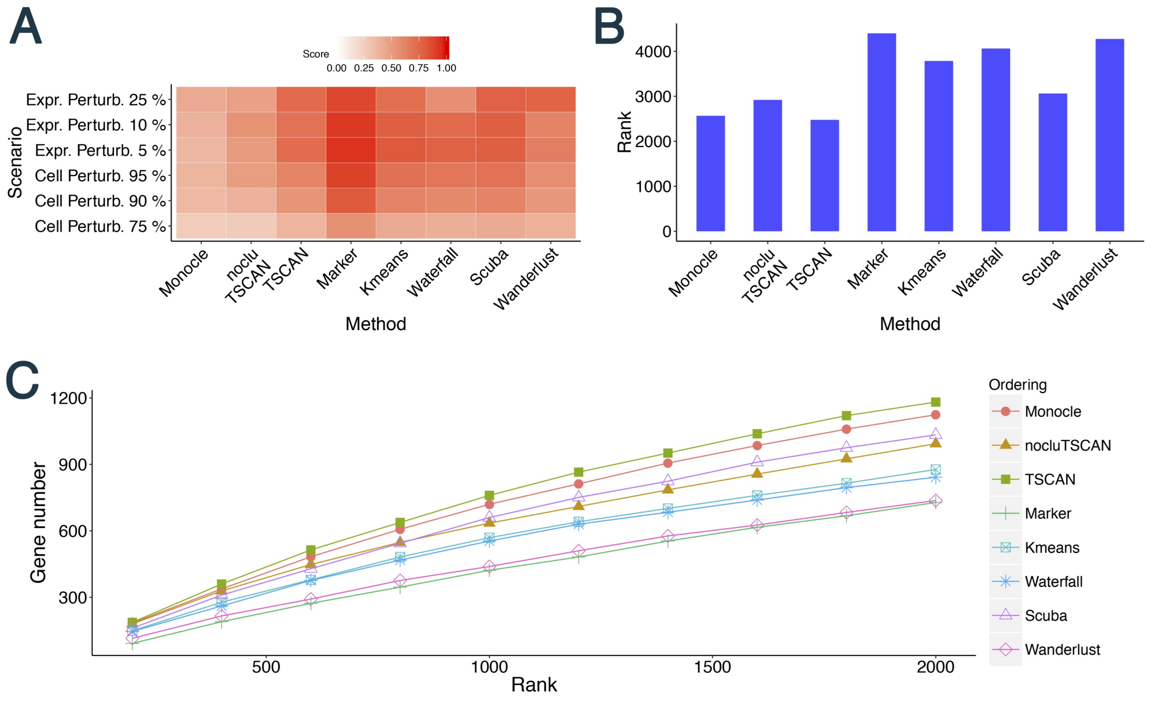


Figure 2-13. Evaluation results for different methods in qNSC dataset. (A) Robustness measured by the average similarity score from 100 independent perturbations. (B) Mean rank of gold standard genes. (C) Number of detected gold standard genes among top differential genes.

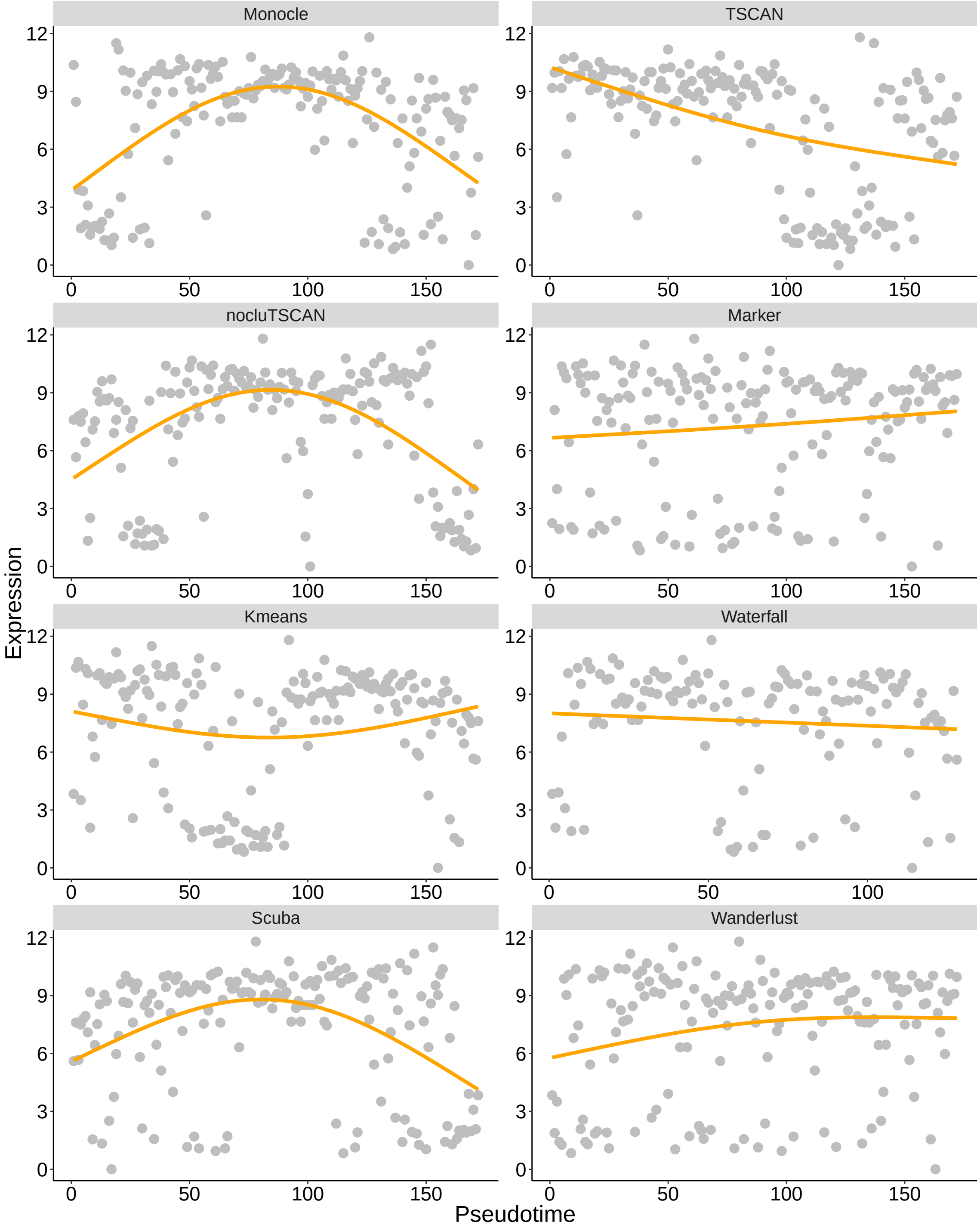
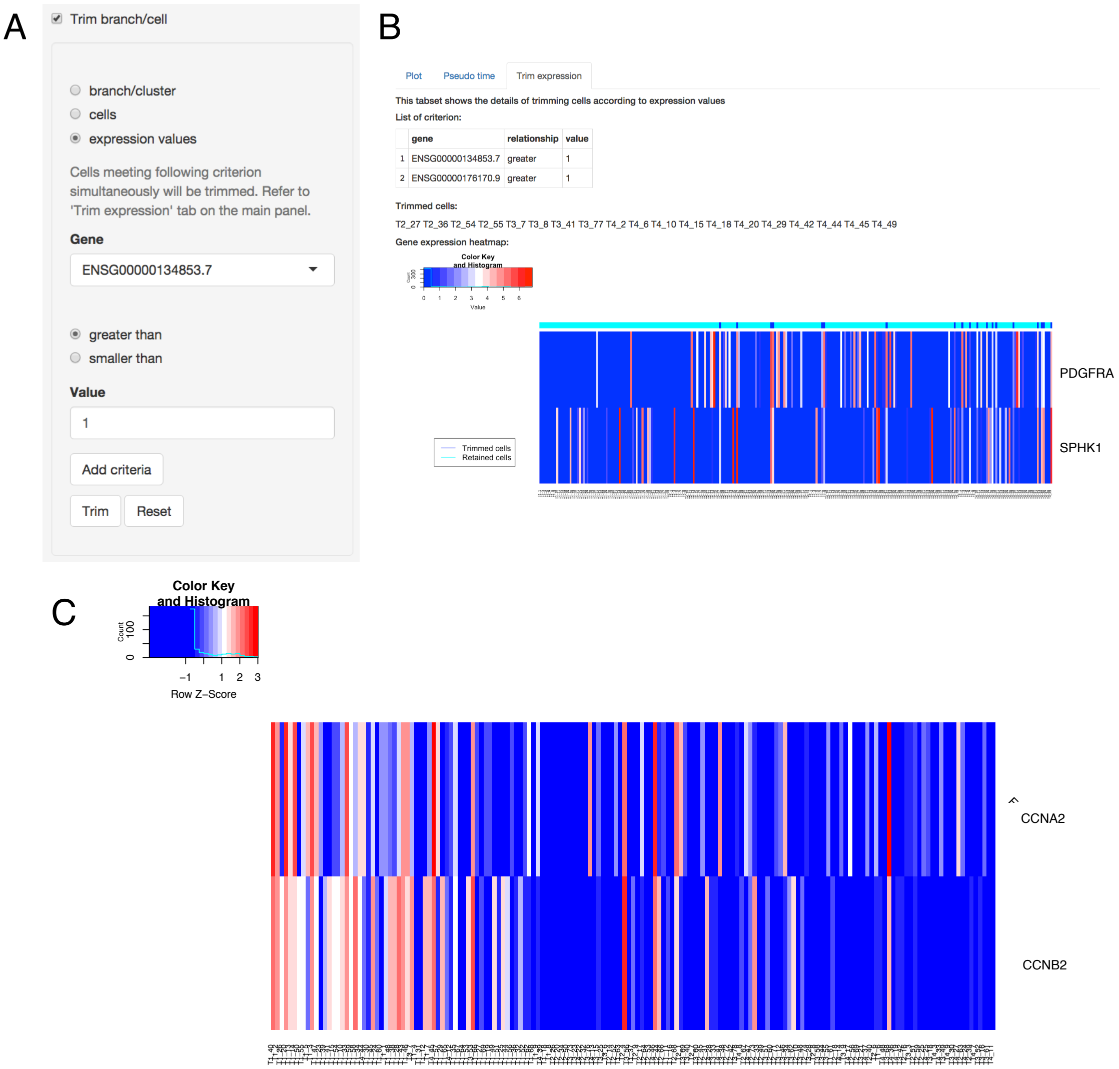


Figure 2-14. SOX9 expression patterns in qNSC dataset. SOX9 expression in each cell is plotted as a function of cell order on the pseudo-time axis. The orange curve is the fitted GAM function.



☑ Trim branch/cell

○ branch/cluster
○ cells
● expression values

Cells meeting following criterion simultaneously will be trimmed. Refer to 'Trim expression' tab on the main panel.

Gene

ENSG00000134853.7

● greater than
○ smaller than

Value

1

This tabset shows the details of trimming cells according to expression values

List of criterion:

	gene	relationship	value
1	ENSG00000134853.7	greater	1
2	ENSG00000176170.9	greater	1

Trimmed cells:

T2_27 T2_36 T2_54 T2_55 T3_7 T3_8 T3_41 T3_77 T4_2 T4_6 T4_10 T4_15 T4_18 T4_20 T4_29 T4_42 T4_44 T4_45 T4_49

Gene expression heatmap:

Figure 2-15. Further demonstration of TSCAN GUI. (A) Users can set up trimming criteria by choosing gene names and specifying expression cutoffs. (B) TSCAN excludes cells that meet all trimming criteria. (C) Users can also visualize the expression of specified genes along pseudo-time as heatmaps.

Chapter 3

Single-cell ATAC-seq Signal Extraction and Enhancement with SCATE

3.1 Introduction

A cell's regulome, defined as the activities of all cis-regulatory elements (CREs) in its genome, contains crucial information for understanding how genes' transcriptional activities are regulated in normal and pathological conditions. Conventionally, regulome is measured using bulk technologies such as chromatin immunoprecipitation coupled with sequencing (ChIP-seq [11]), DNase I hypersensitive site sequencing (DNase-seq [7]) and assay for transposase-accessible chromatin followed by sequencing (ATAC-seq [8]). These technologies measure cells' average behavior in a biological sample consisting of thousands to millions of cells. They cannot analyze each individual cell. When a heterogeneous sample (e.g., a tissue sample) consisting of multiple cell types or cell states is analyzed, these bulk technologies may miss important biological signals carried by only a subset of cells.

Recent innovations in single-cell genomic technologies make it possible to map regulomes in individual cells. For example, single-cell ATAC-seq (scATAC-seq [32, 33]) and single-cell DNase-seq (scDNase-seq [14]) are two technologies for analyzing open chromatin, a hallmark for active cis-regulatory elements, in single cells. Single-cell ChIP-seq (scChIP-seq [34]), on the other hand, allows single-cell analysis of histone modification. Technologies for simultaneously mapping open chromatin along with other -omics modalities are also under active development (e.g., scNMT-seq [67], Pi-ATAC [68], sci-CAR [35]). These single-cell technologies enable scientists to examine a heterogeneous sample with an unprecedented cellular resolution, allowing them to systematically discover and characterize unknown cell subpopulations.

Among the existing single-cell regulome mapping technologies, scATAC-seq is the most widely used one due to its relatively simple and robust protocol and its unparalleled throughput for analyzing a large number of cells. It is adopted by the Human Cell Atlas (HCA) Consortium as a major tool for characterizing regulatory landscape of human cells ([69]).

Data produced by scATAC-seq are highly sparse. For instance, a typical human scATAC-seq dataset contains 10^2–10^4 cells and 10^3–10^5 sequence reads per cell. However, the number of CREs in the genome far exceeds 10^5. Thus, in a typical cell, most CREs do not have any mapped read. For CREs with reads, the number of mapped reads seldom exceeds two (Figure 3-1A,B) because each locus has no more than two copies of assayable chromatin per cell in a diploid genome. Also, existing single-cell regulome mapping technologies including scATAC-seq destroy cells during the assay. Thus, they only get a snapshot of a cell at one time point. However, molecular events such as transcription factor (TF)-DNA binding and their dissociation are temporal stochastic processes. The steady-state activity of a CRE in a cell is determined by the probability that such stochastic events occur over time. Since probability is a continuous measure, the overall activity of a CRE in a cell should be a continuous signal in principle. The sparse and nearly binary scATAC-seq data collected for each CRE at one single time point therefore cannot accurately describe the CRE's continuous steady-state activity in a cell.

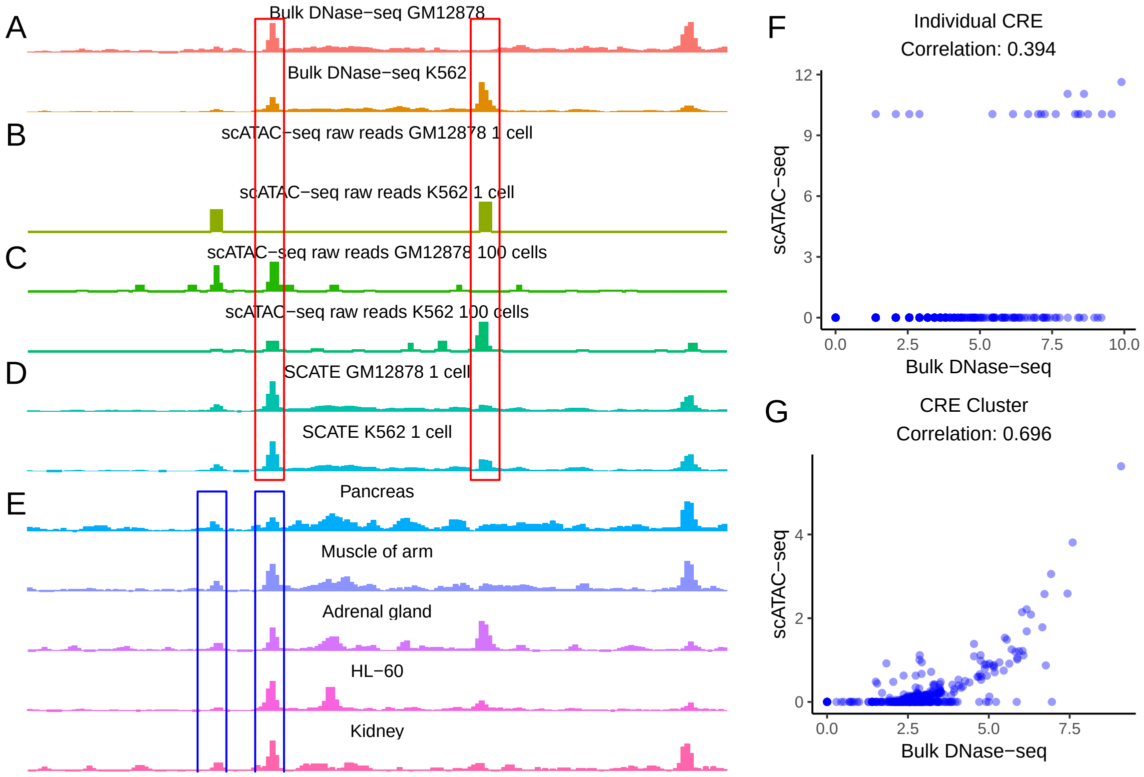


Figure 3-1. Background and motivation. (A)-(D): an example genomic region showing chromatin accessibility in GM12878 and K562 measured by different methods including (A) bulk DNase-seq, (B) scATAC-seq from one single cell, (C) scATAC-seq by pooling 100 cells, (D) SCATE-reconstructed scATAC-seq signal from one single cell. (E): Illustration of CRE-specific baseline activities using the same genomic region. Bulk DNase-seq data from multiple different cell types show that some loci tend to have higher activity than others regardless of cell type (e.g. compare the two loci in blue boxes). (F): At the individual CRE level, the correlation between the log-normalized scATAC-seq read count in one GM12878 cell and the log-normalized bulk GM12878 DNase-seq signal is low (Pearson correlation = 0.394). Each dot is a CRE. (G): After aggregating multiple CREs based on co-activated CRE pathways by SCRAT, the correlation between the CRE pathway activities in one GM12878 cell and the bulk GM12878 DNase-seq signal (both at log-scale) is substantially higher (Pearson correlation = 0.696). Each dot is a CRE pathway.

The discrete, sparse and noisy data pose significant data analysis challenges. Conventional methods developed for bulk data cannot effectively analyze single-cell regulome data [70, 71]. As a result, there is a pressing need for new computational tools for single-cell regulome analysis. Recently, several single-cell regulome analysis methods have been developed. They can be grouped into three categories based on how they deal with the sparsity. (Table 3-I)

Method	Combine CREs	Combine cells	Adaptively tune resolution	Use public bulk data to model baseline	Use Binary or Count data	Primary goal	Reference
SCATE	✓	✓	✓	✓	C	Reconstruct activities of each individual CRE	This paper
chromVAR	✓				C	Cluster cells, identify TF motifs associated with differential accessibility and variability	[70]
SCRAT	✓				C	Cluster cells, identify CRE pathways associated with differential accessibility	[71]
BROCKMAN	✓				B	Summarize data by k-mers and perform principal component analysis on k-mer features to identify co-varying TFs, cluster cells	[72]
Dr.seq2		✓			C	Cluster cells, identify peaks (MACS) in each cell subpopulation	[73]
Cicero		✓			B	Identify correlated pairs of CREs	[74]
Scasat					B	Cluster cells, identify peaks (MACS), differential accessibility analysis	[75]
Destin					B	Cluster cells	[76]
scABC					C	Cluster cells	[77]
PRISM					B	Quantify cell-to-cell variation to identify hyperor hypo-variable genomic features	[78]
cisTopic					B	Represent data using low-dimensional topiccell and region-topic representation, cluster cells and CREs accordingly	[79]

Table 3-I. Comparison of single-cell regulome analysis methods

Methods in category 1, including chromVAR [70], SCRAT [71] and BROCKMAN [72], tackle sparsity by aggregating reads from multiple CREs. Instead of analyzing each CRE, they combine reads from CREs that share either a TF binding motif, a k-mer, or a co-activation pattern in DNase-seq data from the Encyclopedia of DNA Elements (ENCODE) [80, 81]. The aggregated data on motifs, k-mers, or co-activated CRE pathways are then used as features to cluster cells or characterize cell heterogeneity. To demonstrate the effect of combining CREs, Figure 3-1F shows chromatin accessibility in cell line GM12878 computed using non-aggregated data at each individual CRE, and Figure 3-1G shows accessibility computed using SCRAT aggregated data (i.e., average normalized read count across CREs) for each co-activated

CRE pathway. After aggregation, the signal in scATAC-seq became more continuous and showed higher correlation with the bulk DNase-seq-measured accessibility. One major drawback of aggregating multiple CREs is the loss of CRE-specific information. Thus, existing methods in this category do not analyze the activity of each individual CRE.

Methods in category 2, including Dr.seq2 [73] and Cicero [74], tackle sparsity by pooling multiple cells. Dr.seq2 [73] pools cells and applies MACS [82] to the pooled pseudobulk sample to call peaks. Cicero [74] first pools the binary chromatin accessibility profiles from similar cells to create pseudobulk samples. It then uses the pseudobulk samples to study the pairwise correlation among different CREs. Typically, scATAC-seq data pooled from multiple cells are more continuous than data from a single cell, and the pooled data also correlate better with bulk data (Figure 3-1 A-C). Despite this, pooling cells does not fully eliminate sparsity, particularly in a rare cell type with only a few cells. Also, pooling cells may result in loss of cell-specific information. Thus, one may want to only pool cells that are highly similar in order to better characterize a heterogeneous cell population. This could result in grouping cells into many small cell clusters, each with only a few highly similar cells. In that situation, pooling cells alone may not be enough for removing sparsity and accurately estimating activities of individual CREs.

Methods in category 3 directly work with the peak-by-cell read count matrix or its binarized version. For example, Scasat [75] converts the peak-by-cell read count matrix into a binary accessibility matrix and uses this binary matrix to cluster cells. Destin [76] applies weighted principal components and K-means clustering to the binary accessibility matrix to cluster cells. scABC [77] uses the read count matrix to cluster cells via a weighted K-medoids clustering algorithm. PRISM [78] uses the binary accessibility matrix to compute cosine distance between cells and then uses this distance to evaluate the degree of heterogeneity of a cell population. CisTopic

[79] models the binary accessibility matrix using Latent Dirichlet Allocation (LDA). This approach views each cell as a mixture of multiple topics, and each topic is a collection of peak regions and their usage preferences. The topic-cell and region-topic vectors provide a low-dimensional representation of the data. Cells and peaks are then clustered in this low-dimensional space. Category 3 methods typically are designed for specific tasks such as clustering and assessment of sample variability rather than estimating activities of individual CREs.

In summary, while existing methods provide tools for clustering cells, identifying co-accessible CREs, and analyzing sample heterogeneity, they do not address the fundamental issue of accurately reconstructing activities of each individual CRE using sparse data. Knowing activities of each individual CRE is crucial for functional studies. For example, such knowledge can be used to inform the selection of CREs for knock-out or transgenic experiments. In order to facilitate accurate reconstruction of CRE activities using scATAC-seq data, this article introduces a new statistical and analytical framework SCATE (**S**ingle-**C**ell **AT**AC-seq Signal **E**xtraction and Enhancement). SCATE employs a model-based approach to integrate three types of information: (1) co-activated CREs, (2) similar cells, and (3) publicly available bulk regulome data. Unlike the existing methods that either aggregate CREs (category 1) or cells (category 2) but not both, SCATE combines both types of information. SCATE also uniquely uses public regulome data to enhance the analysis and adaptively optimizes the analysis resolution based on the available information in the scATAC-seq data. SCATE is freely available as an open source R package via GitHub. Compared to the existing methods, SCATE can more accurately predict CRE activities and transcription factor binding sites using the sparse data from a single cell (Figure 3-1 B,D) or a rare cell type as we shall demonstrate.

3.2 Methods

3.2.1 Single-cell ATAC-seq data preprocessing

Single-cell ATAC-seq data for GM12878 and K562 cells were obtained from GEO (GSE65360) [32]; Single-cell ATAC-seq data for human hematopoietic cell types were obtained from GEO (GSE96769) [83]; Single-cell ATAC-seq data for mouse brain and thymus were obtained from GEO (GSE111586) [84]. For each cell, paired-end reads were trimmed using the program provided by [32] to remove adaptor sequences. Reads were then aligned to human (hg19) or mouse (mm10) genome using bowtie2 with parameter -X2000. This parameter retains paired reads with insertion up to 2000 base pairs (bps). PCR duplicates were removed using Picard (http://broadinstitute.github.io/picard/).

3.2.2 Genome segmentation

Genome is segmented into 200 base pair (bp) nonoverlapping bins. Bins that overlap with ENCODE blacklist regions are excluded from subsequent analyses since their signals tend to be artifacts [85].

3.2.3 Bulk DNase-seq database (BDDB)

SCATE borrows information from large amounts of publicly available bulk DNase-seq data to improve scATAC-seq analysis. We compiled a database consisting of 404 human and 85 mouse DNase-seq samples obtained from the ENCODE. Take human as an example, we downloaded all ENCODE DNase-seq samples generated by the University of Washington [80] in bam format. Files marked by ENCODE as low quality (marked as “extremely low spot score” or “extremely low read depth” by ENCODE) were filtered out. Technical replicates for each distinct cell type or tissue were merged into one sample. This has resulted in 404 DNase-seq samples representing diverse cell types. Mouse samples were processed similarly.

3.2.4 Compiling cis-regulatory elements (CREs) using bulk data compendium

Given a species and a compendium of bulk regulome samples (e.g., DNase-seq samples in BDDB), SCATE systematically identifies CREs in the genome as follows. Let $y_{i,j}$ denote the raw read count of bin i in sample j. Let L_j be sample j's total read count divided by 10^8 (i.e., the library size in the unit of hundred million. For example, a sample with 200 million reads has $L_j = 2$). We normalize the raw read counts by library size and log2-transform them after adding a pseudocount 1. This results in normalized data $\tilde{y}_{i,j} = \log_2(y_{i,j}/L_j + 1)$. Bin i is called a "signal bin" in sample j if (1) $y_{i,j} \geq 10$, (2) $\tilde{y}_{i,j} \geq 5$, and (3) $\tilde{y}_{i,j}$ is at least five times (three times for mouse) larger than the background signal defined as the mean of $\tilde{y}_{i,j}$s in the surrounding 100 kb region. If a bin is a signal bin in at least one bulk sample, it is labeled as a "known CRE". In this way, all genomic bins are labeled as either "known CREs" or "other bins". 522,173 known CREs for human and 475,865 known CREs for mouse are identified using our bulk DNase-seq compendium. Locations of these CREs are stored in SCATE and provided as part of the software package. Saturation analysis shows that typically a new bulk sample from a new cell type only contributes a small fraction (0.013 % for human and 0.18 % for mouse) of new CREs to the known CRE list (Figure 3-3A). In the three benchmark scATAC-seq datasets used in this article, datasets 1, 2 and 3 would only add 0.050%, 0.0013%, and 0.063% new CREs, respectively, to our known CRE list. For the human hematopoietic differentiation dataset used in the last Results section, the scATAC-seq dataset would only add 0.118 % of new CREs to the known CRE list (Figure 3-3B; the calculation was based on detecting CREs in each cell type separately and then adding the union of all CREs from all cell types in the scATAC-seq data to the known CRE list). This suggests that the majority of a new sample's regulome can be studied by analyzing the precompiled known CREs, which can save user's work on compiling and clustering their own CREs. In this article, SCATE is demonstrated using our precompiled known CRE list, as the performance curves and statistics do not change much by adding new CREs from each scATAC-seq dataset to the analysis.

3.2.5 SCATE model for known CREs in a single cell

Consider scATAC-seq data from one single cell j. Given aligned sequence reads, SCATE will estimate activities of known CREs first. Let $y_{i,j}$ denote the observed read count for CRE i ($i = 1, \ldots, I$) in cell j, and let $\mu_{i,j}$ denote the unobserved true activity. Our goal is to infer the unobserved $\mu_{i,j}$ from the observed data $y_{i,j}$. We assume the following data generative model with three components.

1. *Model for true activity.* The unobserved $\mu_{i,j}$ is modeled as $\log(\mu_{i,j}) = m_i + s_i\delta_{i,j}$. Here m_i and s_i represent CRE i's baseline mean activity and standard deviation (SD). They are used to model the locus-specific but cell-type-independent baseline behavior of each CRE (i.e., the locus effects observed in Figure 3-1E). Since these locus-specific effects cannot be reliably learned using sparse data or data from one cell type, we learn them using the bulk data from diverse cell types in our bulk regulome data compendium (see below). Once they are learned, m_i and s_i are treated as known. The unknown $\delta_{i,j}$ describes CRE i's cell-specific activity after removing locus effects (i.e., $\delta_{i,j} = \frac{\log(\mu_{i,j})-m_i}{s_i}$).

   Due to data sparsity, accurately estimating $\delta_{i,j}$ using the observed data from only one CRE in one cell is difficult. Thus, we impose additional structure on $\delta_{i,j}$s to allow co-activated CREs to share information to improve the estimation. We group CREs into K clusters based on their co-activation patterns across cell types (see below). We assume that CREs in the same cluster share the same

δ. Mathematically, let $\boldsymbol{\delta}_j = (\delta_{1,j}, \dots, \delta_{I,j})^T$ be a column vector that contains $\delta_{i,j}$s from all CREs in cell j. Let $\mathbf{X}$ be a $I \times K$ cluster membership matrix. Each entry of this matrix x_{ik} is a binary variable: $x_{ik} = 1$ if CRE i belongs to cluster k, and $x_{ik} = 0$ otherwise. Let $\beta_{k,j}$ denote the common activity of all CREs in cluster k. Arrange $\beta_{k,j}$s into a column vector $\boldsymbol{\beta}_j = (\beta_{1,j}, \dots, \beta_{K,j})^T$. Our assumption can be represented as $\boldsymbol{\delta}_j = \mathbf{X}\boldsymbol{\beta}_j$. When the cluster number K is smaller than the CRE number I, imposing this additional structure on $\delta_{i,j}$ reduces the number of unknown parameters from I to K. As a result, it increases the average amount of information available for estimating each parameter.

Note that in our model, two CREs with the same δ can still have different activities (i.e., different $\mu_{i,j}$s) because $\log(\mu_{i,j}) = m_i + s_i\delta_{i,j}$. In other words, SCATE allows co-activated CREs to share information through δ, but at the same time it also allows each CRE to keep its own locus-specific baseline characteristics. This is an important feature missing in other existing methods.

Another unique feature of SCATE is that we treat the cluster number K as a tuning parameter and adaptively choose it based on available information to optimize the spatial resolution of the analysis. Unlike SCATE, other existing methods aggregate CREs based on known pathways. For them, K is fixed and the analysis' spatial resolution cannot be tuned and optimized.

2. *Model for technical bias.* Since the locus effects m_i and s_i are learned from the bulk data, we view $\mu_{i,j}$ as the activity one would obtain if one could measure a bulk regulome sample (e.g., bulk DNase-seq) consisting of cells identical to cell j. In scATAC-seq data, $\mu_{i,j}$ is distorted to become $\mu_{i,j}^{sc}$ due to technical biases in single-cell experiments (e.g., DNA amplification bias). We model these unknown technical biases using a cell-specific monotone function $h_j(.)$. In other words, we assume $\log(\mu_{i,j}^{sc}) = h_j(\log(\mu_{i,j}))$. We estimate the unknown function $h_j(.)$ by

comparing scATAC-seq data with the bulk regulome data at CREs that show constant activity across different cell types (see below). Once $h_j(.)$ is estimated, it is assumed to be known.

3. *Model for observed read counts.* We assume that the observed read count $y_{i,j}$ is generated from a Poisson distribution with mean $L_j \mu^{sc}_{i,j}$. Here L_j is the total number of reads in cell j divided by 10^8. It is a cell-specific normalizing factor to adjust for library size.

To summarize, our model assumes:

$$\begin{aligned}
&y_{i,j} \sim Poisson(L_j \mu^{sc}_{i,j})\\
&\log(\mu^{sc}_{i,j}) = h_j(\log(\mu_{i,j}))\\
&\log(\mu_{i,j}) = m_i + s_i \delta_{i,j}\\
&\boldsymbol{\delta}_j = \mathbf{X}\boldsymbol{\beta}_j
\end{aligned} \tag{3.1}$$

For a fixed cluster number K, we fit the model as follows: (1) use the bulk regulome data compendium to learn locus effects m_i and s_i; (2) use scATAC-seq data and the bulk regulome data compendium to learn technical bias function $h_j(.)$ which normalizes scATAC-seq data with the bulk regulome compendium used to learn locus effects; (3) given m_i, s_i and $h_j(.)$, use the observed data $\mathbf{y}$ to estimate $\boldsymbol{\beta}$ which will determine $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$. The estimated $\boldsymbol{\mu}$ provides the final estimates for CRE activities.

In order to optimize the analysis' spatial resolution, SCATE treats the cluster number K as a tuning parameter. CREs are clustered at multiple granularity levels corresponding to different Ks. As K increases, the average number of CREs per cluster decreases. This increases spatial resolution because the cluster activity more resembles the activity of individual CREs. However, increasing K also decreases the amount of information for estimating the activity of each cluster, and thus the estimates become noisier. We use a cross-validation approach to choose the optimal K that balances spatial resolution and estimation uncertainty (see below).

3.2.6 Estimate locus effects m_i and s_i

We estimate locus effects using the rich bulk data from diverse cell types in the bulk regulome compendium. Let $y_{i,j}$ be the observed read count for genomic bin i and bulk sample j ($j = 1, \ldots, J$). L_j represents sample j's library size in the unit of hundred million. For each genomic bin i, locus effects are estimated using the observed counts $\{y_{i,j} : j = 1, \ldots, J\}$. We model $y_{i,j}$ in bulk data as:

$$\begin{aligned} y_{i,j} &\sim Poisson(L_j \mu_{i,j}) \\ \log(\mu_{i,j}) &= m_i + s_i \delta_{i,j} \end{aligned} \tag{3.2}$$

This is similar to the single-cell model above but without the technical bias component. Without additional constraints, m_i and s_i are not identifiable since each bin i has only J observed data points but $J+2$ unknown parameters (i.e., m_i, s_i, and J different $\delta_{i,j}$s). Thus, we further assume $\delta_{i,j} \sim N(0,1)$. This is equivalent to assuming that $\log(\mu_{i,j})$ for bin i is normally distributed, and m_i and s_i are its mean and SD respectively. This assumption is based on observing that CREs' log-normalized read counts after standardization (i.e. subtract m_i and divide by s_i) are approximately normally distributed (Figure 3-2). With this additional constraint, m_i and s_i become identifiable. Since maximum likelihood estimation for all genomic bins in a big genome like human is computationally slow, SCATE employs the method of moments to estimate m_i and s_i. Based on the model and theoretical moments of Poisson and Lognormal distributions, the first and second moments of $y_{i,j}/L_j$ are:

$$\begin{aligned} E\left(\frac{y_{i,j}}{L_j}\right) &= e^{m_i + \frac{1}{2}s_i^2} \\ E\left(\frac{y_{i,j}}{L_j}\right)^2 &= \frac{1}{L_j} e^{m_i + \frac{1}{2}s_i^2} + \left[e^{m_i + \frac{1}{2}s_i^2}\right]^2 e^{s_i^2} \end{aligned} \tag{3.3}$$

By matching the model-based moments to the empirical first two moments of the observed $y_{i,j}/L_j$s, we obtain the following closed-form estimates for m_i and s_i which can be computed efficiently:

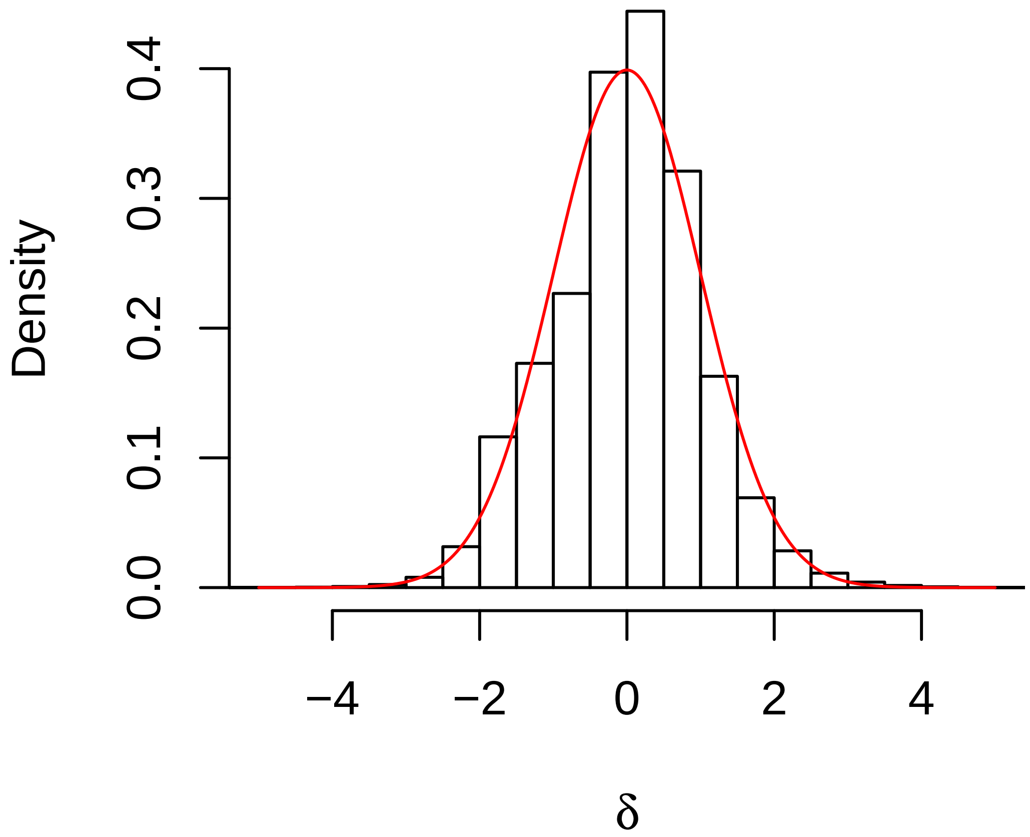


Figure 3-2. The empirical distribution (histogram) of the log-normalized read counts in human BDDB after standardization (i.e., subtract the mean and divide by SD of each CRE) can be fitted well with a normal distribution (red curve).

$$\begin{aligned}\tilde{s}_i &= \sqrt{\log\left(\frac{\sum_j (y_{i,j}/L_j)^2/J - \sum_j (y_{i,j}/L_j^2)/J}{(\sum_j (y_{i,j}/L_j)/J)^2}\right)} \\ \tilde{m}_i &= \log\left(\frac{\sum_j (y_{i,j}/L_j)}{J}\right) - \tilde{s}_i^2/2\end{aligned} \tag{3.4}$$

In rare cases where $\frac{\sum_j (y_{i,j}/L_j)^2/J - \sum_j (y_{i,j}/L_j^2)/J}{(\sum_j (y_{i,j}/L_j)/J)^2} < 1$, the estimates become:

$$
\begin{aligned}
&\tilde{s}_i = 0 \\
&\tilde{m}_i = \log\left(\frac{\sum_j (y_{i,j}/L_j)}{J}\right)
\end{aligned}
\tag{3.5}
$$

3.2.7 Estimate technical bias function $h_j(.)$

The cell-specific technical bias function $h_j(.)$ is estimated using known CREs whose activities do not change much across cell types. All known CREs are sorted according to $\tilde{s}_i$ estimated above which reflects their variability across diverse cell types in the bulk regulome data compendium. We split $\tilde{m}_i$ into ten groups by its 10%, 20%, ..., 100% quantiles, and find 1000 CREs with the smallest $\tilde{s}_i$ in each group. The union set of these 10000 CREs is a set $\mathscr{H}$ of "low-variability" CREs. For these low-variability CREs, their activities are almost constant across cell types. Thus, one can assume that their activities in a new cell are known and approximately equal to $\tilde{m}_i$, and the model for their scATAC-seq read counts in a new cell j can be simplified to:

$$
\begin{aligned}
&y_{i,j} \sim Poisson(L_j \mu^{sc}_{i,j}) \\
&\log(\mu^{sc}_{i,j}) = h_j(\log(\mu_{i,j})) \approx h_j(\tilde{m}_i)
\end{aligned}
\tag{3.6}
$$

We estimate $h_j(.)$ using $y_{i,j}$s from these low-variability CREs. The function $h_j(.)$ is monotonically increasing but has unknown form. We model it using monotone spline [86] (splines2 package in R):

$$
h_j(x) = \alpha_{j,0} + \sum_{t=1}^{T} \alpha_{j,t} I_t(x) \quad s.t.\ \alpha_{j,t} \geq 0\ (t = 1, ..., T)
$$

Here $I_t(x)$ are known I-spline basis functions (which are monotone functions [86]) and $\alpha_{j,t}$s are unknown regression coefficients. The constraints $\alpha_{j,t} \geq 0$ make $h_j(.)$ monotone and non-decreasing. The maximum likelihood estimates for coefficients $\boldsymbol{\alpha}_j = \{\alpha_{j,t} : t = 0, \ldots, T\}$ can then be obtained as:

$$\tilde{\boldsymbol{\alpha}}_j = \arg\max_{\boldsymbol{\alpha}_j} \sum_{i \in \mathscr{H}} [y_{i,j} * h(\tilde{m}_i) - L_j e^{h(\tilde{m}_i)}] \qquad s.t.\ \alpha_{j,t} \geq 0\ (t = 1, ..., T) \tag{3.7}$$

To select the optimal set of basis functions, we try different settings of knots by changing T. We set $T = 1, 2, ..., 6$, respectively, which sets the number of knots from 0 to 5. For each T, the t/T-th quantiles $(t = 1, ..., T - 1)$ of $\tilde{m}_i$ are chosen as the knots. Given the knots, the spline basis functions are then generated by splines2. The T with the smallest Bayesian information criterion (BIC) is chosen to obtain the optimal set of basis functions.

3.2.8 Estimate $\boldsymbol{\beta}$, $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$

Once the locus effects m_i and s_i and technical bias function $h_j(.)$ are estimated, SCATE treats them as known and will then estimate $\boldsymbol{\beta}$. Suppose CREs are grouped into K clusters. The activity for cluster k in cell j, $\beta_{k,j}$, can be estimated using the observed read counts in cell j for all CREs in the cluster. When data are sparse (particularly for clusters with small number of CREs), the maximum likelihood estimate can be unreliable due to its high variance. Thus, consistent with our bulk regulome data model, we impose a prior distribution on $\beta_{k,j}$ to help regularize its estimation: $\beta_{k,j} \sim N(0, 1)$. We then estimate $\beta_{k,j}$ using its posterior mode:

$$\tilde{\beta}_{k,j} = \arg\max_{\beta} \sum_{i \in C(k)} [y_{i,j} h_j(m_i + s_i\beta) - L_j e^{h_j(m_i + s_i\beta)}] - \beta^2/2$$

Here $C(k)$ represents the set of CREs in cluster k. The above optimization involves only one variable β, and thus the computation is not expensive. Estimation of different $\beta_{k,j}$s are handled separately.

Given $\tilde{\beta}_{k,j}$, $\delta_{i,j}$ and $\mu_{i,j}$ can be derived using model (3.1).

3.2.9 Analysis at multiple spatial resolution levels (i.e., multiple Ks)

SCATE analyzes data at multiple spatial resolution levels by setting the cluster number K to different values. To do so, known CREs are clustered based on their co-activation patterns across all samples in the bulk regulome data compendium. Before clustering, CREs' normalized data $\tilde{y}_{i,j}$ are organized as a matrix. Rows of the matrix correspond to CREs and columns correspond to samples. Each row is standardized to have zero mean and unit SD. Then CREs (i.e., rows) are clustered hierarchically at multiple granularity levels. A naive hierarchical clustering of 522,173 CREs (475,865 CREs for mouse) is difficult because it requires computing a distance matrix on the order of $500,000 \times 500,000$. To make the computation tractable, SCATE employs a three-stage clustering approach.

- Stage 1: CREs are grouped into 5000 clusters using K-means clustering (Euclidean distance). Each cluster contains approximately 100 CREs that show similar cross-sample activity patterns. For each cluster, the mean activity of all CREs in each sample is computed. It is then standardized to have zero mean and unit SD across samples.

- Stage 2: To obtain coarser clusters, the 5000 clusters from stage 1 are grouped hierarchically using hierarchical clustering (Euclidean distance, complete agglomeration) based on their mean activity profile. In this way, CREs are hierarchically grouped into 5000, 2500, 1250, 625, 312 and 156 clusters.

- Stage 3: To obtain fine-grained clusters, for each cluster obtained in Stage 1, hierarchical clustering is applied to split CREs in that cluster into smaller clusters. In this way, each cluster from Stage 1 can be divided into 2, 4, 8, ... subclusters until each subcluster contains only one CRE.

CREs' clustering structure for human and mouse obtained using our DNase-seq compendium is stored and provided as part of the SCATE package. Users can use it directly without recomputing them.

3.2.10 Optimizing spatial resolution (K) by cross-validation

SCATE optimizes the spatial resolution of the analysis by choosing the optimal K via cross-validation. For a given K, after clustering CREs, CREs are randomly partitioned into a training set (90% CREs) and a testing set (10% CREs). Next, for each cluster k, CREs in the training set are used to estimate $\beta_{k,j}$ which is the common activity of all CREs in that cluster. Using the estimated $\tilde{\beta}_{k,j}$, the log-likelihood of the test CREs in cluster k can be computed according to model (3.1) because they share the same $\beta_{k,j}$ with training CREs in the same cluster. We perform the same calculations for all clusters and obtain the median log-likelihood of all testing CREs.

The above procedure is run for different values of K. The cluster number K with the largest median log-likelihood in test data is selected as the optimal K.

3.2.11 Postprocessing – SCATE for other genomic bins in a single cell

After estimating activities of known CREs, SCATE will analyze all other bins in the genome. These bins fall into two classes. First, some bins have zero scATAC-seq read count across all cells. For these bins, $\mu_{i,j}$ is estimated to be zero. Second, the remaining bins have at least one read in the scATAC-seq data. For these bins, we estimate $\mu_{i,j}$ using a predictive machine learning approach xgboost (eXtreme Gradient Boosting [87]) where the response variable is the SCATE signal $\tilde{\mu}_{i,j}$ and the predictors are normalized read count $y_{i,j}/L_j$, m_i and s_i. The model is trained using known CREs. The trained model is then applied to bins not included in the known CRE list to make predictions. This will transform the read counts at these bins to a scale consistent with the reconstructed activities for known CREs.

3.2.12 SCATE for multiple cells

When a scATAC-seq dataset contains multiple cells, we first cluster cells using a method similar to our previously published method SCRAT [71]. Before clustering cells, CREs are grouped into 5000 clusters using BDDB as before. For each cell, the average activity of all CREs in each CRE cluster is calculated as in SCRAT. This transforms the scATAC-seq data in each cell into a feature vector consisting of 5000 CRE cluster activities. After quantile normalizing features across cells, features with low-variability across cells are filtered out. To identify low-variability features, for each feature we calculate the mean and SD of its activity across cells. Using the means and SDs of all features, we fit a polynomial regression with degree=3 to describe the relationship between the SD (response) and mean (independent variable). Features for which the observed SD is smaller than the expected SD (from the fitted model) given the mean activity are filtered out. Among the remaining high-variability features, we retain those that have non-zero read count in at least 10% of cells. PCA is then performed on the retained features. The top 50 principal components are then used to perform tSNE. The model-based clustering (mclust in R) [60] is used to perform clustering on tSNE space with default settings. The cluster number is chosen based on the Bayesian Information Criterion in mclust. If users do not want to use the default cluster number or clustering method, SCATE also provides an option to allow them to specify the cluster number by their own or use their own clustering results from other algorithms.

After cell clustering, each cluster consists of a set of similar cells and represents a relatively homogeneous cell subpopulation. SCATE will estimate the regulome profile of each cluster. For each cluster, reads from all cells are pooled together to create a pseudo-cell. The SCATE model for a single cell described above is then applied to the pseudo-cell to estimate CRE activities. The estimated regulome profile of the pooled sample typically will achieve higher spatial resolution than a single cell since (1) the pseudo-cell contains data from more than one cell and (2) SCATE automatically tunes the spatial resolution based on available information. The output of SCATE is the estimated regulome profile for each cell subpopulation.

3.2.13 Peak calling and evaluation

A moving average approach is used to call peaks from the reconstructed regulome profile. Given a moving window size $2W + 1$, the moving average signal for each 200 bp bin is calculated as the average signal of the bin and its $2W$ neighboring bins (W bins on the left and W bins on the right). By default, $W = 1$ which amounts to averaging signals from 3 bins spanning 600 bp in total. In parallel, we also calculate the average signal of $2W + 1$ randomly selected bins (not necessarily neighboring bins) for 100000 times to construct a background distribution for the moving average signal. For a genomic bin with moving average signal s, the false discovery rate (FDR) is estimated as the proportion of background distribution larger than s divided by the observed proportion of genomic bins with signals larger than s. Genomic bins with FDR smaller than 0.05 are identified and consecutive bins are merged into peaks. Peaks are ranked by FDR. For peaks tied with the same FDR, they are ranked further by the moving average signals.

For evaluation, peaks called using signals constructed by different methods are compared with peaks called using bulk regulome data. In the evaluation, we also assessed MACS peak calling on pooled cells. MACS is run with settings –nomodel –extsize 147.

3.2.14 TFBS prediction

TF motifs are downloaded from JASPAR [88]. These motifs were mapped to the genome using CisGenome with likelihood ratio cutoff = 100. Narrow peak files of the corresponding ChIP-seq data in GM12878 and K562 are downloaded from ENCODE. For each TF and cell type, genomic bins with motif were ranked based on reconstructed scATAC-seq signals to predict TFBSs. Genomic bins with motif that overlap with ChIP-seq peaks are used as gold standard.

3.2.15 Processing of benchmark bulk DNase-seq and ATAC-seq data

The benchmark bulk DNase-seq data for GM12878 and K562 (Dataset 1) are obtained from ENCODE. Bulk ATAC-seq data for human CMP and monocytes (Dataset 2) and human hematopoietic cell types in the last example are obtained from GEO under accession GSE74912. Bulk DNase-seq data for mouse brain and thymus (Dataset 3) are obtained from ENCODE.

Bulk DNase-seq samples are processed using the same protocol as DNase-seq data processing in BDDB. For ATAC-seq sample, reads are aligned to human genome hg19 using bowtie with parameters (-X 2000 -m 1). PCR duplicates are removed by Picard (http://broadinstitute.github.io/picard/). The aligned reads are used to obtain bin read counts.

3.2.16 Software

SCATE is freely available as an open source R package via GitHub and licensed under the MIT License:

https://github.com/zji90/SCATE

In terms of computational time, compiling CREs and clustering CREs typically take 1-2 days. Given the CRE list and CREs' clustering structure, running SCATE to reconstruct regulome approximately takes 5 minutes per cell cluster on a computer with 10 computing cores (2.5 GHz CPU/core) and a total of 20GB RAM.

3.3 Results

3.3.1 SCATE model for a single cell

SCATE begins with compiling a list of candidate CREs and grouping co-activated CREs into clusters. Currently, most scATAC-seq data are generated from human and mouse. For user's convenience, for these two species we have constructed a Bulk DNase-seq Database (BDDB) consisting of normalized DNase-seq samples from diverse cell types generated by the ENCODE project. For each species, we compiled putative CREs using BDDB and clustered these CREs based on their co-activation patterns across BDDB samples. Users may augment these precompiled CRE lists by using SCATE-provided functions to (1) add and normalize their own bulk and pseudo-bulk (obtained by pooling single cells) DNase-seq or ATAC-seq samples to BDDB and then (2) re-detect and cluster CREs using the updated BDDB. These functions can also be used to create CRE database for other species. For human and mouse, saturation analyses show that BDDB covers most CREs one would discover in a new DNase-seq or ATAC-seq dataset. On average, a new sample only contributes <0.2% new CREs to our precompiled CRE lists (Figure 3-3). Thus, in order to save time and computation for CRE detection and clustering, users may directly use the precompiled CRE lists in BDDB without significant loss. In this article, our analyses using SCATE are all carried out using these precompiled CREs as the input.

Given a list of CREs, their clustering structure, and scATAC-seq data from a single cell, the SCATE model contains the following key components (Figure 3-4A).

(1) Modeling a CRE's cell-independent but CRE-specific baseline behavior using publicly available bulk regulome data. By analyzing large amounts of ENCODE

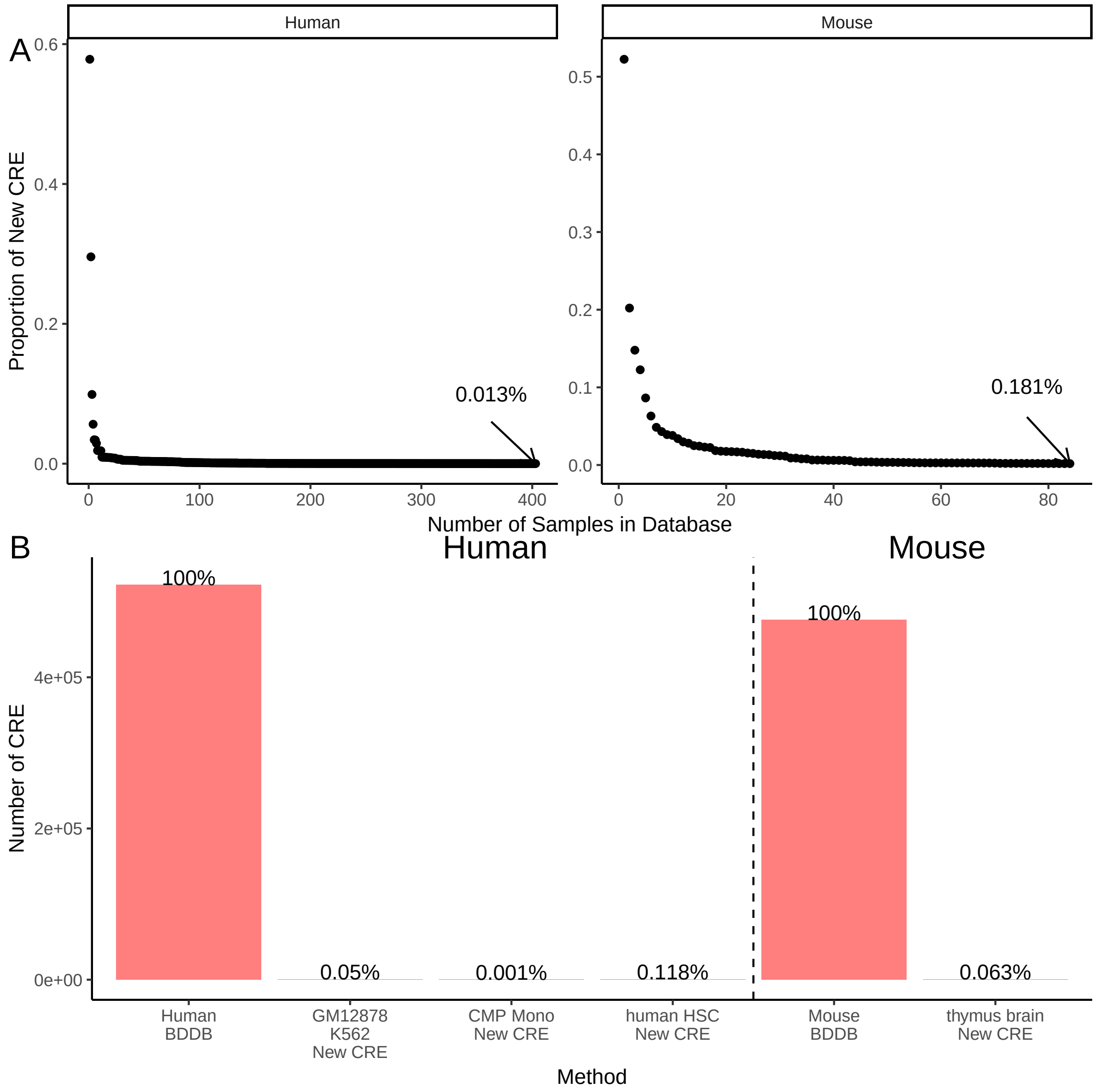


Figure 3-3. Saturation analysis of BDDB CRE lists. (A): As one increases the number of DNase-seq samples in the BDDB database, the proportion of new CREs contributed by adding a new sample gradually decreases. (B): The scATAC-seq datasets analyzed in this study would only add 0.0013%-0.118% new CREs to the precompiled CRE list in BDDB.

DNase-seq data, we found that these bulk data contain invaluable information not captured by the sparse single-cell data. In particular, our recent analysis of DNase-seq data from diverse cell types shows that different CREs have different baseline activities [89]. Some CREs tend to have higher activity levels than others regardless of cell type (Figure 3-1E: compare two CREs in blue boxes). As a result, the mean DNase-seq

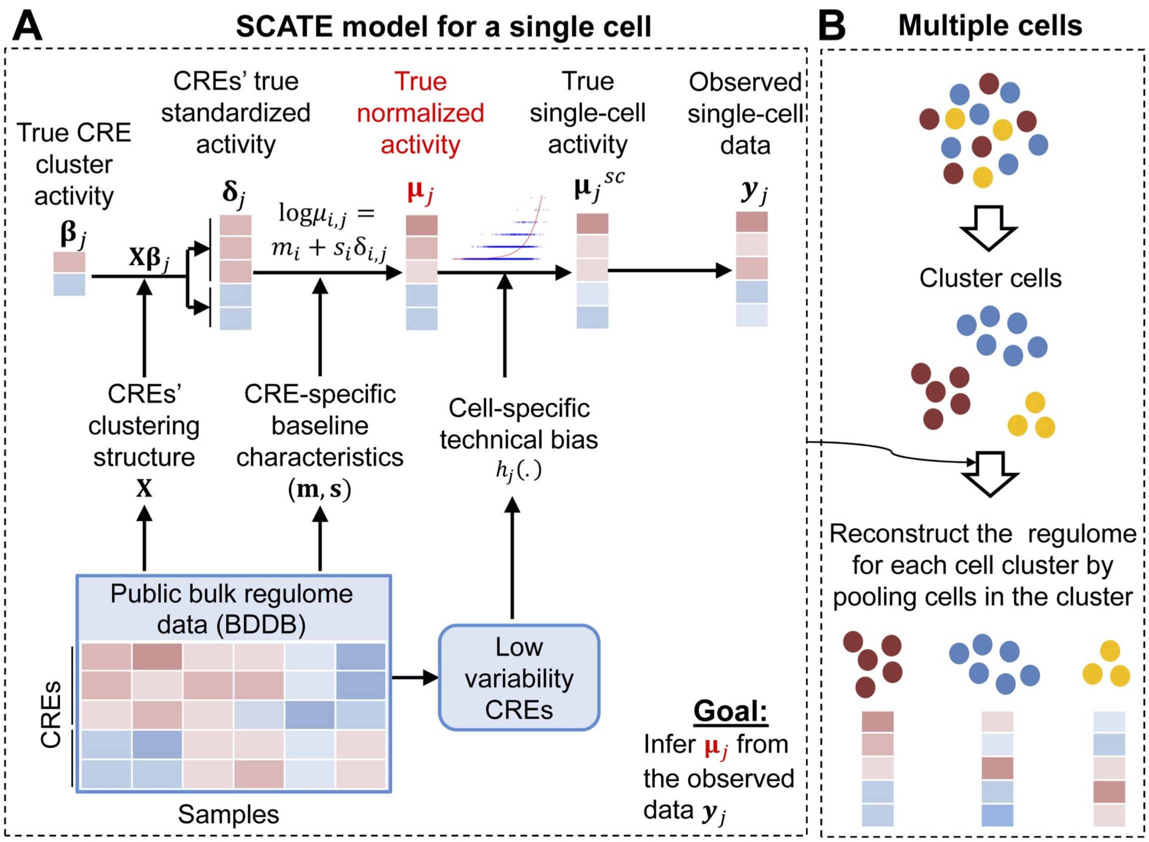

A
SCATE model for a single cell
True CRE cluster activity
$\boldsymbol{\beta}_j$
$\mathbf{X}\boldsymbol{\beta}_j$
CREs' true standardized activity
$\boldsymbol{\delta}_j$
$\log\mu_{i,j} =$
$m_i + s_i\delta_{i,j}$
True normalized activity
$\boldsymbol{\mu}_j$
True single-cell activity
$\boldsymbol{\mu}_j^{sc}$
Observed single-cell data
$\boldsymbol{y}_j$
CREs' clustering structure
$\mathbf{X}$
CRE-specific baseline characteristics
$(\mathbf{m},\mathbf{s})$
Cell-specific technical bias
$h_j(.)$
Public bulk regulome data (BDDB)
CREs
Samples
Low variability CREs
Goal:
Infer $\boldsymbol{\mu}_j$ from the observed data $\boldsymbol{y}_j$
B
Multiple cells
Cluster cells
Reconstruct the regulome for each cell cluster by pooling cells in the cluster

Figure 3-4. SCATE overview. (A): SCATE model for a single cell. (B): SCATE model for multiple cells.

profile across diverse cell types to a large extent can predict the DNase-seq profile in a new cell type, even though such prediction is cell-type-invariant and cannot capture cell-type-specific CRE activities. In [89], we found that the mean DNase-seq profile correlates well with independently measured TF binding activities, indicating that differences in the baseline activity among different CREs captured by the mean DNase-seq profile are real biological signals rather than technical artifacts. These highly reproducible CRE-specific baseline activities cannot be captured by the sparse data in a single cell or by pooling a small number of cells (Figure 3-1B,C,E). Thus, in order to better reconstruct activities of each individual CRE from scATAC-seq, SCATE explicitly models these cell-type-invariant but CRE-specific baseline behaviors by fitting a statistical model to the large compendium of bulk DNase-seq data in

BDDB. This allows us to estimate the baseline mean activity (m_i) and variability (s_i) of each CRE i.

(2) Modeling a CRE's cell-dependent activity by borrowing information from similar CREs. We model the activity of CRE i in cell j, denoted as $\mu_{i,j}$, by decomposing it into two components: a cell-type invariant component that models the baseline behavior (m_i and s_i), and a cell-dependent component $\delta_{i,j}$ for modeling the CRE's cell-specific activity. In other words, $\log(\mu_{i,j}) = m_i + s_i\delta_{i,j}$. The cell-type invariant component is learned from BDDB as described above. The cell-dependent component is learned using scATAC-seq data in each cell. To do so, we leverage CREs' clustering structure. Recall that co-activated CREs are grouped into clusters. We assume that CREs in the same cluster have the same $\delta_{i,j}$. Thus, information is shared across multiple co-activated CREs. Unlike other methods, we only share information through $\delta_{i,j}$ rather than assuming that $\mu_{i,j}$ is the same across similar CREs. In our approach, two CREs in the same cluster have the same δ, but they can have different activities (i.e., different μs) because of the difference in their CRE-specific baseline behaviors.

(3) Bulk and single-cell data normalization. Since CREs' baseline characteristics are learned from bulk DNase-seq data but our goal is to model scATAC-seq data, we need to reconcile differences between these two technologies. To do so, we assume that $\mu_{i,j}$ is the unobserved true activity of CRE i in cell j one would obtain if one could measure a bulk DNase-seq sample consisting of cells identical to cell j. In scATAC-seq data, $\mu_{i,j}$ is distorted to become $\mu^{sc}_{i,j}$ due to technical biases in scATAC-seq compared to bulk DNase-seq. These unknown technical biases are modeled using a cell-specific monotone function $h_j(.)$ such that $\log(\mu^{sc}_{i,j}) = h_j(\log(\mu_{i,j}))$. The observed scATAC-seq read count data are then modeled using Poisson distributions with mean $L_j\mu^{sc}_{i,j}$ where L_j is cell j's library size. The technical bias function $h_j(.)$ normalizes scATAC-seq and bulk DNase-seq data. We developed a method to estimate this unknown function by using CREs whose activities are nearly constant across diverse cell types in BDDB.

Once $h_j(.)$ is estimated, CRE activities $\delta_{i,j}$ and $\mu_{i,j}$ can be inferred by fitting the SCATE model to the observed read count data.

(4) Adaptively optimizing the analysis resolution based on available data. In order to examine the activity of each individual CRE, ideally one would hope to pool as few CREs as possible. However, when data are sparse, pooling too few CREs will lack the power to robustly distinguish biological signals from noise. Thus, the optimal analysis should carefully balance these two competing needs. All existing methods reviewed in category 1 pool CREs based on fixed and predefined pathways (e.g., all motif sites of a TF binding motif). They do not adaptively tune the analysis resolution based on the amount of available information. In SCATE, co-activated CREs are grouped into K clusters. Information is shared among CREs in the same cluster. We uniquely treat K as a tuning parameter and developed a cross-validation procedure to adaptively choose the optimal K based on the available data. When the data is highly sparse, SCATE will choose a small K so that each cluster contains a large number of CREs. As a result, the activity of a CRE will be estimated by borrowing information from many other CREs. This sacrifices some CRE-specific information in exchange for higher estimation precision (i.e., lower estimation variance). When the data is less sparse and more CREs have non-zero read counts, SCATE will choose a large K so that each cluster will contain a small number of CREs. As a result, the CRE activity estimation will borrow information from only a few most similar CREs, and more CRE-specific information will be retained.

(5) Postprocessing. After estimating CRE activities, we will further process all genomic regions outside the input CRE list. SCATE will transform read counts at these remaining regions to bring them to a scale normalized with the reconstructed CRE activities. The transformed data can then be used for downstream analyses such as peak calling, TF binding site prediction, or other whole-genome analyses.

3.3.2 SCATE for a cell population consisting of multiple cells

For a homogeneous cell population with multiple cells, we will pool reads from all cells together to create a pseudo-cell. We will then treat the pseudo-cell as a single cell and apply SCATE to reconstruct CRE activities. Similar to Dr.seq2, this approach combines similar cells to estimate CRE activities. Unlike Dr.seq2, we also combine information from co-activated CREs and public bulk regulome data as described above. Moreover, SCATE adaptively tunes the resolution for combining CREs (i.e. the CRE cluster number K) which is lacking in other methods. As the cell number in the population increases, the sparsity of the pseudo-cell will decrease and the optimal analysis resolution chosen by SCATE typically will increase.

For a heterogeneous cell population, we first group similar cells into clusters. SCATE is then applied to each cell cluster to reconstruct CRE activities by treating the cluster as a homogeneous cell population (Figure 3-4B). By default, SCATE uses model-based clustering [60] to cluster cells, and the cluster number is automatically chosen by the Bayesian Information Criterion (BIC). Since one clustering method is unlikely to be optimal for all applications, we also provide users with the option to adjust the cluster number or provide their own cell clustering. SCATE can be run using user-specified cluster number or clustering results. For example, if users believe that the default clustering does not sufficiently capture the heterogeneity, they could increase the cluster number. In the most extreme case, if one sets the cluster number equal to the cell number, each cluster will become a single cell.

We note that pooling cells in each cluster to create a pseudobulk sample does not mean that the value of single-cell analysis is lost or that scATAC-seq can be replaced by bulk ATAC-seq or DNase-seq. This is because bulk ATAC-seq or DNase-seq analysis of a heterogeneous sample cannot separate different cell subpopulations or discover new cell types. Even if one could use cell sorting to separate cells in a sample by cell type and then apply bulk analysis to each cell type, the sorting relies on known cell type markers and therefore cannot discover new cell types. By contrast, a scATAC-seq experiment coupled with SCATE can identify and characterize different cell populations including potentially new cell types in a heterogeneous sample.

3.3.3 Benchmark data

We compiled three datasets for method evaluation. Dataset 1 consists of human scATAC-seq data from two different cell lines GM12878 (220 cells) and K562 (157 cells) generated by [32]. For this dataset, ENCODE bulk DNase-seq data for GM12878 and K562 were used as the gold standard to evaluate signal reconstruction accuracy. Dataset 2 contains scATAC-seq data from human common myeloid progenitor (CMP) cells (637 cells) and monocytes (83 cells) obtained from [83, 90]. We also obtained bulk ATAC-seq data from human CMP and monocytes generated by [90] and used them as gold standard. Dateset 3 consists of mouse scATAC-seq data from brain (3321 cells) and thymus (7775 cells) generated by [84]. For evaluation, the ENCODE bulk DNase-seq data for mouse brain and thymus were used as gold standard. In all evaluations, we removed the test cell types from the BDDB before running SCATE in order to avoid using the same bulk regulome data in both SCATE model fitting and performance evaluation.

3.3.4 Analysis of a homogeneous cell population - a demonstration

We first demonstrate SCATE analysis of a homogeneous cell population using the GM12878 and K562 data (Dataset 1) as an example. We applied SCATE to each cell type separately. For each cell type, we randomly sampled n (n = 1, 5, 10, 25, 50, 100, etc.) cells and pooled their sequence reads together to run SCATE. CRE activities reconstructed by SCATE were compared with their activities measured by bulk DNase-seq in the corresponding cell type.

Figure 3-5 shows the normalization function $h_j(.)$ learned by SCATE for normalizing scATAC-seq and the BDDB bulk DNase-seq data. Each scatter plot corresponds to a pooled scATAC-seq sample. Different plots represent different cell numbers or cell types. In these plots, each data point is a low-variability CRE with nearly constant activity across BDDB samples. For each CRE, the read count in the pooled scATAC-seq sample (Y-axis) versus the CRE's baseline mean activity in BDDB (X-axis) are shown. The red curve is the SCATE-fitted function ($e^{h_j(.)}$) for modeling technical biases in scATAC-seq. Overall, scATAC-seq read counts were positively correlated with CREs' baseline activities at these low-variability CREs, and the SCATE-fitted normalization functions were able to capture the systematic relationship (i.e., technical biases) between the scATAC-seq and bulk DNase-seq data.

Figure 3-6 shows the number of CRE clusters adaptively chosen by SCATE. For each cell type, there are four plots corresponding to SCATE analyses by pooling different number of cells, with the cell number n shown on top of each plot. For each n, n cells were randomly sampled from the scATAC-seq dataset and pooled. SCATE was applied to the pooled data to automatically choose the CRE cluster number. This procedure was repeated ten times. The histogram shows the empirical distribution of the cluster number chosen by SCATE in these ten independent cell samplings without using any information from the gold standard bulk DNase-seq. As a benchmark, we also ran SCATE by manually setting the CRE cluster number K to different values. For each K, we computed the Pearson correlation between the SCATE-estimated CRE activities in scATAC-seq and the gold standard CRE activities in bulk DNase-seq. The dots in each plot show the correlation coefficients for different Ks, also averaged across the ten independent cell samplings. The dot with the largest correlation coefficient corresponds to the true optimal cluster number. In real applications this true optimal cluster number would be unknown because one would not have the bulk DNase-seq as

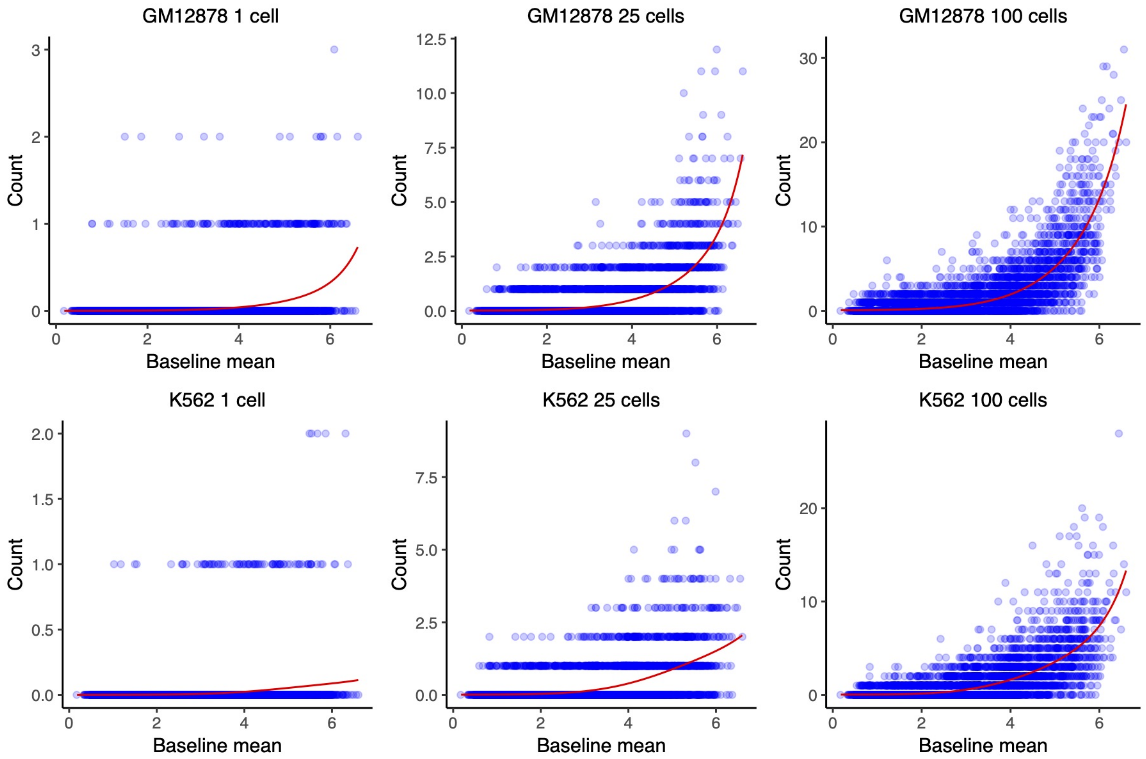


Figure 3-5. Normalization of scATAC-seq and bulk DNase-seq data. The scATAC-seq read counts versus baseline mean activities are shown for low-variability CREs in GM12878 (top panel) and K562 (bottom panel). Each blue dot is a low-variability CRE, defined as a CRE with almost constant activity across diverse cell types in BDDB bulk DNase-seq samples. Different plots correspond to analyses based on pooling different number of cells. In each plot, the red curve is the technical bias function fitted by SCATE.

the gold standard to help with choosing K.

Figure 3-6 shows that the CRE cluster number automatically chosen by SCATE (histogram) typically was close to the true optimal cluster number (the dot with the highest correlation). For instance, for analyzing a single GM12878 cell, the cluster number chosen by SCATE had its mode at 1250, and the true optimal cluster number was 2500. For analyzing 220 GM12878 cells, the cluster number chosen by SCATE had its mode at 521820, and the true optimal cluster number was also 521820.

Figure 3-6 also shows that, as the cell number increases, both the true optimal CRE cluster number and the cluster number chosen by SCATE also increase. Increasing

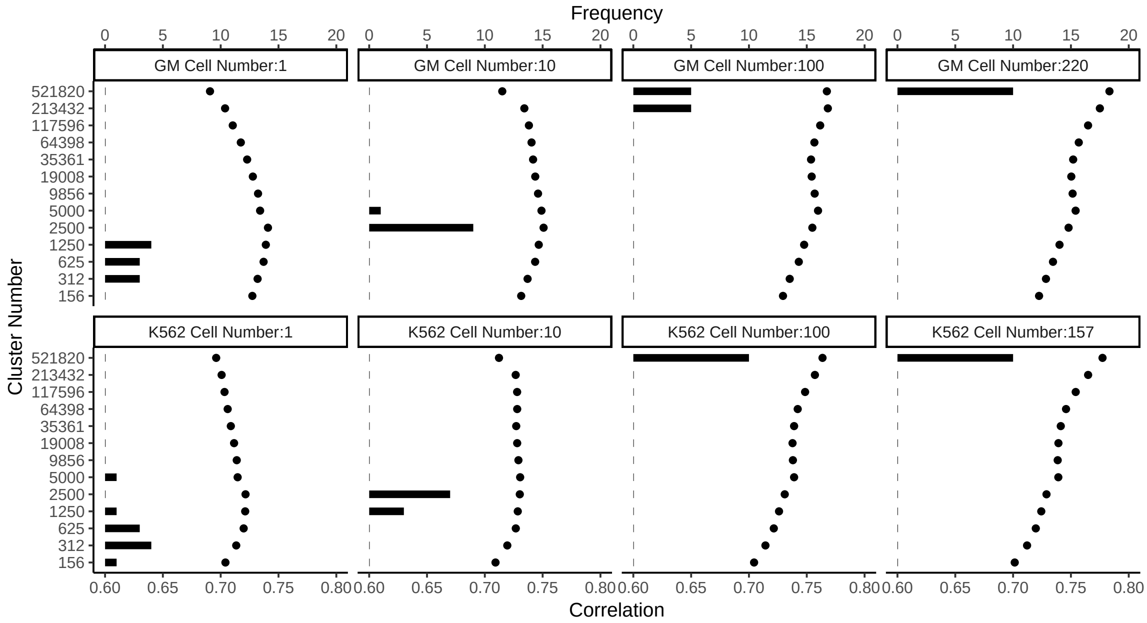


Figure 3-6. Adaptive tuning of analysis resolution. The number of CRE clusters automatically chosen by SCATE via cross-validation (histogram) is compared with the true optimal CRE cluster number determined by external information from the gold standard bulk DNase-seq data (dots). Different plots correspond to different cell types and pooled cell number. In each plot, the histogram shows the CRE cluster number chosen by SCATE in 10 independent cell samplings. The dots show the true correlation between the gold standard bulk DNase-seq signal and the SCATE-reconstructed scATAC-seq signal (both at log-scale) at each CRE cluster number, averaged across the 10 cell samplings. The dot with the highest correlation is the true optimal cluster number.

cluster number implies decreasing cluster size. Thus, SCATE adaptively changes analysis resolution: as more data are available for each CRE, SCATE gradually decreases the number of CREs in each cluster for information sharing. This allows SCATE to maximally retain CRE-specific information.

Figure 3-7 compares SCATE-reconstructed scATAC-seq signal with bulk DNase-seq signal in GM12878 and K562 in an example genomic region. The figure has six columns corresponding to different cell types and different pooled cell numbers. For benchmark purpose, the figure also compares SCATE with a number of other methods, all run based on 200bp non-overlapping genomic windows. Here “Raw reads” displays the scATAC-seq read count pooled across cells for each genomic window. This approach is used by Dr.seq2. Raw read counts are also used by scABC to characterize CRE activities in single cells, but scABC does not pool cells. “Binary” converts read counts in each cell to a binary accessibility vector and then adds up the binary accessibility vectors across cells. This approach is used by Cicero. Binary accessibility is also used by Scasat, Destin, PRISM and cisTopic as their data matrix. ChromVAR, SCRAT and BROCKMAN only analyze and report aggregated CRE pathway activities rather than activities of individual CREs. Thus, they cannot be compared here. However, for our previously developed SCRAT, we were able to modify the codes to estimate CRE activities by directly using pathway activities. This results in three methods, “SCRAT 500 CRE cluster”, “SCRAT 1000 CRE cluster” and “SCRAT 2000 CRE cluster” shown in the figure. Here, CREs were clustered into 500, 1000 or 2000 clusters as in SCRAT using the bulk DNase-seq data in BDDB. For each CRE cluster, the average normalized scATAC-seq read count across all CREs in the cluster was calculated. It was then assigned back to each CRE in the cluster to represent the estimated CRE activity. The “Raw reads” method can be viewed as a special case of the “SCRAT CRE cluster” method when the cluster number is equal to the CRE number (i.e., each CRE is a cluster). “Average DNase-seq” shows the average normalized read count profile of bulk DNase-seq samples in BDDB. It reflects CRE’s baseline mean activity.

Figure 3-7 shows that SCATE-reconstructed scATAC-seq signals accurately captured the variation of CRE activities in bulk DNase-seq across different genomic loci and different cell types, whereas CRE activities estimated using raw read counts, binarized chromatin accessibility, or SCRAT CRE cluster methods all failed to accurately capture the bulk DNase-seq landscape. Interestingly, SCATE was able to use scATAC-seq data from one single cell to accurately estimate CRE activities in bulk DNase-seq. By contrast, the raw read count and binary accessibility methods both failed due to data sparsity (e.g., see regions in blue boxes). The SCRAT CRE cluster method also failed because (1) it assigns the same activity to all CREs in the same

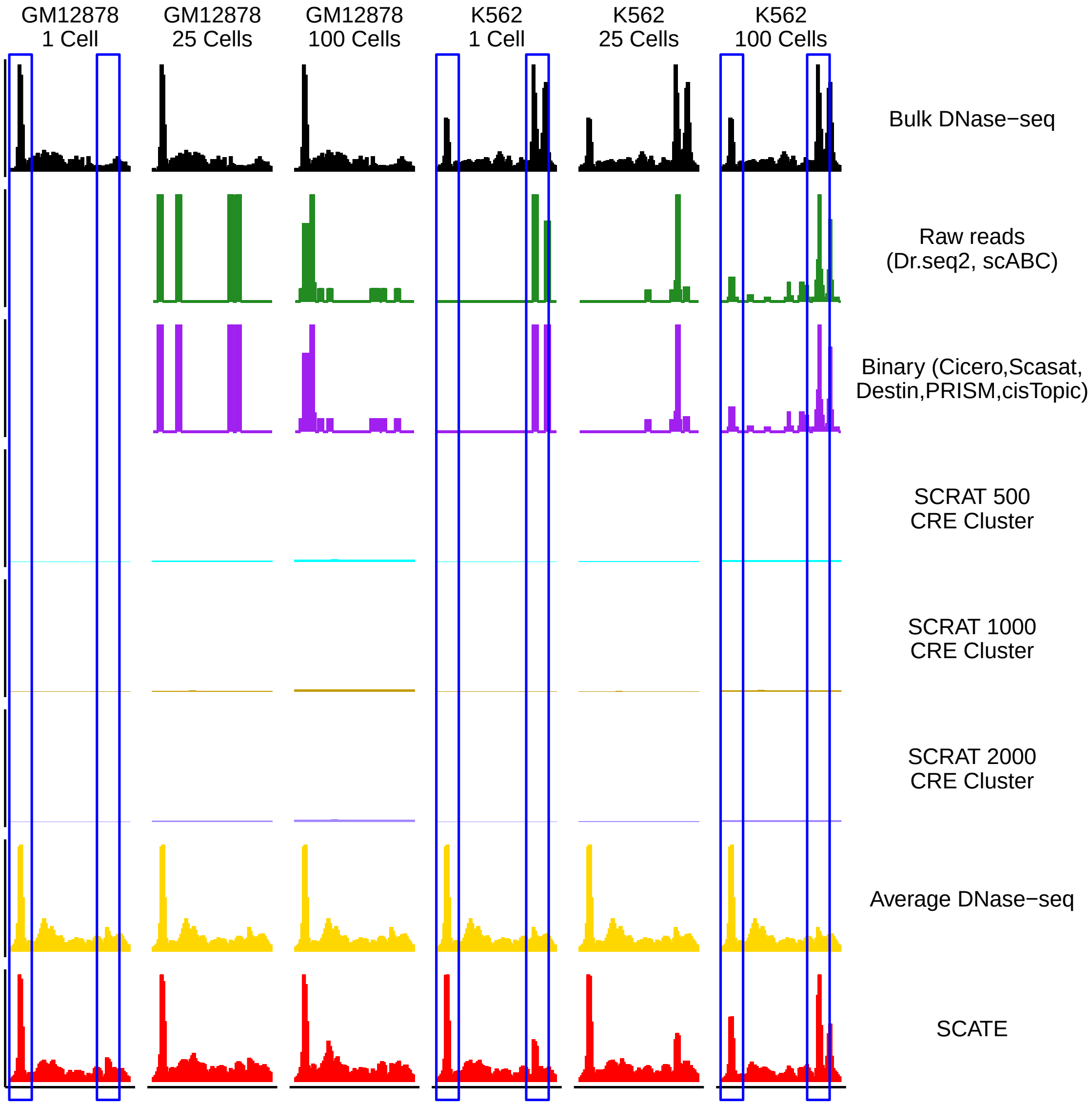


Figure 3-7. Comparison of different methods in an example genomic region. Each row is a method, each column corresponds to a different cell type or pooled cell number. All columns show the same genomic region. The blue boxes highlight two CREs. The left CRE occurs in both GM12878 and K562. It cannot be detected by Raw reads, Binary and SCRAT CRE cluster methods in a single cell, but can be detected by Average DNase-seq and SCATE. The right CRE is K562-specific. It cannot be detected by Average DNase-seq but can be detected by SCATE.

CRE cluster and ignores CRE-specific behaviors, and (2) it does not adaptively tune the analysis resolution as in SCATE to maximally retain CRE-specific signals. The "Average DNase-seq" approach produced relatively continuous signals and captured some variation across genomic loci in the GM12878 and K562 bulk DNase-seq data. However, it was unable to capture cell-type-specific signals, such as those shown in the blue boxes.

3.3.5 Analysis of a homogeneous cell population - a systematic evaluation

Next, we systematically evaluated SCATE and the other methods in all three benchmark datasets by treating the six test cell types as six homogeneous cell populations. The evaluation was based on the correlation with gold standard bulk regulome data, peak calling performance using reconstructed signals, and ability to predict transcription factor binding sites (TFBSs). Note that even though each test cell type could potentially be decomposed further into multiple cell subtypes, we could not conduct the analysis at the cell subtype level because the gold standard bulk regulome data for those cell subtypes are unavailable and the subtype label of each cell is unknown. Thus, for benchmark purpose, here we defined "homogeneous" at a coarser scale and view cells from each test cell type as homogeneous. This is reasonable because according to statistical theory, cells in the same cell population (regardless of the composition of the population) are exchangeable in the sense that, without knowing the finer structure of the population, the expectation of the behavior of any cell randomly drawn from the population is equal to the population's bulk (mean) behavior.

In the first evaluation, we computed the Pearson correlation between the scATAC-seq signals reconstructed by each method and the gold standard bulk signals across all CREs. As one example, Figure 3-8A shows the results based on pooling scATAC-seq data from 10 GM12878 cells. Among all methods, SCATE showed the highest correlation with the bulk gold standard. We performed the same analysis on all six test cell types by pooling different cell numbers. For each cell number, we repeated the analysis ten times using ten independent cell samplings. The median performance of the ten analyses was then compared. Figure 3-8B shows that SCATE consistently outperformed all the other methods and showed the strongest correlation with the bulk gold standards in all test data. When the pooled cell number was small, the improvement of SCATE over many methods was substantial. For instance, for the analysis of one single Monocyte cell, the correlation was 0.22, 0.22, 0.57, 0.57 and 0.57 for Raw reads, Binary, SCRAT 500, 1000 and 2000 CRE cluster methods, respectively. For SCATE, it was 0.67, representing an improvement of 18%∼205% over the other methods. Of note, the Average DNase-seq method performed relatively well in this evaluation when the cell number was small. However, as we will show later, the average DNase-seq profile cannot predict changes in CRE activity between different cell types, but SCATE can.

In the second evaluation, we performed peak calling using scATAC-seq signals reconstucted by SCATE and other methods. Peak calling is a common task in DNase-seq or ATAC-seq data analyses. Its objective is to find genomic regions with significantly enriched signals. We implemented a peak calling algorithm using a moving average approach (see Methods) and applied it to signals reconstructed by each method (SCATE, Raw reads, Binary, SCRAT CRE cluster, and Average DNase-seq). In addition, we also performed peak calling by applying MACS2 [82] to the pseudobulk sample obtained by pooling cells. The peak calling performance of each method was evaluated using the sensitivity versus false discovery rate (FDR) curve, where the "truth" was defined by the peaks called from the bulk gold standard data. Here sensitivity is the proportion of true bulk peaks discovered by scATAC-seq, and FDR is the proportion of scATAC-seq peaks that are false (i.e., not found in bulk peaks). As one example, Figure 3-9A compares the sensitivity-FDR curves of different methods when they were applied to the pooled scATAC-seq data from 25 GM12878 cells. For each curve, we computed the area under the curve (AUC). Figure 3-9B systematically compares the AUCs of all methods in all six test cell types. In each

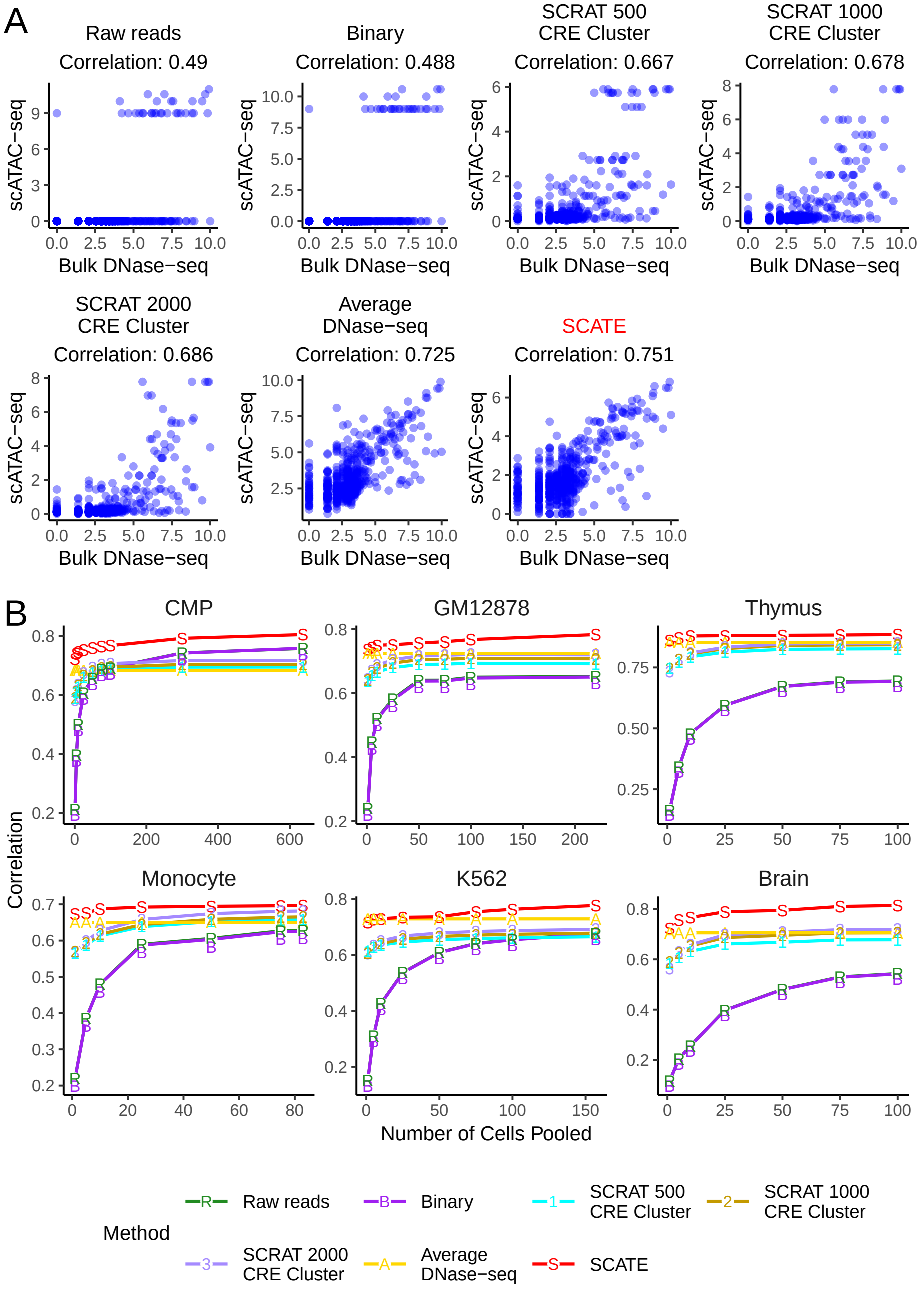


Figure 3-8. Correlation between reconstructed and true CRE activities. (A): Scatterplots showing true bulk CRE activities vs. CRE activities estimated by different methods in an analysis that pools 10 GM12878 cells. In this analysis, both activities are at log-scale. (B): The correlation between the scATAC-seq reconstructed and true bulk regulome for different methods. Each plot corresponds to a test cell type. In each plot, the correlation is shown as a function of the pooled cell number.

plot, the analyses were run by pooling different numbers of cells, and the median AUC from 10 independent cell samplings was plotted as a function of the cell number. Once again, SCATE showed the best overall peak calling performance. When the cell number was small, the improvement was substantial. For analyzing one Monocyte cell, for example, the AUC of SCATE was 0.4, whereas the AUCs for the other methods (except for Average DNase-seq) were all below 0.21. Thus, SCATE improved over these methods by 90% or more.

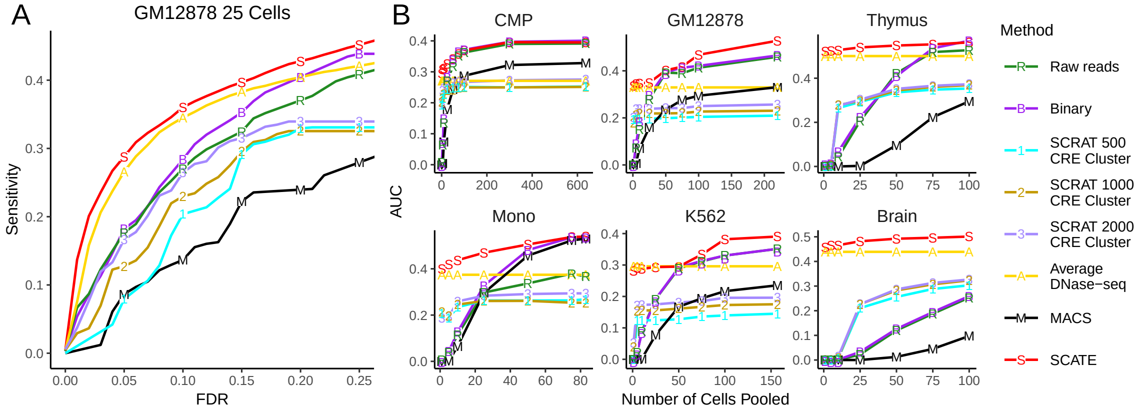


Figure 3-9. Peak calling performance. (A): The sensitivity versus FDR curve is shown for different peak calling methods in an analysis that pools 25 GM12878 cells. (B): The area under the sensitivity-FDR curve (AUC) is shown as a function of pooled cell number for different methods. Each plot corresponds to a different test cell type.

In the third evaluation, we used signals reconstructed by each method to predict TFBSs. We evaluated 28 TFs in GM12878 and 29 TFs in K562. As gold standard, we collected ChIP-seq peaks for these TFs from the ENCODE [80]. For the other cell types, we did not find TF ChIP-seq data suitable for evaluation. Therefore, our TFBS prediciton analysis was focused on GM12878 and K562. To predict TFBSs of a TF, we mapped its motif sites in the genome using CisGenome [91]. Genomic windows overlapping with motif sites were sorted based on their reconstructed scATAC-seq signals. Windows with the highest signals were labeled as predicted TFBSs (Figure 3-10A). Motif-containing windows that overlap with TF ChIP-seq peaks were viewed as gold standard true TFBSs. Based on this, we generated the sensitivity-FDR curve for each TF by gradually relaxing the TFBS calling cutoff. As one example, Figure 3-10B shows the sensitivity-FDR curves of different methods for predicting ELF1 binding sites by pooling scATAC-seq data from 25 GM12878 cells. For each TF and cell type, we performed this analysis using different cell numbers. For each cell number, the median area under the sensitivity-FDR curve (AUC) of 10 independent cell samplings was computed. As two examples, Figure 3-10C shows the AUCs for different methods as a function of pooled cell number for two TFs: ELF1 in GM12878 and JUND in K562. Finally, Figure 3-10D shows the average performance of all 28 TFs in GM12878 and 29 TFs in K562. In all these analyses, SCATE robustly outperformed all the other methods. The overall improvement was substantial (e.g., see K562 in Figure 3-10D).

3.3.6 Analysis of a heterogeneous cell population - demonstration and systematic evaluation

To demonstrate the analysis of a heterogeneous cell population, we mixed GM12878 and K562 cells from Dataset 1 with different ratios to create synthetic samples with different heterogeneity levels. Each synthetic sample had 100 cells representing a mixture of GM12878 and K562 cells. The percentage of GM12878 cells was set to $x =$ 10%, 30% and 50%, respectively. For each percentage x, ten synthetic samples were created using independently sampled cells. The median performance of each method on the ten analyses was compared.

Each synthetic sample was analyzed by first clustering cells using the default cell clustering algorithm in SCATE. SCATE and other methods were then used to estimate CRE activities for each cell cluster. The number of cell clusters automatically determined by SCATE in these samples ranged from 2-5 (Figure 3-11A). Figure 3-11B shows one example in which cells were grouped into 2 clusters.

In order to evaluate whether the analysis can discover the true biology, we first

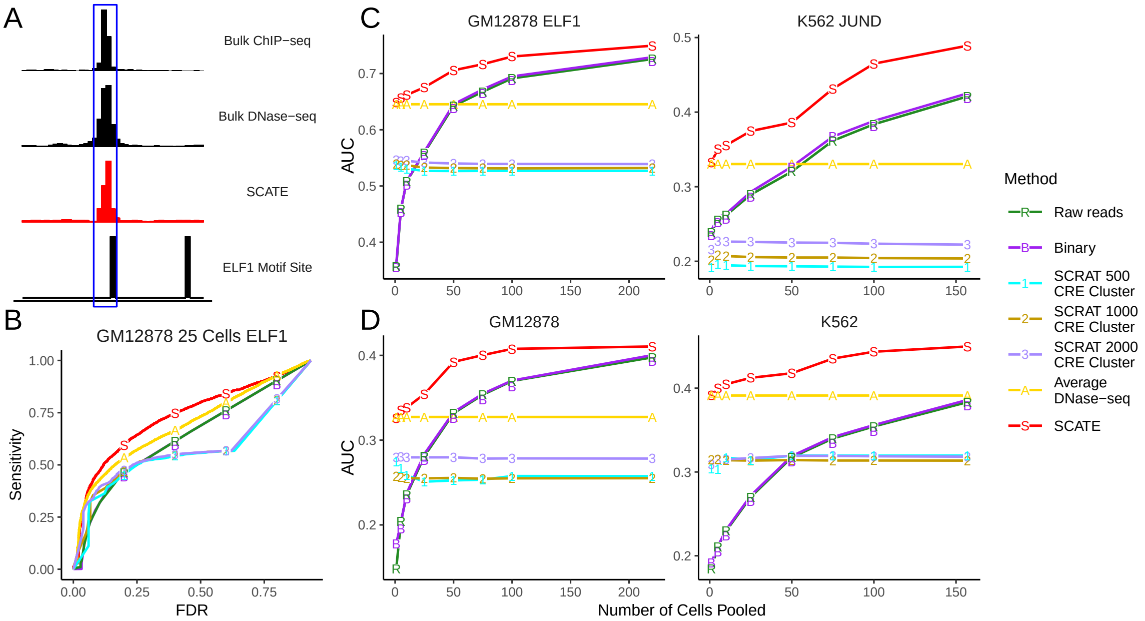


Figure 3-10. TFBS prediction performance. (A): An illustration of TFBS prediction in an example genomic region. The region contains a genomic bin with ELF1 motif and high SCATE-reconstructed CRE activity in GM12878. The bin is predicted as a ELF1 binding site. The prediction can be validated by ELF1 ChIP-seq peak in GM12878. (B): An example sensitivity versus FDR curve for comparing different methods for predicting ELF1 TFBSs in an analysis that pools 25 GM12878 cells. (C): Two examples (ELF1 in GM12878 and JUND in K562) that illustrate the method comparison across different cell numbers. In each example, analyses are performed by pooling different numbers of cells. The median AUC under the sensitivity-FDR curve from 10 independent cell samplings is shown as a function of pooled cell number. (D): The averaged AUC across all TFs is shown as a function of pooled cell number in GM12878 and K562 respectively.

annotated each cell cluster based on its dominant cell type. A cell cluster was labeled as "predicted GM12878" if over 70% of cells in the cluster were indeed GM12878 cells. Similarly, a cell cluster with ≥70% K562 cells was labeled as "predicted K562". All other clusters were labeled as "ambiguous". For a given sample, if at least one cell cluster was labeled as "predicted cell type X" (X = GM12878 or K562), we say that cell type X was detected. Based on this definition, both GM12878 and K562 can be detected in all samples (Figure 3-11C). Note that one cell type may be identified by multiple cell clusters. Given the cell type annotation, we then compared the regulome of each cell type reconstructed by SCATE and other methods. Since all methods used

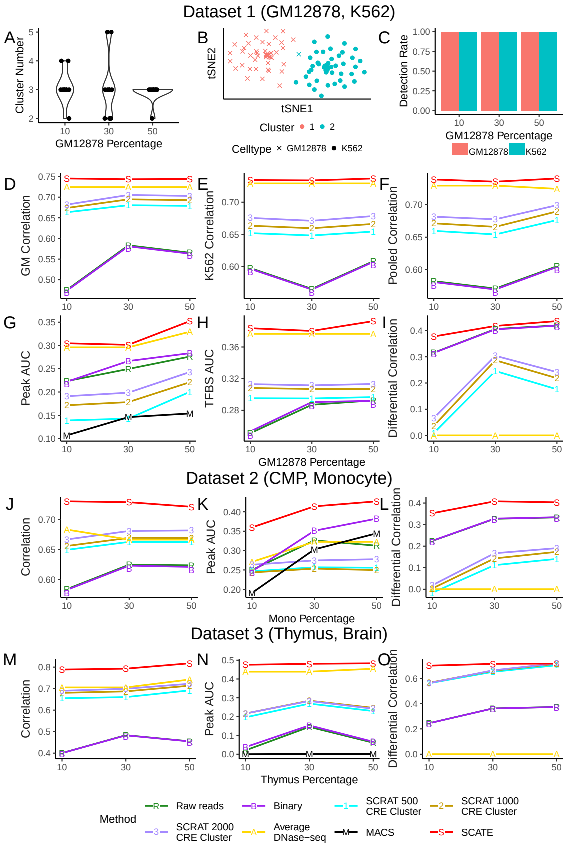


Figure 3-11. Analyses of a heterogeneous cell population. (A): Distribution of cell cluster numbers obtained by SCATE for synthetic samples with different cell mixing proportions. GM12878 and K562 cells are mixed at different proportions. For each mixing proportion, 10 synthetic samples are created and analyzed. (B): An example tSNE plot showing clustering of cells in a synthetic sample. (C): At each cell mixing proportion, the frequency that each cell type is detected in the 10 synthetic samples is shown. (D)-(F): The correlation between the scATAC-seq reconstructed and true bulk regulome in (D) GM12878, (E) K562, and (F) GM12878 and K562 combined for different methods is shown as a function of cell mixing proportion (GM12878 cell percentage). (G): The peak calling AUC (GM12878 and K562 combined) vs. cell mixing proportion. (H): The TFBS prediction AUC (GM12878 and K562 combined) vs. cell mixing proportion. (I): The correlation between the scATAC-seq reconstructed and true bulk differential log-CRE activities is shown as a function of cell mixing proportion. (J)-(L): Similar analyses in samples consisting of human CMP and monocyte cells, including (J) correlation between reconstructed and true bulk log-CRE activities, (K) peak calling AUC, and (L) correlation between predicted and true differential log-CRE activities. (M)-(O): Similar analyses in samples consisting of mouse thymus and brain cells, including (M) correlation between reconstructed and true bulk log-CRE activities, (K) peak calling AUC, and (L) correlation between predicted and true differential log-CRE activities.

the same cell clustering results, the comparison of their signal reconstruction ability is a fair comparison. We conducted four types of comparisons.

First, we asked whether the regulome reconstructed by each method for each predicted cell type can accurately recover the cell type's true regulome measured by the gold standard bulk data. Take GM12878 as an example. For each cell cluster predicted as GM12878, the Pearson correlation between the cluster's reconstructed scATAC-seq signal and the gold standard bulk GM12878 DNase-seq data was computed. If a sample had two or more cell clusters predicted as GM12878, each cluster was analyzed separately. The median correlation of all such clusters in ten independent synthetic samples is shown in Figure 3-11D. SCATE again performed the best. When the proportion of GM12878 cells in a sample was small, the improvement by SCATE was larger. Figure 3-11E shows the same analysis for K562, but the performance was shown as a function of GM12878 cell proportion. Figure 3-11F shows the combined results. Here at each cell mixing proportion, the median scATAC-bulk correlation of all cell clusters predicted either as GM12878 or K562 was shown. In all these analyses, SCATE consistently performed the best.

Second, we conducted peak calling and evaluated each method's ability to recover true peaks in each cell type. Here the truth was defined as peaks called from the gold standard bulk data, and the evaluation was conducted similar to Figure 3-9. Figure 3-11G shows the median AUC of all cell clusters predicted either as GM12878 or K562 as a function of cell mixing proportion. SCATE robustly outperformed the other methods.

Third, we compared different methods in terms of their ability to predict TFBSs. TFBS prediction and evaluation were performed similar to Figure 3-10. The results are shown in Figure 3-11H, in which the median AUC for each method is plotted as a function of cell mixing proportion. SCATE produced the best prediction accuracy.

Last but not least, we applied different methods to predict differential CRE activities between different cell types, which is crucial for characterizing the regulatory landscape of a heterogeneous sample. Here we collected all pairs of cell clusters that were predicted as two different cell types (i.e., one cluster was "predicted GM12878" and the other cluster was "predicted K562"; ambiguous cell clusters were excluded). For each such pair, we computed the difference of reconstructed CRE activities between the two cell clusters. We then compared this predicted difference with the true differential CRE activities derived from the gold standard bulk DNase-seq data for GM12878 and K562. The Pearson correlation between the predicted and true differential signals was calculated. As one example, Figure 3-12 shows the results for a cell cluster pair in a synthetic sample in which 30% of cells was GM12878. SCATE best recovered the differential CRE activities (Correlation = 0.43). Figure 3-11I shows the median correlation across ten independent synthetic samples at each cell mixing proportion. Once again, SCATE performed the best.

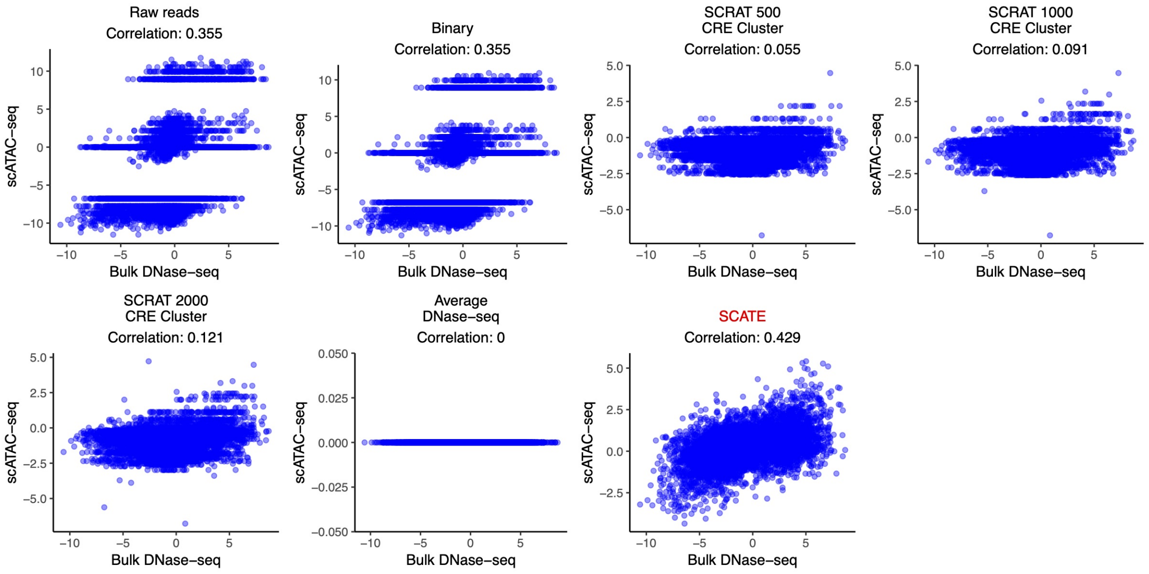


Figure 3-12. An example of predicting differential CRE activities. Scatterplots showing true bulk differential log-CRE activities vs. differential log-CRE activities estimated by different methods in an analysis of a synthetic sample consisting of 30 GM12878 and 70 K562 cells.

We note that the Average DNase-seq method completely failed for predicting differential signals between two cell types (Correlation = 0) (Figs. 9I,10), even though it performed relatively well for estimating CRE activities within one cell type, and peak calling and TFBS prediction in one cell type (Figs. 6,7,8,9F-H). Similarly, each of the other methods may perform well in some datasets or analyses but not in others. SCATE is the only method that robustly performed the best in all our analyses.

Similar to GM12878 and K562 (Dataset 1), we also constructed heterogeneous cell populations using the other two datasets (Datasets 2 and 3) and used them to evaluate different methods. The results are shown in Figure 3-11J-O and Figure 3-13. For these two datasets, we did not perform TFBS prediction due to lack of gold standard ChIP-seq data. For estimating CRE activities (Figure 3-11J,M), peak calling (Figure 3-11K,N) and predicting differential CRE activities (Figure 3-11L,O), SCATE again outperformed all the other methods. In many cases, the improvement was substantial (e.g., Figure 3-11K,L,N,O).

3.3.7 Analysis of scATAC-seq data from human hematopoietic differentiation

To further demonstrate and evaluate SCATE, we analyzed a scATAC-seq dataset generated by [83] which consists of 1920 cells from 8 human hematopoietic cell types for which corresponding bulk ATAC-seq data are available. These cell types include hematopoietic stem cell (HSC), multipotent progenitor (MPP), lymphoid-primed multipotent progenitor (LMPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythrocyte progenitor (MEP) and monocyte (Mono). In this dataset, the true cell type label of each cell was known since cells were obtained by cell sorting. Figure 3-14A shows the tSNE [92] plot of all cells color-coded by their true cell types. In the plot, different cell types were distributed along three major differentiation lineages (myeloid: HSC→MPP→(CMP or LMPP)→GMP→Mono; erythroid:

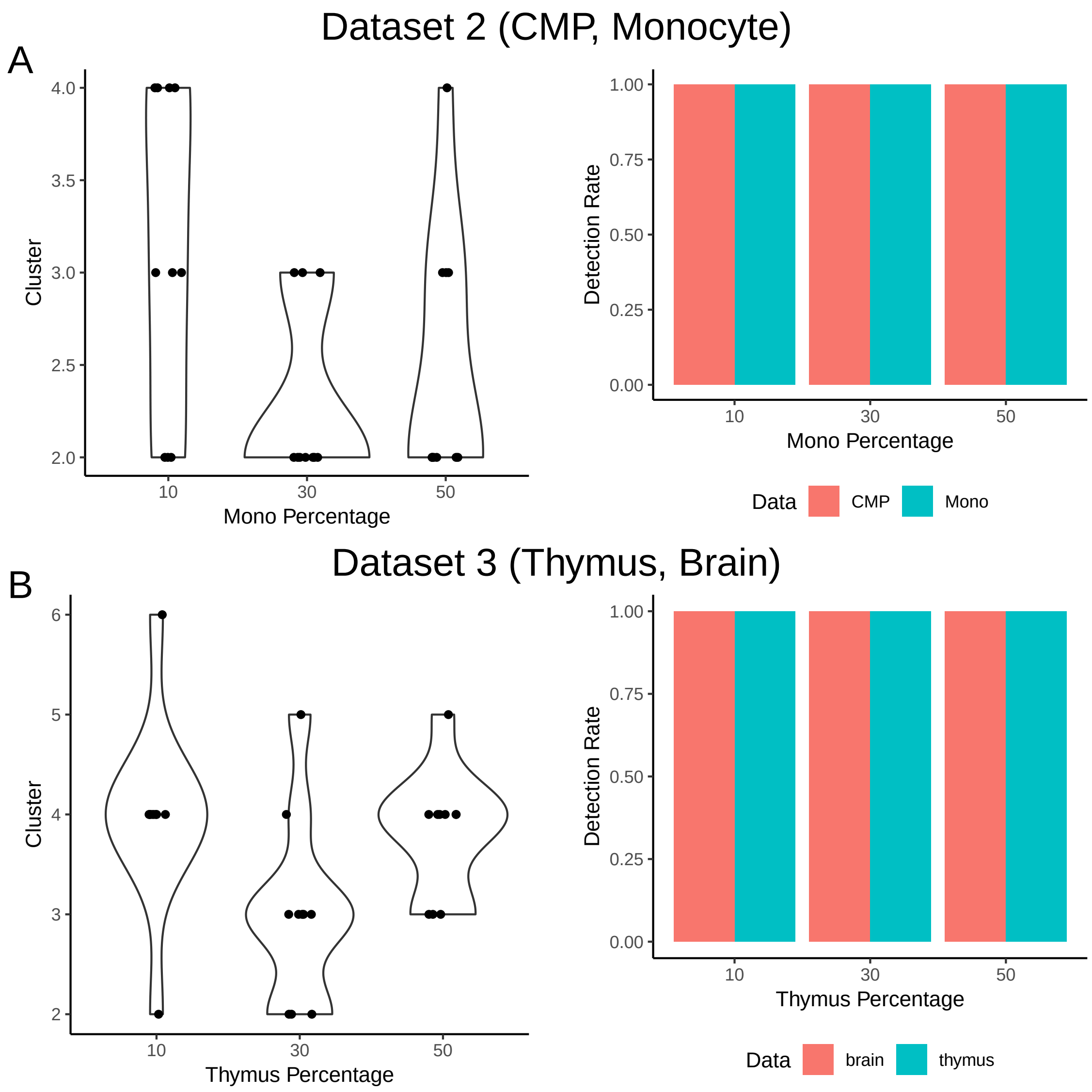


Figure 3-13. Analyses of a heterogeneous cell population created using (A) Dataset 2 and (B) Dataset 3. In each dataset, the left plot shows distribution of cell cluster numbers obtained by SCATE for synthetic samples with different cell mixing proportions. For each mixing proportion, 10 synthetic samples were created and analyzed. The right plot shows the frequency that each cell type is detected in the 10 synthetic samples at each cell mixing proportion.

HSC→MPP→CMP→MEP; lymphoid: HSC→MPP→LMPP→CLP), which are consistent with known biology. For method evaluation, we analyzed all cells together as a heterogeneous cell population and pretended that the cell type labels were unknown. We also downloaded and processed bulk ATAC-seq data for these 8 cell types from

[90] and used them as the gold standard to assess regulome reconstruction accuracy.

Using its default cell clustering method, SCATE identified 14 cell clusters. To evaluate the performance of this unsupervised analysis for recovering true biology, we first assigned a cell type label for each cluster. A cluster was annotated as "predicted cell type X" if the cluster contained at least two cells and the true cell type label of $\geq$70% cells from the cluster was cell type X. Clusters that cannot be annotated using this criterion were labeled as ambiguous. In this way, we were able to unambiguously annotate 9 clusters. Since multiple clusters may be annotated with the same cell type, these 9 annotated clusters corresponded to a total of 6 cell types (Figure 3-14B). For these 9 clusters, one can evaluate signal reconstruction accuracy because the bulk ATAC-seq data for the annotated cell type was available. Each cluster was treated as a homogeneous cell population by SCATE and other methods in our analysis (as one would do in real applications), even though the cluster actually may not be pure and may contain cells from more than one cell types. Figure 3-14D compares the Pearson correlation between the gold standard bulk signal and the CRE activities reconstructed from scATAC-seq by different methods. Each boxplot contains 9 data points corresponding to the 9 cell clusters. Figure 3-14E compares the peak calling performance (AUC under the sensitivity-FDR curve). Figure 3-14F compares the accuracy for predicting differential CRE activities between different cell types. Here each data point in the boxplot is a pair of cell clusters annotated with two different cell types. The Pearson correlation between the gold standard bulk differential signal and differential signal reconstructed from scATAC-seq was computed and compared. In all these analyses, SCATE outperformed the other methods. Figure 3-14J shows an example genomic region in a HSC cell cluster. SCATE most accurately reconstructed the bulk ATAC-seq signal in HSC.

SCATE provides users with the flexibility to specify their own cell cluster number or use their own cell clustering results. The software can reconstruct signals based on

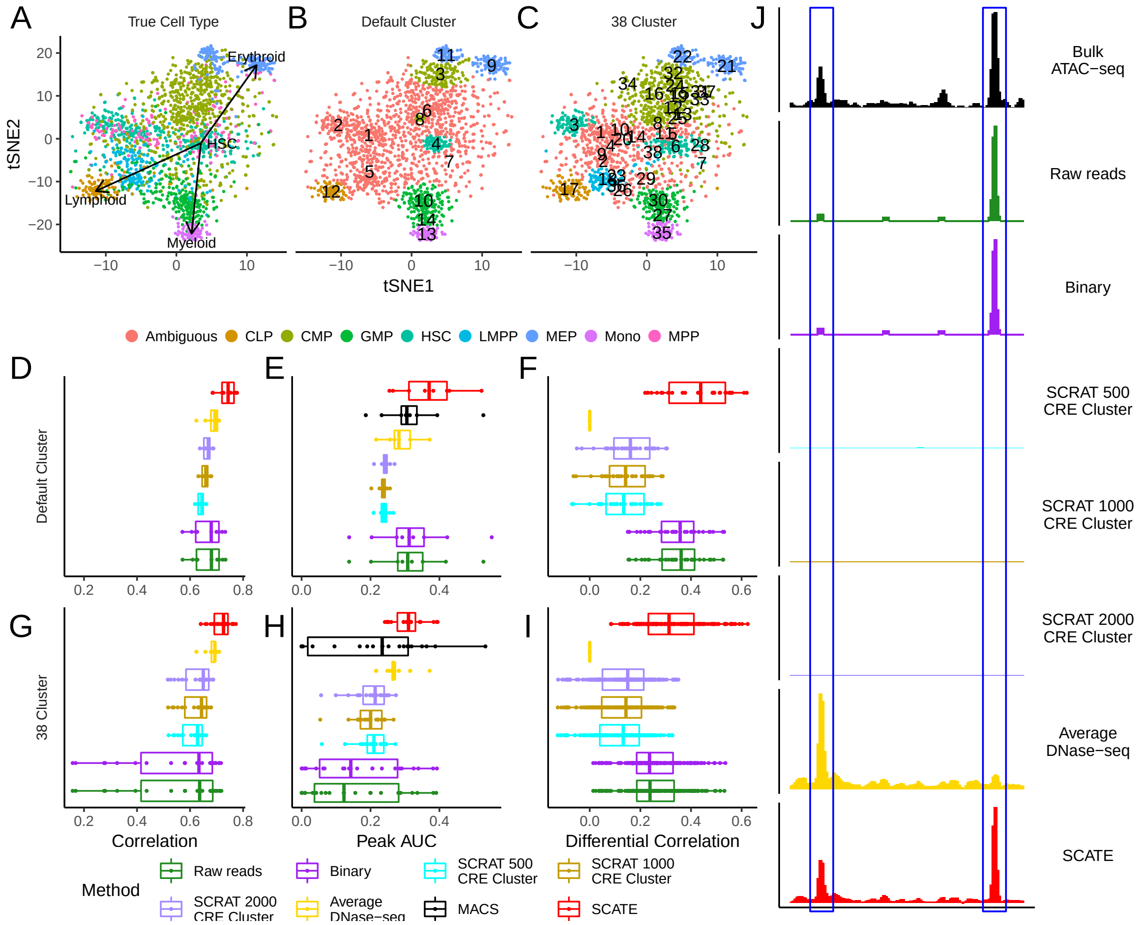


Figure 3-14. Analysis of human hematopoietic differentiation cell types. (A): tSNE plot showing cells color-coded by their true cell types. (B): tSNE plot showing cells color-coded by their predicted cell types. Using the default setting, SCATE grouped cells into 14 clusters (numbers in the plot indicate cluster centers). The clusters that can be unambiguously linked to a cell type are color-coded by cell type. (C): Similar to (B), but cells are clustered using user-specified cluster number (38 clusters). (D)-(F): Regulome reconstruction performance of different methods in the default analysis, including (D) correlation between reconstructed and true bulk log-CRE activities, (E) peak calling AUC, and (F) correlation between predicted and true differential log-CRE activities. (G)-(I): Regulome reconstruction performance using user-specified cluster number (38 clusters), including (G) correlation between reconstructed and true bulk log-CRE activities, (H) peak calling AUC, and (I) correlation between predicted and true differential log-CRE activities. (J): Comparison of different methods in an example genomic region in HSC cell cluster in the default analysis.

user-provided cell cluster number or clustering structure. For instance, suppose one is not satisfied with the default cell clustering and wants to increase the granularity of clustering to make each cluster smaller and more homogeneous, one can manually adjust the cluster number. To demonstrate, we increased the cluster number to 38 so that each cluster had approximately 50 cells on average. After rerunning SCATE, 24 of the 38 cell clusters can be unambiguously annotated, identifying a total of 7 cell types (Figure 3-14C). As a comparison, the default analysis only unambiguously identified 6 cell types. For the unambiguously annotated cell clusters, Figure 3-14G-I compares the performance of different methods for reconstructing CRE activities, peak calling, and estimating differential CRE activities between different cell types. SCATE still delivered the best performance. Since the average cell cluster size became smaller, the performance of some methods decreased substantially in some analyses (e.g., the CRE reconstruction and peak calling accuracy for Raw reads and Binary in Figure 3-14G,H). In these cases, the benefit from SCATE was even more obvious.

3.4 Discussion

In summary, SCATE provides a new tool for analyzing scATAC-seq data. Our analyses show that it robustly outperforms the existing methods for reconstructing activities of each individual CRE. In many cases, the gain can be substantial.

The main novelty of SCATE is its unique strategy to reconstruct CRE activities from sparse data by (1) integrating data from both similar CREs and cells, (2) leveraging the rich information provided by publicly available regulome data, and (3) adaptively optimizing the analysis resolution based on available data. Coupled with appropriate cell clustering, SCATE allows one to systematically characterize the regulatory landscape of a heterogeneous sample via unsupervised identification of cell subpopulations and reconstruction of their chromatin accessibility profile at the single CRE resolution.

Since many methods for clustering cells using scATAC-seq data have been developed

(Table 3-I), cell clustering *per se* is not the focus of this article. In principle, the SCATE model may be coupled with any cell clustering method. While our implementation uses model-based clustering as the default, users are provided with the option to use their own cell clustering results as the input for SCATE.

The basic framework adopted by SCATE to improve the analysis of sparse data by integrating multiple sources of information is general. In principle, a similar approach may also be used to analyze other types of single-cell epigenomic data such as single-cell DNase-seq or ChIP-seq, and possibly single-cell Hi-C [93].

Our current implementation of SCATE is focused on identifying and characterizing cell subpopulations. A future direction is to extend this framework to other types of analyses such as pseudotime analysis [39] to allow the study of CRE activities along continuous pseudotemporal trajectories. Another future direction is to develop new methods that utilize the improved CRE estimation to more accurately reconstruct gene regulatory networks.

Chapter 4

RAISIN: Regression Analysis in Single-cell RNA-Seq with multiple samples

4.1 Introduction

Transcriptome profiling by single-cell RNA-sequencing (scRNA-seq) [20, 45] is rapidly transforming biomedical research. The ability of scRNA-seq to analyze individual cells enables systematic discovery and characterization of known and unknown cell populations in a biological sample. Identifying differentially expressed genes associated with various biological or technical factors such as cell type or experimental condition is one of the most common tasks for analyzing scRNA-seq data [94, 95]. While many early studies only analyze cells from one sample, recent studies increasingly analyze multiple samples such as multiple biological replicates in order to make discoveries generalizable to the population [28, 96]. For analyzing data with multiple samples, it is important to consider both cell-to-cell variation and sample-to-sample variation in order to distinguish true biological signals from noises. However, the most commonly used differential expression (DE) analysis methods either ignore sample-level variation [42] or do not consider cell-level variation [97]. Applying them to multi-sample data will produce unsatisfactory or misleading results.

To solve this problem, we developed RAISIN to support Regression Analysis In SINgle-cell RNA-seq datasets with multiple samples. RAISIN takes raw gene expression counts and experimental design as input and provides a complete preprocessing pipeline consisting of cell and gene filtering, normalization and gene expression imputation. It then aligns cells of the same type across samples and identify cell subpopulations through clustering. DE analysis is then performed using a flexible mixed effects regression framework that accounts for both sample-level and cell-level variances (Figure 4-1A). The classical linear mixed effects model (LMM) [98] does not consider small sample size or small cell number in rare cell populations, which are common in scRNA-seq studies and can lead to poor variance estimation and reduced statistical power. Fitting mixed effects models to large datasets consisting of many samples and millions of cells is also computationally challenging. To address these issues, RAISIN combines the mixed model with a hierarchical model to regularize variances, and a new model fitting algorithm is developed to efficiently handle large datasets.

4.2 Methods

4.2.1 RAISIN overview

Given scRNA-seq data from multiple samples, a basic RAISIN analysis consists of data preprocessing and differential expression detection. The data preprocessing includes cell and gene filtering, normalization, imputation, aligning cells across samples, and clustering cells to identify cell subpopulations. The differential expression detection analyzes each cell subpopulation or compares different cell subpopulations to identify gene expression associated with user-specified biological or experimental variables (e.g., normal vs. disease, age, sex, etc.).

4.2.2 RAISIN data preprocessing

The data preprocessing of RAISIN is a multi-step procedure. It is implemented in a modular fashion so that users can conveniently replace each step by their own functions or new methods. The default preprocessing pipeline is described below. Users have options to change the parameter values.

Cell and gene filtering. By default, cells with less than 5,000 reads are removed. We also remove cells with more than 50,000 reads because an extremely large total read count may indicate a doublet rather than a single cell. Since high mitochondrial gene expression is often associated with low sample quality, cells in which mitochondrial gene reads account for more than 10% of all reads are also filtered out. For gene filtering, we retain genes that have non-zero read count in at least 1% of cells in at least one sample and remove the other genes.

Normalization. The raw read counts are normalized across cells using the cell size factors estimated by SCRAN [99] (using R scran package) which is run across all cells and samples.

Imputation. SAVER [100] is run on SCRAN normalized data in each sample to impute dropouts and quantify gene expression values. The output of this step is log2-transformed gene expression. A pseudocount of 1 is added before log-transformation to avoid log-zero.

Aligning cells across samples. In order to track cells of the same cell type across samples, the Mutual Nearest Neighbors (MNN) [101] approach is used to align cells from different samples. To this end, we first identify genes with (1) expression≥0.1 in at least 1% of all cells across all samples, and (2) positive biological variation (higher variation than expected controlling for mean expression) as determined by the decomposeVar function in scran package. The fastMNN function in scran package is then run using these genes and default settings. This function maps all cells to a common principal component (PC) space and corrects cells' positions in this space to removes systematic differences among samples. In this way, cells of the same type but from different samples are aligned together. By default, MNN generates 50 PCs. A subset of 50 PCs are further chosen for follow-up analysis. The optimal number of dimensions to use is chosen using an elbow method same as in TSCAN [43]. Cells' MNN-corrected coordinates in top L PCs are retained. The optimal L is determined using the piece-wise linear elbow method described in TSCAN [43] and is truncated at 50 (i.e., $L \leq 50$).

Clustering cells to identify cell subpopulations. Using the MNN-corrected coordinates in the top L PCs, cells are clustered using K-means clustering. Users can either specify the cluster number by themselves or let RAISIN to automatically choose the cluster number. To choose the cluster number automatically, K-means clustering is first run using an relatively large initial cluster number K_0 (the default $K_0 = 100$). The K_0 initial clusters are then clustered further using hierarchical clustering and merged along the dendrogram to obtain $k = K_0 - 1$, $K_0 - 2$, ..., 2 clusters. For each cluster number k, we calculate the ratio between the within-cluster sum of squared residuals (RSS) and total data variance (= within-cluster RSS + between-cluster RSS). This ratio, denoted as r_k, characterizes the proportion data variance that cannot be explained by clustering. It decreases with increasing cluster number k. We calculate the difference $r_{k-1} - r_k$ for $k = 2, 3, ..., K_0$. These differences are log10 transformed and grouped into histogram bins. Denote the lower bound of the bin with the largest number of elements as c. The smallest k that satisfies $r_{k-1} - r_k \leq 10^c$ is chosen as the cluster number.

Visualization. To visualize cell clustering, UMAP (umap package in R) is applied to cells' MNN-corrected coordinates in the top L PCs. UMAP is run with its default settings which reduce cells' dimension from L to 2.

4.2.3 RAISIN differential expression (DE) analysis

RAISIN uses a mixed effects regression model with variance shrinkage to detect differential expression. In order to introduce the method, first consider a simple scenario of comparing two sample types (e.g., cancer vs. normal). For such a comparison, RAISIN will analyze each cell subpopulation separately. For a given cell subpopulation, let y_{gsc} be the gene expression value of gene g in sample s and cell c, and let $\mathbf{y}_g$ be the column vector consisting of y_{gsc}s from all samples and cells in the cell subpopulation. Here y_{gsc}s are normalized gene expression values after imputation but without MNN correction because biological differences between different sample types (e.g., normal vs. disease) would be removed from the MNN-corrected expression values. Thus, MNN is only used to align samples to identify cells of the same type across samples.

A conventional linear mixed model (LMM) assumes that

$$\mathbf{y}_g = \mathbf{X}\boldsymbol{\beta}_g + \mathbf{Z}\mathbf{u}_g + \mathbf{e}_g \tag{4.1}$$

Here $\mathbf{X}\boldsymbol{\beta}_g$ models fixed effects, $\mathbf{Z}\mathbf{u}_g$ models the sample-level random effects, and $\mathbf{e}_g$ models the cell-level random effects. The matrices $\mathbf{X}$ and $\mathbf{Z}$ are known experiment design information. $\boldsymbol{\beta}_g$ contains unknown regression coefficients of interest. The random effects $\mathbf{u}_g$ and $\mathbf{e}_g$ are unobserved random vectors with zero mean. Their variances $var(\mathbf{u}_g) = \boldsymbol{\Sigma}_g$ and $var(\mathbf{e}_g) = \boldsymbol{\Omega}_g$ characterize cross-sample variability and cross-cell variability, respectively. Both $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Omega}_g$ are unknown. In this study, they are assumed to be diagonal matrices with block structures such that diagonal elements within the same block are equal but those from different blocks can have different values. For $\boldsymbol{\Omega}_g$, cells from the same sample are treated as a block. For $\boldsymbol{\Sigma}_g$, the block structure is given by users. For example, if samples are from multiple groups (e.g., normal vs. disease), one can treat each group as a block. Under this framework, differential expression is detected by evaluating linear combinations of regression coefficients $\boldsymbol{\beta}_g$.

Let $\mathbf{x}_{sc}^T$ and $\mathbf{z}_{sc}^T$ denote the row corresponding to sample s and cell c in $\mathbf{X}$ and $\mathbf{Z}$ respectively. The model can also be written as

$$y_{gsc} = \mathbf{x}_{sc}^T\boldsymbol{\beta}_g + \mathbf{z}_{sc}^T\mathbf{u}_g + e_{gsc} \tag{4.2}$$

For instance, suppose one compares two normal control samples ($s = 1, 2$) with two tumor samples ($s = 3, 4$), and each sample has two cells ($c = 1, 2$) in the cell subpopulation in question. The model can be written as

$$\begin{bmatrix} y_{g11} \\ y_{g12} \\ y_{g21} \\ y_{g22} \\ y_{g31} \\ y_{g32} \\ y_{g41} \\ y_{g42} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_{g0} \\ \beta_{g1} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{g1} \\ u_{g2} \\ u_{g3} \\ u_{g4} \end{bmatrix} + \begin{bmatrix} e_{g11} \\ e_{g12} \\ e_{g21} \\ e_{g22} \\ e_{g31} \\ e_{g32} \\ e_{g41} \\ e_{g42} \end{bmatrix} \tag{4.3}$$

or

$$y_{gsc} = \beta_{g0} + x_{sc}\beta_{g1} + u_{gs} + e_{gsc} \tag{4.4}$$

where the dummy variable x_{sc} indicates whether a cell comes from a normal sample ($x_{sc} = 0$) or a tumor sample ($x_{sc} = 1$). u_{gs} and e_{gsc} are independent sample-level and cell-level random effects respectively. Finding differential expression between tumor and normal amounts to evaluating whether β_{g1} is equal to zero or not.

In this example, one can assume $var(u_{g1}) = var(u_{g2}) = \sigma_{g1}^2$ and $var(u_{g3}) = var(u_{g4}) = \sigma_{g2}^2$ (i.e., samples of the same type have the same variance), which implies that $\boldsymbol{\Sigma}_g = diag\left\{\sigma_{g1}^2, \sigma_{g2}^2\right\} \otimes \mathbf{I}_{2\times 2} = diag\left\{\sigma_{g1}^2, \sigma_{g1}^2, \sigma_{g2}^2, \sigma_{g2}^2\right\}$. In other words, $\boldsymbol{\Sigma}_g$ is a diagonal matrix with two blocks. Here $\otimes$ denotes Kronecker product and $\mathbf{I}$ denotes an identity matrix. Similarly, one can assume $var(e_{gsc}) = \omega_{gs}^2$ (i.e., cells in the same sample have the same variance conditional on their sample-level mean $\mathbf{x}_{sc}^T\boldsymbol{\beta}_g + \mathbf{z}_{sc}^T\mathbf{u}_g$) and thus $\boldsymbol{\Omega}_g = diag\left\{\omega_{g1}^2, \omega_{g2}^2, \omega_{g3}^2, \omega_{g4}^2\right\} \otimes \mathbf{I}_{2\times 2}$ is a diagonal matrix with four blocks.

In the LMM, the marginal variance of $\mathbf{y}_g$ is $\mathbf{Z}\boldsymbol{\Sigma}_g\mathbf{Z}^T + \boldsymbol{\Omega}_g$, which is no longer a diagonal matrix. Thus, the model can deal with correlation among cells from the same sample. By contrast, the Wilcoxon test used by Seurat, MAST, scDD, and t test used in our benchmark analysis do not consider sample-level variation. This is similar to removing the $\mathbf{Zu}$ component from the LMM model and treating all cells as independent samples for testing differential expression. Since the actual number of independent samples (i.e. effective sample size) is much smaller than the cell number, these methods will underestimate the uncertainty of $\boldsymbol{\beta}_g$ estimates and report overly optimistic p-values and false discovery rates (i.e., the actual error rates can be much higher than the reported error rates). By considering correlation among cells, LMM improves the characterization of the uncertainty of $\boldsymbol{\beta}_g$ estimates and hence can better control the false discovery rates.

The conventional LMM has several limitations. First, it treats $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Omega}_g$ as fixed unknown parameters. When the number of samples or the number of cells in a cell subpopulation is small, the estimates of $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Omega}_g$ have high variability and hence are highly unstable, leading to reduced statistical power. Second, fitting LMM often requires iterative algorithms since closed-form solutions are unavailable except for a few special cases. When the cell number or sample size is large, fitting the model for tens of thousands of genes using the conventional algorithms is computationally intensive. For cell atlases with millions of cells, model fitting can be very slow.

To overcome these limitations, RAISIN extends LMM using an empirical Bayes framework which introduces a number of new components.

First, we reformulate the LMM using cells' average gene expression in each cell subpopulation and sample. For the given cell subpopulation, let n_s be the cell number of the subpopulation in sample s, and $\widetilde{y}_{gs} = \sum_c y_{gsc}/n_s$ be the average expression of cells in the subpopulation in sample s. Let S be the total number of samples. The

LMM is rewritten as

$$\widetilde{\mathbf{y}}_g = \widetilde{\mathbf{X}}\boldsymbol{\beta}_g + \widetilde{\mathbf{Z}}\mathbf{u}_g + \widetilde{\mathbf{e}}_g \tag{4.5}$$

or

$$\widetilde{y}_{gs} = \widetilde{\mathbf{x}}_s^T\boldsymbol{\beta}_g + \widetilde{\mathbf{z}}_s^T\mathbf{u}_g + \widetilde{e}_{gs} \tag{4.6}$$

For instance, the model for the example considered above will become

$$\begin{bmatrix} \widetilde{y}_{g1} \\ \widetilde{y}_{g2} \\ \widetilde{y}_{g3} \\ \widetilde{y}_{g4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_{g0} \\ \beta_{g1} \end{bmatrix} + \begin{bmatrix} u_{g1} \\ u_{g2} \\ u_{g3} \\ u_{g4} \end{bmatrix} + \begin{bmatrix} \widetilde{e}_{g1} \\ \widetilde{e}_{g2} \\ \widetilde{e}_{g3} \\ \widetilde{e}_{g4} \end{bmatrix} \tag{4.7}$$

or

$$\widetilde{y}_{gs} = \beta_{g0} + \widetilde{x}_s\beta_{g1} + u_{gs} + \widetilde{e}_{gs} \tag{4.8}$$

where $var(\mathbf{u}_g) = \boldsymbol{\Sigma}_g = diag\left\{\sigma_{g1}^2, \sigma_{g1}^2, \sigma_{g2}^2, \sigma_{g2}^2\right\}$ and $var(\widetilde{\mathbf{e}}_g) = \widetilde{\boldsymbol{\Omega}}_g = diag\left\{\frac{\omega_{g1}^2}{n_1}, \ldots, \frac{\omega_{g4}^2}{n_4}\right\}$. This model reformulation can substantially reduce the dimension of $\mathbf{y}$ from $\sum_{s=1}^S n_s$ (which can be millions of cells) to S (e.g., a few dozens of samples) and hence simplify the computation.

Second, in order to deal with unstable variance estimates in small cell number scenarios (e.g., in a rare cell subpopulation), we assume that parameters in $\widetilde{\boldsymbol{\Omega}}_g$ from different genes are random variables and their prior distributions are shared across genes. This allows one to derive shrinkage estimators to improve variance estimation by borrowing information across genes. Specifically, let s denote samples and $\left\{\omega_{gs}^2 : s = 1, \ldots, S\right\}$ be the set of unique variance parameters in $\widetilde{\boldsymbol{\Omega}}_g$. We assume

$$\widetilde{e}_{gs} \sim N(0, \frac{\omega_{gs}^2}{n_s}) \tag{4.9}$$

$$\omega_{gs}^2 \sim IG(\theta_s, \phi_s) \tag{4.10}$$

where $N(.,.)$ represents normal distribution, and $IG(.,.)$ represents inverse-gamma distribution whose parameters θ_s and ϕ_s are shared by all genes. We estimate θ_s and

ϕ_s using data from all genes via moment estimators similar to limma. An empirical Bayes shrinkage estimator is then used to estimate ω_{gs}^2 for each individual gene by its posterior mean.

Third, in order to deal with unstable variance estimates in small sample size scenarios, we assume that parameters in $\mathbf{\Sigma}_g$ are random variables whose prior distributions do not depend on specific genes. Assume $\mathbf{\Sigma}_g$ has L blocks and use l to index the block. Let $\left\{\sigma_{gl}^2 : l = 1, \ldots, L\right\}$ be the set of unique variance parameters in $\mathbf{\Sigma}_g$. For a sample s that belongs to variance block l, we assume

$$u_{gs} \sim N(0, \sigma_{gl}^2) \tag{4.11}$$

$$\sigma_{gl}^2 \sim Gamma(\alpha_l, \gamma_l) \tag{4.12}$$

The parameters α_l and γ_l in the prior distribution are gene-independent and are estimated using all genes via moment estimation. σ_{gl}^2 is then estimated using its posterior mean. Due to multi-level variance modeling, the posterior mean of σ_{gl}^2 does not have a closed-form. Thus, Gauss-Laguerre quadrature is used to obtain a numerical approximation. In theory, one could also use inverse-gamma distribution as the prior for σ_{gl}^2. However, computing Gauss-Laguerre quadrature under the inverse-gamma assumption empirically is unstable numerically. Thus, gamma distribution is used instead since it makes the computation numerically stable.

Fourth, suppose the goal is to evaluate whether a linear combination of regression coefficients $\boldsymbol{a}^T\boldsymbol{\beta}_g$ is equal to zero. We assume that a priori each gene has probability p to be non-differential ($H_0 : \boldsymbol{a}^T\boldsymbol{\beta}_g = 0$) and probability $1 - p$ to be differential ($H_1 : \boldsymbol{a}^T\boldsymbol{\beta}_g \neq 0$). When a gene is differential, assume $\boldsymbol{a}^T\hat{\boldsymbol{\beta}}_g/\sqrt{var(\boldsymbol{a}^T\hat{\boldsymbol{\beta}}_g)} \sim N(0, 1 + \tau^2)$. We use an Expectation-Maximization algorithm to estimate p and τ^2. The posterior probability for H_1 is used to detect and rank DE genes. Treating the posterior probability of H_0 as a local false discovery rate, a global FDR can also be calculated as in [102] to compare with other methods.

Note that besides the single-cell methods discussed before (Seurat, MAST, scDD, t test), another existing approach to run DE analysis is to pool cells in each cell subpopulation and then analyze cells' average expression as bulk samples using existing bulk DE methods such as limma, DESeq2, and edgeR. This approach ignores the cell-level variability, which is similar to removing the $\widetilde{\mathbf{e}}$ component from the LMM. When the cell-level variability is comparable to the sample-level variability, ignoring cell-level variability will substantially underestimate the uncertainty of $\boldsymbol{\beta}_g$ estimates, which can lead to incorrect error rate estimates and reduced statistical power. Note also that variance shrinkage has been used in the past in linear model (LM) settings (e.g., limma, DESeq2 and edgeR). However, the LM only requires one level of variance modeling, whereas the LMM requires multi-level variance modeling. Imposing prior distributions on both the sample-level and cell-level variances makes the model fitting complicated. It is difficult to directly apply algorithms in limma, DESeq2 and edgeR. One solution to fitting the model is to use the fully Bayesian approach and run Markov Chain Monte Carlo. However, this approach is slow and not scalable to large datasets. In order to make the model fitting scalable, we developed a computationally efficient multi-step fitting algorithm that sequentially estimates $\widetilde{\boldsymbol{\Omega}}_g$, $\widetilde{\boldsymbol{\Sigma}}_g$, $\boldsymbol{\beta}$, p, τ^2, and the posterior probability of DE.

The regression framework adopted by RAISIN is flexible. Besides comparing two groups of samples, it can also be used to analyze the association between gene expression and any other categorical or continuous variables. One can also add covariates to the model to adjust for potential confounding. For instance, one can formulate a model to identify DE associated with age after accounting for experimental batches:

$$\widetilde{y}_{gs} = \beta_{g0} + \beta_{g1} \times age_s + \beta_{g2} \times batch_s + u_{gs} + \widetilde{e}_{gs} \tag{4.13}$$

Here, the design matrix $\widetilde{\mathbf{X}}$ will contain 1 (for intercept), *age* and *batch*. The DE will be detected by evaluating β_{g1}.

RAISIN can be further generalized to allow L groups of arbitrary random factors:

$$\widetilde{\mathbf{y}}_g = \widetilde{\mathbf{X}}\boldsymbol{\beta}_g + \left[\begin{array}{cccc} \widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2 & ... & \widetilde{\mathbf{Z}}_L \end{array}\right] \left[\begin{array}{c} \mathbf{u}_{g1} \\ \mathbf{u}_{g2} \\ ... \\ \mathbf{u}_{gL} \end{array}\right] + \widetilde{\mathbf{e}}_g \tag{4.14}$$

$$var([\mathbf{u}_{g1}, \mathbf{u}_{g2}, ..., \mathbf{u}_{gL}]) = diag(\sigma^2_{g1}, ..., \sigma^2_{g1}, \sigma^2_{g2}, ..., \sigma^2_{g2}, ..., \sigma^2_{gL}, ..., \sigma^2_{gL})$$

This formulation allows more flexible types of differential analysis. Below gives an example of identifying differential genes between two cell subpopulations k_1 and k_2. Let $y_{gsc,k}$ denote gene expression for gene g, sample s, and cell c in cell subpopulation k. Let $\widetilde{y}_{gs,k}$ be gene g's average expression in sample s across cells in cell subpopulation k. For instance, suppose there are four samples and no other covariates to adjust for, the model would be

$$\left[\begin{array}{c} \widetilde{y}_{g1,k_1} \\ \widetilde{y}_{g2,k_1} \\ \widetilde{y}_{g3,k_1} \\ \widetilde{y}_{g1,k_2} \\ \widetilde{y}_{g2,k_2} \\ \widetilde{y}_{g4,k_2} \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{array}\right] \left[\begin{array}{c} \beta_{g0} \\ \beta_{g1} \end{array}\right] + \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} u_{g11} \\ u_{g12} \\ u_{g13} \\ u_{g14} \\ u_{g21} \\ u_{g22} \end{array}\right] + \left[\begin{array}{c} \widetilde{e}_{g1,k_1} \\ \widetilde{e}_{g2,k_1} \\ \widetilde{e}_{g3,k_1} \\ \widetilde{e}_{g1,k_2} \\ \widetilde{e}_{g2,k_2} \\ \widetilde{e}_{g4,k_2} \end{array}\right] \tag{4.15}$$

where $var(\mathbf{u}_g) = \boldsymbol{\Sigma}_g = diag\left\{\sigma^2_{g1}, \sigma^2_{g1}, \sigma^2_{g1}, \sigma^2_{g1}, \sigma^2_{g2}, \sigma^2_{g2}\right\}$

$var(\widetilde{\mathbf{e}}_g) = \widetilde{\boldsymbol{\Omega}}_g = diag\left\{\frac{\omega^2_{g1,k_1}}{n_{1,k_1}}, \ldots\right\}$.

Methods to estimate $\omega^2_{gs,k}$ and σ^2_{gi} can be found below. When there are covariates, they can be added to the design matrix $\widetilde{\mathbf{X}}$.

4.2.4 Estimating θ_s,ϕ_s and ω^2_{gs}

Denote v_{gs} as the sample variance across all cells for sample s and gene g.

Let $d_s = n_s - 1$, we have $v_{gs}|\omega^2_{gs} \sim \frac{\omega^2_{gs}}{d_s}\chi^2_{d_s}$

By assumption, $\omega^2_{gs} \sim IG(\theta_s, \phi_s)$

$$
\begin{aligned}
p(v_{gs}) &= \int p(v_{gs}|\omega_{gs}^2)p(\omega_{gs}^2)\,\mathrm{d}\omega_{gs}^2 \\
&= \frac{1}{v_{gs}B(d_s/2,\theta_s)}\sqrt{\frac{(2\theta_s)^{2\theta_s}(\frac{\theta_s}{\phi_s}d_s v_{gs})^{d_s}}{(\frac{\theta_s}{\phi_s}d_s v_{gs}+2\theta_s)^{2\theta_s+d_s}}}
\end{aligned}
$$

where $B(.,.)$ denotes beta distribution.

Thus, $\frac{\theta_s}{\phi_s}v_{gs} \sim F(d_s, 2\theta_s)$ and $\frac{1}{2}log(\frac{\theta_s}{\phi_s}v_{gs}) \sim z(d_s, 2\theta_s)$

where $z(.,.)$ denotes Fisher's z-distribution.

$E[log(v_{gs})] = \log(2\phi_s/d_s) + \psi(d_s/2) - \psi(\theta_s)$

$\mathrm{var}[log(v_{gs})] = \psi'(d_s/2) + \psi'(\theta_s)$

where $\psi(.)$ and $\psi'(.)$ are the digamma and trigamma functions respectively.

If $\mathrm{var}[log(v_{gs})] - \psi'(d_s/2) > 0$

$\theta_s = \psi'^{-1}(\mathrm{var}[log(v_{gs})] - \psi'(d_s/2))$

$\phi_s = exp(E[log(v_{gs})] - \psi(d_s/2) + \psi(\theta_s)) * d_s/2$

Since $\omega_{gs}^2|v_{gs} \sim IG(\theta_s + d_s/2, d_s v_{gs}/2 + \phi_s)$

Let $\alpha_s = \theta_s + d_s/2$,$\beta = d_s v_{gs}/2 + \phi_s$

$E[\omega_{gs}^2|v_{gs}] = \int_0^\infty \frac{(d_s v_{gs}/2+\phi_s)^{\theta_s+d_s/2}}{\Gamma(\theta_s+d_s/2)} x^{-(\theta_s+d_s/2)} exp(-\frac{d_s v_{gs}/2+\phi_s}{x})\,\mathrm{d}x$

If $\theta + d/2 > 1$, $E[\omega_{gs}^2|v_{gs}] = \frac{d_s v_{gs}/2+\phi_s}{\theta_s+d_s/2-1}$. Otherwise $E[\omega_{gs}^2|v_{gs}]$ is derived using Gauss–Laguerre quadrature.

If $\mathrm{var}[log(v_{gs})] - \psi'(d_s/2) \leq 0$

$E[\omega_{gs}^2|v_{gs}] = exp(\bar{e})$, where $e = \log(v_{gs})$

$E[\omega_{gs}^2|v_{gs}]$ serves as the estimate of ω_{gs}^2.

For situations where $n_s = 1$, $E[\omega_{gs}^2|v_{gs}] = E[\omega_{gs'}^2|v_{gs'}]$ s.t. $s' = argmin_{j:n_j>1}(\|\beta_s - \beta_j\|_2)$

4.2.5 Estimating α_l,γ_l and σ_{gl}^2

Suppose there are L groups of random effects.

α_l, γ_l, σ^2_{gl} are estimated iteratively for each group l in the order of o_l (i.e. group l with smallest o_l is estimated first and group l with largest o_l is estimated last). Here $o_l = \sum_{i,j} I(\mathbf{Z}_l[i,j] > 0)$.

To perform the estimation for group l, suppose groups in set $\mathbf{A}_l$ are already estimated, and groups in set $\mathbf{B}_l$ are not estimated yet ($l \notin \mathbf{B}_l$).

Let $\mathbf{C}_l = \{c | \exists\ i \in \{l, \mathbf{B}_l\}\ s.t.\ \mathbf{Z}_i[c,.] \neq \mathbf{0}\}$

Let $\widetilde{\mathbf{X}}_l = \widetilde{\mathbf{X}}[\mathbf{C}_l,.]$

Let $\widetilde{\mathbf{Z}}_{i,l} = \widetilde{\mathbf{Z}_i}[\mathbf{C}_l,.]$ for $i = 1, 2, ..., L$

Let $\tilde{\mathbf{\Sigma}}_{gl} = \tilde{\mathbf{\Sigma}}_g[\mathbf{C}_l, \mathbf{C}_l]$

Let $\tilde{\mathbf{\Omega}}_{gl} = \tilde{\mathbf{\Omega}}_g[\mathbf{C}_l, \mathbf{C}_l]$

Let $\tilde{\mathbf{y}}_{gl} = \tilde{\mathbf{y}}_g[\mathbf{C}_l]$

Let $\tilde{\mathbf{e}}_{gl} = \tilde{\mathbf{e}}_g[\mathbf{C}_l]$

Find $\mathbf{K}_l$ such that $\mathbf{K}_l\widetilde{\mathbf{X}}_l = 0$, $\mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l} = 0$ for $i \in \mathbf{B}_l$, and $\mathbf{K}_l\mathbf{K}_l^T = I$.

$\mathbf{K}_l\tilde{\mathbf{y}}_{gl} = \mathbf{K}_l\widetilde{\mathbf{X}}_l\boldsymbol{\beta}_g + \sum_{i\in\{\mathbf{A}_l,\mathbf{B}_l,l\}} \mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l}\tilde{\mathbf{u}}_{gi} + \mathbf{K}_l\tilde{\mathbf{e}}_{gl} = \sum_{i\in\mathbf{A}_l} \mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l}\tilde{\mathbf{u}}_{gi} + \mathbf{K}_l\widetilde{\mathbf{Z}}_{l,l}\tilde{\mathbf{u}}_{gl} + \mathbf{K}_l\tilde{\mathbf{e}}_{gl}$

$\mathbf{K}_l\tilde{\mathbf{y}}_{gl} | \tilde{\mathbf{\Sigma}}_{gl}, \tilde{\mathbf{\Omega}}_{gl} \sim N(\mathbf{0}, \sigma^2_{gl}\mathbf{K}_l\widetilde{\mathbf{Z}}_{l,l}(\mathbf{K}_l\widetilde{\mathbf{Z}}_{l,l})^T + \sum_{i\in\mathbf{A}_l} \sigma^2_{gi}\mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l}(\mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l})^T + \mathbf{K}_l\tilde{\mathbf{\Omega}}_{gl}\mathbf{K}_l^T)$

Note that here σ^2_{gi} are already estimated for $i \in \mathbf{A}_l$

Let p_{igl} be the ith element of the vector $\mathbf{K}_l\tilde{\mathbf{y}}_{gl}$

Let q_{il} be the ith diagonal element of the matrix $\mathbf{K}_l\widetilde{\mathbf{Z}}_{l,l}(\mathbf{K}_l\widetilde{\mathbf{Z}}_{l,l})^T$

Let r_{igl} be the ith diagonal element of the matrix $(\sum_{i\in\mathbf{A}_l} \sigma^2_{gi}\mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l}(\mathbf{K}_l\widetilde{\mathbf{Z}}_{i,l})^T + \mathbf{K}_l\tilde{\mathbf{\Omega}}_{gl}\mathbf{K}_l^T)$

$p_{igl} | \sigma^2_{gl} \sim N(0, \sigma^2_{gl}q_{il} + r_{igl})$

Let $E[\sigma^2_{gl}] = M_l$, $E[(\sigma^2_{gl})^2] = V_l$

$E[p^2_{igl}] = E[E[p^2_{igl} | \sigma^2_{gl}]] = E[\sigma^2_{gl}]q_{il} + r_{igl} = M_l q_{il} + r_{igl}$

Using methods of moment:

$\hat{M}_l = (\sum_{i,g} \frac{p^2_{igl} - r_{igl}}{q_{il}})/(I * G)$

$E[p_{igl}^4] = E[E[p_{igl}^4|\sigma_{gl}^2]] = E[3(\sigma_{gl}^2 q_{il} + r_{igl})^2] = 3E[(\sigma_{gl}^2 q_{il})^2 + 2(\sigma_{gl}^2 q_{il}) r_{igl} + r_{igl}^2] = 3V_l q_{il}^2 + 6M_l q_{il} r_{il} + 3r_{il}^2$

Using methods of moment:

$\hat{V}_l = \sum_{i,g}(\frac{p_{igl}^4 - 3r_{il}^2 - 6M_l q_{il} r_{il})}{3q_{il}^2}/(I * G)$

$\alpha_l = M_l^2/(V_l - M_l^2)$

$\gamma_l = M_l/(V_l - M_l^2)$

Using Gauss-Laguerre quadrature to calculate the following two integrals:

$P(\mathbf{K}_l \widetilde{\mathbf{y}}_{gl}) = \int_R P(\mathbf{K}_l \widetilde{\mathbf{y}}_{gl}|\sigma_{gl}^2) P(\sigma_{gl}^2)\, \mathrm{d}\sigma_{gl}^2$

$E_{\sigma_{gl}^2|\mathbf{K}_l \widetilde{\mathbf{y}}_{gl}}[\sigma_{gl}^2] = \int_R \sigma_{gl}^2 P(\mathbf{K}_l \widetilde{\mathbf{y}}_{gl}, \sigma_{gl}^2)/P(\mathbf{K}_l \widetilde{\mathbf{y}}_{gl})\, \mathrm{d}\sigma_{gl}^2$

$E_{\sigma_{gl}^2|\mathbf{K}_l \widetilde{\mathbf{y}}_{gl}}[\sigma_{gl}^2]$ serves as an estimate of σ_{gl}^2.

4.2.6 Hypothesis testing

To test whether a certain contrast $\mathbf{a}^T \boldsymbol{\beta}_g$ is zero:

$H_0 : \mathbf{a}^T \boldsymbol{\beta}_g = 0$

$H_1 : \mathbf{a}^T \boldsymbol{\beta}_g \neq 0$

$\hat{\boldsymbol{\beta}}_g = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \tilde{\mathbf{y}}_g$

Let $\mathbf{k} = \mathbf{a}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$

The test statistics: $t_g = \boldsymbol{a}^T \hat{\boldsymbol{\beta}}_g / \sqrt{var(\boldsymbol{a}^T \hat{\boldsymbol{\beta}}_g)} = \mathbf{k} \widetilde{\mathbf{y}}_g / \sqrt{\mathbf{k} \widetilde{\mathbf{Z}} \boldsymbol{\Sigma}_g \widetilde{\mathbf{Z}}^T \mathbf{k}^T + \mathbf{k} \widetilde{\boldsymbol{\Omega}}_g \mathbf{k}^T}$

Under H_0:

t_g asymptotically follows $N(0, 1)$ with large sample size.

Under H_1:

t_g asymptotically follows $N(0, 1 + \tau^2)$ with large sample size.

Denote $z_g = 0$ when gene g is non-differential (H_0), $z_g = 1$ when gene g is differential (H_1).

$P(z_g = 1) = p$

Use E-M algorithm to estimate p and τ^2.

E-step:

$L = \prod_g [N(t_g, 0, 1+\tau^2)p]^{I(z_g=1)}[N(t_g, 0, 1)(1-p)]^{I(z_g=0)}$

$log(L) = l = \sum_g I(z_g = 1)log[N(t_g, 0, 1+\tau^2)p] + I(z_g = 0)log[N(t_g, 0, 1)(1-p)] = \sum_g[I(z_g = 1)[-log(2\pi)/2 - log(1+\tau^2)/2 - \frac{t_g^2}{2(1+\tau^2)} + log(p)] + I(z_g = 0)[-log(2\pi)/2 - \frac{t_g^2}{2} + log(1-p)]]$

Let $M_g = E_{z_g|t_g,(\tau^2)^{(t)},p^{(t)}}[I(z_g = 1)]$

$= P[z_g = 1|t_g, (\tau^2)^{(t)}, p^{(t)}] = P[t_g|z_g = 1, (\tau^2)^{(t)}, p^{(t)}]P[z_g = 1|(\tau^2)^{(t)}, p^{(t)}]/P[t_g|(\tau^2)^{(t)}, p^{(t)}]$

$= N(t_g, 0, 1 + (\tau^2)^{(t)})p^{(t)}/[N(t_g, 0, 1 + (\tau^2)^{(t)})p^{(t)} + N(t_g, 0, 1)(1 - p^{(t)})]$

$E_{z_g,|t_g,(\tau^2)^{(t)},p^{(t)}}[l] = C + \sum_g M_g[-log(1+\tau^2)/2 - \frac{t_g^2}{2(1+\tau^2)} + log(p)] + (1-M_g)log(1-p)$

Where C is some constant.

M-step:

$p = \frac{\sum_g M_g}{G}$

$\tau^2 = \frac{\sum_g M_g t_g^2}{\sum_g M_g} - 1$

If $\tau^2 < 0$, τ^2 is set to be 0.

When E-M converges, M_g is treated as posterior probability for differential.

4.2.7 Benchmark data collection and processing

The bone marrow scRNA-seq data from the Human Cell Atlas (HCA) were downloaded from [103] and aligned to human hg19 genome using Cell Ranger [24] version 2.1.1. The data are analyzed using the default RAISIN pipeline. For this analysis, bulk RNA-seq data (count matrix) of FACS-sorted hematopoietic cell types were downloaded from the Gene Expression Omnibus (GEO accession number: GSE74246). The counts were converted to TPM and log2 transformed after adding a pseudocount of 1. DESeq2 was used to call DE genes in bulk RNA-seq (FDR cutoff=0.05).

4.2.8 Analysis of HCA bone marrow data

For this dataset, differential expression are detected by RAISIN with FDR < 0.05. To benchmark the performance, we annotated the cell type of each cell subpopulation using cell-type specific marker genes which were derived from bulk RNA-seq data from 13 FACS-sorted hematopoietic cell types. In bulk RNA-seq, gene expression profiles of replicate samples were averaged for each cell type. For each pair of cell types k_1 and k_2, genes were ranked based on differences in gene expression between the two cell types, and the top 100 genes upregulated in each cell type were obtained as a marker gene set.

We first assigned an initial cell type label for each individual cell. For each marker gene set, genes' average expression was calculated for each bulk RNA-seq sample. These data were arranged as a matrix $\mathbf{B}$, where each row represents a marker gene set and each column represents a bulk RNA-seq sample. Similarly, the averaged expression of each marker gene set was also calculated for each cell in the scRNA-seq data. Denote the resulting matrix as $\mathbf{C}$, where each row represents a marker gene set and each column represents a cell. For both matrices, each row was standardized across samples or cells to have zero mean and unit variance. For each cell i, the Spearman correlation between the i-th column of $\mathbf{C}$ and each column of $\mathbf{B}$ was calculated. If the maximum correlation was above 0.6, we assigned the cell type corresponding to the maximum correlation to cell i. Otherwise, the cell was not assigned any cell type.

We then assigned cell type for each cell cluster. For each cluster, the proportion of cells from each cell type was computed. The cell type with the largest proportion was identified. If this largest proportion was larger than 0.7, then this cell type was used to annotate the cell cluster. Otherwise, the cell cluster was annotated as unknown cell type. After cell type annotation, the performance of differential analysis of scRNA-seq in each cell cluster was evaluated.

4.2.9 Simulation study

We generated simulation data based on the HCA bone marrow scRNA-seq dataset. Eight simulated samples were created using randomly sampled cells from the eight bone marrow samples. For each bone marrow sample, cells were drawn from four cell types including common lymphoid progenitor (CLP), monocyte, erythroid and hematopoietic stem cell (HSC) to create a simulated sample consisting of four cell types.

Baseline simulation. The eight simulated samples were partitioned into two groups, denoted as $\mathcal{S}_1$ and $\mathcal{S}_2$. Each group contained four samples. Differences in cell proportion between the two sample groups were introduced for monocyte and HSC, but not for CLP and erythroid. To implement this, each simulated sample s was created by randomly drawing a_s CLP cells, b_s erythroid cells, c_s HSC cells and d_s monocyte cells from the corresponding HCA bone marrow sample. Here a_s was a random integer uniformly drawn from the interval $[5, 10]$. The a_ss for 8 samples were independently generated. Similarly, b_s was a random integer uniformly distributed in interval $[40, 50]$. c_s was also an random integer, but its distribution was different for the two sample groups. For samples in $\mathcal{S}_1$, c_s was uniformly drawn from $[5, 10]$. For samples in $\mathcal{S}_2$, c_s was uniformly drawn from $[40, 50]$. Similarly, d_s was a random integer with different distributions for the two sample groups. For samples in $\mathcal{S}_1$, d_s was uniformly drawn from $[40, 50]$. For samples in $\mathcal{S}_2$, c_s was uniformly drawn from $[5, 10]$. After sampling cells, the expression profiles (including raw read counts, scran normalized values, and SAVER imputed values) of the sampled cells were carried over to the simulated sample. RAISIN cell clustering was then performed using the SAVER imputed gene expression to group all cells into four clusters. This baseline simulation procedure generated two groups of simulated samples with differential cell proportion for HSC and monocyte and non-differential cell proportion for CLP and erythroid. In this baseline simulation, the gene expression profile of each cell type was not expected to be differential between the two sample groups because $\mathcal{S}_1$ and $\mathcal{S}_2$ were obtained by partitioning samples of the same type (i.e., they were all bone marrow samples).

Simulation 1. In order to benchmark detection of DE between two groups of samples, we further introduced differentially expressed genes on top of the samples generated by the baseline simulation. We simulated a total of 48 datasets through combinations of 3 different DE gene proportions, 4 cell types, and 4 different magnitude of differential signals. Let G denote the total gene number, and p be the proportion of genes that are differential. In each simulation dataset, one cell type was chosen to introduce DE, and the other three cell types remained the same and thus did not contain DE. For the chosen cell type, DE was introduced to $p * G$ randomly chosen genes so that $\frac{p}{2} * G$ genes were upregulated in sample group $\mathcal{S}_1$ and the other $\frac{p}{2} * G$ genes were upregulated in $\mathcal{S}_2$. The DE signal for the $\frac{p}{2} * G$ genes upregulated in $\mathcal{S}_1$ was introduced as follows. For the cell type in question, let $\mathcal{C}_1$ denote the set of all cells in sample group $\mathcal{S}_1$, and let $\mathbf{Y}_1$ denote the expression matrix of the $\frac{p}{2} * G$ selected genes in cells in $\mathcal{C}_1$. We first randomly sampled the same number of cells with the same cell type from the bone marrow scRNA-seq data and denote this new set of cells as $\mathcal{C}_2$. In $\mathcal{C}_2$, we removed genes with zero expression across all cells. The remaining genes were stratified into four equal-sized groups based on each gene's average expression across cells in $\mathcal{C}_2$. The four strata corresponded to genes with expression from high to low. We then picked up a stratum and randomly sampled $\frac{p}{2} * G$ genes from the stratum. Let $\mathbf{Y}_2$ denote the expression matrix of these $\frac{p}{2} * G$ genes in cells in $\mathcal{C}_2$. Note that the matrix dimension of $\mathbf{Y}_2$ was the same as the dimension of $\mathbf{Y}_1$. We added $\mathbf{Y}_2$ to $\mathbf{Y}_1$ and used their sum to replace $\mathbf{Y}_1$ in the original data matrix. In this way, upregulation was introduced to $\frac{p}{2} * G$ genes in sample group $\mathcal{S}_1$. Depending on which of the four gene strata was chosen from $\mathcal{C}_2$ to generate $\mathbf{Y}_2$, four different magnitudes of differential expression can be introduced. Using a similar procedure, the DE signal for the $\frac{p}{2} * G$ genes upregulated in $\mathcal{S}_2$ was introduced. This creates one simulation dataset. By selecting different cell types to simulate DE (there were 4 cell types in total), setting DE gene proportion to 3 different values ($p = 0.02, 0.1, 0.2$) and introducing 4 different magnitudes of DE signals, a total of 48 simulation datasets were created. Different DE analysis methods were then run on each dataset to detect DE between the two sample groups $\mathcal{S}_1$ and $\mathcal{S}_2$.

Simulation 2. In order to benchmark detection of DE between two cell types, we introduced DE genes on top of the samples generated by the baseline simulation as follows. Given a cell type pair, in order to create a clean non-differential background, we first randomly sampled 10% of all genes as the evaluation gene set T. For each gene in T, we then randomly permuted cells' expression values across the two cell types within each sample. After this step, all genes in T should be non-differential between the two cell types. We then added DE to $p * T$ genes in gene set T using the same approach as in simulation 1, but only in T. Our performance evaluation was based on genes in T only because the true differential status of the remaining 90% genes not included in T was unknown. The reason we only chose 10% of genes as the evaluation gene set is that if we chose too many genes and made them non-differential using permutation, the two cell types would become the same and could not be separated into two cell populations by cell clustering. The procedure above creates one simulation dataset. By selecting different cell type pairs to simulate DE (there were 6 cell type pairs in total), setting DE gene proportion to 3 different values ($p = 0.02, 0.1, 0.2$) and introducing 4 different magnitudes of DE signals, a total of 72 simulation datasets were created. Different DE analysis methods were then run on each dataset to detect DE between two cell types.

4.2.10 Performance evaluation by AUC and FDR difference

To evaluate a method's overall ability to detect DE genes, the sensitivity (y-axis) was plotted as a function of FDR (x-axis). The area under the sensitivity-FDR curve

Table 4-I. List of differential methods compared.

Method	Type	Preprocessing	Reference
limma	Bulk RNA-seq	Raw Count, SAVER Imputed	[97]
DESeq2	Bulk RNA-seq	Raw Count	[104]
edgeR	Bulk RNA-seq	Raw Count	[105]
t test	scRNA-seq	Raw Count, SAVER Imputed	[106]
wilcoxon test (Seurat)	scRNA-seq	Raw Count, SAVER Imputed	[42]
MAST	scRNA-seq	Raw Count, SAVER Imputed	[95]
limma (dupcor)	scRNA-seq	Raw Count, SAVER Imputed	[107]
scDD	scRNA-seq	Raw Count, SAVER Imputed	[94]
DESeq2	scRNA-seq	Zinbwave	[108]
edgeR	scRNA-seq	Zinbwave	[108]

(AUC) was calculated for each method. The calculation only considers the curve up to FDR ≤ 0.25 since in practice users usually only care about findings with relatively small FDR. For simulations, the true DE status of each gene was known. Thus, sensitivity and FDR were computed using genes' true DE status.

To evaluate whether a method can accurately estimate FDR, we computed the difference between the real FDR and reported FDR. For each method, this difference was plotted as a function of real FDR. We computed the area under the curve up to real FDR ≤ 0.25 and called this area "FDR difference". If the FDR difference is negative, the real FDR overall is smaller than the reported FDR, and the method is conservative. If the FDR difference is positive, the real FDR overall is larger than the reported FDR, and the method is too optimistic and reports misleading error rates.

4.3 Results

4.3.1 Simulation study

We compared RAISIN and the most commonly used DE methods in simulations where in silico differential signals were added to non-differential background constructed using real scRNA-seq data from biological replicates. Among the compared methods, Wilcoxon test (used by Seurat [42]), t-test, MAST [95] and scDD [94] ignore sample-level variance. They treat cells from biological replicates as if they were from one sample. They were run using both SCRAN [99] normalized data without imputation and SAVER [109] imputed data. DESeq2 [104], edgeR [105] and limma [97] are bulk DE methods. They do not consider cell-level variance. To run them, cells in each sample and cell cluster were pooled to create a pseudo-bulk sample. Pseudo-bulk samples were then analyzed as if they were bulk samples. DESeq2 and edgeR are based on modeling read counts. Thus, they were run using both counts and ZINB-WaVE [108] corrected data. Limma also provides a LMM (limmacell [rename-limma-LMM]) originally designed for handling random effects of microarray probesets. Since limma accepts both continuous data and discrete counts, limma and limmaLMM were run using both read counts (normalized by total library size) and SAVER imputed values. RAISIN were run in four different modes that either use scran normalized unimputed data or SAVER imputed data as input, and with or without variance regularization. RAISIN without variance regularization (RAISIN-LMM) reduces to the classical LMM. For comparing the same cell cluster between two groups of samples, all methods except for RAISIN and RAISIN-LMM failed to control false discovery rates (FDR) (Figure 4-1C). RAISIN with variance regularization substantially outperformed RAISIN-LMM without variance regularization in terms of the sensitivity-FDR curve characterized by the area under the curve (AUC) (Figure 4-1C).

4.3.2 HCA bone marrow data

To test RAISIN in real data, we analyzed Human Cell Atlas (HCA) [69] bone marrow scRNA-seq data from 8 healthy donors. After sample alignment, we identified 44 cell clusters distributed along three major hematopoietic differentiation lineages consistent with the known biology (Figure 4-2A). We performed a null DE analysis by randomly partitioning the samples into two groups. The analysis was run both using all samples and cells and randomly subsampled samples and cells. There should be no DE genes

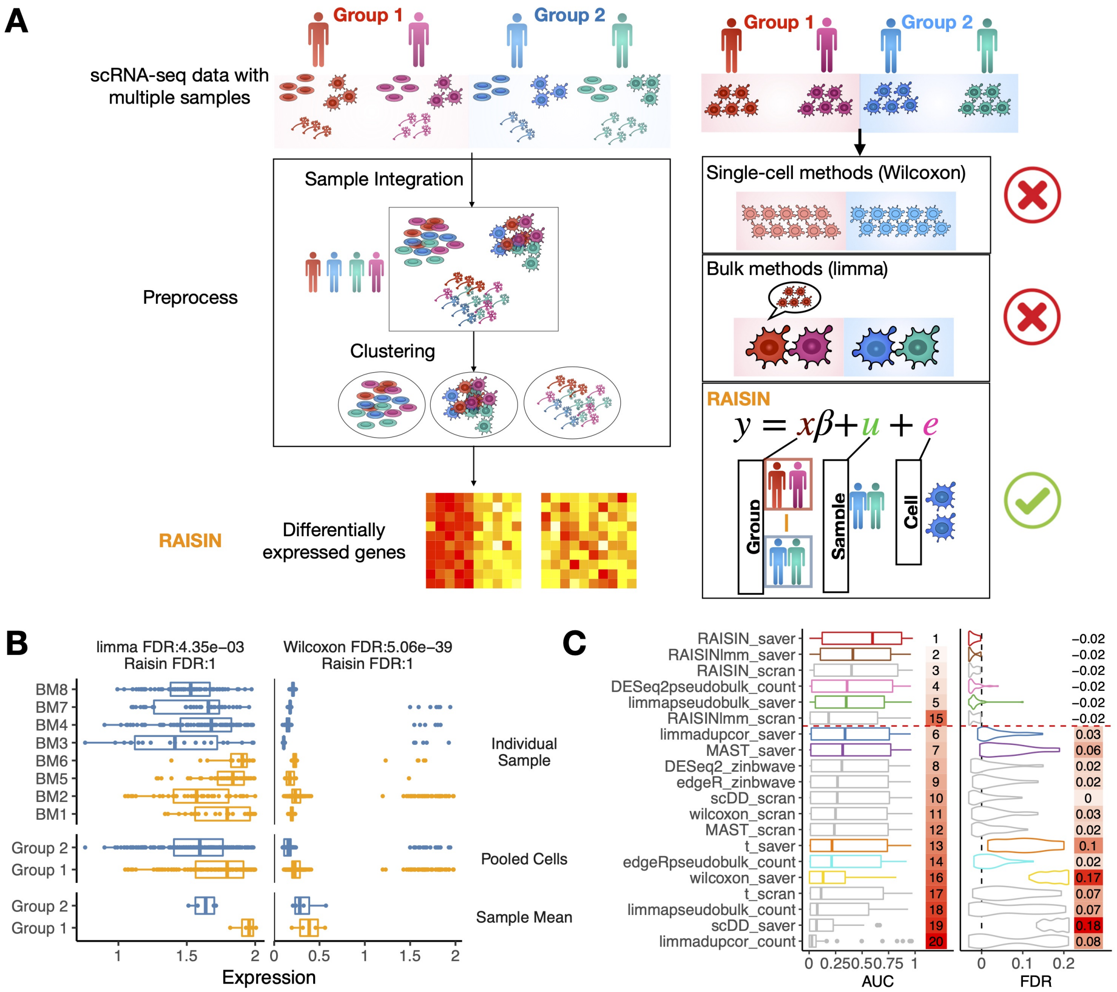


Figure 4-1. A. Schematic of RAISIN algorithm. B. Methods that ignore cell-level variability (limma) or sample-level variability (Wilcoxon test) yield false positives. C. AUROC and FDR difference of different methods in a simulation study

between the two sample groups. However, most methods reported over 100 DE genes at their claimed 5% FDR cutoff in at least one analysis (Figure 4-2C). Here only RAISIN (SAVER or SCRAN) and RAISIN-LMM (SCRAN) reported fewer than 100 DE genes. Figure 4-1B left panel shows an example to illustrate why methods that ignore cell-level variability (e.g., limma) failed. This gene is non-differential since the cell-level variability is larger than the observed difference between groups. However, when cells from each sample are collapsed into a bulk sample, this variability is not reflected in the averaged bulk expression which appeared to be differential between the two groups. This yields an overly optimistic FDR of $4.35 * 10^{-3}$ by limma. Figure 4-1B right panel shows an example to illustrate why methods that ignore sample-level variability (e.g., Wilcoxon test) failed. This gene is non-differential since the sample-level variability is larger than the observed difference between groups. However, when cells from biological replicates are treated as if they were independently drawn from one sample, the degrees of freedom of the hypothesis test are falsely determined by the cell number which is large. This yields an overly optimistic FDR of $5.06 * 10^{-39}$ by Wilcoxon test. In both cases, RAISIN reported an FDR of 1.

The regression framework adopted by RAISIN is flexible. Besides comparing two sample groups or two cell types, it can also be used to analyze the association between gene expression and any other categorical or continuous variables. One can also add covariates to the model to adjust for potential confounding (Online Methods). Figure 4-2B compares computation time of different methods for comparing two sample groups in a single cell cluster. The time increases as a function of cell number. The LMM used by limma (limmadupcor) and MAST are not scalable to atlas-scale datasets, and the classical LMM (RAISIN-LMM) is slow. Compared to the classical LMM, RAISIN is 8 times faster and can handle a two-group comparison of a cell cluster with 10^5 cells in 12 minutes. The computational efficiency of RAISIN is in between Wilcoxon test and bulk DE analysis methods (i.e., limma, DESeq2, edgeR). Thus, RAISIN not only improves statistical power and false discovery rate estimation of the DE analysis, but is also scalable to atlas-level analyses.

4.4 Discussion

With the reduced cost of single-cell sequencing technologies, single-cell RNA-seq data with multiple samples start to emerge in recent years. Most existing methods to identify differential genes consider only sample-level variability or cell-level variability. These methods fail to control for false discovery rate and have impaired statistical

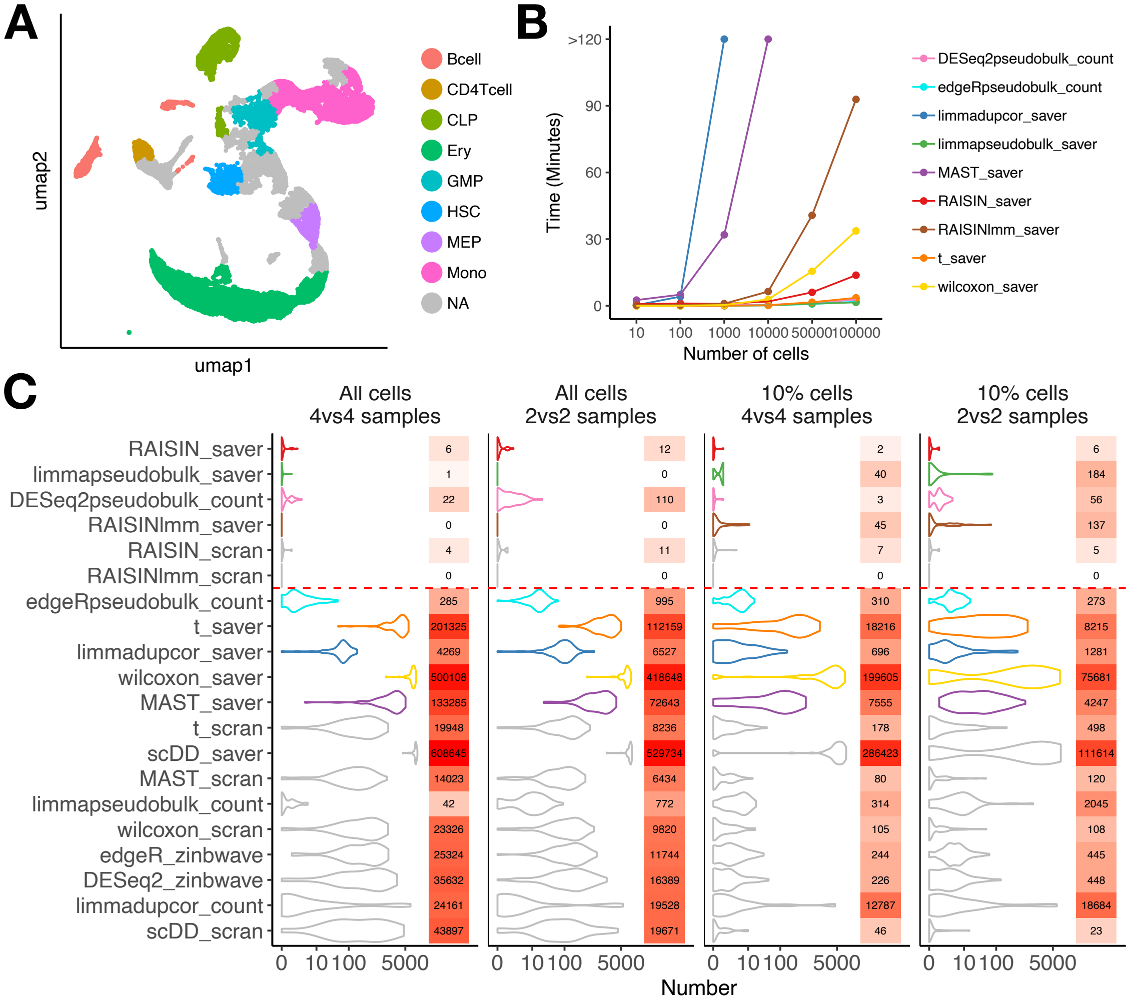


Figure 4-2. A. UMAP of the HCA dataset B. Computational time (y-axis) with different number of cells in the dataset (x-axis). C. Number of false positives with different number of cells and different number of samples.

power. The classical linear mixed model is able to consider both sample-level and cell-level variability, but it scales poorly and has an unstable variance estimate with a small number of cells or samples. To address these issues, we developed a novel statistical model, RAISIN, that performs differential analysis for single-cell gene expression data with multiple samples. RAISIN models both sample-level and cell-level variability, and combines Bayesian shrinkage estimators to stabilize variance estimate with a small number of cells and samples. RAISIN also improves the scalability of the classical linear mixed model by reducing the dimensionality of the data. RAISIN has the best statistical power while controls the false positive rate among all existing methods. The differential genes identified by RAISIN can be used to study the molecular mechanism that differentiates different groups of samples for each cell type.

In the future, RAISIN can be further extended to study the differential activities of other types of genomic information. For example, combined with the predicted chromatin accessibility using gene expression [89], RAISIN is able to identify cis-regulatory elements that have differential activities across groups of samples. A differential cis-regulatory element in the promoter region of a differential gene may help explain the mechanism of the gene's differential expression. Similar analyses can also be done for DNA methylation, histone modification and other types of epigenomic signals.

Conclusions and general discussion

Single-cell sequencing has become a powerful tool in biomedical research. It grants researchers unprecedented resolution in studying cell diversity, cell differentiation, and many other biological processes. Analyzing data from single-cell sequencing is challenging, and new statistical and machine learning methods are needed. In this thesis work, we developed statistical methods that computationally order cells to infer the underlying biological process, enhance the highly sparse single-cell ATAC-seq data to better infer the gene regulatory programs, and perform differential analysis to identify molecular signatures that differentiate different groups of samples. These methods can extract useful information from the highly sparse, noisy and complex data from single-cell sequencing. Such information may provide crucial insights into the biological process.

Single-cell sequencing technologies are evolving rapidly. New types of technologies keep emerging, which continuously brings new challenges in analyzing the data. Novel statistical and computational methods also provide new approaches to extracting useful biological and clinical information from the data. Thus, the development of new statistical approaches is equally important as the development of new single-cell sequencing technology. In the near future, it will be especially important to develop new methods in three directions.

First, highly scalable methods need to be developed to efficiently integrate and analyze single-cell data from multiple samples. Thanks to the advancement of single-cell sequencing technology, the cost of single-cell experiments continue to decrease.

In recent years, large-scale datasets from single-cell RNA-seq with many samples or patients start to emerge [28, 96]. Analytical methods have been developed to address certain issues for such datasets, such as methods to integrate data across individuals [41, 42] and method developed in this thesis to identify genes with differential expression. However, there are still many issues that remain to be solved. For example, many current methods are only tested and applied for datasets with thousands to millions of cells. Their scalability may not be able to handle datasets with billions or even larger amounts of cells, which may appear in the near future with the next generation of single-cell sequencing technology (e.g. celsee). While the integration of gene expression profile across multiple samples has been relatively well studied, how to integrate other types of genomic information such as T cell receptor sequence from multiple individuals remains an open question. While the current methodology development has been focused on comparing the average expression between groups of samples, the method to identify genes or molecular signatures with differential variation or differential pattern along pseudotime across groups of samples is still lacking.

Second, methods that can integrate single-cell data from multiple modalities need to be developed. While single-cell gene expression profiling is relatively mature and has been widely used, other types of single-cell technologies such as single-cell ATAC-seq and single-cell DNA methylation are still immature and less prevalent. Thus, computationally predicting one type of genomic information using the other is a useful complement to the experimental approach. For example, we have already demonstrated that it is possible to predict chromatin accessibility using gene expression information for single cells with reasonable accuracy [110]. A similar idea can be used to predict other types of epigenetic signals such as histone modification, DNA methylation, and DNA 3D structure. Since currently technology still cannot reliably measure multiple data modalities in one single cell, the prediction methods can also serve as an important bridge between different modalities from experimental data.

For example, to align data from single-cell RNA-seq and single-cell ATAC-seq, we can predict the chromatin accessibility for single-cell RNA-seq data and align it with the experimental chromatin accessibility from single-cell ATAC-seq. Other potential directions include to combine CRISPR-based perturbations with single-cell RNA-seq [111] and single-cell ATAC-seq [112] to better study gene regulatory network, as well as combine spatial information with single-cell sequencing data to study spatial transcriptomics and spatial epigenomics.

Third, methods need to be developed to translate the information from single-cell data to discoveries in biology and improvement in clinical practice. For example, highly scalable machine learning methods such as deep neural network can be used to link patients' response to certain therapy with single-cell data, and to elucidate the molecular mechanism explaining different treatment outcomes. This information can be used to find new biomarkers to predict patients' early outcomes or to improve the therapy to benefit more patients. Single-cell data can also be used to deconvolve bulk sequencing data collected from a large population cohort. This information can be further used to adjust for confounding effects of different cell type compositions in association studies such as epigenome-wide association studies, which leads to more accurate identification of epigenetic marks associated with certain traits.

In summary, new statistical and computational methods to extract useful information from the single-cell sequencing data are still much needed. Combined with new single-cell sequencing technologies, these methods will ultimately lead to a deeper understanding of basic biology, as well as the improvement of clinical practice and public health.

References

1. Levine, M. & Tjian, R. Transcription regulation and animal diversity. *Nature* **424,** 147–151 (2003).
2. Matlin, A. J., Clark, F. & Smith, C. W. Understanding alternative splicing: towards a cellular code. *Nature reviews Molecular cell biology* **6,** 386–398 (2005).
3. Deribe, Y. L., Pawson, T. & Dikic, I. Post-translational modifications in signal integration. *Nature structural & molecular biology* **17,** 666–672 (2010).
4. Lee, T. I. & Young, R. A. Transcriptional regulation and its misregulation in disease. *Cell* **152,** 1237–1251 (2013).
5. Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics* **17,** 333 (2016).
6. Wang, Z., Gerstein, M. & Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet.* **10,** 57–63 (2009).
7. Boyle, A. P. *et al.* High-resolution mapping and characterization of open chromatin across the genome. *Cell.* **132,** 311–322 (2008).
8. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat Methods.* **10,** 1213–1218 (2013).
9. Giresi, P. G., Kim, J., McDaniell, R. M., Iyer, V. R. & Lieb, J. D. FAIRE (Formaldehyde-Assisted Isolation of Regulatory Elements) isolates active regulatory elements from human chromatin. *Genome research* **17,** 877–885 (2007).
10. Klemm, S. L., Shipony, Z. & Greenleaf, W. J. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics* **20,** 207–220 (2019).
11. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-wide mapping of in vivo protein-DNA interactions. *Science.* **316,** 1497–1502 (2007).
12. Frommer, M. *et al.* A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands. *Proceedings of the National Academy of Sciences* **89,** 1827–1831 (1992).
13. Consortium, E. P. *et al.* The ENCODE (ENCyclopedia of DNA elements) project. *Science.* **306,** 636–640 (2004).
14. Jin, W. *et al.* Genome-wide detection of DNase I hypersensitive sites in single cells and FFPE tissue samples. *Nature.* **528,** 142 (2015).

15. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics. *Nature methods* **10,** 1213 (2013).

16. Frank, N. Y., Schatton, T. & Frank, M. H. The therapeutic promise of the cancer stem cell concept. *The Journal of clinical investigation* **120,** 41–50 (2010).

17. Li, L. & Clevers, H. Coexistence of quiescent and active adult stem cells in mammals. *science* **327,** 542–545 (2010).

18. Choi, P. J., Cai, L., Frieda, K. & Xie, X. S. A stochastic single-molecule event triggers phenotype switching of a bacterial cell. *Science* **322,** 442–446 (2008).

19. Saitou, M., Barton, S. C. & Surani, M. A. A molecular programme for the specification of germ cell fate in mice. *Nature* **418,** 293–300 (2002).

20. Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods.* **6,** 377 (2009).

21. Islam, S. *et al.* Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Res.* **21,** 1160–1167 (2011).

22. Ramsköld, D. *et al.* Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nat Biotechnol.* **30,** 777 (2012).

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

24. Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8,** 14049 (2017).

25. Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360,** 176–182 (2018).

26. Zeisel, A. *et al.* Molecular architecture of the mouse nervous system. *Cell* **174,** 999–1014 (2018).

27. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol.* **32,** 381 (2014).

28. Azizi, E. *et al.* Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell* **174,** 1293–1308 (2018).

29. Plass, M. *et al.* Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science* **360,** eaaq1723 (2018).

30. Moffitt, J. R. *et al.* Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. *Science* **362,** eaau5324 (2018).

31. Park, J. *et al.* Single-cell transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease. *Science* **360,** 758–763 (2018).

32. Buenrostro, J. D. *et al.* Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature.* **523,** 486–490 (2015).

33. Cusanovich, D. A. *et al.* Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science.* **348,** 910–914 (2015).

34. Rotem, A. *et al.* Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state. *Nat Biotechnol.* **33,** 1165 (2015).

35. Cao, J. *et al.* Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science.* **361,** 1380–1385 (2018).

36. Gawad, C., Koh, W. & Quake, S. R. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics* **17,** 175 (2016).

37. Gravina, S., Dong, X., Yu, B. & Vijg, J. Single-cell genome-wide bisulfite sequencing uncovers extensive heterogeneity in the mouse liver methylome. *Genome biology* **17,** 150 (2016).

38. Tu, A. A. *et al.* TCR sequencing paired with massively parallel 3 RNA-seq reveals clonotypic T cell signatures. *Nature immunology* **20,** 1692–1699 (2019).

39. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nature biotechnology* **37,** 547 (2019).

40. Hou, W., Ji, Z., Ji, H. & Hicks, S. C. A Systematic Evaluation of Single-cell RNA-sequencing Imputation Methods. *bioRxiv* (2020).

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

42. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36,** 411 (2018).

43. Ji, Z. & Ji, H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic acids research* **44,** e117–e117 (2016).

44. Chen, Z. *et al.* TCF-1-centered transcriptional network drives an effector versus exhausted CD8 T cell-fate decision. *Immunity* **51,** 840–855 (2019).

45. Tang, F. *et al.* Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell RNA-Seq analysis. *Cell Stem Cell.* **6,** 468–478 (2010).

46. Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L. & Wold, B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods.* **5,** 621 (2008).

47. Schena, M., Shalon, D., Davis, R. W. & Brown, P. O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science.* **270,** 467–470 (1995).

48. Schulze, A. & Downward, J. Navigating gene expression using microarrays—a technology review. *Nat Cell Biol.* **3,** E190 (2001).

49. Simpson, E. H. The interpretation of interaction in contingency tables. *J Roy Stat Soc B Met.* **13,** 238–241 (1951).

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

51. Saliba, A.-E., Westermann, A. J., Gorski, S. A. & Vogel, J. Single-cell RNA-seq: advances and future challenges. *Nucleic Acids Res.* **42,** 8845–8860 (2014).

52. Amir, E.-a. D. *et al.* viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat Biotechnol.* **31,** 545 (2013).

53. Bendall, S. C. *et al.* Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell.* **157,** 714–725 (2014).

54. Qiu, P. *et al.* Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nat Biotechnol.* **29,** 886 (2011).

55. Marco, E. *et al.* Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proc Natl Acad Sci U S A.* **111,** E5643–E5650 (2014).

56. Jaitin, D. A. *et al.* Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science.* **343,** 776–779 (2014).

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

58. Dalerba, P. *et al.* Single-cell dissection of transcriptional heterogeneity in human colon tumors. *Nat Biotechnol.* **29,** 1120 (2011).

59. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat Methods.* **11,** 740 (2014).

60. Fraley, C. & Raftery, A. E. Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc.* **97,** 611–631 (2002).

61. Fraley, C., Raftery, A. E., Murphy, T. B. & Scrucca, L. *mclust version 4 for R: normal mixture modeling for model-based clustering, classification, and density estimation* tech. rep. (Technical Report, 2012).

62. Wood, S. N. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *J Roy Stat Soc B Met.* **73,** 3–36 (2011).

63. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J Roy Stat Soc B Met.* **57,** 289–300 (1995).

64. Amit, I. *et al.* Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science.* **326,** 257–263 (2009).

65. Shin, J. *et al.* Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell.* **17,** 360–372 (2015).

66. Haghverdi, L., Buettner, F. & Theis, F. J. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics.* **31,** 2989–2998 (2015).

67. Clark, S. J. *et al.* scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. *Nat Commun.* **9,** 781 (2018).

68. Chen, X. *et al.* Joint single-cell DNA accessibility and protein epitope profiling reveals environmental regulation of epigenomic heterogeneity. *Nat Commun.* **9,** 4590 (2018).

69. Regev, A. *et al.* The Human Cell Atlas. *Elife.* **6,** e27041 (2017).

70. Schep, A. N., Wu, B., Buenrostro, J. D. & Greenleaf, W. J. chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nat Methods.* **14,** 975 (2017).

71. Ji, Z., Zhou, W. & Ji, H. Single-cell regulome data analysis by SCRAT. *Bioinformatics.,* btx315 (2017).

72. De Boer, C. G. & Regev, A. BROCKMAN: deciphering variance in epigenomic regulators by k-mer factorization. *BMC Bioinformatics.* **19,** 253 (2018).

73. Zhao, C., Hu, S., Huo, X. & Zhang, Y. Dr. seq2: A quality control and analysis pipeline for parallel single cell transcriptome and epigenome data. *PLoS One.* **12,** e0180583 (2017).

74. Pliner, H. A. *et al.* Cicero predicts cis-regulatory DNA Interactions from single-cell chromatin accessibility data. *Mol Cell.* **71,** 858–871 (2018).

75. Baker, S. M., Rogerson, C., Hayes, A., Sharrocks, A. D. & Rattray, M. Classifying cells with Scasat, a single-cell ATAC-seq analysis tool. *Nucleic Acids Res.* (2018).

76. Urrutia, E., Chen, L., Zhou, H. & Jiang, Y. Destin: toolkit for single-cell analysis of chromatin accessibility. *Bioinformatics.,* btz141. (2019).

77. Zamanighomi, M. *et al.* Unsupervised clustering and epigenetic classification of single cells. *Nat Commun.* **9,** 2410 (2018).

78. Cai, S., Georgakilas, G. K., Johnson, J. L. & Vahedi, G. A cosine similarity-based method to infer variability of chromatin accessibility at the single-cell level. *Front Genet.* **9,** 319 (2018).

79. González-Blas, C. B. *et al.* cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nat Methods.* **16,** 397 (2019).

80. Consortium, E. P. *et al.* An integrated encyclopedia of DNA elements in the human genome. *Nature* **489,** 57 (2012).

81. Yue, F. *et al.* A comparative encyclopedia of DNA elements in the mouse genome. *Nature* **515,** 355 (2014).

82. Zhang, Y. *et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* **9,** R137 (2008).

83. Buenrostro, J. D. *et al.* Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell.* (2018).

84. Cusanovich, D. A. *et al.* A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell.* **174,** 1309–1324 (2018).

85. Amemiya, H. M., Kundaje, A. & Boyle, A. P. The ENCODE Blacklist: Identification of Problematic Regions of the Genome. *Scientific Reports* **9,** 9354 (2019).

86. Ramsay, J. O. *et al.* Monotone regression splines in action. *Statistical Science.* **3,** 425–441 (1988).

87. Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system* in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), 785–794.

88. Sandelin, A., Alkema, W., Engström, P., Wasserman, W. W. & Lenhard, B. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* **32,** D91–D94 (2004).

89. Zhou, W. *et al.* Genome-wide prediction of DNase I hypersensitivity using gene expression. *Nat Commun.* **8,** 1038 (2017).

90. Corces, M. R. *et al.* Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nature genetics* **48,** 1193–1203 (2016).

91. Ji, H. *et al.* An integrated software system for analyzing ChIP-chip and ChIP-seq data. *Nature biotechnology* **26,** 1293–1300 (2008).

92. Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9,** 2579–2605 (2008).

93. Ramani, V. *et al.* Massively multiplex single-cell Hi-C. *Nat Methods.* **14,** 263 (2017).

94. Korthauer, K. D. *et al.* A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome biology* **17,** 222 (2016).

95. Finak, G. *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome biology* **16,** 278 (2015).

96. Li, H. *et al.* Dysfunctional CD8 T cells form a proliferative, dynamically regulated compartment within human melanoma. *Cell* **176,** 775–789 (2019).

97. Ritchie, M. E. *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research* **43,** e47–e47 (2015).

98. Bates, D., Mächler, M., Bolker, B. & Walker, S. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823* (2014).

99. Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology* **17,** 75 (2016).

100. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications* **9,** 997 (2018).

101. Haghverdi, L., Lun, A. T., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology* **36,** 421 (2018).

102. Efron, B. *et al.* Size, power and false discovery rates. *The Annals of Statistics* **35,** 1351–1377 (2007).

103. *Human Cell Atlas 1M immune cells* https://data.humancellatlas.org/explore/projects/cc95ff89-2e68-4a08-a234-480eca21ce79.

104. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* **15,** 550 (2014).

105. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139–140 (2010).

106. Student. The probable error of a mean. *Biometrika,* 1–25 (1908).

107. Smyth, G. K., Michaud, J. & Scott, H. S. Use of within-array replicate spots for assessing differential expression in microarray experiments. *Bioinformatics* **21,** 2067–2075 (2005).

108. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications* **9,** 284 (2018).

109. Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods* **15,** 539–542 (2018).

110. Zhou, W., Ji, Z., Fang, W. & Ji, H. Global prediction of chromatin accessibility using small-cell-number and single-cell RNA-seq. *Nucleic acids research* **47,** e121–e121 (2019).

111. Dixit, A. *et al.* Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167,** 1853–1866 (2016).

112. Rubin, A. J. *et al.* Coupled single-cell CRISPR screening and epigenomic profiling reveals causal gene regulatory networks. *Cell* **176,** 361–376 (2019).

Curriculum Vitae

Zhicheng Ji

Department of Biostatistics	Mobile Phone:	(410) 736-0905
Bloomberg School of Public Health	Email:	zji4@jhu.edu
Johns Hopkins University	Homepage:	http://www.zji90.com

Education

- Ph.D. in Biostatistics, Johns Hopkins Bloomberg School of Public Health, 2020 (expected)

  Thesis Advisor: Hongkai Ji, Ph.D.

- M.S.E. in Computer Science, Johns Hopkins Whiting School of Engineering, 2020 (expected)

- Sc.M. in Biostatistics, Johns Hopkins Bloomberg School of Public Health, 2015 (transferred to Ph.D. program)

- B.S. in Statistics, Fudan University, 2013

Honors and Awards

- Margaret Merrell Award, Department of Biostatistics, Johns Hopkins University, 2018

  Recognizes outstanding research by a doctoral student; Unique recipient

- June B. Culley Award, Department of Biostatistics, Johns Hopkins University, 2018

  Honors outstanding achievement by a doctoral student on schoolwide examination paper; Unique recipient

- Runner-up, ENCODE-DREAM in vivo Transcription Factor Binding Site Prediction Challenge, 2017

  Team leader; 4th place out of 50 teams

- ASA Section on Statistics in Genomics and Genetics Distinguished Student Paper Award, 2016

- Top Performers, Prostate Cancer DREAM Challenge, 2015

- Kocherlakota Award, Department of Biostatistics, Johns Hopkins University, 2014

  Honors outstanding achievement by a master's student on the first-year comprehensive examination; Unique recipient

- First-class Scholarship, Fudan University, 2012

Publications

Google Scholar

* indicates equal contributions

Journal Articles and Articles under Review

1. Wenpin Hou, **Zhicheng Ji**, Hongkai Ji and Stephanie Hicks. A Systematic Evaluation of Single-cell RNA-sequencing Imputation Methods. Genome Biology, Under review bioRxiv

2. **Zhicheng Ji**, Weiqiang Zhou, Wenpin Hou and Hongkai Ji. Single-cell ATAC-seq signal extraction and enhancement with SCATE. Genome Biology, Under review bioRxiv

3. Jifeng Zhang, Shoubao Yan, Cheng Jiang, **Zhicheng Ji**, Chenrun Wang and Weidong Tian. (2020) Network Properties of Cancer Prognostic Gene Signatures in the Human Protein Interactome. Genes. 11(3): 247.

4. Jiajia Zhang*, **Zhicheng Ji***, Justina Caushi*, Margueritta El Asmar*, Valsamo Anagnostou, Tricia Cottrell, Hok Yee Chan, Prerna Suri, Haidan Guo, Taha Merghoub, Jamie Chaft, Joshua Reuss, Ada Tam, Richard Blosser, Mohsen Abu-Akeel, John-William Sidhom, Ni Zhao, Jinny Ha, David Jones, Kristen Marrone, Jarushka Naidoo, Edward Gabrielson, Janis Taube, Victor Velculescu, Julie Brahmer, Franck Housseau, Matthew Hellmann, Patrick Forde, Drew Pardoll, Hongkai Ji, and Kellie Smith. (2020) Compartmental analysis of T cell clonal dynamics as a function of pathologic response to neoadjuvant PD-1 blockade in resectable NSCLC. *Clinical Cancer Research*, 26(6):1327-1337

5. Rachel K. Lex*, **Zhicheng Ji***, Kristin N. Falkenstein*, Weiqiang Zhou, Joanna L. Henry, Hongkai Ji and Steven A. Vokes. (2020) GLI transcriptional repression regulates tissue-specific enhancer activity in response to Hedgehog signaling. *eLife*, 9:e50670

6. Kimberly E. Stephens, Weiqiang Zhou, **Zhicheng Ji**, Zhiyong Chen, Shaoqiu He, Hongkai Ji, Yun Guan and Sean D. Taverna. (2019) Sex differences in gene regulation in the dorsal root ganglion after nerve injury. *BMC Genomics*. 20:147

7. Zeyu Chen*, **Zhicheng Ji***, Shin Foong Ngiow, Sasikanth Manne, Zhangying Cai, Alexander C. Huang, John Johnson, Ryan P. Staupe, Bertram Bengsch, Caiyue Xu, Sixiang Yu, Makoto Kurachi, Ramin S. Herati, Laura A. Vella,

Jennifer E. Wu, Omar Khan, Erietta Stelekati, Laura M. Mclan, Chi Wai Lau, Xiaolu Yang, Shelley L. Berger, Golnaz Vahedi, Hongkai Ji and E. John Wherry. (2019) TCF-1-Centered Transcriptional Network Drives an Effector versus Exhausted CD8 T Cell-Fate Decision. *Immunity*, 51(5): 840-855.e5

Featured in ACIR and ScienceDaily

8. Weiqiang Zhou, **Zhicheng Ji**, Weixiang Fang and Hongkai Ji. (2019) Global prediction of chromatin accessibility using small-cell-number and single-cell RNA-seq. *Nucleic Acids Research*, 47(19):e121.

9. Weiqiang Zhou, Ben Sherwood, **Zhicheng Ji**, Yingchao Xue, Fang Du, Jiawei Bai, Mingyao Ying, and Hongkai Ji. (2017) Genome-wide prediction of DNase I hypersensitivity using gene expression. *Nature Communications*. 8(1):1038

10. Zheng Kuang, **Zhicheng Ji**, Jef D. Boeke and Hongkai Ji. (2017) Dynamic motif occupancy (DynaMO) analysis identifies transcription factors and their binding sites driving dynamic biological processes. *Nucleic Acids Research*. 46(1): e2

11. Fang Han, Hongkai Ji, **Zhicheng Ji** and Honglang Wang. (2017) A provable smoothing approach for high dimensional generalized regression with applications in genomics. *Electronic Journal of Statistics*. 11(2):4347-4403

12. Justin Guinney et al. (2017) Prediction of overall survival for patients with metastatic castration-resistant prostate cancer: development of a prognostic model through a crowdsourced challenge with open clinical trial data. *The Lancet Oncology*. 18(1):132-142 (Listed as consortium coauthor)

13. **Zhicheng Ji***, Weiqiang Zhou* and Hongkai Ji. (2017) Single-cell regulome data analysis by SCRAT. *Bioinformatics*. 33(18):2930-2932

14. Qiang Li, Rachel K. Lex, HaeWon Chung, Simone M. Giovanetti, **Zhicheng Ji**, Hongkai Ji, Maria D. Person, Jonghwan Kim and Steven A. Vokes. (2016) The pluripotency factor NANOG binds to GLI proteins and represses Hedgehog-mediated transcription. *Journal of Biological Chemistry*, 291(13):7171-82

15. Jacqueline L. Norrie, Qiang Li, Swanie Co, Bau-Lin Huang, Susan Mackem, Ding Ding, **Zhicheng Ji**, Mark T. Bedford, Antonella Galli, Hongkai Ji and Steven A. Vokes. (2016) PRMT5 is necessary to form distinct cartilage identities in the knee and long bone. *Development*. 143(24):4608-4619

16. **Zhicheng Ji** and Hongkai Ji. (2016) TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research*. 44(13): e117
    Winner of ASA Section on Statistics in Genomics and Genetics Distinguished Student Paper Award
    220+ Citations on Google Scholar

17. Xiumei Hong, Christine Ladd-Acosta, Ke Hao, Ben Sherwood, Hongkai Ji, Corinne A. Keet, Rajesh Kumar, Deanna Caruso, Xin Liu, Guoying Wang, Zhu Chen, Yuelong Ji, Guanyun Mao, Sheila Ohlsson Walker, Tami R. Bartell, **Zhicheng Ji**, Yifei Sun, Hui-Ju Tsai, Jacqueline A. Pongracic, Daniel E. Weeks and Xiaobin Wang. (2016) Epigenome-wide association study links site-specific DNA methylation changes with cow's milk allergy. *The Journal of Allergy and Clinical Immunology*, 138(3):908-911.e9

18. Guoying Wang, Frank B. Hu, Kamila B. Mistry, Cuilin Zhang, Fazheng Ren, Yong Huo, David Paige, Tami Bartell, Xiumei Hong, Deanna Caruso, **Zhicheng Ji**, Zhu Chen, Yuelong Ji, Colleen Pearson, Hongkai Ji, Barry Zuckerman, Tina L. Cheng and Xiaobin Wang. (2016) Association between maternal prepregnancy body mass index and plasma folate concentrations with child metabolic health. *JAMA Pediatrics*. 170(8): e160845

19. Detian Deng, Yu Du, **Zhicheng Ji**, Karthik Rao, Zhenke Wu, Yuxin Zhu and Yates Coley. (2016) Predicting survival time for metastatic castration resistant prostate cancer: An iterative imputation approach. *F1000research.* 5:2672

20. **Zhicheng Ji**, Steven A. Vokes, Chi V. Dang and Hongkai Ji. (2015) Turning publicly available gene expression data into discoveries using gene set context analysis. *Nucleic Acids Research*, 44(1): e8

Book Chapters

21. **Zhicheng Ji** and Hongkai Ji. (2019) Pseudotime reconstruction using TSCAN. *Computational Methods for Single-Cell Data Analysis*, 115-124. Springer

22. Jiajia Zhang, **Zhicheng Ji** and Kellie Smith. (2019) Analysis of TCR β CDR3 sequencing data for tracking anti-tumor immunity. *Methods in Enzymology.* Elsevier.

Software

Methods for analyzing single-cell genomic data

- SCATE: Single-cell ATAC-seq signal extraction and enhancement [Github]
- TSCAN: Pseudo-time reconstruction in single-cell RNA-seq analysis [Bioconductor] [Github] [GUI]
- BIRD: Big data regression for predicting DNase I hypersensitivity [Github]
- SCRAT: Single-cell regulome analysis tool [Github] [GUI]
- STIP: State transition inference prediction [Github]
- iXplore: Reproducible interactive data exploration tool [GUI]
- SEPA: Single-cell gene expression pattern analysis [Bioconductor] [Github] [GUI]

- SIMEX: Single-cell immune profiling and gene expression [GUI]

Methods for analyzing large-scale multi-modal genomic data

- GSCA: Gene set context analysis [Bioconductor] [Github] [GUI]
- BIRD: Big data regression for predicting DNase I hypersensitivity [Github]
- DynaMO: Dynamic motif occupancy analysis [Github]
- GEOsearch: Extendable search engine for gene expression omnibus [Bioconductor] [Github] [GUI]

Methods for high-dimensional statistics

- RMRCE: Regularized maximum rank correlation estimator [Github]

Software designed for teaching

- Statistics toolbox [Apple App Store]
- Graderanalytics [GUI]

Teaching

- Guest lecturer, Statistics in Genomics. 2018, 2019, 2020
- Lead teaching assistant and lab instructor, Statistical Methods in Public Health. 2016-2019
- Teaching assistant, Statistical Methods in Public Health. 2014, 2015

Editorial Activities

Journal Referee

- Bioinformatics
- Statistics in Biosciences

Presentations

Contributed Talks

- Single-cell ATAC-seq signal extraction and enhancement with SCATE. *ENAR*, March, 2020
- Reproducible interactive data visualization and exploration with iXplore. *ENAR*, March, 2017
- Reproducible interactive data visualization and exploration with iXplore. *The 10th International Chinese Statistical Association International Conference*, December, 2016
- TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Joint Statistical Meeting*, August, 2016

Posters

- Single-cell ATAC-seq signal extraction and enhancement with SCATE. *RECOMB/ISCB Conference on Regulatory & Systems Genomics*, December, 2018
- Turning publicly available gene expression data into discoveries using gene set context analysis. *The American Society of Human Genetics Annual Meeting*, October, 2015

- Turning publicly available gene expression data into discoveries using gene set context analysis. *International Genetic Epidemiology Society Annual Meeting*, October, 2015