Title
Gene Selection Methods for Single-cell Sequencing Data Analysis

Author
Li, Kexin

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Gene Selection Methods for Single-cell Sequencing Data Analysis

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Kexin Li

2023

ABSTRACT OF THE DISSERTATION

Gene Selection Methods for Single-cell Sequencing Data Analysis

by

Kexin Li
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2023
Professor Jingyi Li, Chair

Since the advent of single-cell RNA sequencing (scRNA-seq) technologies around 15 years ago, they have become a powerful tool to characterize cell-to-cell heterogeneity within a cell population in various biological systems, and have revolutionized transcriptomic studies. A typical scRNA-seq dataset contains thousands to tens of thousands of genes; however, a subset of genes are usually sufficient for representing the underlying biological variations of cells that are aligned with researchers' various interest. The sufficiency can be explained by three reasons: (1) highlighting and enhancing biological signals, (2) improving the interpretability of analysis results, and (3) reducing the number of genes to save computational or human resources. Hence, a number of gene selection methods have been performed in various tasks, for instance, informative gene selection for cell clustering and post-clustering differentially expressed gene identification for cell type annotation. However, existing efforts have not fully addressed the problems: among the genes selected by the existing methods, many are irrelevant, redundant, or insignificant. Gene selection for certain single-cell analysis tasks with biological meaningful interpretation and statistical rigor remains challenging. This dissertation aims to address them in two projects.

My first project focuses on the informative gene selection in general scRNA-seq data analysis, and extends an application to guide targeted gene profiling design. Unlike scRNA-seq, targeted gene profiling has a strong requirement for a limited number (often no more than hundreds) of genes to be specified before sequencing. In Chapter 2, we propose the

single-cell Projective Non-negative Matrix Factorization (scPNMF) method, which leverages the PNMF algorithm and adds a unique feature of basis selection. scPNMF outperforms existing informative gene selection methods in that its selected, limited number of genes better distinguish cell types, and it enables the alignment of new targeted gene profiling data with reference data in a low-dimensional space to facilitate the prediction of cell types in the new data.

My second project discusses post-clustering differentially expressed (DE) gene identification for cell-type annotation tasks. Here the selected genes serve as potential cell-type marker genes, by matching with the canonical ones, they are crucial in determining the cell types in single-cell sequencing data. Despite the popularity of the typical two-step analysis workflow: first, clustering; second, finding DE genes between cell clusters, an issue known as "double dipping"–the same data is used twice to define cell clusters and find DE genes–exists here and leads to false-positive cell-type marker genes when the cell clusters are spurious. To overcome this challenge, in Chapter 3, we propose ClusterDE, a post-clustering DE method for controlling the false discovery rate (FDR) of identified DE genes regardless of clustering quality, which can work as an add-on to popular pipelines such as Seurat. The core idea of ClusterDE is to generate real-data-based synthetic null data containing only one cluster, as contrast to the real data, for evaluating the whole procedure of clustering followed by a DE test. ClusterDE is fast, transparent, and adaptive to a wide range of clustering algorithms and DE tests. Besides scRNA-seq data, ClusterDE is generally applicable to post-clustering DE analysis, including single-cell multi-omics data analysis.

The dissertation of Kexin Li is approved.

Wei Li

Chad J. Hazlett

Mark S. Handcock

Jingyi Li, Committee Chair

University of California, Los Angeles

2023

To my parents and my cat Riesling

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Many people have been integral to my doctoral journey. I am profoundly grateful for their guidance, encouragement, and unwavering support.

Foremost on this list is my advisor, Dr. Jingyi Jessica Li. Her unending passion for pursuing science, deep dive into analytical details, and open mind for exploring this rapidly changing world have been instrumental in shaping my research interests. I am deeply grateful for her tremendous support and boundless patience throughout my studies at UCLA; without them, I cannot go this far. Beyond academia, her attributes will continue to indelibly influence my future career and life.

I would like to extend my appreciation to Dr. Mark S. Handcock, Dr. Chad Hazlett, Dr. Wei Li, and Dr. Janet Sinsheimer for their roles on my doctoral committee and consistently valuable insights on my research. Among them, the passing of Dr. Janet Sinsheimer has created an irreplaceable void, yet her contributions remain cherished and unforgettable.

I am grateful to all current and previous members of the JSB lab for their assistance and stimulating discussions, particularly Dongyuan Song and Dr. Xinzhou Ge. They are not only collaborators but also my best friends. Their ingenuity, sagacity, and diligence have encouraged me in the past and will continue to do so in the future. I wish them the best of luck with their academic careers ahead and any future endeavors.

Last but not least, I sincerely thank my friends and family for their unconditional love and unwavering companionship throughout this demanding journey. My dear friends stood by me every step of the way, offering comfort and inspiration. Above all, my heartfelt gratitude goes to my parents and my cat Riesling. Your belief in me, coupled with your emotional support, has been and will continue to be my bedrock.

VITA

2014–2018	B.S. in Pure and Applied Mathematics, Department of Mathematical Science, Tsinghua University.
2018–2021	Teaching Assistant, Department of Statistics, University of California, Los Angeles.
2021–2023	Graduate Student Researcher, Department of Statistics, University of California, Los Angeles.

PUBLICATIONS

(* indicates equal contribution.)

Song, D.*, **Li, K.***, Hemminger, Z., Wollman, R., and Li, J.J. (2021). scPNMF: sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling. Bioinformatics 37 (Supp_1): i358-i366.

Ma, W., Kim, S., Chowdhury, S., Li, Z., Yang, M., Yoo, S., Petralia, F., Jacobsen, J., Li, J.J., Ge, X., **Li, K.**, and et al.. (2021). DreamAI: algorithm for the imputation of proteomics data. bioRxiv, pp.2020-07.

Song, D.*, **Li, K.***, Ge, X., and Li, J.J. (2023). ClusterDE: a post-clustering differential expression (DE) method robust to false-positive inflation caused by double dipping. bioRxiv.

CHAPTER 1

Introduction

Single-cell RNA sequencing (scRNA-seq) technologies have enabled gene expression measurement at an unprecedented single-cell resolution, and opened a new frontier to studying cell-to-cell heterogeneity and differentiation trajectories in various biological systems [1], which has led to important scientific discoveries over the years [2, 3]. Moreover, single-cell targeted gene profiling technologies, which we define to include all technologies that measure only a specific set of genes' expression levels in individual cells, are gaining popularity for their low costs, high sensitivity, and extra information. Examples of targeted gene profiling include smFISH [4] and MERFISH [5], which can capture spatial information, BART-Seq [6] that has a lower cost per cell, and HyPR-seq [7] that exhibits a higher sensitivity for detecting lowly expressed genes.

Rapid advances in single-cell sequencing technologies have resulted in thousands of computational methods developed for various tasks in the field [8], where major tasks like cell clustering and cell type annotations are usually performed. The state-of-the-art scRNA-seq analysis pipelines such as the R package `Seurat` [9] and the Python module `Scanpy` [10] both involve these tasks as essential intermediate steps. Starting from the very beginning, a typical scRNA-seq dataset can be viewed as an RNA molecule count matrix, in which various numbers of cells are the observational units, and the thousands to tens of thousands of genes are the features. As the number of genes/features is large, the gene selection methods are conventionally adopted with different criteria in different tasks.

In the cell clustering tasks, researchers usually first select a subset of genes, which we call informative genes, that are sufficient for representing the underlying biological variations of cells. This procedure can be justified in two ways: first, variations of many genes are

not related to the biological variations of interest [11, 12]; second, many genes have strongly correlated expression levels, suggesting that one gene may represent a group of genes without much loss of information [13]. Practically, gene selection is recognized as aiding downstream analysis by highlighting the biological signals in scRNA-seq datasets [9, 14]. Besides scRNA-seq data analysis, informative gene selection is also crucial for designing single-cell targeted gene profiling experiments, in which a limited number (often no more than hundreds) of genes are to be specified before sequencing. However, by inspecting existing popular methods, we found a few key limitations: (1) They are all designed to select a relatively large number of genes, for example, 2000 in Seurat [9] and 700-900 in SCMarker [15]. Thus, their performance in selecting a small number of genes in targeted gene profiling remains unclear. (2) Their selected genes lack functional interpretability.

Another major task in scRNA-seq data analysis is to annotate cell types and understand their biological differences. In practice, the standard workflow includes two steps: (1) cell clustering to find potential cell types, and (2) finding differentially expressed (DE) genes between cell clusters as potential cell-type marker genes. Although this post-clustering differential expression (DE) procedure is widely adopted in the single-cell field, researchers have realized that this procedure is conceptually problematic. For instance, Seurat [9] contains the warning message that "P values should be interpreted cautiously, as the genes used for clustering are the same genes tested for differential expression." This issue is commonly referred to as "double dipping," meaning that the same gene expression data are used twice, once to define cell clusters and once to identify DE genes, thus leading to an inflated false discovery rate (FDR) in identifying post-clustering DE genes as putative cell-type marker genes when the cell clusters are spurious.

We find a few instrumental drawbacks or lack of statistical rigor in the current single-cell sequencing data analysis conventions by diving deep into the gene selection methods developed for various tasks. Among the genes selected by the existing methods, many are irrelevant, redundant, or insignificant to the biological variations under investigation. Therefore, we are aiming to get clear what genes researchers are really interested in, propose new approaches to fill in the gap, and provide users with biological meaningful interpretation

and statistical rigor guarantee.

This dissertation will focus on the selection of the above two types of genes. For selecting the informative genes, we start with a clear motivation–to facilitate gene selection for targeted gene profiling design, and proposed scPNMF. For identifying the cell-type marker genes, we proposed ClusterDE, a post-clustering DE method robust to inflated FDR issues due to double dipping, even when the cell clusters are spurious.

1.1 scPNMF: sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling

The first part of my dissertation focuses on informative gene selection. The procedure is widely performed in scRNA-seq data analysis ahead of common tasks such as clustering. We summarize the advantages of informative gene selection into three: (1) enhancing biological signals by removing unwanted technical variations, (2) improving the interpretability of analysis results by focusing on informative genes, and (3) reducing the number of genes to save computational resources.

Besides scRNA-seq data analysis, informative gene selection is also crucial for designing single-cell targeted gene profiling experiments, where only a specific set of genes' expression levels in individual cells. Unlike scRNA-seq, targeted gene profiling requires a limited number (often no more than hundreds) of genes to be specified before sequencing. However, it remains an open and challenging question to optimize the gene selection for targeted gene profiling under a gene number limitation.

In Chapter 2, we propose scPNMF, an unsupervised method to select informative genes from scRNA-seq data, and can possibly guide targeted gene profiling experimental design. Leveraging the Projective Non-negative Matrix Factorization (PNMF) algorithm [16], scPNMF outputs a non-negative sparse weight matrix that can project cells in a high-dimensional scRNA-seq dataset onto a low-dimensional space, which corresponds to bases that each correspond to a group of co-expressed genes. Compared with the original PNMF,

a unique feature of scPNMF is basis selection: scPNMF uses correlation screening and multimodality testing to remove the bases that cannot reveal potential cell clusters in the scRNA-seq dataset. Two major advantages of scPNMF over the existing gene selection methods are: First, its selected, limited number of informative genes can better distinguish cell types. Second, it enables the alignment of new targeted gene profiling data with reference data in a low-dimensional space to facilitate the prediction of cell types in the new data. Comprehensive benchmark studies demonstrate that scPNMF outperforms existing gene selection methods in cell clustering and cell type prediction accuracy for targeted gene profiling data.

1.2 ClusterDE: a post-clustering differential expression method robust to false-positive inflation caused by double dipping

The second part of my dissertation focuses on the identification of post-clustering differentially expressed (DE) genes as potential cell-type markers. This is a key step in a major task in scRNA-seq data analysis–cell type annotation. Typically, a clustering algorithm is applied to find putative cell types as clusters, and then a statistical differential expression (DE) test is employed to identify the differentially expressed (DE) genes between the cell clusters.

Although this procedure is popular, more and more researchers have realized that this procedure is conceptually problematic, and referred to this issue as "double dipping", meaning that the same gene expression data are used twice to define cell clusters and DE genes. An extreme scenario to demonstrate the invalidity of "double dipping" is where only a single cell type exists, and no genes should be identified as between-cell-type DE genes. However, as clustering is based on gene expression data, certain genes would be correlated with the resulting cell clusters if their expression patterns drive the clustering. Hence, these genes would have different conditional distributions in the two cell clusters and subsequently be identified as between-cell-cluster DE genes, but they are false-positive between-cell-type DE genes and further false putative cell-type marker genes. Therefore, this double dipping is-

sue would inflate the false discovery rate (FDR), the expected proportion of false-positive between-cell-type DE genes among all identified DE genes.

Two attempts to solve the double-dipping issue include the truncated normal (TN) test [17] and the Countsplit method [18]. Despite their claims to achieve the well-calibrated P values, our findings indicate that their P values are anti-conservative in the presence of gene-gene correlations, a real scRNA-seq data feature these methods have not considered. As a result, the P value calibration issue would lead to inflated FDRs when the TN test and Countsplit are applied to real scRNA-seq data. Another category of methods developed to circumvent the double-dipping issue is the cluster-free DE tests that try to bypass the cell clustering step [19–24]. However, it is important to note that these methods do not aim to identify potential cell types, and the identified DE genes cannot be interpreted as marker genes for specific cell types. Thus, the cluster-free DE tests are not comparable with the DE genes identified post-clustering. Another stream of methods has been developed to assess the quality of clustering results, e.g., the "purity" of a cluster or if two clusters should be merged [25–29]. However, these methods do not provide a direct statistical test for identifying DE genes, and it remains difficult to determine the threshold for clustering quality above which double dipping is not a concern.

Motivated by this, we focus on addressing the original inflated FDR issue when using post-clustering DE genes as cell-type marker genes. In Chapter 3, we propose ClusterDE, a post-clustering DE method for controlling the false discovery rate (FDR) of identified DE genes regardless of clustering quality, which can work as an add-on to popular pipelines such as Seurat. The core idea of ClusterDE is to generate real-data-based synthetic null data containing only one cluster, as contrast to the real data, for evaluating the whole procedure of clustering followed by a DE test. Using comprehensive simulation and real data analysis, we show that ClusterDE has not only solid FDR control but also the ability to identify cell-type marker genes as top DE genes and distinguish them from housekeeping genes. ClusterDE is fast, transparent, and adaptive to a wide range of clustering algorithms and DE tests.

1.3 Summary

During my doctoral study, I developed the aforementioned two statistical methods that aim to select certain genes aligned with researchers' interests in single-cell sequencing data analysis. The details of these projects will be described in Chapter 2–3 of this dissertation.

In single-cell sequencing data analysis, gene selection problem with different focuses has always been a fun topic; moreover, the correlation and the comparison between the different definitions of "interesting genes", conceptually and practically, are a future direction to research into. Hopefully, it will provide researchers in this field with a more accurate and interpretable tool to decipher gene functions and find proper interesting genes.

CHAPTER 2

scPNMF: sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling

2.1 Introduction

The recent development of single-cell RNA sequencing (scRNA-seq) technologies provides unprecedented opportunities to decipher transcriptome heterogeneity among individual cells [2, 30, 31]. A typical scRNA-seq dataset contains thousands to tens of thousands of genes; however, a subset of genes, which we call **informative genes**, are usually sufficient for representing the underlying biological variations of cells in the dataset for two reasons. First, variations of many genes are not related to the biological variations of interest. For instance, fluctuations in the expression levels of housekeeping genes are irrelevant to cell types [11, 12]. Second, many genes have strongly correlated expression levels, suggesting that one gene may represent a group of genes without much loss of information [13]. Therefore, for scRNA-seq data analysis, informative gene selection has three advantages: (1) enhancing biological signals by removing unwanted technical variations, (2) improving the interpretability of analysis results by focusing on informative genes, and (3) reducing the number of genes to save computational resources.

Besides scRNA-seq data analysis, informative gene selection is also crucial for designing single-cell targeted gene profiling experiments, which we define to include all technologies that measure only a specific set of genes' expression levels in individual cells. Unlike scRNA-seq, targeted gene profiling requires a limited number (often no more than hundreds) of genes to be specified before sequencing. Examples of targeted gene profiling include spatial technologies (e.g., smFISH [4] and MERFISH [5]) and non-spatial technologies (e.g., BART-Seq

[6], HyPR-seq [7] and 10x-Genomics Targeted Gene Expression). Compared with scRNA-seq, targeted gene profiling technologies have advantages such as capturing spatial information (by smFISH and MERFISH), having a lower cost per cell (by BART-Seq), and exhibiting a higher sensitivity for detecting lowly expressed genes (by HyPR-seq). However, it remains an open and challenging question to optimize the gene selection for targeted gene profiling under a gene number limitation.

Given the importance of informative gene selection, researchers have developed many gene selection methods for scRNA-seq data. Most existing methods select genes based on the relationship between per-gene expression means and per-gene expression variances (with the mean and the variance of each gene calculated across cells). Popular example methods include variance stabilization transformation (vst) [32] and mean-variance plot (mvp) in the R package `Seurat` [33], as well as modelGeneVar in the R package `scran` [34]. These methods select highly variable genes (HVG) that have large expression variances in relation to their expression means. Other methods use various metrics of gene importance instead of the per-gene expression variance. For example, M3Drop selects the genes that have zero expression levels in many cells [35]; GiniClust selects the genes with large Gini indices of expression levels [36]; SCMarker selects the genes that have expression levels bi/multi-modally distributed and are co-expressed or mutually-exclusively expressed with some other genes [15]. A common limitation of these existing methods is that they are all designed to select a relatively large number of genes; thus, their performance in selecting a small number of genes remains unclear. For instance, in Seurat, the default gene number is 2000; SCMarker selects 700-900 genes in its exemplar applications [15]. All these gene numbers are much greater than 200, the maximum gene number allowed by multiple targeted gene profiling technologies. Therefore, existing gene selection methods may not be suitable for selecting genes for targeted gene profiling. Another drawback of these methods is that their selected genes lack functional interpretability; that is, their selected genes are not categorized as functional gene groups.

In addition to these gene selection methods, linear dimensionality reduction methods, such as principal component analysis (PCA) and non-negative matrix factorization (NMF),

can also be used for gene selection. Specifically, genes can be selected based on their contributions to the projected low dimensions found by PCA or NMF [37–39]. Although many variants of PCA and NMF algorithms have been developed for scRNA-seq data analysis, they are not designed for gene selection [40–46].

Here we propose an unsupervised method, scPNMF, to simultaneously select informative genes and project scRNA-seq data onto an interpretable low-dimensional space. Leveraging the Projective Non-negative Matrix Factorization (PNMF) algorithm [16], scPNMF combines the advantages of PCA and NMF by outputting a non-negative sparse weight matrix that can project cells in a high-dimensional scRNA-seq dataset onto a low-dimensional space. Unlike the weight matrix (a.k.a., loading matrix) found by PCA, the non-negative sparse weight matrix output by scPNMF involves bases that each correspond to a group of co-expressed genes. Compared with the original PNMF, a unique feature of scPNMF is basis selection: scPNMF uses correlation screening and multimodality testing to remove the bases that cannot reveal potential cell clusters in the input scRNA-seq dataset. There are two functionalities of scPNMF: (1) given a pre-specified gene number and a scRNA-seq dataset, scPNMF selects informative genes based on its weight matrix; (2) given a targeted gene profiling dataset containing the informative genes, scPNMF projects this dataset onto the same low-dimensional space of a reference scRNA-seq dataset containing cell type labels, thus enabling cell type annotation on the targeted gene profiling dataset. Comprehensive benchmark shows that scPNMF outperforms existing gene selection methods in two aspects. First, the informative genes selected by scPNMF lead to the most accurate cell clustering. Second, the informative genes and weight matrix of scPNMF lead to the best cell type prediction accuracy for targeted gene profiling data. Therefore, scPNMF is a powerful gene selection method that can guide the experimental design and data analysis of single-cell targeted gene profiling.

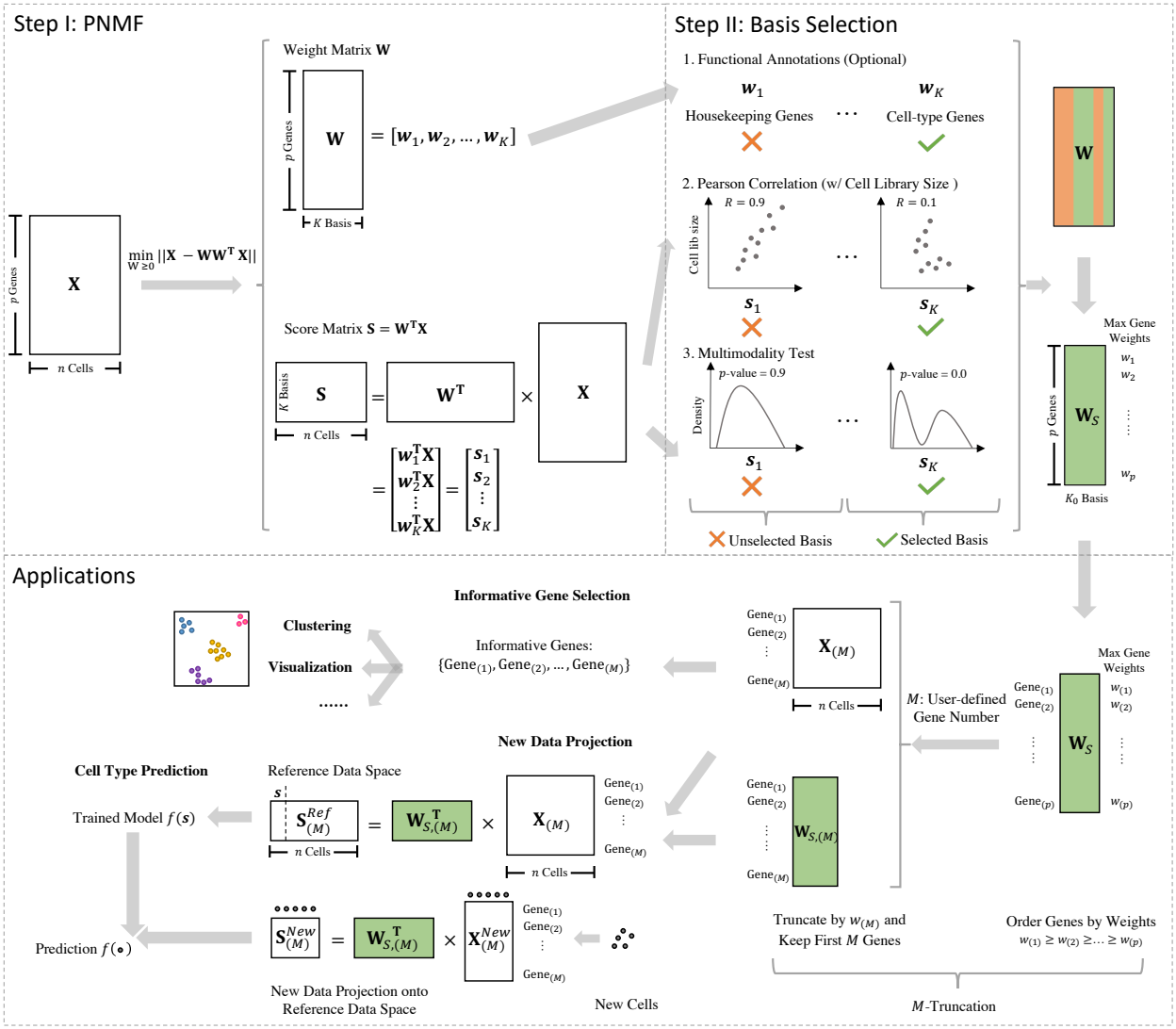


Figure 2.1: An overview of scPNMF.

Taking a log-transformed gene-by-cell count matrix as the input, scPNMF first learns a low-dimensional sparse weight matrix **W** and a low-dimensional cell embedding matrix **S**. Second, it removes the bases irrelevant to cell type variations by examining bases' functional annotations (optional), Pearson correlations with cell library sizes, and multimodality. Given a user-defined gene number M, scPNMF performs M-truncation to facilitate two main applications: (1) selecting the desired number of informative genes; (2) projecting new targeted gene profiling data onto the low-dimensional space defined by reference scRNA-seq data. The details are in the "scPNMF methodology" section.

2.2 scPNMF methodology

The core of scPNMF is to learn a low-dimensional embedding of cells so that the bases of the low-dimensional space correspond to sparse and mutually exclusive gene groups, and that genes in each group are co-expressed and thus functionally related. Fig. 2.1 illustrates the overall workflow of scPNMF. The input of scPNMF is a log-transformed gene-by-cell

count matrix measured by scRNA-seq. There are two main steps in scPNMF: (I) it learns a low-dimensional sparse weight matrix by PNMF; (II) it selects bases in the weight matrix based on functional annotations (optional), correlation screening, and multimodality testing to remove uninformative bases that cannot distinguish cell types. The output of scPNMF includes (1) the selected weight matrix, a sparse and mutually exclusive encoding of genes as new, low dimensions, and (2) the score matrix containing embeddings of input cells in the low dimensions. The selected weight matrix has two main applications: (1) extracting informative gene for downstream analyses, such as cell clustering and new marker gene identification, and (2) projecting new targeted gene profiling data for data integration and cell type annotation.

2.2.1 scPNMF step I: PNMF

In this section, we review the PNMF algorithm [16, 47] as the foundation of scPNMF. We first compare the formulation of PNMF with that of principal component analysis (PCA) and non-negative matrix factorization (NMF), and we show that PNMF has the advantages of both PCA and NMF so that it can be a useful tool for scRNA-seq data analysis. Next, we introduce our PNMF implementation.

Given a log-transformed count matrix $\mathbf{X} \in \mathbb{R}_{\geqslant 0}^{p \times n}$, whose p rows correspond to genes and whose n columns represent cells, and a positive integer $K \leqslant p$, PNMF aims to find a K-dimensional space, whose dimensions correspond to non-negative, sparse and mutually exclusive linear combinations of the p genes, so that projecting the n cells onto the K-dimensional space does not cause much information loss (i.e., projecting the K-dimensional embeddings of the n cells back to the original p-dimensional space can largely restore the original n cells). PNMF tackles this task by solving the optimization problem:

$$\min_{\mathbf{W} \in \mathbb{R}_{\geqslant 0}^{p \times K}} \left\| \mathbf{X} - \mathbf{W}\mathbf{W}^{\mathsf{T}}\mathbf{X} \right\|, \tag{2.1}$$

where $\| \cdot \|$ denotes the Frobenius matrix norm. The solution $\mathbf{W}$ is referred to as a **weight matrix**. Each column of $\mathbf{W}$ is a **basis**, whose p entries are the weights of the p genes.

PNMF requires all weights to be non-negative, leading to a sparse $\mathbf{W}$ with most weights as zeros.

PCA is similar to PNMF but does not require all weights to be non-negative. We can write the optimization problem of PCA as

$$\min_{\mathbf{W}\in\mathbb{R}^{p\times K},\mathbf{W}^\intercal\mathbf{W}=\mathbf{I}} \|\mathbf{X}-\mathbf{W}\mathbf{W}^\intercal\mathbf{X}\|, \tag{2.2}$$

whose solution $\mathbf{W}$ is also a weight matrix but not sparse, and $\mathbf{W}$ is often referred to as the loading matrix.

A common property of PNMF and PCA is that the transpose of their weight matrix, $\mathbf{W}^\intercal \in \mathbb{R}^{K\times p}$, can be used to project a new cell with p gene measurements, $\mathbf{x}\in\mathbb{R}^p$, onto the K-dimensional space as $\mathbf{W}^\intercal\mathbf{x}$.

In contrast to PMNF and PCA, NMF finds two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ so that their product approximates the original matrix $\mathbf{X}$. NMF solves the optimization problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{p\times K}_{\geqslant 0},\mathbf{H}\in\mathbb{R}^{K\times n}_{\geqslant 0}} \|\mathbf{X}-\mathbf{W}\mathbf{H}\|, \tag{2.3}$$

whose solution $\mathbf{W}$ still has K columns representing bases, and $\mathbf{H}$ has n columns as K-dimensional embeddings of the n cells. Due to the non-negative constraint on $\mathbf{W}$ and $\mathbf{H}$, $\mathbf{W}$ is a sparse matrix [48]. However, the transpose $\mathbf{W}^\intercal$ cannot be used as a projection matrix from the original p-dimensional space to a K-dimensional space. The reason is that, if $\mathbf{W}^\intercal$ is a projection matrix, then by the definition of $\mathbf{H}$ we have $\mathbf{W}^\intercal\mathbf{X}=\mathbf{H}$, which would convert the objective function (2.3) of NMF to the objective function (2.1) of PNMF. In other words, PNMF is a constrained version of NMF by requiring $\mathbf{W}^\intercal$ to be a projection matrix. Hence, PNMF inherits the property of NMF by having non-negative, sparse bases that are mostly mutually exclusive (i.e., different bases correspond to different gene groups). Moreover, based on the similarities of the objective functions of PNMF (2.1) and PCA (2.2), we can see that PNMF also resembles PCA by finding a weight matrix whose transpose can serve as a projection matrix and whose bases are largely orthogonal to each other. Table 2.1

summarizes the properties of PNMF, PCA, and NMF.

Table 2.1: Comparison of the properties of PNMF, PCA and NMF

	PNMF	PCA	NMF
Optimization Problem	$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{X}\|$ s.t. $\mathbf{W} \geqslant 0$	$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{X}\|$ s.t. $\mathbf{W}^\mathsf{T}\mathbf{W} = \mathbf{I}$	$\min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|$ s.t. $\mathbf{W}, \mathbf{H} \geqslant 0$
Non-negativity	Yes	No	Yes
Sparsity	Very high	Low	High
Mutually Exclusiveness	Very high	Low	High
New Data Projection	Yes	Yes	No

In the context of scRNA-seq data analysis, the above advantages of PNMF lead to an interpretable and useful weight matrix $\mathbf{W}$. First, the high sparsity of $\mathbf{W}$ makes each basis (column) depend on only a small set of genes, which has been defined as a **meta-gene** for NMF [49]. Second, the mutual exclusiveness of $\mathbf{W}$ makes different bases correspond to different gene sets, easing the interpretation of bases as meta-genes or functional units. Third, the projection matrix $\mathbf{W}^\mathsf{T}$ allows the alignment of new data to reference data, thus facilitating cell type annotation on the new data.

Algorithm 1 Pseudocode of PNMF implementation in scPNMF

Initialize: $\mathbf{W} = \text{abs}(\mathbf{W}_{\text{PCA}}) \in \mathbb{R}^{p\times K}_{\geqslant 0}$

while not converge **do**

for $i = 1, \cdots, p$; $k = 1, \cdots, K$ **do**

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{2\left(\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{W}\right)_{ik}}{\left(\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{W}\right)_{ik} + \left(\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{W}\right)_{ik}}$$

$$\mathbf{W} \leftarrow \frac{1}{\|\mathbf{W}\|_2}\mathbf{W}$$

Output: $\mathbf{W} \in \mathbb{R}^{p\times K}_{\geqslant 0}$, $\mathbf{S} = \mathbf{W}^\mathsf{T}\mathbf{X} \in \mathbb{R}^{K\times n}_{\geqslant 0}$

Algorithm 1 summarizes the key steps of PNMF implementation in scPNMF. Our implementation mainly follows the two papers that proposed the PNMF algorithm [16, 47], and we change the initialization of $\mathbf{W}$ to the weight matrix found by PCA, $\mathbf{W}_{\text{PCA}}$, with the absolute value taken on every entry. Our initialization is motivated by the desired orthogonality of bases (i.e., columns of $\mathbf{W}$).

With the weight matrix $\mathbf{W} \in \mathbb{R}^{p\times K}_{\geqslant 0}$ learned by PNMF, we obtain the **score matrix** $\mathbf{S} = \mathbf{W}^\mathsf{T}\mathbf{X} \in \mathbb{R}^{K\times n}_{\geqslant 0}$, whose K rows correspond to the bases and whose n columns represent

the cells. Specifically, the j-th column of $\mathbf{S}$ is the K-dimensional embedding of the j-th cell; the k-th row of $\mathbf{S}$, denoted by $\boldsymbol{s}_k^\mathsf{T}$, contains the *scores* (i.e., coordinates) of all n cells in the k-th basis:

$$\boldsymbol{s}_k = \boldsymbol{w}_k^\mathsf{T}\mathbf{X}\,, \tag{2.4}$$

where $\boldsymbol{w}_k$ is the k-th column of $\mathbf{W}$, $k = 1, \ldots, K$.

The low rank K needs to be pre-specified in PNMF, same as in PCA and NMF, A larger K preserves more information in $\mathbf{X}$ but also removes less noise (technical variation of cells that is not of biological interest), impedes the interpretation of $\mathbf{W}$ (more bases are more difficult to interpret), and increases the computational burden. To choose K in a data-driven way, we propose an orthogonality measure, which shows that $K = 20$ is a reasonable choice for multiple scRNA-seq datasets (section S2.7.1.1).

2.2.2 scPNMF step II: basis selection

The second key step of scPNMF is to select informative bases among the K bases found by PNMF (i.e., columns of $\mathbf{W}$ and rows of $\mathbf{S}$) to remove unwanted variations of cells (e.g., variations irrelevant to cell types). The columns of $\mathbf{W}$ enjoy high sparsity and mutual exclusiveness; that is, each column contains positive weights corresponding to a unique small set of genes, so it is expected to reflect a certain biological function. However, some biological functions may not be relevant to the cell heterogeneity of interest, e.g., cell type composition. Motivated by this, we propose three strategies for selecting informative bases (columns of $\mathbf{W}$ and rows of $\mathbf{S}$): functional annotations (optional), correlations with cell library sizes, and tests of multimodality.

2.2.2.1 Strategy 1: examine bases by functional annotations (optional)

The first, optional strategy is to annotate the biological function(s) of each basis in the weight matrix. For example, scPNMF may apply gene ontology (GO) analysis to the top 10% genes with the highest weights in each basis (column of $\mathbf{W}$) and record the enriched GO terms as the basis' functional annotation. Then, users with prior knowledge can interpret

the functional annotation on each basis and decide whether or not to remove the basis. For example, if the goal is to delineate cell types in scRNA-seq data, a basis corresponding to cell-cycle genes should be removed because they would obscure the distinction of cell types.

However, it is worth noting that filtering bases by biological annotations is optional in scPNMF. Conservative users can keep all K bases output by PNMF and directly use data-driven basis selection (section 2.2.2.2). For our results in this paper, scPNMF removes the bases corresponding to well-known housekeeping genes (section S2.7.2).

2.2.2.2 Data-driven strategies

2.2.2.2.1 Strategy 2: examine bases by correlations with cell library sizes

Note that the input of scPNMF is a log-transformed unnormalized count matrix for users' convenience. Hence, scPNMF does not adjust for cell library sizes in the computation of $\mathbf{W}$ and $\mathbf{S}$ in step I. (For a detailed discussion on why scPNMF uses unnormalized data as input, see section S2.7.6.) Given that the variance of cell library sizes contributes to unwanted variations of cells [32], it is necessary to remove the bases whose corresponding rows in $\mathbf{S}$ are strongly correlated with cell library sizes.

We use the total log-transformed counts to approximate the library size of each cell, and we calculate the Pearson correlation between each $\boldsymbol{s}_k$ and the library sizes of n cells. The strategy is to retain the bases whose Pearson correlations are under a pre-defined threshold, which we set to 0.7 based on empirical observations (section S2.7.1.2).

2.2.2.2.2 Strategy 3: examine bases by multimodality tests

Another data-driven strategy is to retain the bases whose corresponding scores are multimodally distributed. If a basis' score vector (row in $\mathbf{S}$) contains n scores with a multimodality pattern, then it is likely to distinguish cell types and should be retained. To implement this strategy, we use the ACR test [50] to check the multimodality of each basis' score vector. The null hypothesis is that the score vector contains n scores sampled from a unimodal

distribution, and the alternative hypothesis is that the distribution has more than one mode. After performing multiple multimodality tests, one per basis, we use the Benjamini-Hochberg procedure to set a P value threshold by controlling the false discovery rate under 1%. The bases whose P values are under this threshold will be retained.

In summary, scPNMF step II allows users to use strategy 1 to filter out uninformative bases based on functional annotations if available; then it implements data-driven strategies 2 and 3 to further remove bases that have strong correlations with cell library sizes and exhibit unimodality patterns. The retained bases will have their corresponding columns in $\mathbf{W}$ selected and stacked into the *selected weight matrix* $\mathbf{W}_S \in \mathbb{R}^{p\times K_0}_{\geqslant 0}$, where K_0 is the number of selected bases.

2.2.3 Applications of scPNMF output: informative gene selection and new data projection

The selected weight matrix $\mathbf{W}_S$ output by scPNMF has two main applications: selection of a desired number of informative genes and projection of new targeted gene profiling data onto the low-dimensional space defined by $\mathbf{W}_S$. Given a gene number M (e.g., 200), scPNMF uses M-truncation, a step to select M rows in $\mathbf{W}_S$, resulting in M informative genes and a **truncated, selected weight matrix** $\mathbf{W}_{S,(M)} \in \mathbb{R}^{M\times K_0}_{\geqslant 0}$ for new data projection.

2.2.3.1 M-truncation and informative gene selection

We denote the desired number of informative genes by $M \in \mathbb{N}$, with $M \leqslant \#$ of non-zero rows in $\mathbf{W}_S$. M-truncation has three steps.

1. For each gene i, calculate its largest weight w_i across bases in $\mathbf{W}_S$:

$$w_i = \max_{k=1,\ldots,K_0} (\mathbf{W}_S)_{ik}, \qquad i = 1, 2, \ldots, p\,. \tag{2.5}$$

2. Order genes by their maximum weights $w_{(1)} \geqslant w_{(2)} \geqslant \cdots \geqslant w_{(p)}$ and set the truncation threshold as $w_{(M)}$. Identify the first M genes as **informative genes**.

3. Construct the truncated, selected weight matrix $\mathbf{W}_{S,(M)}$:

   (1) Truncate the selected weight matrix $\mathbf{W}_S$ by setting all $(\mathbf{W}_S)_{ik} < w_{(M)}$ to be 0;

   (2) Keep the M rows with non-zero entries; stack them by row into $\mathbf{W}_{S,(M)}$ based on the order of the informative genes.

In short, scPNMF selects informative genes based on their maximum weights in the selected bases. The rationale is that a gene's maximum weight reflects the gene's contribution to the establishment of the K_0-dimensional space, which preserves the n cells' biological variations of interest. Hence, genes with larger maximum weights are more informative in the sense of encoding cells' biological variations. An important application of informative gene selection is to guide the design of targeted gene profiling experiments.

2.2.3.2 New data projection

Given the selected M informative genes, once new cells are measured by targeted gene profiling on these genes, $\mathbf{W}_{S,(M)}$ can be used to project the new cells onto the K_0-dimensional space where the cells in the input scRNA-seq data are embedded in. If the input data has cell type annotations, we refer to the input data as **reference data**, then we can predict the new cells' types from the types of the cells in the reference data. In detail, new data projection has the following steps:

1. Apply scPNMF with M-truncation to input, reference data $\mathbf{X} \in \mathbb{R}_{\geqslant 0}^{p\times n}$ with n cells to obtain the truncated, selected weight matrix $\mathbf{W}_{S,(M)}$. Construct $\mathbf{X}_{(M)} \in \mathbb{R}_{\geqslant 0}^{M\times n}$ as a submatrix of $\mathbf{X}$, with rows corresponding to the rows of $\mathbf{W}_{S,(M)}$, i.e., the M informative genes. Hence, the K_0-dimensional embeddings of the n cells in the reference data are the columns of

$$\mathbf{S}_{(M)}^{\text{Ref}} = \mathbf{W}_{S,(M)}^{\mathsf{T}} \times \mathbf{X}_{(M)} \in \mathbb{R}^{K_0\times n}\,. \tag{2.6}$$

2. Denote the targeted gene profiling data of n' new cells with M informative genes measured by $\mathbf{X}_{(M)}^{\text{New}} \in \mathbb{R}_{\geqslant 0}^{M\times n'}$. Note that $\mathbf{X}_{(M)}^{\text{New}}$ contains log-transformed counts and

has rows (genes) corresponding to the rows of $\mathbf{X}_{(M)}$. Project the n' cells to the K_0-dimensional space by

$$\mathbf{S}^{\text{New}}_{(M)} = \mathbf{W}^{\mathsf{T}}_{S,(M)} \times \mathbf{X}^{\text{New}}_{(M)} \in \mathbb{R}^{K_0 \times n'}. \tag{2.7}$$

3. (Optional) Normalize $\mathbf{S}^{\text{New}}_{(M)}$ and $\mathbf{S}^{\text{Ref}}_{(M)}$ to remove batch effects, if existent, by using a single-cell integration method such as Harmony [51].

Now the n reference cells and the n' new cells are in the same K_0-dimensional space with biological variations preserved. Then a classifier can be trained on the n reference cells' types and $\mathbf{S}^{\text{Ref}}_{(M)}$ for cell type prediction, and it can be used to predict the new n' cells' types from $\mathbf{S}^{\text{New}}_{(M)}$.

2.3 Results

2.3.1 scPNMF outputs a sparse and functionally interpretable representation of scRNA-seq data

We first demonstrate that scPNMF step I, PNMF, outputs a sparse and functionally interpretable gene encoding of cells. We use the FregGold dataset [52], which consists of three cell types (three human lung adenocarcinoma cell lines), and set the basis number $K = 5$ for demonstration purposes. Both PCA and PNMF learn a weight matrix that can project the original scRNA-seq data onto a 5-dimensional space. Unlike the weight matrix of PCA that has no zero entries, the weight matrix of PNMF is non-negative, highly sparse, containing 42.6% of entries as zeros, and has bases that are largely mutually exclusive (i.e., non-zero entries in different columns correspond to different rows/genes) (Fig. 2.2a). Compared with NMF, PNMF also has greater sparsity and mutual exclusiveness in bases (section S2.7.7). GO enrichment analysis shows that high weight genes in each PNMF basis are enriched with conceptually-similar GO terms, and high weight genes in different PNMF bases are enriched with conceptually-different GO terms (Fig. 2.2b). This result indicates that PNMF bases correspond to gene groups with distinct functions. On the contrary, the PCA bases do not have good functional interpretations: the high weight genes in each PCA basis are

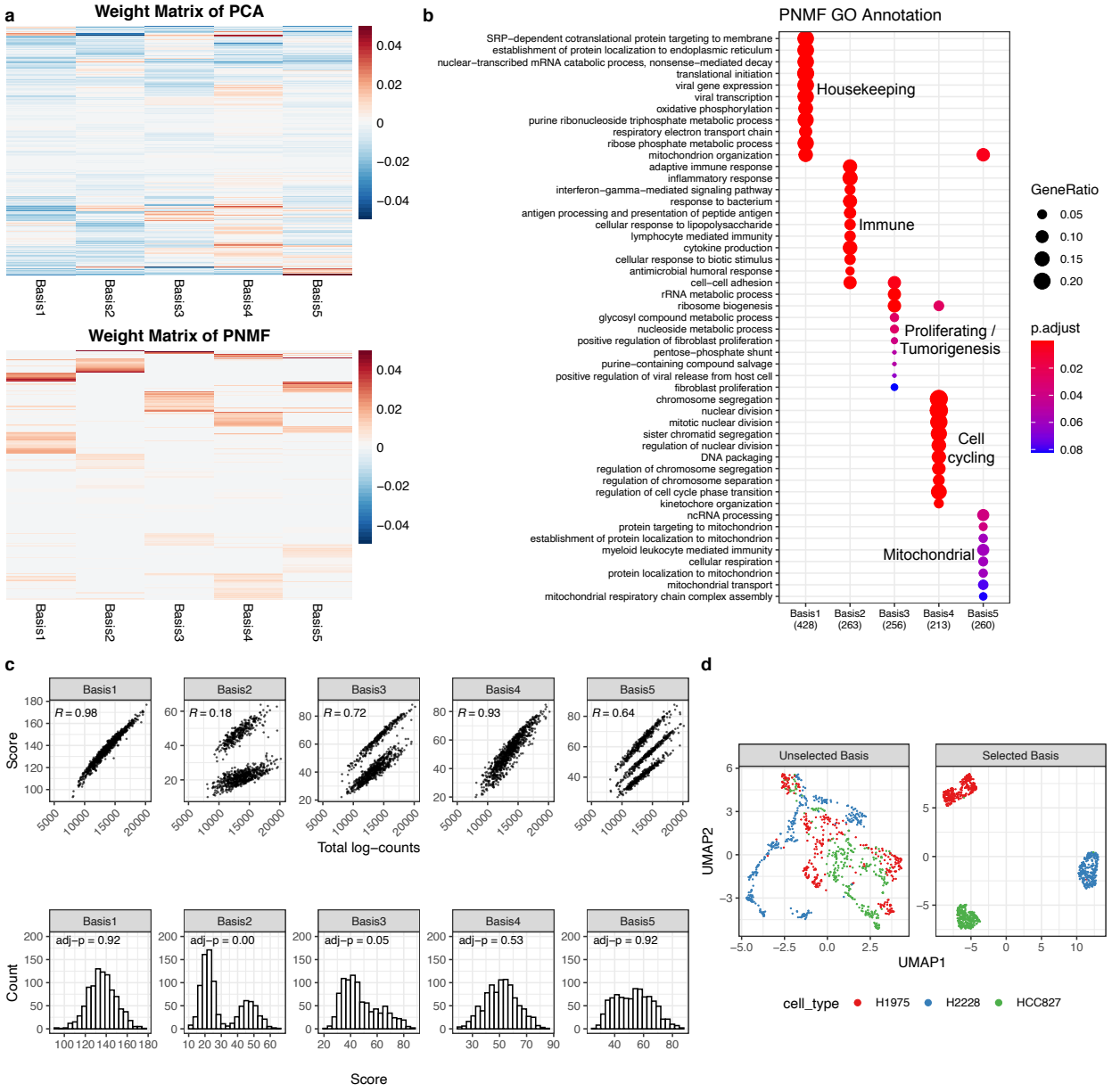


Figure 2.2: Illustration of the sparse and interpretable projection found by scPNMF.

We use the FregGold dataset as an example. (a) Comparison of the weight matrices of PCA and PNMF. Heatmaps visualize the learned weight matrices of PCA (top) and PNMF (bottom), where rows are genes and columns are bases. Red represents positive weights while blue represents negative weights. The rows are ordered by gene-wise hierarchical clustering. Compared to PCA, the weight matrix of PNMF is strictly non-negative, much more sparse and mutually exclusive between bases. (b) GO analysis result of each basis in the weight matrix of PNMF. Texts in black boxes summarize the functions of genes in each basis. The enriched GO terms are almost mutually exclusive, implying that each basis represents a unique gene functional cluster. (c) Statistical tests on each basis in the score matrix of PNMF. Top row: scatter plots of scores and total log-counts (cell library sizes). Each dot represents a cell. Cell scores in bases 1 and 4 are highly correlated with cell library sizes. Bottom row: histograms of cell scores in each basis. Scores in bases 2 and 3 show strong multimodality patterns (adjusted P value $\leqslant 0.05$). (d) UMAP visualizations of cells based on high weight genes in the unselected bases 1 and 4 and those in the selected bases 2, 3, and 5. Genes in the unselected bases completely fail to distinguish the three cell types, while genes in the selected bases lead to a clear separation of the three cell types.

not enriched with conceptually-similar GO terms, and different PCA bases share many high weight genes (Fig. 2.11).

To further analyze the PNMF bases, we list the top 10 high weight genes in each basis (Table S2.4), from which we identify many well-known genes with important functions. For instance, basis 1 contains classic housekeeping genes, such as *GAPDH* [53] and ribosomal protein genes (*RP-*) [54]; basis 3 contains well-known tumor-related genes, including *EGFR* [55] and *CDK4* [56]. In particular, the cells of the HCC827 cell line (one of the three cell types) have overall high scores in basis 3 (Fig. 2.12), a reasonable result because the HCC827 cell line contains an *EGFR* activating mutation [57]. In summary, scPNMF step I outputs bases representing sparse and functionally interpretable gene sets.

2.3.2 Basis selection is an essential step in scPNMF

Here we explain why basis selection is an essential step in scPNMF. In the last section, we show that each PNMF basis of the FregGold dataset approximately represents one functional gene group. It is well known that housekeeping genes (basis 1) and cell-cycle genes (basis 4) are usually irrelevant to cell type distinctions. However, such biological knowledge is not always available or certain. Therefore, scPNMF mainly relies on the two data-driven strategies: correlations with cell library sizes and multimodality tests (section 2.2.2.2) for selecting informative bases.

Fig. 2.2c visualizes the two strategies: cell scores in bases 1 and 4 are highly correlated with cell library sizes (Pearson correlations > 0.9); cell scores in bases 2 and 3 show strong evidence as multi-modally distributed (adjusted$P$value < 0.05). Hence, strategy 1 will not retain bases 1 and 4, and strategy 2 will not retain bases 1, 4, and 5; together, bases 1 and 4 will be removed, and bases 2, 3, and 5 will be selected. To verify the effectiveness of basis selection, we use UMAP to visualize cells based on the top 50 high weight genes in the unselected bases 1 and 4 vs. those in the selected bases 2, 3, and 5 (Fig. 2.2d). We observe that the top genes in the unselected bases completely fail to separate the three cell types, while the top genes in the selected bases perfectly distinguish the three cell types.

This result strongly supports that basis selection is a necessary step of scPNMF. If cell type labels are provided, users may use a strategy alternative to "correlations with cell library sizes" by regressing out the cell library sizes in a cell-type-specific manner from every basis (section S2.7.6).

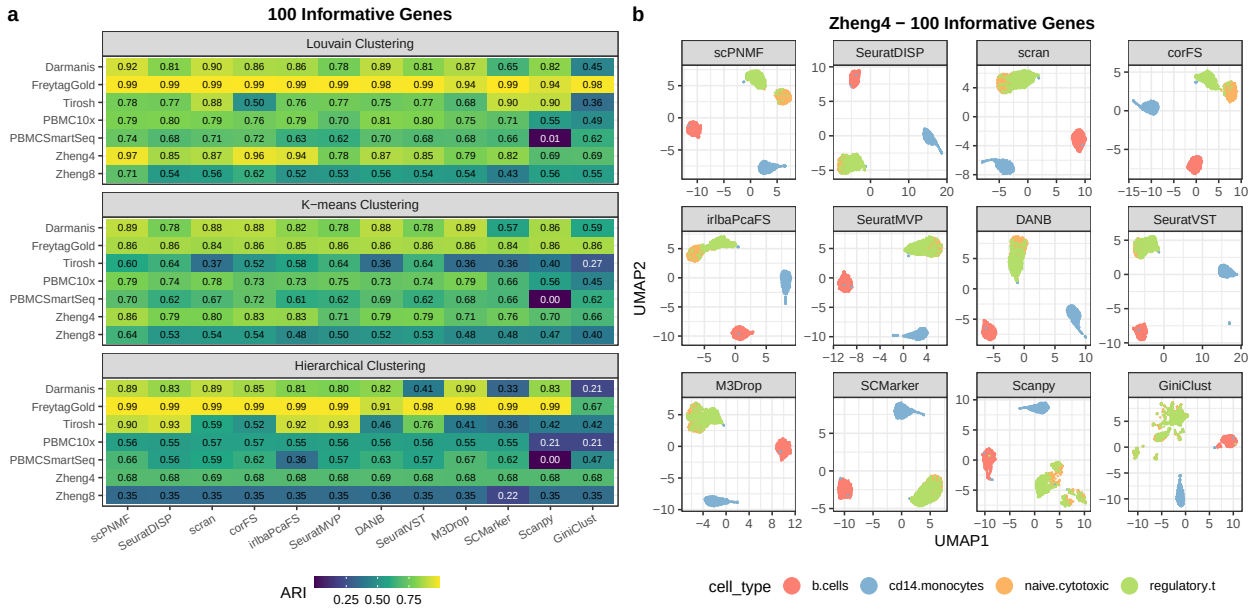


Figure 2.3: Benchmarking scPNMF against 11 informative gene selection methods on seven scRNA-seq datasets.

(a) Clustering accuracies (ARI values) of three clustering methods based on the informative genes selected. Gene selection methods are ordered from left to right by their average ARI across the three clustering methods and the seven datasets. (b) UMAP visualization of cells in the Zheng4 dataset based on 100 informative genes selected by each method. Genes selected by scPNMF lead to a clear separation between naive cytotoxic T cells and regulatory T cells, while the genes selected by other methods do not.

2.3.3 scPNMF outperforms state-of-the-art gene-selection methods on diverse scRNA-seq datasets

In this section, we demonstrate scPNMF's capacity for informative gene selection. We comprehensively benchmark scPNMF against 11 other single-cell informative selection methods (Table S2.5) on seven scRNA-seq datasets (Table S2.3) using three clustering methods (Louvain clustering, K-means clustering, and hierarchical clustering). For fair benchmarking, the seven scRNA-seq datasets cover both unique molecule identifier (UMI) and non-UMI protocols and include various biological samples. Using the adjusted Rank index (ARI) as the metric of clustering accuracy, we calculate the ARI values of the three clustering methods

on each dataset using 100 informative genes selected by each gene selection method, as 100 genes are commonly used in targeted gene profiling.

Fig. 2.3a shows that scPNMF has overall the highest ARI values across datasets and clustering methods. In particular, scPNMF has the highest average ARI value with each clustering method (Louvain: 0.83; K-means: 0.74; hierarchical clustering: 0.69) and the highest overall average ARI (0.75) across datasets and clustering methods. Note that the mean of the overall average ARI values of all methods except scPNMF is only 0.66.

We further show the UMAP visualization of cells in the Zheng4 dataset based on the informative genes selected by each of the 12 gene selection methods (Fig. 2.3b). Only scPNMF leads to a clear separation of naive cytotoxic T cells and regulatory T cells, while the informative genes selected by other methods except corFS and irlbaPcaFS cannot distinguish the two cell types at all.

We also compare the 12 methods under a varying number of informative genes: 20, 50, 200, and 500, the commonly used gene numbers in targeted gene profiling. We observe that the overall average ARI values of scPNMF are consistently higher than those of other methods, across all informative gene numbers (Fig. 2.13 and Fig. 2.14). We apply the same benchmarking framework to scPNMF and its variant, where PNMF is replaced by NMF, and find that scPNMF performs consistently better (section S2.7.7). Moreover, compared with other methods, scPNMF leads to more stable overall average ARI values under varying numbers of informative genes, indicating its stronger robustness to the gene number constraint of targeted gene profiling. These results strongly support the superior performance of scPNMF as an informative gene selection method.

2.3.4 scPNMF guides targeted gene profiling experimental design and cell-type prediction

In this section, we demonstrate how scPNMF can guide the selection of genes to be measured in a targeted gene profiling experiment, and how scPNMF enables subsequent cell type annotation on the targeted gene profiling data. We design two case studies with paired scRNA-seq

reference data and "pseudo" targeted gene profiling data, whose per-cell sequencing depth is higher than that of the corresponding scRNA-seq data.

In the first case study, we use the Zheng8 dataset (measured by the 10x protocol) as the reference dataset. To generate the pseudo targeted gene profiling data, we use a new single-cell gene expression simulator that captures gene correlations, scDesign2 [58], to generate data with a 100-time higher per-cell sequencing depth. In the second case study, we use the PBMC10x dataset (measured by 10x protocol) as the reference dataset, and we use PBMCSmartseq (measured by Smart-Seq2) as the pseudo targeted gene profiling data because Smart-Seq2 has a higher per-gene sequencing depth than 10x does. In both case studies, for each gene selection method, the corresponding pseudo targeted gene profiling datasets only contain the M informative genes selected by the method.

We benchmark scPNMF against the 11 gene selection methods in terms of cell type prediction on the pseudo targeted gene profiling data. To avoid the bias for a specific classification algorithm, we apply three popular algorithms for cell type prediction: random forest [59], k-nearest neighbors (KNN) [60], and support vector machine (SVM) [60]. In each case study, we first train each classification algorithm on the low-dimensional embeddings of the reference cells $\mathbf{S}^{\text{Ref}}_{(M)}$ given the $M = 100$ informative genes selected by each gene selection method. Then we apply the trained classifier to the low-dimensional embeddings of the cells in the pseudo targeted gene profiling data $\mathbf{S}^{\text{New}}_{(M)}$. Table 2.2 shows that scPNMF leads to the highest average prediction accuracy (0.81) across six combinations (two case studies $\times$ three classification algorithms). Moreover, scPNMF achieves the highest accuracy in each combination except Zheng8 + random forest where it is the second best. These results confirm that scPNMF effectively guides the selection of genes to measure in targeted gene profiling experiments, and it enables accurate cell type annotation on newly generated targeted gene profiling datasets.

Table 2.2: Prediction accuracy of cell types based on 100 informative genes selected by 12 gene selection methods in the two case studies with paired reference scRNA-seq data and targeted gene profiling data

Method	Zheng8			PBMC			Average Accuracy
	RandomForest	KNN	SVM	RandomForest	KNN	SVM	
scPNMF	0.85 (0.83,0.87)	<u>0.80</u> (0.78,0.83)	<u>0.87</u> (0.85,0.89)	<u>0.84</u> (0.79,0.88)	<u>0.84</u> (0.79,0.88)	<u>0.67</u> (0.61,0.73)	<u>0.81</u>
M3Drop	0.85 (0.83,0.87)	<u>0.80</u> (0.77,0.83)	<u>0.87</u> (0.84,0.89)	<u>0.84</u> (0.79,0.88)	0.77 (0.71,0.82)	0.63 (0.57,0.69)	0.79
SeuratDISP	0.84 (0.81,0.86)	0.78 (0.75,0.81)	0.86 (0.84,0.88)	0.80 (0.75,0.84)	0.75 (0.70,0.80)	0.64 (0.58,0.70)	0.78
corFS	0.80 (0.77,0.82)	0.75 (0.73,0.78)	0.82 (0.80,0.85)	0.82 (0.77,0.86)	0.81 (0.76,0.86)	0.62 (0.56,0.68)	0.77
GiniClust	<u>0.86</u> (0.83,0.88)	0.79 (0.76,0.81)	0.86 (0.83,0.88)	0.80 (0.75,0.84)	0.76 (0.71,0.81)	0.53 (0.47,0.60)	0.75
scran	0.79 (0.76,0.81)	0.72 (0.69,0.75)	0.82 (0.80,0.85)	0.78 (0.72,0.82)	0.73 (0.67,0.78)	0.67 (0.61,0.72)	0.75
SeuratMVP	0.83 (0.81,0.85)	0.77 (0.74,0.80)	0.85 (0.82,0.87)	0.82 (0.77,0.86)	0.74 (0.69,0.79)	0.47 (0.40,0.53)	0.74
Scanpy	0.79 (0.77,0.82)	0.71 (0.68,0.74)	0.80 (0.78,0.83)	0.80 (0.75,0.84)	0.76 (0.71,0.81)	0.52 (0.46,0.58)	0.73
SCMarker	0.77 (0.74,0.79)	0.68 (0.65,0.71)	0.74 (0.71,0.77)	0.77 (0.71,0.81)	0.71 (0.65,0.76)	0.45 (0.39,0.52)	0.69
SeuratVST	0.73 (0.70,0.76)	0.68 (0.65,0.71)	0.75 (0.73,0.78)	0.74 (0.68,0.79)	0.68 (0.63,0.74)	0.40 (0.34,0.46)	0.67
DANB	0.71 (0.68,0.73)	0.69 (0.66,0.71)	0.75 (0.73,0.78)	0.73 (0.67,0.78)	0.74 (0.68,0.79)	0.28 (0.23,0.34)	0.65
irlbaPcaFS	0.68 (0.65,0.71)	0.61 (0.58,0.64)	0.71 (0.68,0.74)	0.71 (0.65,0.76)	0.77 (0.71,0.82)	0.16 (0.12,0.21)	0.61

Parentheses are 95% confidence intervals. Highest number within each column is labeled by underline.

2.4 Discussion

We propose scPNMF, an unsupervised gene selection and data projection method for scRNA-seq data. The major goal of scPNMF is to select a fixed number of informative genes to distinguish cell types and guide gene selection for targeted gene profiling experiments. Moreover, scPNMF can project a new targeted gene profiling dataset with the selected genes to the low-dimensional space that embeds a reference scRNA-seq dataset. We perform a comprehensive benchmark to evaluate scPNMF in terms of informative gene selection against the state-of-the-art gene selection methods. Our results show that scPNMF consistently outperforms 11 existing methods for a wide range of informative gene numbers (from 20 to 500) on diverse scRNA-seq datasets. We also demonstrate that the informative genes selected by scPNMF can effectively guide gene selection for targeted gene profiling and lead to accurate cell type annotation on targeted gene profiling data based on reference scRNA-seq data. In addition to the 11 methods, we compare scPNMF to the factorial single-cell latent variable model (f-scLVM) [61], both conceptually and empirically, to clarify their differences

and further illustrate the unique strength of scPNMF (see section S2.7.8).

Besides gene selection and data projection, scPNMF also works as a dimensionality reduction method with good interpretability. Each dimension in the low-dimensional space found by scPNMF can be considered as a new functional "feature" (as a linear combination of correlated and thus functionally related genes). Moreover, the mutual exclusiveness makes the PNMF bases used in scPNMF advantageous over the PCA bases in terms of removing confounding effects. For example, cell-cycle genes obscure the identification of cell types and should be removed from low-dimensional embeddings of cells. For PCA, cell-cycle genes affect many PCA bases, so the popular scRNA-seq pipeline Seurat implements a complicated approach that first calculates "cell-cycle scores" and then regresses each basis (principal component) on these scores to remove the effects of cell-cycle genes [33]. In contrast, cell-cycle genes are concentrated in only one PNMF basis, so it is easy to remove that basis to clear the effects of cell-cycle genes. Therefore, scPNMF has great potential in deciphering cell heterogeneity in single-cell data by working as an interpretable dimensionality reduction method.

The current implementation of scPNMF focuses on single-cell gene expression data. Considering the rapid development of single-cell multi-omics technologies, we plan to extend scPNMF to accommodate other technologies that measure other genomics features such chromatin accessibility landscapes measured by single-cell ATAC-seq [62], or even to integrate data across multi-omics datasets. Another note is that the multimodality test for basis selection in scPNMF only accounts for discrete cell types but not continuous cell trajectories. Therefore, other tests or strategies are needed to select informative bases to capture biological variations along continuous cell trajectories.

An important question for gene selection is: how many genes should be selected as informative genes to fully capture the biological variations of interest? In our studies, we observe that, after the informative gene number reaches 200, the clustering accuracies based on the selected informative genes plateau for most gene selection methods including scPNMF. Therefore, 200 genes may be sufficient for capturing biological variations in scRNA-seq data. However, it remains challenging to decide the minimum number of informative genes, given

that the underlying cell sub-population structure is data-specific and might be complex. We plan to explore this problem in the future with the possible use of information theory.

2.5 Code and data availability

The R package and the tutorials of scPNMF are available at https://github.com/JSB-UCLA/scPNMF. The source code and data for reproducing the results are available at: https://doi.org/10.5281/zenodo.4797997 [63].

2.6 Acknowledgments

This chapter is based on my joint work with Dongyuan Song, Dr. Zachary Hemminger, Dr. Roy Wollman, and my Ph.D. advisor Dr. Jingyi Jessica Li. Thanks Dongyuan for bringing up this idea of leveraging the PNMF algorithm for the gene selection task during his rotation at Dr. Roy Wollman's lab. For the methodology development, Dongyuan and I made equal contributions. For the results, I contributed to all the benchmark studies for all informative gene selection methods on diverse scRNA-seq datasets. Dongyuan contributed to the results of cell-type prediction in targeted gene filing data, and the biological interpretations.

2.7 Supplementary materials

S2.7.1 Choice of parameters and robustness analysis

S2.7.1.1 Low rank K

In the development of scPNMF, motivated by the objective function of the PNMF method,

$$\min_{\mathbf{W}\in\mathbb{R}_{\geqslant 0}^{p\times K}} \|\mathbf{X}-\mathbf{W}\mathbf{W}^{\mathsf{T}}\mathbf{X}\|, \tag{S2.8}$$

PNMF aims to inherit the advantages of PCA such as the basis orthogonality and the ability to project new data. However, a key constraint in PCA, $\mathbf{W}^{\mathsf{T}}\mathbf{W}=\mathbf{I}$, is relaxed to satisfy the

constraint $\mathbf{W} \geqslant 0$ in PNMF. To make PNMF closer to PCA and thus approximately achieve these two nice properties, we propose to use the normalized difference between $\mathbf{W}^\mathsf{T}\mathbf{W}$ and $\mathbf{I}$ to measure the orthogonality of $\mathbf{W}$:

$$dev.ortho = \|\mathbf{I} - \mathbf{W}^\mathsf{T}\mathbf{W}\|/K^2\,. \tag{S2.9}$$

It naturally gives rise to a method to determine the number of bases, K: first perform PNMF for a sequence of K's; second, for each K, we calculate the *dev.ortho* measure for the corresponding $\mathbf{W} \in \mathbb{R}^{p\times K}_{\geqslant 0}$; third, we plot *dev.ortho* against K. Users can decide K when *dev.ortho* reaches stability or there is a clear elbow in the graph.

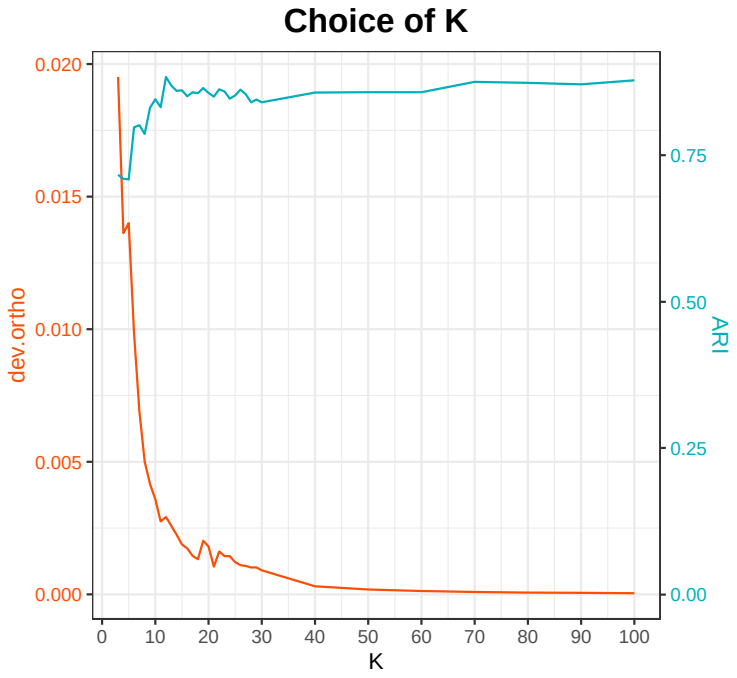


Figure 2.4: Comparison of *dev.ortho* and K-means ARI against low rank K on Zheng4 [1] dataset.

In Fig. 2.4, using the Zheng4 [1] dataset, we demonstrate that (1) the *dev.ortho* measure is highly correlated with the performance of $\mathbf{W}$ in the downstream analysis; (2) in real data application, the *dev.ortho* measure shows a clear elbow pattern, which can help users determine K.

Empirically, we see that *dev.ortho* reaches stability at $K = 20$ for most scRNA-seq data.

For the purpose of guiding users and saving computational time, we set the default number of bases in scPNMF to be $K = 20$.

S2.7.1.2 R_0: threshold for correlations between score vectors and cell library sizes in "scPNMF step II: basis selection"

In real data applications, the threshold for correlations between score vectors and cell library sizes in "scPNMF step II: basis selection," R_0, needs to be pre-defined. We consider thresholds with one decimal digit resolution $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ because of the convention in the field. By running the K-means clustering on the seven datasets (see Table S2.3) and applying these thresholds, as shown in Fig. 2.5, we suggest setting $R_0 = 0.7$ for $K \geqslant 10$, and more conservatively, $R_0 = 0.8$ when the basis number K is small ($K < 10$).

Table S2.3: Overview of datasets used in this study

Dataset	Sequencing protocol	Gene #	Cell #	Cell type #	True label	Description	Ref
Darmanis	Smart-Seq2	13256	420	8	No	Human adult cortical samples	[2]
FreytagGold	10xGenomics Chromium	15410	925	3	Yes	Mixture of human lung adenocarcinoma cell lines	[3]
Tirosh	Smart-Seq2	11934	2887	6	No	Human melanoma tumors	[4]
PBMC10x	10xGenomics Chromium	11714	3308	9	No	Human peripheral blood mononuclear cells. 10x-v2 for sample 1 in the original paper.	[5]
PBMCSmartSeq	Smart-Seq2	17479	273	6	No	Human peripheral blood mononuclear cells. Smart-Seq2 for sample 1 in the original paper.	[5]
Zheng4	10xGenomics GemCode	2192	3994	4	Yes	Mixture of human peripheral blood mononuclear cells	[1, 6]
Zheng8	10xGenomics GemCode	2390	3994	8	Yes	Mixture of human peripheral blood mononuclear cells	[1, 6]

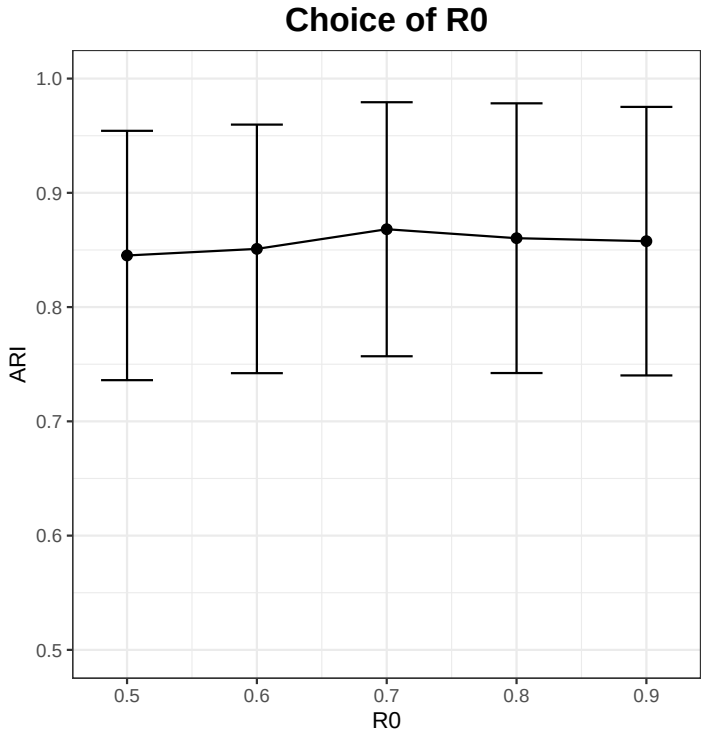


Figure 2.5: Comparison of K-means ARI against R_0, the threshold for correlations between score vectors and cell library sizes in scPNMF step II: basis selection. The mean ARI and the error bars are calculated across seven datasets (See Table S2.3).

S2.7.2 Functional annotation

We use the R package `clusterProfiler` [7] to perform the GO analysis. We set the gene ontology as "biological processes (BP)" and the adjusted P value cutoff as 0.1. The output GO terms are simplified by `clusterProfiler`.

In this paper, we only perform a very conservative filtering based on functionality. We define the common housekeeping gene list to include *ACTB*, *ACTG1*, *B2M*, *GAPDH*, and *MALAT1*. If a basis' top 10 highly weighted genes contain any of these five genes, this basis will be filtered out.

S2.7.3 Data preprocessing

scPNMF only performs minimum data preprocessing to avoid information loss. Denote a scRNA-seq count matrix as $\mathbf{X}^C \in \mathbb{N}^{p\times n}$, with rows representing p genes and columns representing n cells. scPNMF creates the log count matrix $\mathbf{X} \in \mathbb{R}_{\geqslant 0}^{p\times n}$ by taking the log

transformation of $\mathbf{X}^C$ with a pseudo count of 1:

$$\mathbf{X}_{ij} = \log\left(\mathbf{X}^C_{ij} + 1\right), \qquad i = 1, \ldots, p;\ j = 1, \ldots, n. \tag{S2.10}$$

scPNMF takes the log count matrix $\mathbf{X} \in \mathbb{R}^{p\times n}_{\geqslant 0}$ as the input. With the log transformation, the effect of a few extremely large counts will be alleviated, and the transformed values will have more Gaussian-like distributions, a common assumption assumed by many methods. We introduce the pseudo count of 1 to avoid negative infinite values in the later PNMF optimization step.

For the scRNA-seq data used in this chapter (Table S2.4), we filter out the genes that are expressed in fewer than 5% of the cells, and then we filter out the cells that are expressed in fewer than 5% of the remaining genes. Additionally, *MALAT1*, mitochondrial genes, and ribosomal genes are removed from two datasets, PBMC10x and PBMCSmartSeq, according to the reference paper [5]. Users may customize the filtering process before they input the log count matrix $\mathbf{X}$ into scPNMF.

Table S2.4: Top 10 high weight genes in each PNMF basis of the FretagGold dataset

Basis	Gene symbol	Description
1	*RPS2, TMSB4X, GAPDH, RPL41, RPL13, FTH1, MALAT1, COX2, RPL10, RPS18*	Highly expressed housekeeping genes
2	*CD74, PTGR1, HLA-B, ALDH3A1, C15orf48, LCN2, IGFBP3, SAA1, CXCL1, HLA-DRA*	Immune-related genes
3	*SEC61G, CDK4, CCN1, G0S2, ELOC, VOPP1, EGFR, F3, CDKN2A, EPCAM*	Tumor-related genes (oncogenes, tumor suppressor genes)
4	*H4C3, CKS1B, HMGB2, SMC4, PTTG1, KPNA2, CCNB1, CDKN3, CKS2, CDC20*	Genes related to mitotic cell cycle
5	*HSPB1, UBE2S, CALD1, TMEM256, FIS1, ISOC2, ZNHIT1, C20orf27, NDUFA3, PPP2R1A*	Genes related to mitochondrion

S2.7.4 Details about informative gene selection and cell clustering

In this paper, we compare scPNMF with 11 other informative gene selection methods (Table S2.5). Some gene selection methods cannot let users pre-define an arbitrary gene number; for such methods (e.g., SCMarker [8]), we adjust their tuning parameters until their output gene numbers approximately equal the desired gene number.

Table S2.5: Overview of informative gene selection methods used in this study

Method	User-defined gene #	Language	Package	Reference
corFS	Yes	R	`M3Drop` (version 1.14.0)	[9]
DANB	Yes	R	`M3Drop` (version 1.14.0)	[9]
GiniClust	Yes	R	`M3Drop` (version 1.14.0)	[9]
irlbaPcaFS	Yes	R	`M3Drop` (version 1.14.0)	[9]
M3Drop	Yes	R	`M3Drop` (version 1.14.0)	[9, 10]
Scanpy	Yes	Python	`Scanpy` (version 1.6.0)	[11]
SCMarker	No	R	`SCMarker`[1]	[8]
scran	Yes	R	`scran` (version 1.18.3)	[12]
SeuratDISP	Yes	R	`Seurat` (version 3.2.2)	[13, 14]
SeuratMVP	No	R	`Seurat` (version 3.2.2)	[13]
SeuratVST	Yes	R	`Seurat` (version 3.2.2)	[13]
f-scLVM	No	R	`slalom` (version 1.10.0)	[15]

1: Due to failure in `SCMarker` R package installation, we run the R script downloaded from https://github.com/KChen-lab/SCMarker on September 17, 2020.

We apply three clustering algorithms, Louvain clustering (by Seurat), K-means clustering (by R function `kmeans`), and hierarchical clustering (by R function `hclust`). We perform PCA on informative genes and use the top 20 PCs for cell clustering. We use $U = \{u_1, \ldots, u_P\}$ to denote the true partition of P classes and $V = \{v_1, \ldots, v_K\}$ to denote the partition given by clustering results. Let n_i and n_j be the numbers of observations in class u_i and cluster v_j respectively, and n_{ij} denotes the number of observations in both class u_i and cluster v_j. The adjusted Rand index (ARI) is calculated as

$$\frac{\sum_{i=1}^{P}\sum_{j=1}^{K}\binom{n_{ij}}{2} - \left[\sum_{i=1}^{P}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K}\binom{n_{\cdot j}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{P}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{K}\binom{n_{\cdot j}}{2}\right] - \left[\sum_{i=1}^{P}\binom{n_{i\cdot}}{2}\sum_{j=1}^{K}\binom{n_{\cdot j}}{2}\right] / \binom{n}{2}} \tag{S2.11}$$

where $n = \sum_{i=1}^{P} n_{i\cdot} = \sum_{j=1}^{K} n_{\cdot j}$. An ARI value close to 1 means more accurately inferred clusters. Regarding the choice of tuning parameter values (the resolution parameter r in Louvain clustering and the number of clusters k in K-means and hierarchical clustering), we consider the following parameter values:

$$r \in \{0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}, \quad k \in \{2, 3, 4, \cdots, 15\}, \tag{S2.12}$$

and we use the average of the top three high ARI values (across the parameter combinations)

as the final output.

S2.7.5 Details about new data projection and cell type prediction

We use two datasets, Zheng8 and PBMC10x, as the reference scRNA-seq datasets. For the Zheng8 dataset, we first use scDesign2 [16] to learn the underlying parameters, and then we simulate a new dataset with the same genes and cell types but a 100-time larger sequencing depth compared to the Zheng8 dataset. For the PBMC10x dataset, we use the PBMCSmartSeq dataset, which measures the exact same example by Smart-seq2 and contains all genes measured in PBMC10x. Given M selected genes, the simulated Zheng8 and PBMC10x are pruned to contain only those genes, and the pruned datasets serve as the "pseudo" targeted gene profiling datasets that only have the M genes measured.

For cell type prediction, we project every targeted gene profiling dataset and its scRNA-seq reference onto the same low-dimensional space, which mainly follows the idea of scPred [17]. When applying scPNMF, we use the weight matrix $\mathbf{W}_{S,(M)}$ to project both the reference dataset and the targeted gene profiling dataset. For other gene selection methods, we first subset the reference dataset with only M selected genes, run PCA to obtain a weight matrix $\mathbf{W}_{\text{PCA}}$, and then use it to project both the reference dataset and targeted gene profiling dataset, both containing only M genes. After obtaining the two sets of low-dimensional embeddings of reference and targeted gene profiling datasets, we run the Harmony algorithm [18] to remove the technical variations between these two sets of low-dimensional embeddings. Then we apply three classification algorithms, random forest (`rf`), k-nearest neighbors (`knn`), and support vector machine with radial kernel (`svmRadial`) in the R package `caret` [19], for cell type prediction. The tuning parameters are selected by 5-fold cross-validation with three repeats.

S2.7.6 Data normalization by cell library size

S2.7.6.1 Why scPNMF does not use normalized data as input

By default, scPNMF takes the raw data without normalization (e.g., regressing out the cell library size [14]). In practice, scPNMF can be applied to such pre-processed scRNA-seq data, and then it does not need to remove the factors correlated with cell library size in its basis selection step. However, we have two reasons to prefer the default procedure:

1. Normalizing by cell library size is inappropriate for targeted gene profiling. In scRNA-seq, the cell library size is the total count in a cell. However, in targeted gene profiling, the cell sequencing depth can not be accurately estimated since only a small subset of genes is captured. For instance, `Seurat` claims that in the analysis of spatial data (a type of targeted gene profiling data), "force each data point to have the same underlying 'size' after normalization, can be problematic" [20]. To make sure that the genes selected based on scRNA-seq are informative for designing targeted gene profiling experiments, we prefer to use raw data without normalizing cell library sizes.

2. Cell library size can be informative for distinguishing cell types. Some studies have observed that cell library sizes are significantly different between some cell types and thus serve as a useful feature for distinguishing them [21, 22], a phenomenon we have also encountered in our data analysis. Therefore, normalization by cell library size is not always desirable. scPNMF avoids this issue by using unnormalized data, and if a factor is correlated with cell library size but also shows a significant multimodal pattern, the factor will be preserved for downstream analysis.

Besides, there are a few data-driven procedures in the scPNMF workflow. As the following steps do not require the common Gaussian assumptions, explicitly or implicitly, then applying the method to the raw count data without any variance-stabilizing transformations does not violate certain assumptions. Besides, empirical results show that using the raw count data as input for scPNMF has better performance.

S2.7.6.2 Normalization on the score matrix output by scPNMF

If users want to directly remove the effects of cell library size, they may choose the option of "regressing out cell library size by cell type" in the `scPNMF` package. The reason why we do not set it as the default is that scPNMF is designed as an unsupervised method **without cell type information**. Therefore, if cell type labels are not provided, it is impossible to regress out cell library size in a cell-type-specific manner. To overcome this no-cell-type-label issue, scPNMF applies a new clustering algorithm, "K-lines clustering" [23], to identify more than one linear relationship, if existent, between a basis and cell library size. Below we describe our algorithm.

In the **score matrix** $\mathbf{S} = \mathbf{W}^\mathsf{T}\mathbf{X} \in \mathbb{R}_{\geqslant 0}^{K\times n}$, whose K rows correspond to bases and whose n columns represent cells, the k-th row of $\mathbf{S}$, denoted by $\boldsymbol{s}_k^\mathsf{T}$, contains the **scores** (i.e., coordinates) of n cells in the k-th basis. For $\boldsymbol{s}_k^\mathsf{T}$, we assume that it is composed of cell groups $C_1, \ldots, C_{N_k}$, which correspond to either pre-defined cell types or clusters obtained by K-lines clustering. That is, $C_1 \cup \cdots \cup C_{N_k} = \{1, \ldots, n\}$. Therefore, for group C_r, $r = 1, \ldots, N_k$, we fit a linear model:

$$s_{ik} = \beta_{0k}^{(r)} + \beta_{1k}^{(r)} d_i + \epsilon_i, \quad i \in C_r\,,$$

where d_i is cell i's library size, and obtains estimates $\hat{\beta}_{0k}^{(r)}$ and $\hat{\beta}_{1k}^{(r)}$, as well as residuals e_i, $i \in C_r$. Then, we define the "corrected" score of cell i in group C_r as

$$u_{ik} = \hat{\beta}_{0k}^{(r)} + \hat{\beta}_{1k}^{(r)} \bar{d}^{(r)} + e_i, \quad i \in C_r\,,$$

where $\bar{d}^{(r)}$ is the mean cell library size in group C_r. The corrected score matrix $\mathbf{U}$ is used for downstream analysis such as dimensionality reduction. We observe that, using the corrected scores, the "stretching" shape within each cell type is removed, and cell types are better distinguished in UMAP visualization (Fig. 2.6).

Although this correction is useful, we argue that it should be used with caution since the results depend on cell type/cluster labels. In an unsupervised setting, we recommend users follow the basis selection criteria we described in our paper.

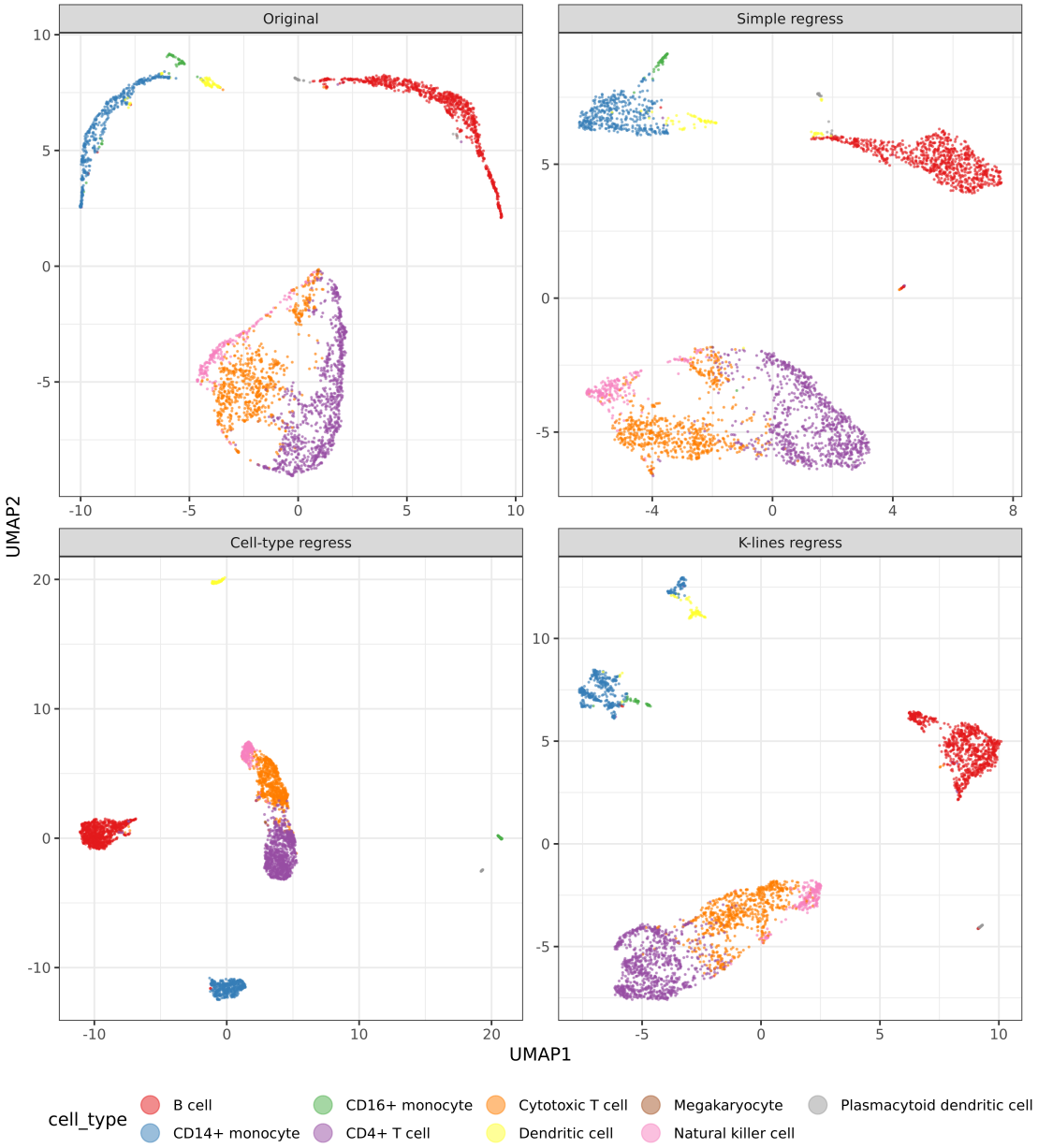


Figure 2.6: UMAP visualization of the cell score matrice **S** before correction and its corrected versions after regressing out cell library size in the PBMC10x dataset.

Cell types are marked with colors. Original: the original score matrix **S** without correction; Simple regress: simply regressing out cell library size with all cells in one group, i.e., $N_k = 1$; Cell-type regress: regressing out cell library size within each cell type, i.e., $C_1, \ldots, C_{N_k}$ are cell types; K-lines regress: regressing out the library size within each K-lines cluster, i.e., $C_1, \ldots, C_{N_k}$ are cell clusters.

S2.7.7 Comparison between PNMF and NMF

PNMF, the first step of `scPNMF`, outputs a much more sparse representation of a scRNA-seq dataset than NMF does. Using the FregGold dataset [3] and $K = 5$ bases, we demonstrate

that the weight matrix of PNMF is highly sparse (42.6% zeros) and has largely mutually exclusive bases, while the zero proportion in the weight matrix of NMF is only 1.1%, and the bases are much less mutually exclusive (Fig. 2.7). These results suggest that PNMF bases are concentrated on a small set of genes and correspond to gene groups with distinct functions. Moreover, we demonstrate that, when applied to the seven scRNA-seq datasets (Table S2.3), `scPNMF` outperforms its variant that replaces PNMF by NMF in the first step (Fig. 2.8)

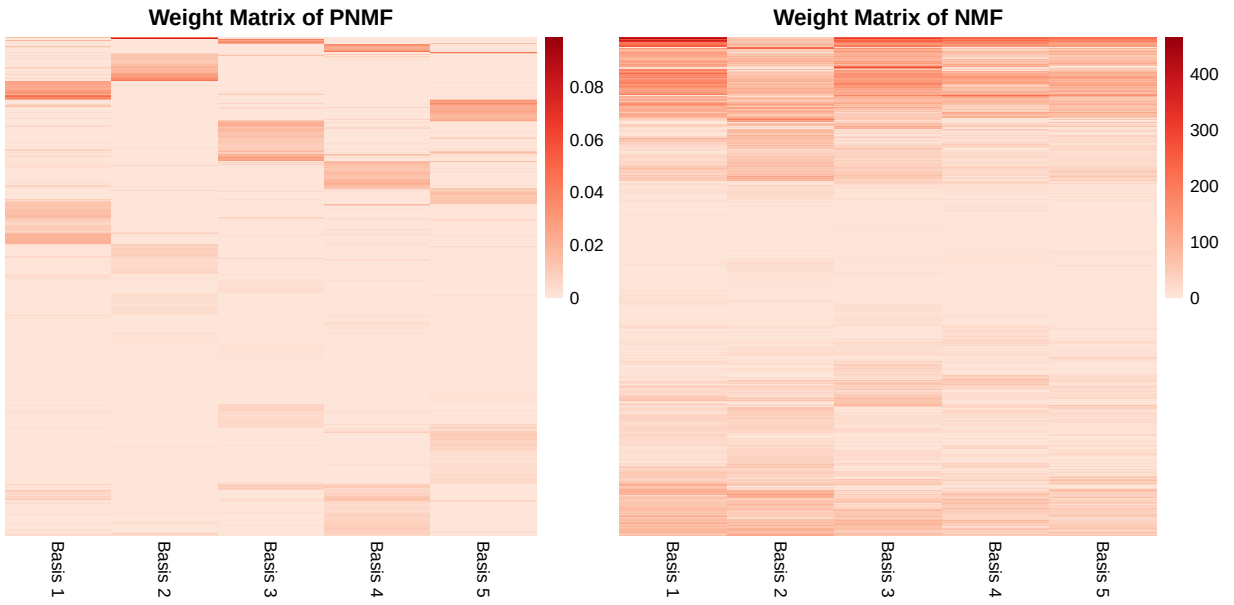


Figure 2.7: Weight matrices of PNMF and NMF. Rows are genes ordered by hierarchical clustering, and columns are bases.

`scPNMF` also has the functionality of outputting a projection matrix $\mathbf{W}$ that can project new cells onto the latent space, which `scPNMF` learns from reference cells. This functionality enables the alignment of new data with the reference data in the same low-dimensional space and facilitates cell type prediction in the new data. In contrast, NMF does not output a projection matrix, and the basis-by-cell matrix it outputs does not satisfy the requirement of a projection matrix.

The reason why we cannot simply select genes from the NMF weight matrix and use these genes to align new data with reference data is that the selected gene number (usually in hundreds) would be much greater than the number of bases in `scPNMF`'s projection matrix. Otherwise, if the selected gene number is too small, we would lose biological information for

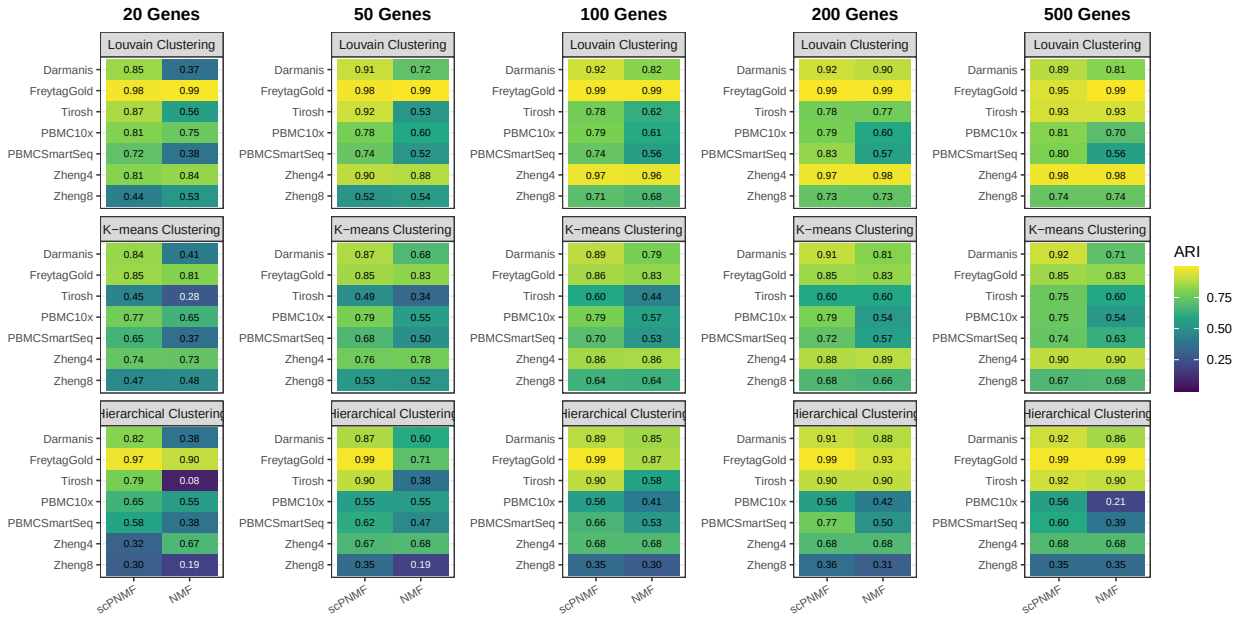


Figure 2.8: Benchmarking `scPNMF` and its variant, where PNMF is replaced by NMF, in selecting 20, 50, 100, 200, and 500 genes for cell clustering.

aligning cells, not to mention that it is not straightforward to select a small number of genes from a not-so-sparse NMF weight matrix. Due to the well-known curse of dimensionality, we deem it reasonable to use a low-dimensional space, instead of hundreds of genes, to align cells. `scPNMF` essentially combines gene selection and dimensionality reduction into one step by directly providing the projection matrix, and we demonstrate that `scPNMF` has good performance in applications.

S2.7.8 Comparison with f-scLVM

The factorial single-cell latent variable model (f-scLVM) is a Bayesian method based on factor analysis that can jointly refine gene set annotations and infer factors without annotation. Similar to our scPNMF, f-scLVM indeed can also learn sparse and interpretable factors [15]. However, scPNMF differs from f-scLVM in its required input data, main goal, and model construction. As a result, scPNMF performs better in informative gene selection for targeted gene profiling, which is its major goal. Moreover, we find scPNMF more computationally efficient than f-scLVM.

S2.7.8.1 Differences in input data

While scPNMF only requires a gene-by-cell count matrix as input, f-scLVM additionally requires pre-defined gene sets for its model fitting.

S2.7.8.2 Differences in main goal and model

scPNMF aims to select a limited number of informative genes for targeted gene profiling based on existing scRNA-seq data. It finds the set of informative genes by learning a low-dimensional embedding of cells so that the bases correspond to sparse and mutually exclusive gene groups, and further selecting bases based on functional annotations (optional), correlation screening, and multimodality testing to remove uninformative bases that cannot distinguish cell types.

In contrast, f-scLVM focuses more on decomposing scRNA-seq datasets into interpretable components. It jointly infers both annotated and unannotated factors, including confounders, and refines the pre-defined gene sets in a data driven manner. The model can be written as:

$$\mathbf{Y} = \underbrace{\sum_{c=1}^{C} \mathbf{u}_c \mathbf{V}_c^\mathsf{T}}_{\text{cell covariates}} + \underbrace{\sum_{a=1}^{A} \mathbf{p}_a \mathbf{R}_a^\mathsf{T}}_{\text{annotated factors}} + \underbrace{\sum_{h=1}^{H} \mathbf{s}_h \mathbf{Q}_h^\mathsf{T}}_{\text{unannotated factors}} + \mathbf{\Psi} \tag{S2.13}$$

$$= \mathbf{X}\mathbf{W}^\mathsf{T} + \mathbf{\Psi}. \tag{S2.14}$$

Here, $\mathbf{Y}$ denotes the cell-by-gene gene expression matrix; the vectors $\mathbf{u}_c$, $\mathbf{p}_a$, $\mathbf{s}_h$ correspond to known cell covariates, as well as cell states for annotated and unannotated factors; and $\mathbf{V}_c$, $\mathbf{R}_a$, $\mathbf{Q}_h$ are the corresponding regulatory weights of a given factor on all genes; the matrix $\mathbf{\Psi}$ denotes residual noise. We then collapse the vectors of factors and weights into activation matrices $\mathbf{X} = [\mathbf{u}_1, \ldots, \mathbf{u}_C, \mathbf{p}_1, \ldots, \mathbf{p}_A, \mathbf{s}_1, \ldots, \mathbf{s}_H]$ and $\mathbf{W} = [\mathbf{V}_1, \ldots, \mathbf{V}_C, \mathbf{R}_1, \ldots, \mathbf{R}_A, \mathbf{Q}_1, \ldots, \mathbf{Q}_H]$.

f-scLVM is not originally designed for selecting informative genes from the gene expression

matrix alone. Although the f-scLVM authors have discussed about identifying an augmented gene set specific to each factor, the identified genes are for interpreting the factors but not for capturing the overall biological variations of cells or distinguishing cell types.

S2.7.8.3 Results for informative gene selection

Although there is no description of informative gene selection in the f-scLVM paper, inspired by the bilinear model structure (eq. (S2.14)), we have used f-scLVM in two ways to select n informative genes from its estimated weight matrix $\mathbf{W}$, whose columns are factors' loading vectors and rows are genes. Note that $\mathbf{W}$ is not a direct output of the `f-scLVM` software package.

- Across-factor: select informative genes based on their maximum loadings across factors (i.e., take the maximum of each row of $\mathbf{W}$; then pick the n genes with the largest row maxima).
- Per-factor: select top $\sim \lceil n/K \rceil$ informative genes for each factor, where K is the number of factors (i.e., pick the $\sim \lceil n/K \rceil$ genes with the largest loadings in each column of $\mathbf{W}$); then take union of the K informative gene sets. Note that the union may contain fewer than n genes due to the possible overlaps of gene sets.

The f-scLVM results are based on R package `slalom` (version 1.10.0) and default parameter values (Gene set annotations: the MSigDB core processes database (hallmark gene sets H) v7.2; number of hidden factors: 5; minimum number of genes to retain a gene set: 10). Similar as in section 2.3.3, we comprehensively benchmark scPNMF and f-scLVM on seven scRNA-seq datasets (Table S2.3) using three clustering methods (Louvain clustering, K-means clustering, and hierarchical clustering). Using the adjusted Rank index (ARI) as the metric of clustering accuracy, we calculate the ARI values of the three clustering methods on each dataset using 20, 50, 100, 200, and 500 selected informative genes, which are the commonly used gene numbers in targeted gene profiling.

Fig. 2.9 shows that scPNMF consistently has the highest overall ARI values across

datasets and clustering methods. scPNMF leads to more stable overall average ARI values under varying numbers of informative genes, indicating its stronger robustness to the gene number constraint of targeted gene profiling. It is worth noting that scPNMF works well even when the number of informative genes is as small as 20.

Similar as in section 2.3.3, Fig. 2.10 shows the UMAP visualization of cells in the Zheng4 dataset based on the 100 informative genes selected by scPNMF and f-scLVM. scPNMF leads to a clear separation of naive cytotoxic T cells and regulatory T cells, while f-scLVM-Across-factor and f-scLVM-Per-factor cannot, even though f-scLVM-Per-factor incorporates slightly more (123) informative genes.

S2.7.8.4 Computational time

scPNMF is more time-efficient than f-scLVM across diverse scRNA-seq datasets. We have run both software packages on a PC with 3.4 GHz Quad-Core Intel Core i5 and 8GB RAM. The f-scLVM results are based on R package `slalom` (version 1.10.0) and default parameter values. In Table S2.6, we see that scPNMF runs 2.2X $\sim$ 100X faster than f-scLVM.

Table S2.6: Running time of scPNMF and f-scLVM in minutes

Dataset	scPNMF Running Time (mins)	f-scLVM Running Time (mins)
Darmanis	27.95	178.57
FreytagGold	38.87	474.61
Tirosh	23.47	1172.91
PBMC10x	22.87	733.36
PBMCSmartSeq	48.88	107.92
Zheng4	0.90	53.89
Zheng8	1.12	111.89

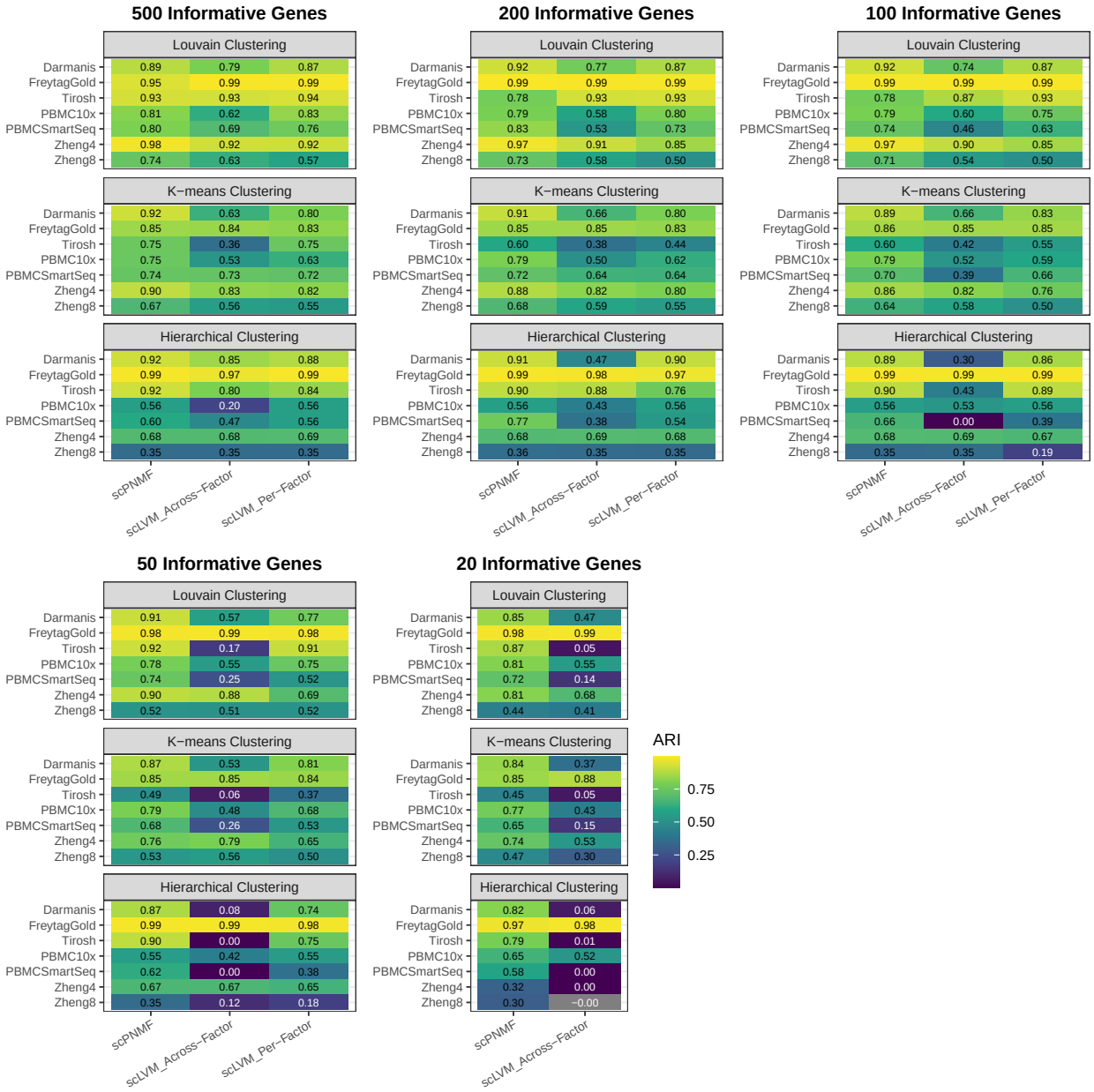


Figure 2.9: Benchmarking scPNMF and f-scLVM using 20, 50, 100, 200, 500 genes.

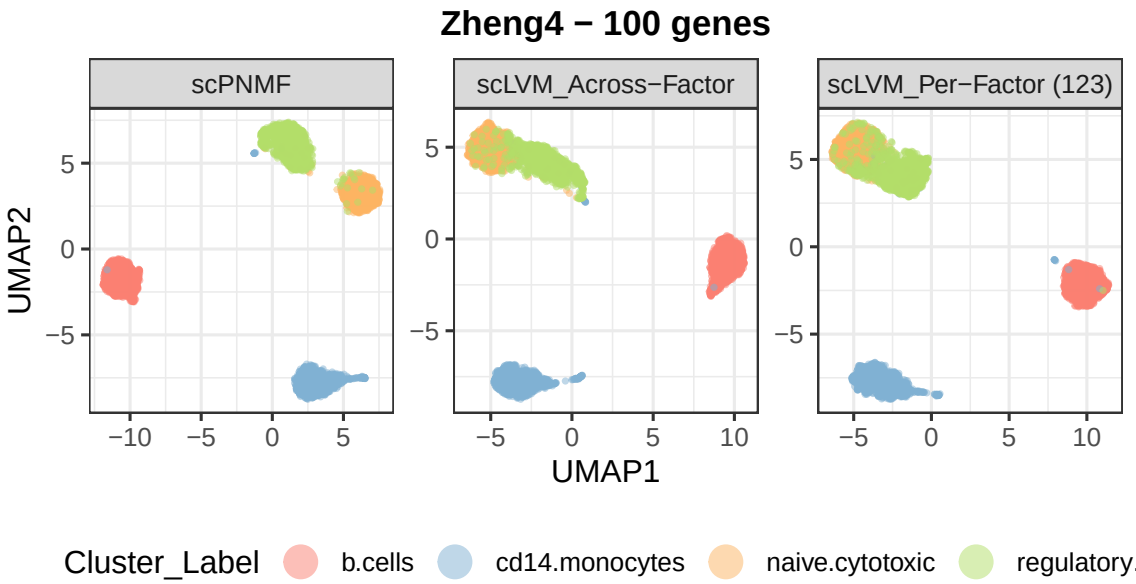


Figure 2.10: UMAP visualization of cells in the Zheng4 dataset based on 100 informative genes selected by scPNMF and f-scLVM.

S2.7.9 Supplementary figures

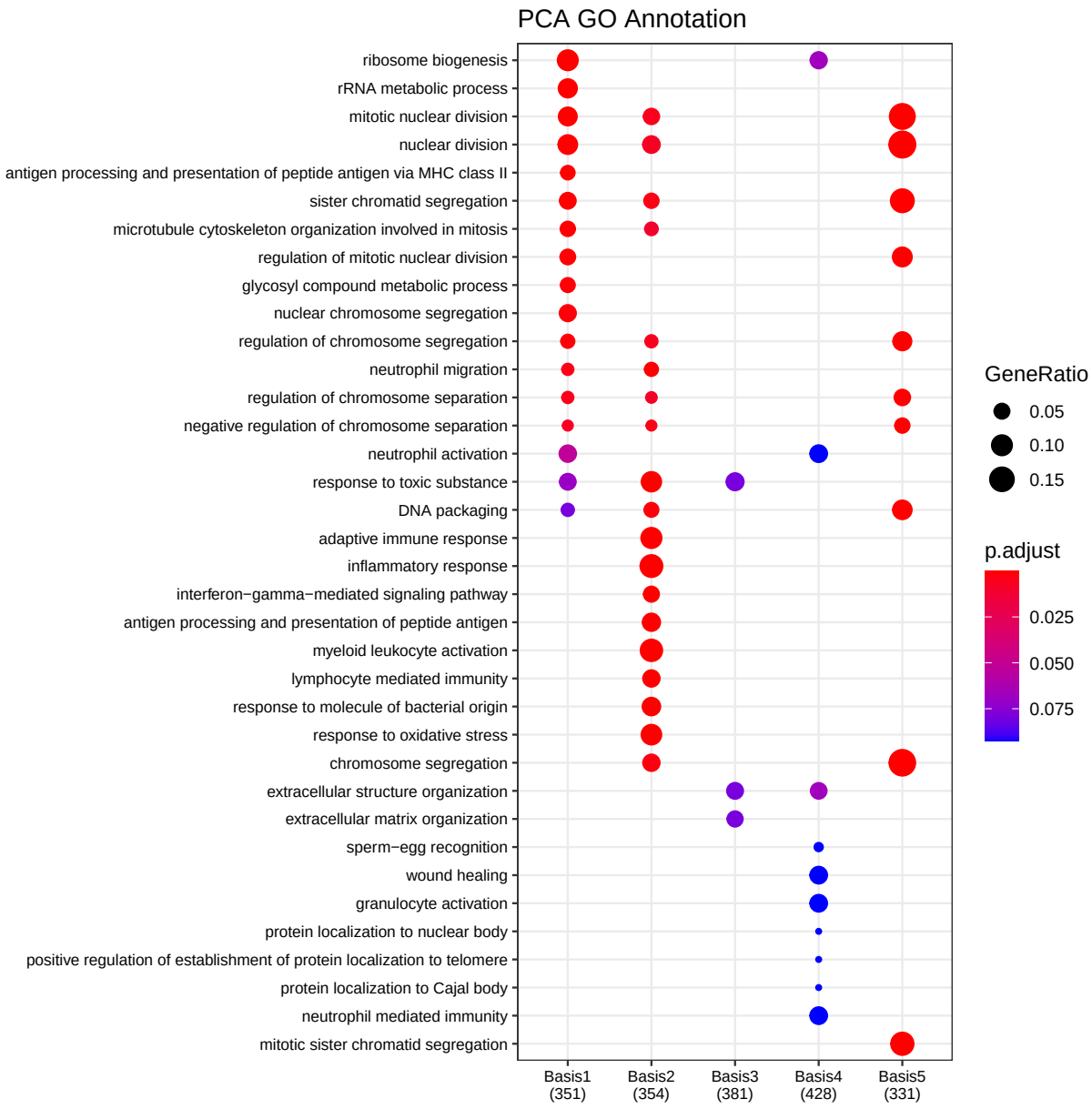


Figure 2.11: GO annotation on weight matrix of PCA. The enriched GO terms between bases are largely overlapped.

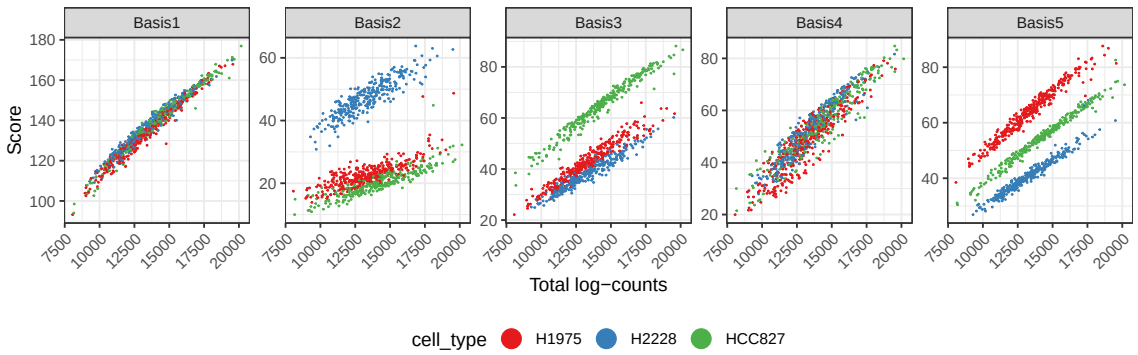


Figure 2.12: scPNMF scores versus total log-counts of FregGold dataset colored by cell types. Basis 2 distinguishes H2228 from the other two cell types, and basis 3 distinguishes HCC827 from the other two cell types.

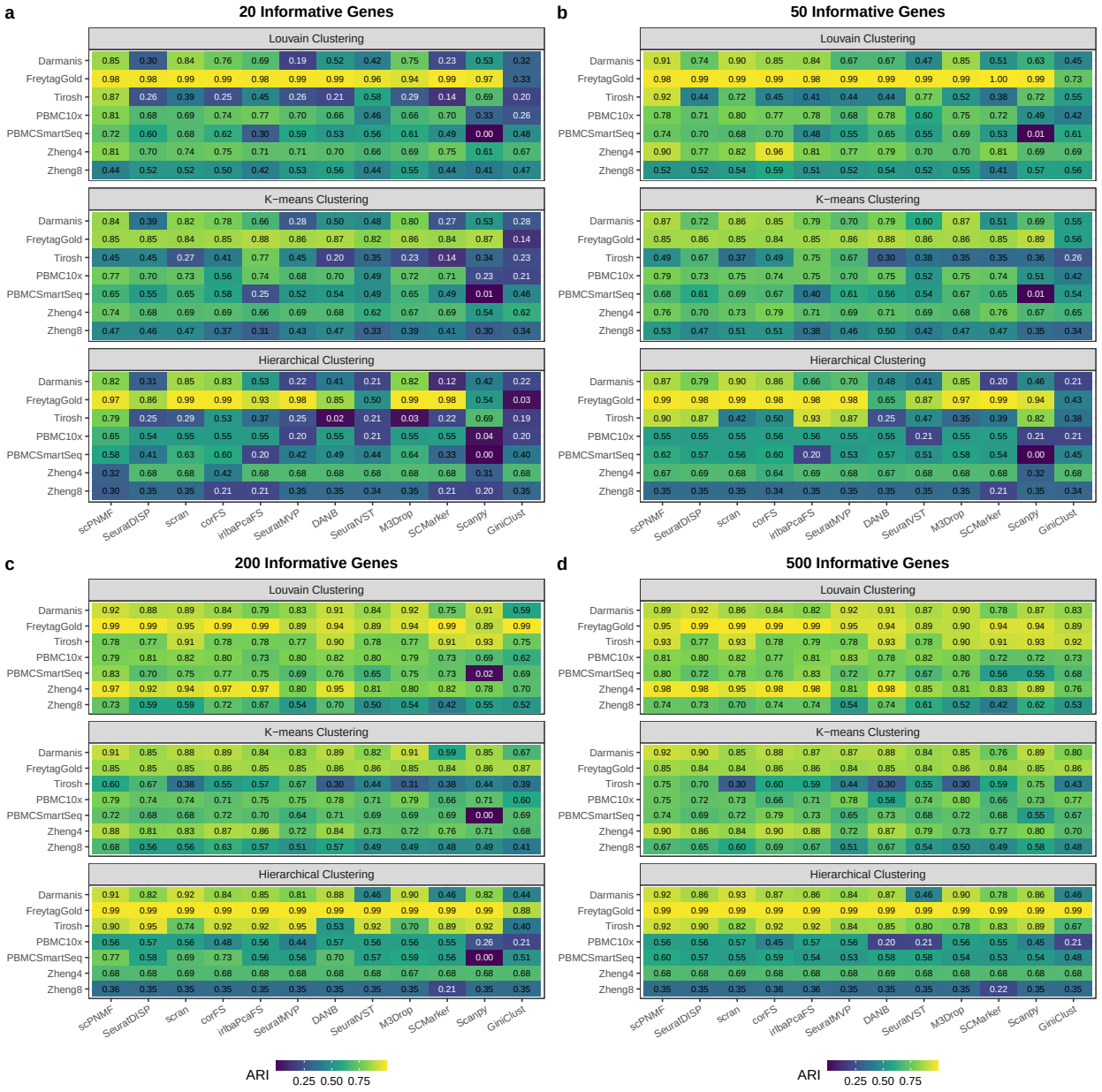


Figure 2.13: Benchmarking scPNMF and other informative gene selection methods using 20, 50, 200, 500 genes.

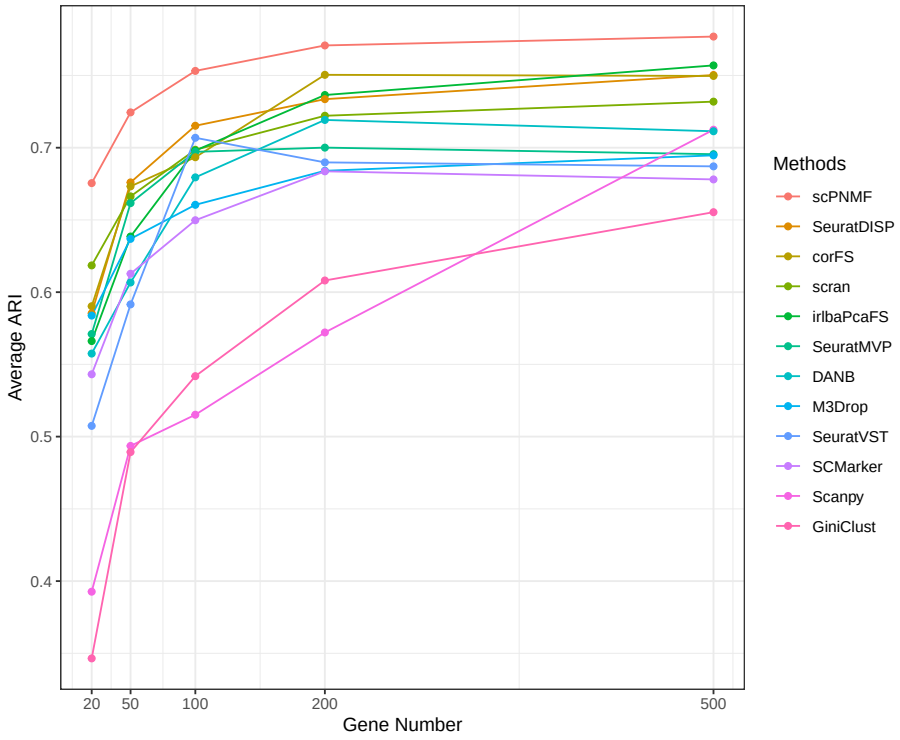


Figure 2.14: Comparison of overall average ARI of different methods versus gene numbers. The y-axis indicates the average ARI values across seven datasets and three clustering methods for each gene selection methods.

CHAPTER 3

ClusterDE: a post-clustering differential expression method robust to false-positive inflation caused by double dipping

3.1 Introduction

The recent development of single-cell RNA-seq (scRNA-seq) technologies has revolutionized transcriptomic studies by providing unprecedented pictures of gene expression within individual cells. A major task of scRNA-seq data analysis is to annotate cell types and understand their biological differences. Hence, the standard workflow of analyzing scRNA-seq data includes two steps: (1) clustering cells to find potential cell types, and (2) finding differentially expressed (DE) genes between cell clusters as potential cell-type marker genes [64, 65].

Although this post-clustering differential expression (DE) procedure is used in the state-of-the-art scRNA-seq analysis pipelines such as the R package `Seurat` [66] and the Python module `Scanpy` [10], researchers have realized that this procedure is conceptually problematic. For instance, Seurat contains the warning message that "P values should be interpreted cautiously, as the genes used for clustering are the same genes tested for differential expression." This issue is commonly referred to as "double dipping," meaning that the same gene expression data are used twice to define cell clusters and DE genes, thus leading to an inflated false discovery rate (FDR) in identifying post-clustering DE genes as putative cell-type marker genes when the cell clusters are spurious.

We illustrate the double-dipping issue in Fig. 3.1a, a scenario where only a single cell

type exists, and no genes should be identified as between-cell-type DE genes. However, as clustering is based on gene expression data, certain genes would be correlated with the resulting cell clusters if their expression patterns drive the clustering. Hence, these genes would have different conditional distributions in the two cell clusters and subsequently be identified as between-cell-cluster DE genes, but they are false-positive between-cell-type DE genes. Therefore, this double-dipping issue would inflate the false discovery rate (FDR), the expected proportion of false-positive between-cell-type DE genes among all identified DE genes.

Two attempts to solve the double-dipping issue include the truncated normal (TN) test [17] and the Countsplit method [18]. The first method TN test has five steps: (1) splitting cells into two sets: training cells and test cells; (2) applying a clustering algorithm to the training cells to find two clusters; (3) training a support vector machine classifier on the training cells to predict a cell's cluster label from the cell's gene expression vector; (4) using the trained classifier to predict the test cells' cluster labels; (5) finding DE genes between the two test cell clusters using the TN test. Instead of splitting cells, the second method Countsplit splits the scRNA-seq count matrix into two count matrices of the same dimensions (cells and genes)—a training matrix and a test matrix—by a procedure called data thinning [67]. Since the two matrices have exactly one-to-one matched cells, Countsplit finds cell clusters by applying a clustering algorithm to the training matrix, and it subsequently identifies DE genes by applying a DE test to the test matrix given the cell clusters. Despite the claims made by the TN test and Countsplit that they can provide well-calibrated P values, uniformly distributed between 0 and 1 under the null hypotheses, our findings indicate that their P values are anti-conservative in the presence of gene-gene correlations (section Results). The reason behind this issue is that the validity check of P values in the TN test and Countsplit papers relied on simulation studies that implicitly assumed genes to be independent [17, 18], an assumption that does not hold in real scRNA-seq data. As a result, the P value calibration issue would lead to inflated FDRs when the TN test and Countsplit are applied to real scRNA-seq data.

In addition to the TN test and Countsplit, several cluster-free DE tests have been devel-

oped to circumvent the double-dipping issue by bypassing the cell clustering step [19–24]. However, it is important to note that these cluster-free methods do not aim to identify potential cell types. Consequently, the DE genes identified by these methods cannot be interpreted as marker genes for specific cell types, unlike the DE genes identified after clustering. For instance, for the genes that demonstrate a multimodality pattern, it is difficult to match the modes with the potential cell types identified in the previous clustering step. In other words, the cluster-free DE genes and the post-clustering DE genes serve different purposes and are not conceptually comparable. Another stream of methods has been developed to assess the quality of clustering results, e.g., the "purity" of a cluster or if two clusters should be merged [25–29]. However, these methods do not provide a direct statistical test for identifying DE genes, and it remains difficult to determine the threshold for clustering quality above which double dipping is not a concern. In this study, we focus on addressing the inflated FDR issue when using post-clustering DE genes as cell-type marker genes. Hence, we do not consider cluster-free DE tests and clustering assessment methods as competing alternatives in our investigation.

Here we introduce ClusterDE, a post-clustering DE method for identifying potential cell-type marker genes by avoiding the inflated FDR issue due to double dipping. It is worth noting that ClusterDE is not designed to replace any existing pipelines for clustering followed by DE analysis (e.g., Seurat); instead, ClusterDE works simply as an add-on to an existing pipeline for achieving more reliable discoveries. In particular, ClusterDE controls the FDR for identifying cell-type marker genes even when the cell clusters are spurious. As an efficient and interpretable method, ClusterDE adapts to the most widely used pipelines Seurat [66] and Scanpy [10], which include a wide range of clustering algorithms and DE tests. We benchmarked ClusterDE against the default Seurat (which includes double dipping), the TN test, and Countsplit, each of which includes a cell clustering step and a DE analysis step. Specifically, to align with the prevailing practices in single-cell data analysis, we employed the default Seurat clustering algorithm (which involves data processing steps followed by the Louvain algorithm) for cell clustering; for DE analysis, we evaluated five widely used DE tests (e.g., the Wilcoxon rank-sum test and the two-sample t test) included in the `Seurat`

package, with the exception of the TN test, which utilizes its own DE test. Our benchmarking results demonstrate that ClusterDE is the only method that effectively controls the FDR across varying thresholds. Moreover, ClusterDE achieves comparable or superior statistical power compared to the other three methods. When applied to the scRNA-seq data of five homogeneous cell lines, ClusterDE successfully avoids finding false-positive DE genes. In contrast, Seurat, the TN test, and Countsplit yield thousands of DE genes due to double dipping. Moreover, when applied to a well-studied peripheral blood mononuclear cells (PBMC) scRNA-seq dataset with two biological replicates and four protocols, ClusterDE excels at discovering the cell-type marker genes of $CD14^+$ monocytes and $CD16^+$ monocytes as its top DE genes, while Seurat's top DE genes contain many housekeeping genes. Besides the ability to control the FDR and identify cell-type marker genes, ClusterDE has a notable practical advantage for allowing users to dissect an abstract statistical null hypothesis as concrete synthetic null data, so users can decide whether the synthetic null data accurately reflects the negative control scenario they have in mind, and if not, how the synthetic null generation should be adjusted.

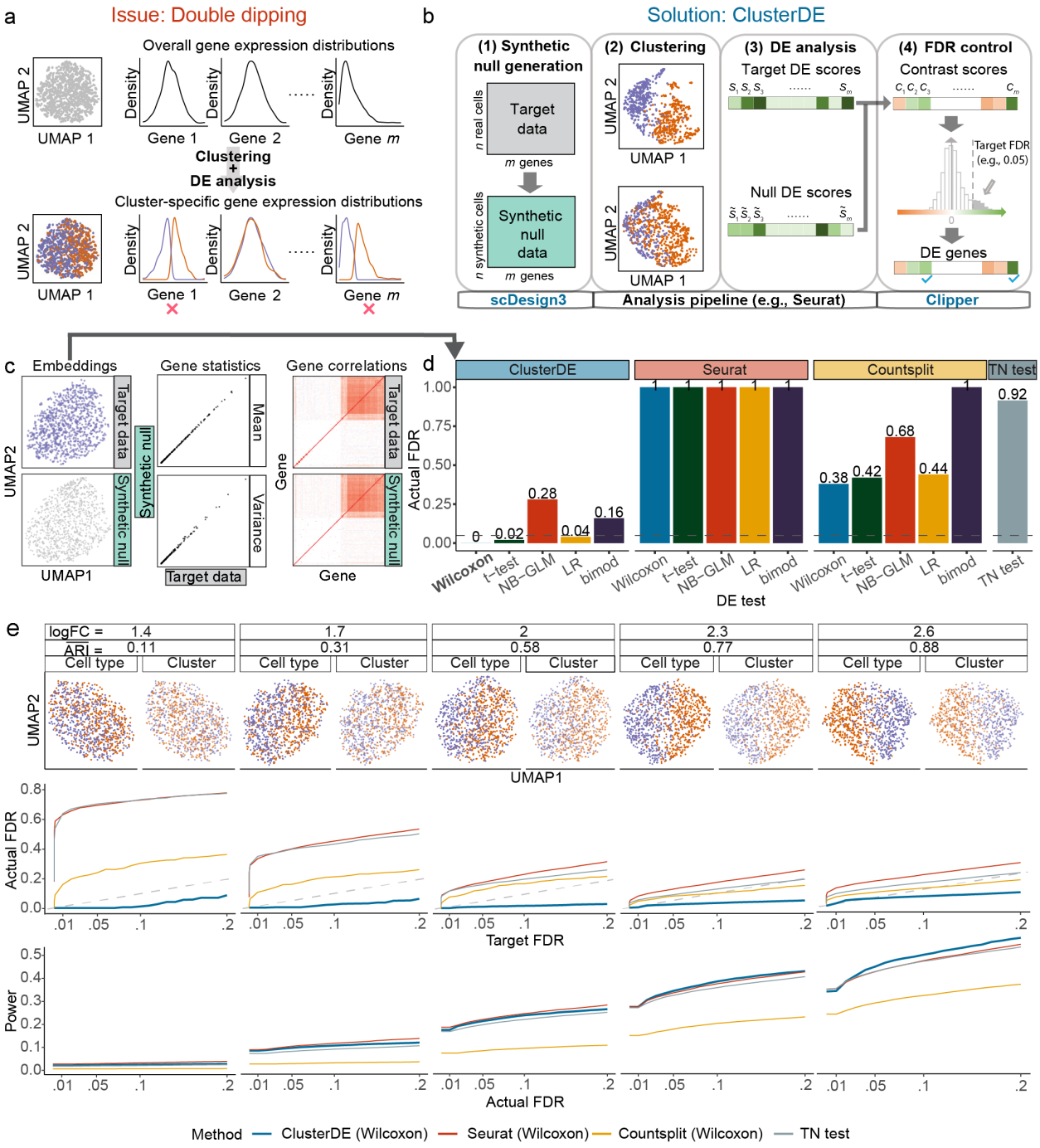


Figure 3.1: ClusterDE is a solution to the double-dipping issue in post-clustering DE analysis.

a, An illustration of the double-dipping issue. Each gene's expression follows an unimodal distribution when cells come from a homogeneous cell type. However, if clustering divides the cells into two clusters, certain genes are "forced" to have different distributions between the two clusters. **b**, An overview of the ClusterDE test. Given the "target data" (real data), ClusterDE employs the simulator scDesign3 [68] to generate the corresponding "synthetic null data," which contains synthetic cells from one "hypothetical" cell type (the null hypothesis) to mimic the real cells but fill any gap between real cell types if existent. Then ClusterDE applies a clustering algorithm followed by a DE test to both the target data and the synthetic null data in parallel, yielding two DE scores for each gene (a "target DE score" and a "null DE score"). Finally, ClusterDE uses the FDR-control method Clipper [69] to calculate a contrast score based on the two DE scores for each gene. ClusterDE identifies DE genes as those whose contrast scores exceed the threshold, which is determined by finding a contrast score threshold (represented by the vertical dashed line) based on the contrast score distribution and the desired target FDR (e.g., 0.05). **c**, When the target data contained cells from a single type (simulation; see ClusterDE methodology "Simulation setting with one cell type and zero true DE genes"), the synthetic null data generated by ClusterDE resembled the target data well in terms of UMAP cell embeddings (left), per-gene expression mean and variance statistics (middle), and gene-gene correlations (right). **d**, On the target data in c, ClusterDE (with five DE tests) outperformed existing methods—including Seurat (which does not consider double dipping), Countsplit (which aims to address double dipping and works with any DE test), and TN test (which aims to address double dipping and has its own DE test)—in FDR control. The horizontal dashed line indicates the target FDR of 0.05. The five DE tests are the Wilcoxon rank sum test (Wilcoxon), t-test, negative binomial generalized linear model (NB-GLM), logistic regression model predicting cluster membership with likelihood-ratio test (LR), and likelihood-ratio test for single cell gene expression (bimod). **e**, The FDRs and power of ClusterDE and the existing methods under various severity levels of double dipping. The log fold change (logFC) summarizes the average gene expression difference between two cell types in simulation (see ClusterDE methodology "Simulation setting with two cell types and 200 true DE genes"). Corresponding to a small logFC, a small adjusted Rand index (ARI) represents a bad agreement between cell clusters and cell types, representing a more severe double-dipping issue. Across various severity levels of double dipping, ClusterDE controlled the FDRs under the target FDR thresholds (diagonal dashed line) and achieved comparable or higher power compared to the existing methods at the same actual FDRs.

3.2 ClusterDE methodology

3.2.1 Notations for the double-dipping problem in post-clustering DE analysis

The target data is denoted by $\mathbf{Y} = [Y_{ij}] \in \mathbb{N}_{\geqslant 0}^{n \times m}$, a cell-by-gene Unique Molecular Identifier (UMI) count matrix with n cells as rows, m genes as columns, and Y_{ij} as the UMI count of gene $j = 1, \ldots, m$ in cell $i = 1, \ldots, n$. We treat each cell i as an observation, which is an m-dimensional vector $\mathbf{Y}_i = (Y_{i1}, \cdots, Y_{im})^\mathsf{T}$.

In our formulation of the post-clustering DE problem, the n cells belong to two latent cell types and are partitioned into two clusters by a clustering algorithm. Accordingly, we use $Z_i \in \{0, 1\}$ to denote cell i's latent cell type.

We define the "ideal DE test" as the one that decides whether a gene has equal mean expression in two cell types. For gene j, we assume that $\{(Y_{ij}|Z_i = 0)\}_{i=1}^n$ share the same mean denoted by $\mu_{0j} = \mathbb{E}[Y_{ij}|Z_i = 0]$, and $\{(Y_{ij}|Z_i = 1)\}_{i=1}^n$ share the same mean denoted by $\mu_{1j} = \mathbb{E}[Y_{ij}|Z_i = 1]$. Then the ideal DE test has the following null hypothesis H_{0j} and alternative hypothesis H_{1j}:

$$H_{0j} : \mu_{0j} = \mu_{1j} \quad \text{vs.} \quad H_{1j} : \mu_{0j} \neq \mu_{1j} \, .$$

Hence, gene j is a true DE gene if and only if H_{0j} does not hold. When all n cells belong to one cell type only, all m null hypotheses, $H_{01}, \ldots, H_{0m}$, hold simultaneously.

However, since Z_i's are unobserved, standard single-cell data analysis partitions cells into two clusters using a clustering algorithm g (e.g., the Louvain algorithm in Seurat) applied to $\mathbf{Y}$. We use $\widehat{Z}_i = g_{\mathbf{Y}}(\mathbf{Y}_i) \in \{0, 1\}$ to denote cell i's cluster membership, where $g_{\mathbf{Y}} : \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\} \to \{0, 1\}$ is the clustering function, constructed from the clustering algorithm g and the data $\mathbf{Y}$, that maps a cell's gene expression vector to a cluster membership.

After cell clustering, standard single-cell analysis performs a DE test for each gene based on $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ and $\widehat{Z}_1, \ldots, \widehat{Z}_n$. In other words, the data $\mathbf{Y}$ is used twice (in clustering and DE analysis), referred to as the "double-dipping (DD) issue." The standard post-clustering DE analysis used in the Seurat pipeline has the DD issue, and it tackles a statistical test

different from the ideal DE test. Specifically, for gene j, we denote $\mu_{0j}^{\mathrm{DD}} = \mathbb{E}[Y_{ij}|\widehat{Z}_i = 0]$ and $\mu_{1j}^{\mathrm{DD}} = \mathbb{E}[Y_{ij}|\widehat{Z}_i = 1]$, two parameters that are the same for all $i = 1, \ldots, n$. Then, the post-clustering DE method in Seurat corresponds to the following null hypothesis H_{0j}^{DD} and alternative hypothesis H_{1j}^{DD}:

$$H_{0j}^{\mathrm{DD}} : \mu_{0j}^{\mathrm{DD}} = \mu_{1j}^{\mathrm{DD}} \quad \text{vs.} \quad H_{1j}^{\mathrm{DD}} : \mu_{0j}^{\mathrm{DD}} \neq \mu_{1j}^{\mathrm{DD}} .$$

Hence, gene j would be detected as a false-positive cell-type marker gene if H_{0j}^{DD} is rejected but H_{0j} holds, leading to an inflated FDR in identifying cell-type marker genes. Fig. 3.7 provides a toy example illustration of this issue.

3.2.2 ClusterDE step 1: synthetic null generation

Previous findings indicate that, in a single cell type, each gene's UMI counts can be fitted well by a negative binomial (NB) distribution [70–72], and all genes' UMI counts can be well approximated by a multivariate NB (MVNB) distribution specified by the Gaussian copula [68]. Based on these findings, in ClusterDE, the null model that indicates a single "hypothetical" cell type is an MVNB distribution specified by the Gaussian copula. In ClusterDE step 1, the null model would be fitted on the real data $\mathbf{Y}$ by scDesign3 [68], and subsequently, synthetic null data would be sampled from the fitted null model. The intuition behind this null model is that cells of a single cell type constitute a sample from a homogenous population, in which every gene's marginal count distribution is NB, and the gene-gene correlation structure is specified by the Gaussian copula. In addition, since scDesign3 supports many other choices of marginal distributions, ClusterDE can also generate synthetic null data from multivariate Gaussian, multivariate Poisson, multivariate Zero-Inflated Poisson, and multivariate Zero-Inflated Negative Binomial distribution.

Note that the idea of fitting a null model on real data, regardless of whether the real data was generated from the null model, is the core idea of the commonly used likelihood-ratio test in statistics [73], in which the maximum likelihood under the null hypothesis is estimated from the real data. Then the null maximum likelihood is compared with the alternative max-

imum likelihood, which is also estimated from the real data under a more flexible alternative hypothesis. Finally, the null hypothesis is only rejected if the null maximum likelihood is significantly smaller than the alternative maximum likelihood. ClusterDE generalizes this idea by sampling synthetic null data from the null model fitted by maximum likelihood estimation on the real data, so any clustering-followed-by-DE pipeline, however complicated, can be applied to the synthetic null data in parallel to the real data. Then a contrastive strategy can identify trustworthy DE genes as those whose DE scores are significantly higher from the real data than the synthetic null data.

Fig. 3.2 illustrates the synthetic null generation process detailed below. In the R package `ClusterDE`, this step 1 is implemented by the R package `scDesign3` (version 0.99.0) [68].

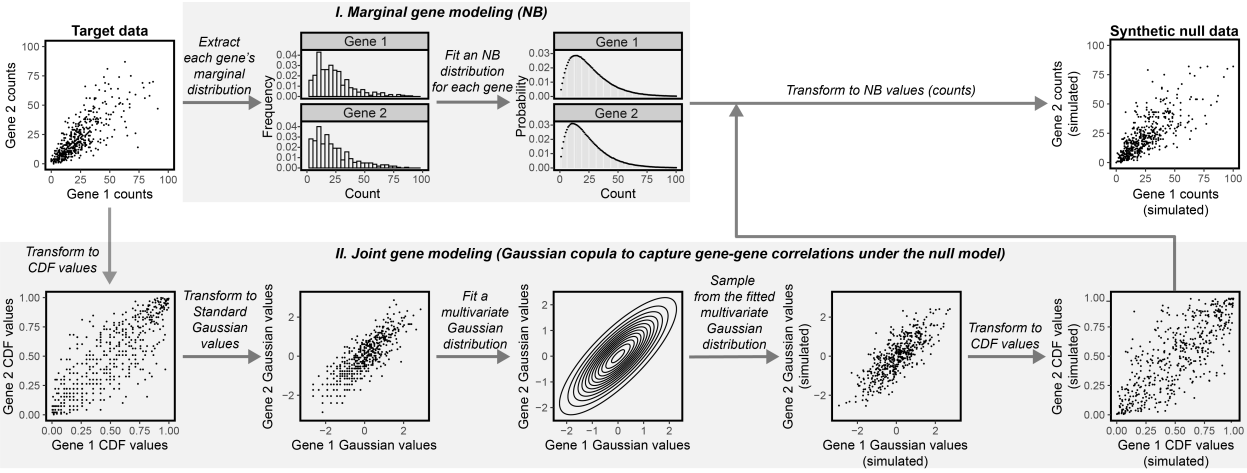


Figure 3.2: The generation process of synthetic null data from target data (top left) by scDesign3.

We show the bivariate case (with two genes) for illustration purposes, and the real case is high-dimensional with thousands of genes. The null model consists of two parts: for marginal gene modeling, each gene's counts follow a negative binomial (NB) distribution; for joint gene modeling, the genes' dependence structure is specified by the Gaussian copula. For the marginal gene modeling part (top), a negative binomial (NB) distribution is fitted to each gene's counts in the target data, obtaining the two NB parameters (mean and dispersion) for each gene. For the joint gene modeling part (bottom), there are three steps. First, each gene's counts in the target data are transformed into cumulative distribution function (CDF) values, via the fitted NB distribution or the counts' empirical distribution (if the target cell number is large), so the gene's CDF values are uniform between 0 and 1. Second, each gene's CDF values are transformed into quantiles of the standard Gaussian $N(0, 1)$ distribution. Third, a multivariate Gaussian distribution (a bivariate Gaussian distribution for illustration) is fitted to the transformed Gaussian values of the many genes whose correlations are to be modeled. The correlation matrix of the fitted multivariate Gaussian distribution specifies the Gaussian copula. After the marginal and joint modeling, the generation of the same genes' synthetic counts takes three steps. First, the genes' standard Gaussian values are jointly sampled from the fitted multivariate Gaussian distribution. Second, each gene's standard Gaussian values are transformed into the CDF values of the standard Gaussian distribution. Third, each gene's CDF values are transformed into quantiles of the gene's fitted NB distribution and thus become counts, which constitute the synthetic null data (top right).

1. **The null model: MVNB specified by the Gaussian copula**

   Under the null model, we assume that Y_{ij}, gene j's UMI count in cell i, independently

follows the $\mathrm{NB}(\mu_j, \sigma_j)$ distribution with the probability mass function:

$$\mathbb{P}(Y = y; \mu_j, \sigma_j) = \frac{\Gamma\left(y + \frac{1}{\sigma_j}\right)}{\Gamma\left(\frac{1}{\sigma_j}\right)\Gamma(y+1)} \left(\frac{1}{1+\sigma_j\mu_j}\right)^{\frac{1}{\sigma_j}} \left(\frac{\sigma_j\mu_j}{1+\sigma_j\mu_j}\right)^{y} ;\ y \in \{0, 1, 2, \cdots\},$$

where μ_j and σ_j are the mean and dispersion parameters of the NB distribution. That is,

$$Y_{1j},\ \cdots,\ Y_{nj} \overset{\text{i.i.d.}}{\sim} \mathrm{NB}(\mu_j, \sigma_j),$$

with "i.i.d." short for "independent and identically distributed," meaning that the n cells' counts for gene j represent a random sample from $\mathrm{NB}(\mu_j, \sigma_j)$.

Denoting F_j as the cumulative distribution function (CDF) of $\mathrm{NB}(\mu_j, \sigma_j)$, $j = 1, \ldots, m$, the MVNB distribution specified by the Gaussian copula is

$$\begin{aligned} &\left(\Phi^{-1}(F_1(Y_{11})), \cdots, \Phi^{-1}(F_m(Y_{1m}))\right)^{\mathsf{T}}, \\ &\qquad\qquad \vdots \qquad\qquad \overset{\text{i.i.d.}}{\sim} N_m(\mathbf{0}, \mathbf{R}), \\ &\left(\Phi^{-1}(F_1(Y_{n1})), \cdots, \Phi^{-1}(F_m(Y_{nm}))\right)^{\mathsf{T}}, \end{aligned}$$

where Φ is the CDF of the standard Gaussian distribution $N(0, 1)$, and $N_m(\mathbf{0}, \mathbf{R})$ is an m-dimensional Gaussian distribution with an m-dimensional $\mathbf{0}$ mean vector and an m-by-m correlation matrix $\mathbf{R}$ (which is also the covariance matrix because all m Gaussian variables have unit variances). This null model assumes that, after each gene is transformed to a standard Gaussian random variable, the n cells represent a random sample from an m-dimensional Gaussian distribution with zero means, unit variances, and a correlation matrix $\mathbf{R}$.

In summary, the null model parameters include $\{\mu_j, \sigma_j\}_{j=1}^m$ and $\mathbf{R}$.

2. **Fitting the null model to real data (parameter estimation)**

First, the parameters $\{\mu_j, \sigma_j\}_{j=1}^m$ are estimated by employing the maximum likelihood estimation for the m NB distributions: using $Y_{1j}, \ldots, Y_{nj}$ to estimate μ_j and σ_j as $\widehat{\mu}_j$ and $\widehat{\sigma}_j$, respectively, $j = 1, \ldots, m$. Based on $\{\widehat{\mu}_j, \widehat{\sigma}_j\}_{j=1}^m$, the corresponding CDFs are

denoted as $\widehat{F}_1, \ldots, \widehat{F}_m$.

Second, to estimate $\mathbf{R}$, each Y_{ij} is first transformed as $U_{ij} = V_{ij} \cdot \widehat{F}_j(Y_{ij}) + (1 - V_{ij}) \cdot \widehat{F}_j(Y_{ij} + 1)$, where $V_{ij} \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0, 1]$, so that $U_{ij} \sim \text{Uniform}[0, 1]$. This procedure is referred to as the "distribution transform" to convert a discrete random variable Y_{ij} to a continuous Uniform$[0, 1]$ random variable [74]. Then, $\mathbf{R}$ is estimated as the sample correlation matrix of

$$\left(\Phi^{-1}(U_{11}), \cdots, \Phi^{-1}(U_{1m})\right)^{\mathsf{T}}, \ \cdots, \ \left(\Phi^{-1}(U_{n1}), \cdots, \Phi^{-1}(U_{nm})\right)^{\mathsf{T}}$$

and denoted as $\widehat{\mathbf{R}}$.

In summary, the fitted null model parameters include $\{\widehat{\mu}_j, \widehat{\sigma}_j\}_{j=1}^m$ and $\widehat{\mathbf{R}}$.

3. **Sampling from the fitted null model (synthetic null data generation)**

First, n Gaussian vectors of m dimensions are independently sampled $N_m(\mathbf{0}, \widehat{\mathbf{R}})$ as

$$(\widetilde{Z}_{11}, \cdots, \widetilde{Z}_{1m})^{\mathsf{T}}, \ \cdots, \ (\widetilde{Z}_{n1}, \cdots, \widetilde{Z}_{nm})^{\mathsf{T}}.$$

Second, The n Gaussian vectors are converted to NB count vectors as

$$\widetilde{\mathbf{Y}}_1 := \left(\widehat{F}_1^{-1}(\Phi(\widetilde{Z}_{11})), \cdots, \widehat{F}_m^{-1}(\Phi(\widetilde{Z}_{1m}))\right)^{\mathsf{T}},$$

$$\vdots$$

$$\widetilde{\mathbf{Y}}_n := \left(\widehat{F}_1^{-1}(\Phi(\widetilde{Z}_{n1})), \cdots, \widehat{F}_m^{-1}(\Phi(\widetilde{Z}_{nm}))\right)^{\mathsf{T}},$$

which represent the n synthetic null cells, each of which contains m genes' synthetic null counts sampled from the null model.

In summary, the real data is an n-by-m count matrix $\mathbf{Y}$ with the n real cells $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ as the rows, while the synthetic null data is also an n-by-m count matrix $\widetilde{\mathbf{Y}}$ with the n synthetic null cells $\widetilde{\mathbf{Y}}_1, \ldots, \widetilde{\mathbf{Y}}_n$ as the rows. Note that there is no one-to-one correspondence between the real cells and the synthetic null cells because the synthetic null cells are independently sampled from the null model.

3.2.3 ClusterDE step 2: cell clustering

While ClusterDE allows any clustering algorithm, to align with the most common practice, we used the R package `Seurat` (version 4.2.0) for cell clustering in the results. That is, we applied the default Seruat clustering to the target data and the synthetic null data in parallel, obtaining two cell clusters in each dataset respectively.

Specifically, the default Seurat clustering includes the following steps applied to both the target data and the synthetic null data, each of which is stored as a Seurat object. We denote each Seurat object as `Seurat.obj`.

1. Normalize each cell to have a total count of 10,000; then perform log(normalized count+ 1) transformation.
   ```
   NormalizeData(Seurat.obj, normalization.method = "LogNormalize",
   scale.factor = 10000)
   ```

2. Select 2,000 highly variable genes.
   ```
   FindVariableFeatures(Seurat.obj, selection.method = "vst",
   nfeatures = 2000)
   ```

3. Scale the data.
   ```
   ScaleData(Seurat.obj)
   ```

4. Run PCA on the data.
   ```
   RunPCA(Seurat.obj, features = VariableFeatures())
   ```

5. Compute cells' k-nearest neighbors.
   ```
   FindNeighbors(Seurat.obj, dims = 1:30, nn.method = "rann", k.param = 20)
   ```

6. Perform Louvain clustering on the cells.
   ```
   FindClusters(Seurat.obj, resolution)
   ```
   Since the Louvain clustering cannot pre-specify the cluster number, we tried resolutions starting from the default resolution of 0.5 and adjusted the resolution until two clusters were found.

After applying the above clustering procedure, we obtained the cluster labels $\widehat{Z}_1, \ldots, \widehat{Z}_n$ from the target data $\mathbf{Y}$, and $\widetilde{Z}_1, \ldots, \widetilde{Z}_n$ from the synthetic null data $\widetilde{\mathbf{Y}}$, respectively, where $\widehat{Z}_i, \widetilde{Z}_i \in \{0, 1\}$, $i = 1, \ldots, n$. Again, there exists no one-to-one correspondence between $\widehat{Z}_1, \ldots, \widehat{Z}_n$ and $\widetilde{Z}_1, \ldots, \widetilde{Z}_n$.

3.2.4 ClusterDE step 3: DE analysis

ClusterDE allows any DE test. In the results, we used five DE tests included in the Seurat function `FindMarkers`, including the Wilcoxon rank-sum test (Wilcoxon, the default test), t-test, negative binomial generalized linear model (NB-GLM), logistic regression model predicting cluster membership with likelihood-ratio test (LR), and likelihood-ratio test for single-cell gene expression (bimod, [75]).

Given a DE test (e.g., the Wilcoxon rank-sum test), on the target data, ClusterDE computes a P value P_j for each gene j for testing the null hypothesis $H_{0j}^{\mathrm{DD}} : \mu_{0j}^{\mathrm{DD}} = \mu_{1j}^{\mathrm{DD}}$, where $\mu_{0j}^{\mathrm{DD}} = \mathbb{E}[Y_{ij} | \widehat{Z}_i = 0]$ and $\mu_{1j}^{\mathrm{DD}} = \mathbb{E}[Y_{ij} | \widehat{Z}_i = 1]$. Then the target DE score of gene j is defined as $S_j := -\log_{10} P_j$.

In parallel, on the synthetic null data, ClusterDE calculates a P value $\widetilde{P}_j$ for each gene j for testing the null hypothesis $\widetilde{H}_{0j}^{\mathrm{DD}} : \widetilde{\mu}_{0j}^{\mathrm{DD}} = \widetilde{\mu}_{1j}^{\mathrm{DD}}$, where $\widetilde{\mu}_{0j}^{\mathrm{DD}} = \mathbb{E}[\widetilde{Y}_{ij} | \widetilde{Z}_i = 0]$ and $\widetilde{\mu}_{1j}^{\mathrm{DD}} = \mathbb{E}[\widetilde{Y}_{ij} | \widetilde{Z}_i = 1]$. Then the null DE score of gene j is defined as $\widetilde{S}_j := -\log_{10} \widetilde{P}_j$.

In summary, the m genes have the target DE scores $S_1, \ldots, S_m$ and the null DE scores $\widetilde{S}_1, \ldots, \widetilde{S}_m$.

3.2.5 ClusterDE step 4: FDR control

Given the target DE scores $S_1, \ldots, S_m$ and the null DE scores $\widetilde{S}_1, \ldots, \widetilde{S}_m$, we use the FDR-control method Clipper to identify DE genes given a target FDR threshold $q \in (0, 1)$ [69]. Given a set of identified DE genes, the FDR is defined as

$$\mathrm{FDR} := \mathbb{E}\left[\frac{\#\text{ false discoveries}}{\#\text{ discoveries } \vee 1}\right],$$

where $a \vee b$ is defined as the maximum of two numbers a and b.

To ensure a valid FDR control, Clipper requires each gene to have a contrast score such that the true non-DE genes have contrast scores symmetric about zero. In ClusterDE, gene j's contrast score C_j is defined as

$$C_j := S_j - \widetilde{S}_j \,.$$

Then ClusterDE uses Clipper to find a contrast score cutoff T within $\mathcal{C}$ (i.e., the set of non-zero contrast score values) given the target FDR threshold q:

$$T := \min \left\{ t \in \mathcal{C} : \frac{|\{j : C_j \leqslant -t\}| + 1}{|\{j : C_j \geqslant t\}| \vee 1} \leqslant q \right\}$$

and outputs $\{j \in \{1, \cdots, m\} : C_j \geqslant T\}$ as discoveries. Here $|A|$ defines the size of a set A. The FDR control of this contrast-score thresholding procedure was from the knockoffs method [76].

Under the assumption that the majority of genes are non-DE genes, we would expect that the distribution of all genes' contrast scores has a mode at zero, so the symmetry requirement of Clipper is satisfied. That is, in the ideal scenario, slightly less than 50% of all genes' contrast scores should be negative. However, in some real data scenarios, this symmetry requirement might not hold. For example, the contrast score distribution might have a positive mode such that too few contrast scores are negative, leading to inflated false discoveries made by Clipper. Or it could be that the contrast score distribution has a negative mode such that too many contrast scores are negative, leading to a loss of statistical power. Hence, in practice, ClusterDE verifies the symmetry assumption by employing Yuen's trimmed mean test (using the function `yuen.t.test()` from the R package `PariedData` (version 1.1.1)). This test examines the symmetry of the contrast score distribution after excluding the smallest 10% and largest 10% of the contrast scores.

If symmetry is rejected by Yuen's trimmed mean test, ClusterDE applies an adjustment to the contrast score distribution so that the symmetry requirement can approximately hold.

In detail, ClusterDE applies the "robust fitting of linear models" (using the function `rlm()` from the R package `MASS` (version 7.3-60)) to adjust the null DE scores; that is, a linear model is fitted between the target DE scores (the response variable y) and the null DE scores (the explanatory variable x), and the fitted values (the predicted response variable $\hat{y}$) are taken as the adjusted null DE scores. Then the adjusted contrast scores, defined as the differences between the target DE scores and the adjusted null DE scores, would better satisfy the symmetry requirement.

Since we would like to be conservative regarding the adjustment of contrast scores, ClusterDE uses the one-sided ("greater than") Yuen's trimmed mean test at the significance level of 0.001. Hence, adjustment is performed only when too few contrast scores are negative, a scenario that would lead to inflated false discoveries made by Clipper.

3.3 Results

3.3.1 ClusterDE uses a contrastive strategy to identify reliable DE genes robust to double dipping

The ClusterDE test consists of four major steps (Fig. 3.1b), with its core idea being to establish a negative control for the entire computational pipeline that includes cell clustering followed by DE analysis. This contrastive strategy enables the identification of trustworthy DE genes by comparing the result from real-data analysis with that from the negative-control analysis. To implement this strategy, we introduce a null model that assumes the cells of interest (i.e., the cells divided into two clusters and subject to DE analysis, referred to as the "target data") are from a homogeneous cell type, where no between-cell-type DE genes should be detected.

In step 1 of ClusterDE, we use the model-based simulator scDesign3 [68] to generate "synthetic null data" that mimic the target data but represent a homogeneous cell type, with the same number of cells and the same genes as in the target data. Fig. 3.2 illustrates the synthetic null generation process, with the mathematical details described in section Clus-

terDE methodology. Fig. 3.1c and Fig. 3.4 show that the synthetic null data preserve the per-gene mean and variance statistics, as well as the gene-gene correlations in the target data. Meanwhile, irrespective of the clustering pattern in the target data, the synthetic null data exhibit a homogeneous cell cluster, which is specified as the "null model" for a single cell type in scDesign3.

In steps 2 and 3 of ClusterDE, users have the flexibility to specify a clustering algorithm and a DE test, respectively, to analyze the target data and the synthetic null data in parallel. For example, users may use the Seurat pipeline for clustering and DE analysis. These two steps yield a "target DE score" and a "null DE score" for each gene. Specifically, we define a gene's DE score as a summary statistic measuring the difference of the gene's expression values in two clusters; a higher DE score indicates that the gene is more likely DE. For example, the DE score is by default defined as the negative logarithm of the P value obtained from a statistical DE test (e.g., the Wilcoxon rank-sum test).

Finally, in step 4 of ClusterDE, a "contrast core" is computed for each gene by subtracting the gene's null DE score from its target DE score. True non-DE genes are expected to have contrast scores symmetrically distributed around 0. Then ClusterDE uses the FDR control method Clipper [69] to determine a contrast score cutoff corresponding to a target FDR (e.g., 0.05). Genes with contrast scores greater than or equal to the cutoff are identified as DE genes.

3.3.2 ClusterDE achieves reliable FDR control and good statistical power under double dipping

We conducted extensive simulation studies to validate ClusterDE as a post-clustering DE method with reliable FDR control under double dipping. We also compared ClusterDE with Seurat, the most widely used analysis pipeline that involves double dipping, and two existing methods that attempted to address the double-dipping issue—the TN test [17] and Countsplit [18]. In the cell clustering step of all four methods, we used the default Seurat clustering as in most scRNA-seq data analyses. In the DE analysis step of ClusterDE, Seurat,

and Countsplit, we considered five DE tests in the `Seurat` package: the Wilcoxon rank-sum test (Wilcoxon; the default option in the `Seurat` package), the two-sample t test (t-test), the negative binomial generalized linear model (NB-GLM), the logistic regression (LR), and the likelihood-ratio test (bimod). The TN test is an exception because it uses its own TN test in the DE analysis step. As Seurat, Countsplit, and the TN test all output a P value for each gene, we applied the Benjamini-Hochberg (BH) procedure to all genes' P values to find a P value cutoff given a target FDR (e.g., 0.05). Genes with P values less than or equal to the cutoff are identified as DE genes.

In the first simulation setting, which represents the most severe double-dipping scenario, we simulated the target data from a single cell type by mimicking the naïve cytotoxic T cells in a real dataset [77] (Fig. 3.1c top left; see section "Simulation designs"), where any identified DE genes should be considered false discoveries. At the target FDR of 0.05, all three existing methods—Seurat, Countsplit, and the TN test—were unable to control the actual FDR under 0.05 (Fig. 3.1d). As expected, the double-dipping approach employed by Seurat exhibited the worst performance, with all five DE tests yielding actual FDRs of 1. Although Countsplit and the TN test were designed to overcome the FDR inflation issue caused by double dipping, their actual FDRs still far exceeded 0.05. The reason is that their P values are anti-conservative in the presence of gene-gene correlations (Fig. 3.5 right), although their own simulation studies verified their P value validity under unrealistic settings where genes are assumed to be independent [17, 18]. In contrast, ClusterDE successfully controlled the FDRs under 0.05 for three out of the five DE tests: Wilcoxon, t-test, and LR (Fig. 3.1d). We verified that the contrast scores calculated in step 4 of ClusterDE satisfied the symmetry requirement around zero (Fig. 3.5 left). Although ClusterDE did not control the actual FDRs of the NB-GLM and bimod tests under 0.05 due to possible violations of these two tests' parametric modeling assumptions on this dataset, the FDR inflation of ClusterDE for these two tests was much less severe than that of Countsplit (ClusterDE's actual FDRs 0.28 and 0.16 vs. Countsplit's actual FDRs 0.68 and 1 for NB-GLM and bimod, respectively) (Fig. 3.1d).

In the second simulation setting, we generated datasets with varying degrees of double

dipping, still by mimicking the naïve cytoxic T cells in a real dataset [77] (Fig. 3.1e top; see section "Simulation designs"). Each dataset consists of two synthetic cell types with pre-specified 200 true DE genes with varying expression level differences between the cell types, and the overall difference is summarized as the log fold change (logFC). A larger logFC indicates a greater distinction between the two cell types. After the default Seurat clustering algorithm is applied to each dataset to identify two cell clusters, the agreement between the cell clusters and the cell types is measured by the adjusted Rand index (ARI). A smaller ARI represents a more severe degree of double dipping, as illustrated by the UMAP visualizations (Fig. 3.1e top row). Since Wilcoxon is the default DE test in Seurat and yielded the best FDR control for both ClusterDE and Countsplit, we used Wilcoxon as the DE test in ClusterDE, Seurat, and Countsplit, while the TN test uses its own DE test. The results in Fig. 3.1e show that ClusterDE consistently controlled the actual FDRs across a range of target FDR thresholds under varying degrees of double dipping. In contrast, Seurat, Countsplit, and the TN test failed to control the actual FDRs under the target thresholds, and as expected, exhibited greater FDR inflation when the degree of double dipping is more severe (Fig. 3.1e middle row). Notably, ClusterDE achieved comparable or superior statistical power to Seurat, Countsplit, and the TN test at the same actual FDR levels (Fig. 3.1e bottom). These conclusions remained to hold when ClusterDE, Seurat, and Countsplit were used with the other four DE tests (t-test, NB-GLM, LR, and bimod) in the DE analysis step (Fig. 3.8). Moreover, to reflect the fact that cell types mostly have unbalanced cell numbers in real data, we further simulated target data in which the two synthetic cell types have size ratios of $1:4$ and $1:9$. In these two unbalanced scenarios, we still found ClusterDE to outperform the other three methods in terms of FDR control across target FDR thresholds and under varying degrees of double dipping. In particular, ClusterDE consistently exhibited solid FDR control and comparable or superior statistical power to the other three methods when used with Wilcoxon as the DE test (Fig. 3.9–Fig. 3.10).

Technically, ClusterDE shares with the knockoffs methods the concept of controlling the FDR by generating *in silico* negative control data [76]. The knockoffs methods are a suite of statistical methods developed for identifying important features in a high-dimensional pre-

dictive model, a supervised learning setting different from our one-test-per-gene test setting. Roughly, the knockoffs methods generate knockoff data from real data in such a way that each feature is no longer correlated with the outcome variable given the other features, while the feature-feature correlations are preserved in the knockoff data. We applied the default model-X knockoffs method [78] to the simulated datasets—treating genes as features and the cell cluster label as the outcome variable; the results indicate that, although this method controlled the FDR, it always led to zero statistical power, making it impractical for DE gene identification. Moreover, we used the model-X knockoffs method and permutations (where each gene is independently permuted across all cells) as two alternative strategies to scDesign3 for the synthetic null generation in step 1 of ClusterDE, followed by steps 2–4 of ClusterDE. Fig. 3.11 shows a comparison of the target data with the synthetic null data generated by each of the three strategies. Compared with the target data, the synthetic null data generated by scDesign3 preserved per-gene mean and variance statistics and gene-gene correlations. In contrast, the synthetic null data generated by the model-X knockoffs method did not preserve gene mean and variance statistics, and the synthetic null data generated by permutations did not preserve gene-gene correlations. Hence, only the synthetic null cells generated by scDesign3 preserved the 2D UMAP cell embedding topology of the target cells except for filling the gap, if existent, between the target cell types. Our results on the simulated datasets demonstrate that scDesign3 led to the most solid FDR control and the best statistical power among the three strategies for synthetic null generation (Fig. 3.6).

To address the practical concern about the randomness involved in generating synthetic null data (a random sampling process from the null model fitted on target data), we conducted an analysis to assess the robustness of DE genes identified by ClusterDE. The results show that the DE genes identified by ClusterDE remain relatively stable and robust to the randomness (Fig. 3.12).

In summary, the above simulation studies confirm that ClusterDE is a flexible and stable method that effectively controls the FDR under varying degrees of double dipping while maintaining good statistical power.

3.3.3 ClusterDE identifies cell-type marker genes and excludes housekeeping genes from its top DE genes

We applied ClusterDE to multiple real scRNA-seq datasets of different types to demonstrate how it can enhance the statistical rigor and biological relevance of findings from the post-clustering DE analysis. The following real data applications showcase the effectiveness of ClusterDE in identifying meaningful DE genes and improving the reliability of DE gene identification.

In the first application, we collected five datasets of pure cell lines [79, 80], so the cells in each dataset can be trusted as a homogeneous population that should not be divided into more than one cluster (Fig. 3.3a left). Hence, any post-clustering DE genes identified from these datasets should not be interpreted as between-cell-type DE genes. We used these five datasets as real-data negative examples to demonstrate the inflated FDRs of existing methods and the effectiveness of ClusterDE in removing the FDR inflation. As a sanity check of ClusterDE, we first verified that the synthetic null data resembled the target data (Fig. 3.3a right). Applying ClusterDE, Seurat, Countsplit, and the TN test to the five datasets, we found that all methods except ClusterDE identified thousands of DE genes, in many cases even more than 50% of all genes, indicating severely inflated false discoveries at the target FDR of 5%. In contrast, ClusterDE found zero DE genes in 22 out of 25 cases when used with the five DE tests (Wilcoxon, t-test, NB-GLM, LR, and bimod) on the five datasets. In particular, ClusterDE with Wilcoxon consistently found zero DE genes from the five datasets. Hence, we set Wilcoxon as the default DE test in ClusterDE.

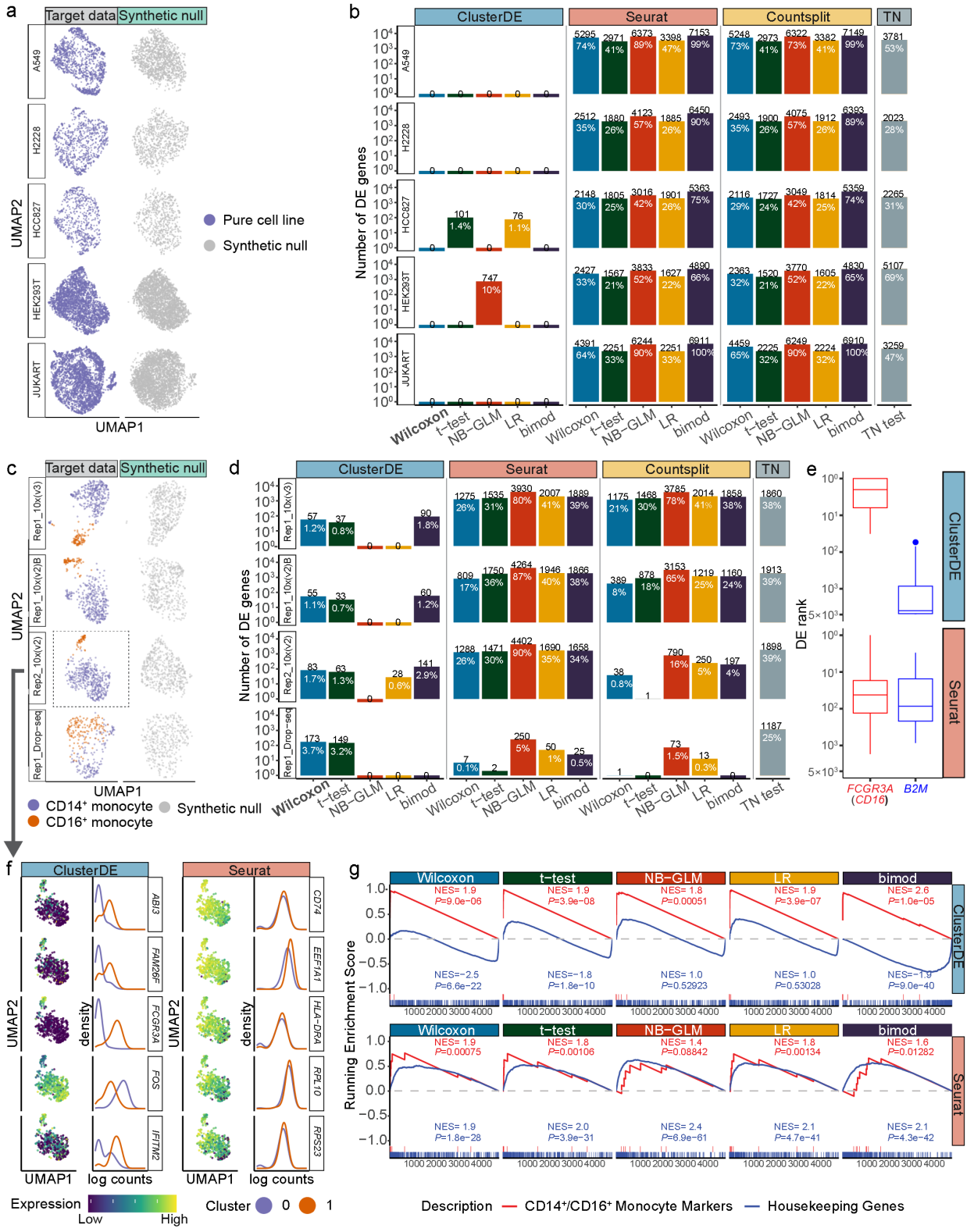


Figure 3.3: ClusterDE achieves reliable FDR control and good statistical power in identifying DE genes from real scRNA-seq data.

a, UMAP visualizations of target data (left) and synthetic null data (right) of five cell lines. **b**, Numbers of DE genes (at the target FDR of 0.05) identified by ClusterDE and the existing methods. While all existing methods found numerous "false" DE genes within a single cell line, ClusterDE made no false discoveries when used with most DE tests. The numbers in black and white indicate the number of DE genes and the proportions of DE genes among all genes, respectively. The five DE tests are the Wilcoxon rank sum test (Wilcoxon), t-test, negative binomial generalized linear model (NB-GLM), logistic regression model predicting cluster membership with likelihood-ratio test (LR), and likelihood-ratio test for single-cell gene expression (bimod). **c**, UMAP visualizations of target data (left) and synthetic null data (right) for four datasets containing two monocyte subtypes: $CD14^+$ monocytes and $CD16^+$ monocytes. The synthetic null data captured the global topology of the real cells in the target data while filling the gap between the two cell subtypes. The grey dashed box labels the dataset used in f and g. **d**, ClusterDE identified DE genes between the two cell subtypes. The numbers in black and white indicate the number of DE genes and the proportions of DE genes among all genes, respectively. **e**, The ranks of two exemplary genes (a monocyte subtype marker *FCGR3A* in red and a well-known housekeeping gene *B2M* in blue) in the DE gene lists of ClusterDE and Seurat across the five DE tests and the four datasets in c. In each boxplot representing the distribution of 20 ranks, the center horizontal line represents the median, and the box limits represent the upper and lower quartiles. **f**, The top DE genes identified by ClusterDE exhibited distinct expression patterns in the two cell clusters identified by Seurat clustering, a phenomenon not observed for the top DE genes identified by Seurat. For ClusterDE and Seurat, the top DE genes are defined as the common DE genes found by the five DE tests in d at the target FDR of 0.05. The UMAP plots show each top DE gene's normalized expression levels in the dataset "Rep2_10x(V2)" (marked by the dashed box in c; see section Real data analysis "Dimensionality reduction and visualization "). The density plots depict each top DE gene's normalized expression distributions in the two cell clusters. **g**, Gene set enrichment analysis (GSEA) of the ranked DE gene lists identified by ClusterDE and Seurat with five DE tests from the dataset Rep2_10x(V2). The red lines represent the enrichment of the $CD14^+/CD16^+$ monocyte marker gene set, and the blue lines represent the enrichment of the housekeeping gene set. The normalized enrichment score (NES) reflects the direction and magnitude of enrichment, and the P value indicates the significance of enrichment.

In the second application, we collected eight PBMC datasets of $CD14^+/CD16^+$ monocytes [81] to demonstrate that ClusterDE can effectively detect known or potential marker genes of the two cell subtypes. The eight datasets were generated from two technical replicates by four unique molecular identifier (UMI) based scRNA-seq protocols (10X Genomics Versions 2 and 3, Drop-seq, and inDrop). After applying the default Seurat clustering to identify two clusters in each of the eight datasets, we found four datasets to have relatively accurate clustering results (ARI > 0.5; Fig. 3.3c left, Fig. 3.13 top), while the other four datasets had clusters poorly matched with the two monocyte subtypes (ARI < 0.2; Fig. 3.13 bottom). Hence, we expected that an effective post-clustering DE method would be able to detect meaningful marker genes for monocyte subtypes in the first four datasets, but we did not expect the same level of effectiveness for the latter four datasets. Hence, we focused on the analysis results of the first four datasets. As a sanity check of ClusterDE, we first verified that the synthetic null data resembled the target data but had the gap filled between $CD14^+$ monocytes and $CD16^+$ monocytes, representing a single "hypothetical" cell type in each dataset (Fig. 3.3c right). Applied to the first four datasets with relatively accurate clustering results, ClusterDE with Wilcoxon identified 55–173 DE genes (1–4% of all genes) at the target FDR of 5%, while Seurat and Countsplit identified 1–1,288 DE genes (0–26% of all genes), and the TN test consistently identified at least 1,187 genes (25% of all genes) (Fig. 3.3d). Given our knowledge that the two monocyte subtypes are not drastically different, we did not expect thousands of genes to be identified as potential subtype marker genes. Hence, we deemed the number of DE genes identified by ClusterDE to be more reasonable.

Examining the post-clustering DE genes identified by ClusterDE or Seurat across the five DE tests on the four datasets (so ClusterDE and Seurat each had 20 DE gene lists), we found that ClusterDE better distinguished known subtype marker genes from housekeeping genes than Seurat did. This distinction was evident in the ranking of specific genes in the DE gene lists. For example, we considered the genes *FCGR3A* (*CD16*), a canonical marker for distinguishing $CD14^+$ monocytes and $CD16^+$ monocytes, and *B2M*, a widely recognized housekeeping gene expressed across various cell types [82]. Notably, ClusterDE consistently ranked *FCGR3A* among its top DE genes (with ranks approximately between 1 and 10) while

placing *B2M* consistently low in its DE gene lists (with ranks below 1,000 in most cases) (Fig. 3.3e top). In contrast, Seurat ranked the two genes similarly (with ranks between 10 and 100) in its DE gene lists (Fig. 3.3e bottom), making it impossible to discern which of the two genes is more likely a subtype marker without prior knowledge.

Next, using one of the four datasets "Rep2_10x(V2)" as an example, we examined the five most frequently identified post-clustering DE genes (defined based on the top 50 DE genes identified by each of the five DE tests) by ClusterDE or Seurat (Fig. 3.3f). Again, the two clusters were found by the default Seurat clustering, and ClusterDE and Seurat both used these two clusters for post-clustering DE analysis. Our analysis found that the five genes identified by ClusterDE all exhibited distinct distributions of normalized expression levels between the two clusters, while the five genes identified by Seurat all had almost indistinguishable distributions between the two clusters (Fig. 3.3f). Further, we examined the enrichment of two gene sets—known $CD14^+/CD16^+$ monocyte markers and housekeeping genes—in the post-clustering DE gene lists outputted by ClusterDE and Seurat. The gene set enrichment analysis (GSEA) revealed that the known monocyte markers had strong enrichment in the top-ranked DE genes identified by ClusterDE, exhibiting a clear distinction from the housekeeping genes (Fig. 3.3g top). In contrast, the monocyte makers exhibited less enrichment in the top-ranked DE genes identified by Seurat; what is worse, they had a similar enrichment trend as the housekeeping genes, indicating that Seurat had the monocyte markers and the housekeeping genes hardly distinguishable in its ranked DE gene list (Fig. 3.3g bottom). The GSEA results on the other three datasets confirmed that ClusterDE better distinguished the monocyte markers from the housekeeping genes than Seurat (Fig. 3.15).

Considering the common analysis practice that only the top k DE genes (e.g., $k = 100$) are used for further investigation, we summarized the numbers of monocyte markers and housekeeping genes among the top $k = 1$ to 100 DE genes identified by ClusterDE or Seurat across the five DE tests on the four datasets. Fig. 3.16 shows that ClusterDE found more monocyte markers and fewer housekeeping genes among the top DE genes than Seurat. To further explain why ClusterDE can better distinguish monocyte markers and housekeeping

genes, we used the minus-average (MA) plots [83] to demonstrate the effectiveness of using synthetic null as a contrast to remove housekeeping genes from the top DE genes. From the MA plots (Fig. 3.17), we observed that four exemplary housekeeping genes (*ACTB*, *ACTG1*, *B2M*, and *GAPDH*; marked in blue in Fig. 3.17) had both large target DE scores and large null DE scores, resulting in close-to-zero contrast scores, so these genes were not found by ClusterDE as top DE genes. However, these four genes were found by Seurat as top DE genes due to their large target DE scores. On the other hand, we examined four exemplary monocyte markers (*CD14*, *FCGR3A*, *MS4A7*, and *LYZ*; marked in red in Fig. 3.17) and found them to have large target DE scores but small null DE scores, so they were identified as top DE genes by ClusterDE.

3.4 FDR control theory of ClusterDE

In this section, we prove why ClusterDE asymptotically controls the FDR and avoids the FDR inflation issue due to double dipping. Recall that ClusterDE is based on the single-cell simulator scDesign3 [68] for generating the in silico negative control data and the P-value-free FDR control framework Clipper [69] for finding DE genes.

To achieve the FDR control of ClusterDE, the summary statistics from the original dataset $\mathbf{T} = (t_1, \ldots, t_m)$, and the summary statistics from the negative control dataset $\mathbf{T}' = (t_1', \ldots, t_m')$ should satisfy the following assumptions:

Assumption 1. *For each non-DE gene j, the distribution of C_j is symmetric around 0.*

Assumption 2. *We use $\mathcal{N}$ to denote the set of non-DE genes, and define $m_0 = \text{card}(\mathcal{N})$. We assume that the contrast scores $\{C_1, \ldots, C_m\}$ are continuous random variables, and there exist a constant $c > 0$, and $\alpha \in (0, 2)$ such that*

$$\text{Var}\left(\sum_{j \in \mathcal{N}} \mathbb{1}(C_j > t)\right) \leqslant c m_0^{\alpha}, \forall t \in \mathcal{R}$$

Assumption 2 only restricts the correlation among the non-DE genes. We note that if the contrast scores have constant pairwise correlation, or can be clustered into a fixed number

of groups so that their within-group correlation is a constant, α has to be 2 and assumption 2 does not hold. Except for these extreme cases, assumption 2 holds in broad settings. For example, if the features are independent:

$$\mathrm{Var}\left(\sum_{j\in\mathcal{N}} \mathbb{1}(C_j > t)\right) = \sum_{j\in\mathcal{N}} \mathrm{Var}\left(\mathbb{1}(C_j > t)\right) \leqslant cm_0$$

Here we propose an assumption that is stronger than assumption 2, but enjoys a better interpretability:

Assumption 3. *Let r denote the number of contrast scores which has a non-zero correlation with at least another contrast score. We assume that* $\lim_{m\to\infty} r/m_0 \to 0$.

Intuitively we can interpret the assumed correlation structure as a matrix whose most off-diagonal entries are 0 except for a small block. We can prove that Assumption 2 holds if Assumption 3 holds:

Let $\mathcal{M} \subset \mathcal{N}$ denote the genes within the correlated small block, $\mathrm{card}(\mathcal{M}) = r$.

$$\begin{aligned}\mathrm{Var}\left(\sum_{j\in\mathcal{N}} \mathbb{1}(C_j > t)\right) &= \mathrm{Var}\left(\sum_{j\in\mathcal{M}} \mathbb{1}(C_j > t)\right) + \mathrm{Var}\left(\sum_{j\in\mathcal{N}\setminus\mathcal{M}} \mathbb{1}(C_j > t)\right) \\ &\leqslant c_1 r^2 + c_2(m_0 - r)\end{aligned}$$

Theorem 1. *We define* $FDP(t) = \frac{\mathrm{card}(\{j\in\mathcal{N}: C_j \geqslant t\})}{\mathrm{card}(\{j: C_j \geqslant t\}) \vee 1}$, $\widehat{FDP}(t) = \frac{\mathrm{card}(\{j: C_j \leqslant -t\})}{\mathrm{card}(\{j: C_j \geqslant t\}) \vee 1}$, $\tau_q = \min\{t > 0 : \widehat{FDP}(t) \leqslant q\}$. *For any FDR threshold $q \in (0,1)$, assume that there exist a constant $t_q > 0$ such that $\mathbb{P}(FDP(t_q) \leqslant q) \to 1$ as $m \to \infty$. Then under Assumptions 1 and 2's:*

$$FDP(\tau_q) \leqslant q + o_m(1)$$

$$\limsup_{m\to\infty} FDR(\tau_q) \leqslant q$$

The existence of $t_q > 0$ such that $\mathbb{P}(FDP(t_q) \leqslant q) \to 1$ as $m \to \infty$ implies that the data-dependent cutoff τ_q is bounded with probability approaching 1, thus does not diverge

to infinity. For the detailed proof of Theorem 1, please see section S3.8.6. We prove that the contrast score in ClusterDE satisfies the assumptions for mirror statistics in data splitting. Thus, this FDR control theory also applies to ClusterDE.

3.5 Discussion

In conclusion, ClusterDE is an effective solution to the double-dipping issue in post-clustering DE analysis. We note that ClusterDE focuses on identifying potential cell-type marker genes for cell-type annotation, so ClusterDE does not aim to capture the within-cell-type heterogeneity that reflects continuous cell state changes. Notably, ClusterDE adapts to a wide range of clustering algorithms and DE tests. Through extensive simulation studies and real data analysis, we demonstrated that ClusterDE effectively avoids false discoveries caused by double dipping and identifies biologically meaningful cell-type markers. For post-clustering DE analysis with more than two clusters, we recommend using ClusterDE in a stepwise manner, possibly following a cell cluster hierarchy constructed based on cluster similarities (Fig. 3.18). That is, users compare a pair of ambiguous clusters at each step, so the post-clustering DE genes can be used to decide whether the two clusters are biologically meaningful and should be distinct. Finally, while ClusterDE focuses on the double-dipping problem in the post-clustering DE analysis, the concept of synthetic null data (*in silico* negative control) can be readily extended to other analyses also affected by double dipping, such as post-pseudotime DE analysis [84] and data integration analysis. As double dipping is almost surely unavoidable in single-cell data analysis due to the lack of external knowledge, we proposed a general strategy to reduce false discoveries caused by double dipping by setting up synthetic null data and using a contrastive strategy to find more reliable discoveries.

3.6 Code and data availability

The R package ClusterDE is available at `https://github.com/SONGDONGYUAN1994/ClusterDE`. The tutorials of ClusterDE are available at `https://songdongyuan1994.github.io/ClusterDE/`

docs/index.html. The source code and data for reproducing the results are available at: http://doi.org/10.5281/zenodo.8161964 [85]. The pre-processed datasets are available at https://figshare.com/articles/dataset/ClusterDE_datasets/23596764.

3.7 Acknowledgments

This chapter is based on my joint work with Dongyuan Song, Dr. Xinzhou Ge, and my Ph.D. advisor Dr. Jingyi Jessica Li. Firstly, thanks Dongyuan for the development of scDesign3, and thanks Xinzhou for the development of Clipper, which are the two key fragments of the ClusterDE method. For the methodology development, we made equal contributions: Xinzhou provided instructions on applying Clipper, Dongyuan made suggestions on the implementation of scDesign3 and the generation of synthetic null, and I mainly implemented different variants of the ClusterDE method. For the results, I contributed to the simulation studies, and Dongyuan contributed to real data studies.

3.8 Supplementary materials

S3.8.1 Practical guidelines for ClusterDE usage

ClusterDE is designed to find potential cell-type marker genes via pairwise comparisons of cell clusters that might be ambiguous. In practice, we recommend using ClusterDE in the following steps.

1. Given a set of cell clusters, find two clusters that may be defined as potential cell types or subtypes. If users use Seurat, they may use the function `BuildClusterTree` to construct a hierarchy of the clusters and examine two leaf clusters whose distinctions are ambiguous.

2. Given the two chosen cell clusters, construct a data subset that contains only the cells in these two clusters.

3. Input the data subset as the "target data" into ClusterDE.

4. Examine the DE genes outputted by ClusterDE and decide whether the two cell clusters are biologically meaningful cell types or subtypes.

It is worth noting that ClusterDE does not provide an automatic decision about whether two clusters should be merged, unlike the methods that directly assess the quality of clusters [1–5]. Instead, ClusterDE focuses on identifying trustworthy post-clustering DE genes as potential cell-type marker genes, enabling researchers to gain biological insights into clusters by investigating the specific genes that distinguish the clusters. Hence, in contrast to the clustering quality assessment methods, ClusterDE empowers researchers to explore the functional and molecular characteristics of clusters.

Specifically, in step 3 of the above procedure, users have the option to input the cell cluster labels in the target data (the default option in ClusterDE), or they can allow the target data to be re-clustered by ClusterDE. If the default option is used, then ClusterDE performs clustering on the synthetic null data only, and the target DE scores will be calculated based

on the input cell clusters. Otherwise, ClusterDE performs clustering on the target data and the synthetic null data in parallel, but the downside of this approach is that the target cell clusters might not be identical to the input cell clusters of users' interest.

S3.8.2 Simulation designs

To benchmark post-clustering DE methods in terms of the FDR and statistical power, we needed ground truths of DE genes and non-DE genes. Hence, we used the R package `scDesign3` (version 0.99.0) [6] to generate realistic synthetic scRNA-seq data containing true DE genes and non-DE genes, based on the model parameters learned from real scRNA-seq data. Under each simulation setting, we generated 200 synthetic replicates.

For each replicate, we simulated a dataset with $n = 998$ cells and $m = 9239$ genes, the same dimensions as those of the naïve cytotoxic T cells in the Zhengmix4eq dataset [7] after the default Seurat preprocessing step that removed the genes expressed at very low levels. In the following, we let i and j denote the indices of cells and genes, respectively, $i = 1, \ldots, n$; $j = 1, \ldots, m$.

The first step was to estimate the following model parameters from the naïve cytotoxic T cells in the Zhengmix4eq dataset by scDesign3 [6]. For details of the model formulation, please refer to the previous section ClusterDE step 1: synthetic null generation.

- Per-gene NB mean parameter $\mu_j \in \mathbb{R}^+$, $j = 1, \cdots, m$;
- Per-gene NB dispersion parameter $\sigma_j \in \mathbb{R}^+$, $j = 1, \cdots, m$;
- Gene-gene Gaussian copula correlation matrix $\mathbf{R} \in [0, 1]^{m \times m}$.

Given the mode model parameters include $\{\widehat{\mu}_j, \widehat{\sigma}_j\}_{j=1}^m$ and $\widehat{\mathbf{R}}$, the next steps belonged to two settings: (1) one cell type with zero true DE genes; (2) two cell types with 200 true DE genes.

Simulation setting with one cell type and zero true DE genes

All of the $n = 998$ cells were simulated from one cell type with an MVNB distribution specified by the Gaussian copula, whose correlation matrix was $\widehat{\mathbf{R}}$, so gene j's counts followed the NB distribution with mean $\widehat{\mu}_j$ and dispersion $\widehat{\sigma}_j$, $j = 1, \cdots, m$. For the simulation details, please refer to the previous section ClusterDE step 1: synthetic null generation. The R package `scDesign3` [6] was used to simulate a cell-by-gene count matrix $\mathbf{Y} \in \mathbb{N}_{\geqslant 0}^{n \times m}$, which was used as the one-cell-type target data (Fig. 3.1c) in simulation studies.

Simulation setting with two cell types and 200 true DE genes

All of the $n = 998$ cells were designed to belong to two cell types, with each cell type having its own MVNB distribution specified by the Gaussian copula. For each replicate, we randomly specified 200 true DE genes to have different mean parameters μ_j^0 and μ_j^1 based on the estimate $\widehat{\mu}_j$.

For the two cell types, we simulated three cell-type size ratios $r \in \{1, 4, 9\}$ such that cells $i = 1, \ldots, \left[\frac{n}{r+1}\right]$ were designed to be of cell type 0, and cells $i = \left[\frac{n}{r+1}\right] + 1, \ldots, n$ were designed to be of cell type 1. For each replicate, the set 200 true DE genes were specified with the index set $J_{\mathrm{DE}} \subset \{1, \cdots, m\}$.

For each specified true DE gene $j \in J_{\mathrm{DE}}$, we set its mean parameter in cell type 0 as the estimate, i.e., $\mu_j^0 = \widehat{\mu}_j$. Then we modified its mean parameter in cell type 1, μ_j^1, using a pre-specified log fold change logFC with a 50% probability of up-regulation and a 50% probability of down-regulation:

$$\mu_j^1 = \begin{cases} \widehat{\mu}_j \times 2^{\mathrm{logFC}}, & \text{if } Z_j = 1 \\ \widehat{\mu}_j \times 2^{-\mathrm{logFC}}, & \text{if } Z_j = 0 \end{cases}, \quad \text{with } Z_j \sim \mathrm{Ber}(0.5), \quad j \in J_{\mathrm{DE}}\,.$$

For the remaining true non-DE genes, we set

$$\mu_j^0 = \mu_j^1 = \widehat{\mu}_j, \quad j \in \{1, \cdots, m\} \backslash J_{\mathrm{DE}}\,.$$

The parameter logFC determines the differences between the two cell types, and it is expected to have an inverse relationship with the severity level of double dipping (that is, the more different the two cell types, the less severe the double dipping). Hence, we simulated two cell types with a sequence of logFC values

$$\text{logFC} = 1.05, 1.1, \cdots, 1.95, 2, 2.1, \cdots, 2.9, 3.$$

For each logFC value, we simulated cells from cells 0 and 1, each with an MVNB distribution specified by the Gaussian copula, whose correlation matrix was $\widehat{\mathbf{R}}$. That is, gene j's counts in cell types 0 and 1 followed NB distributions with different mean parameters μ_j^0 and μ_j^1, respectively, and the same dispersion parameter $\widehat{\sigma}_j$, $j = 1, \cdots, m$. For the simulation details, please refer to the previous section ClusterDE step 1: synthetic null generation. The R package `scDesign3` [6])was used to simulate a cell-by-gene count matrix $\mathbf{Y} \in \mathbb{N}_{\geqslant 0}^{n \times m}$, which was used as the two-cell-type target data (Fig. 3.4) in simulation studies.

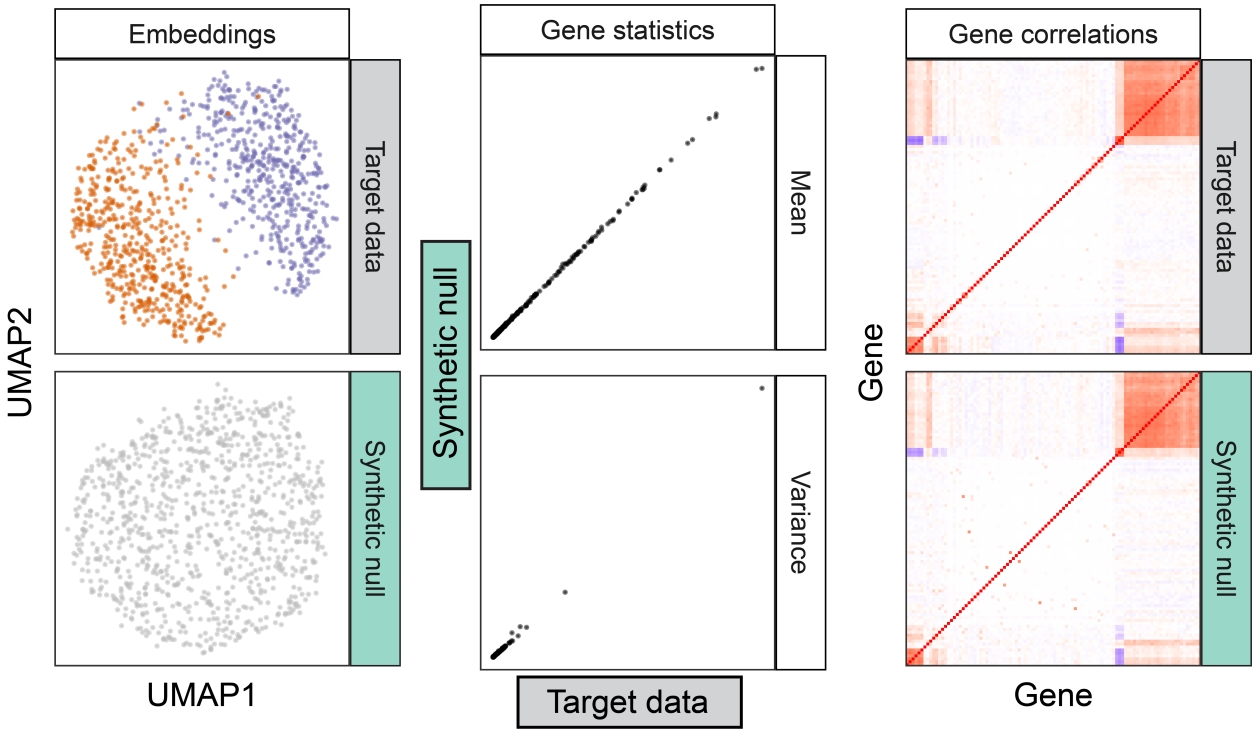


Figure 3.4: When the target data contains cells from two cell types (simulation; see ClusterDE methodology "Simulation setting with one cell type and zero true DE genes"), the synthetic null data generated by ClusterDE fills the gap between the two cell types but resembles the target data in other visual aspects of UMAP cell embeddings (left), per-gene expression mean and variance statistics (middle), and gene-gene correlations.

S3.8.3 Real data analysis

Collection of real data

We collected five scRNA-seq datasets of cell lines, including the three datasets of A549, H2228, and HCC827 from the study [8] and downloaded from the link https://github.com/LuyiTian/sc_mixology/tree/master/data, and the two datasets HEK293T and JUKART from the study [9] and downloaded from https://cf.10xgenomics.com/samples/cell-exp/1.1.0/jurkat/jurkat_filtered_gene_bc_matrices.tar.gz and https://cf.10xgenomics.com/samples/cell-exp/1.1.0/293t/293t_filtered_gene_bc_matrices.tar.gz.

We also collected eight peripheral blood mononuclear cell (PBMC) datasets from the study [10], which were downloaded from https://github.com/satijalab/seurat-data. The datasets were from the same biological sample with two technical replicates (Rep1/Rep2) measured by four protocols (10X Genomics Versions 2 and 3, Drop-seq, and inDrop). In each dataset, we selected the cells with cell type labels "$CD14^+$ monocytes" and "$CD16^+$ monocytes."

Data preprocessing

We filtered out lowly expressed genes. For the cell line datasets A549, H2228, and HCC827, we removed the genes expressed in fewer than 20% cells. For the cell line datasets HEK293T and JUKART, we removed genes expressed in fewer than 10% cells. For PBMC datasets, we removed genes expressed in fewer than 10% of the selected monocyte cells. When performing the default Seurat clustering, Seurat automatically removed the cells with fewer than three genes expressed and the genes expressed in fewer than 200 cells.

Dimensionality reduction and visualization

To visualize the high-dimensional single-cell data, we first applied the PF-logPF transformation to a cell-by-gene count matrix [11]. We then used the R package `irlba` (version 2.3.5.1) to calculate the top 50 principal components (PCs) of the transformed matrix. Next, we

used the R package `umap` (version 0.2.10.0) to project the cells from the 50-dimensional PC space to the 2-dimensional UMAP space.

When comparing the target data and the synthetic null data, we calculated the PCs and UMAPs jointly by concatenating the two datasets so the target cells and synthetic null cells were projected to the same 2-dimensional UMAP space.

We used the R package `ggplot2` (version 3.4.2) to make all plots.

For the UMAP visualizations in Fig. 3.3f, we truncated each gene's normalized expression levels to be below the 99-th percentile to better visualize the gene expression pattern.

Gene set enrichment analysis

We used the R package `clusterProfiler` (4.4.4) to perform the gene set enrichment analysis (GSEA); the test method was `fgsea`, and the number of permutations was 100,000.

The gene set "$CD14^+/CD16^+$ Monocyte Markers" was from the original study [10] and downloaded from https://bitbucket.org/jerry00/scumi-dev/raw/61f7f001d20b2fc8fa7c2f4f4147bff1b0d620d8/R/marker_gene/human_pbmc_marker.rda. The gene set "Housekeeping Genes", `HSIAO_HOUSEKEEPING_GENES`, was downloaded from the Molecular Signature Database (MSigDB); the source study was [12].

Validity checks of the contrast scores of ClusterDE and the P values of Seurat, Countsplit, and the TN test

For ClusterDE, the major assumption is that the contrast scores of true non-DE genes are symmetric around zero. In Fig. 3.5 left, we checked the symmetry of the contrast scores of ClusterDE using the five DE tests (Wilcoxon, t-test, NB-GLM, LR, and bimod; corresponding to Fig. 3.5a–e left) in a simulated one-cell-type dataset where all genes are true non-DE genes (see Simulation designs " Simulation setting with one cell type and zero true DE genes"; the dataset is one of the 200 synthetic replicates).

For Seurat, Countsplit, and the TN test methods, their FDR control validity requires that

the P values of true non-DE genes follow the Uniform$[0, 1]$ distribution. First, we divided the genes in the same simulated dataset into two groups by applying hierarchical clustering (using the default R function `hclust()`) to the estimated correlation matrix $\widehat{\mathbf{R}}$ used in the Gaussian copula. Due to the block pattern of $\widehat{\mathbf{R}}$ (Fig. 3.1c), the two groups include the genes that are highly correlated and those that are not much correlated, respectively. We examined the P values of the genes in the two groups separately. Note that all genes in this simulated dataset are true non-DE genes. In Fig. 3.5 middle and right, we plotted the histograms of the P values and the quantile-quantile plots (Q-Q plots) of the negative log-transformed P values of Seurat, Countsplit (both using the five DE tests; corresponding to Fig. 3.5a–e) and the TN test (using its own test; the same panels plotted five time in Fig. 3.5a–e). We also used the R function `KL.empirical` (from the R package `entropy` (version 1.3.1)) to calculate the empirical Kullback–Leibler divergence (KL div.) between the P value distribution and the theoretical Uniform$[0, 1]$ distribution. A larger Kullback–Leibler divergence value represents a more severe violation of the P value uniformity assumption. The results show that Countsplit and the TN test had close-to-uniform P values in the uncorrelated gene group, but their P values exhibited a severe departure from the uniform distribution in the correlated gene group.

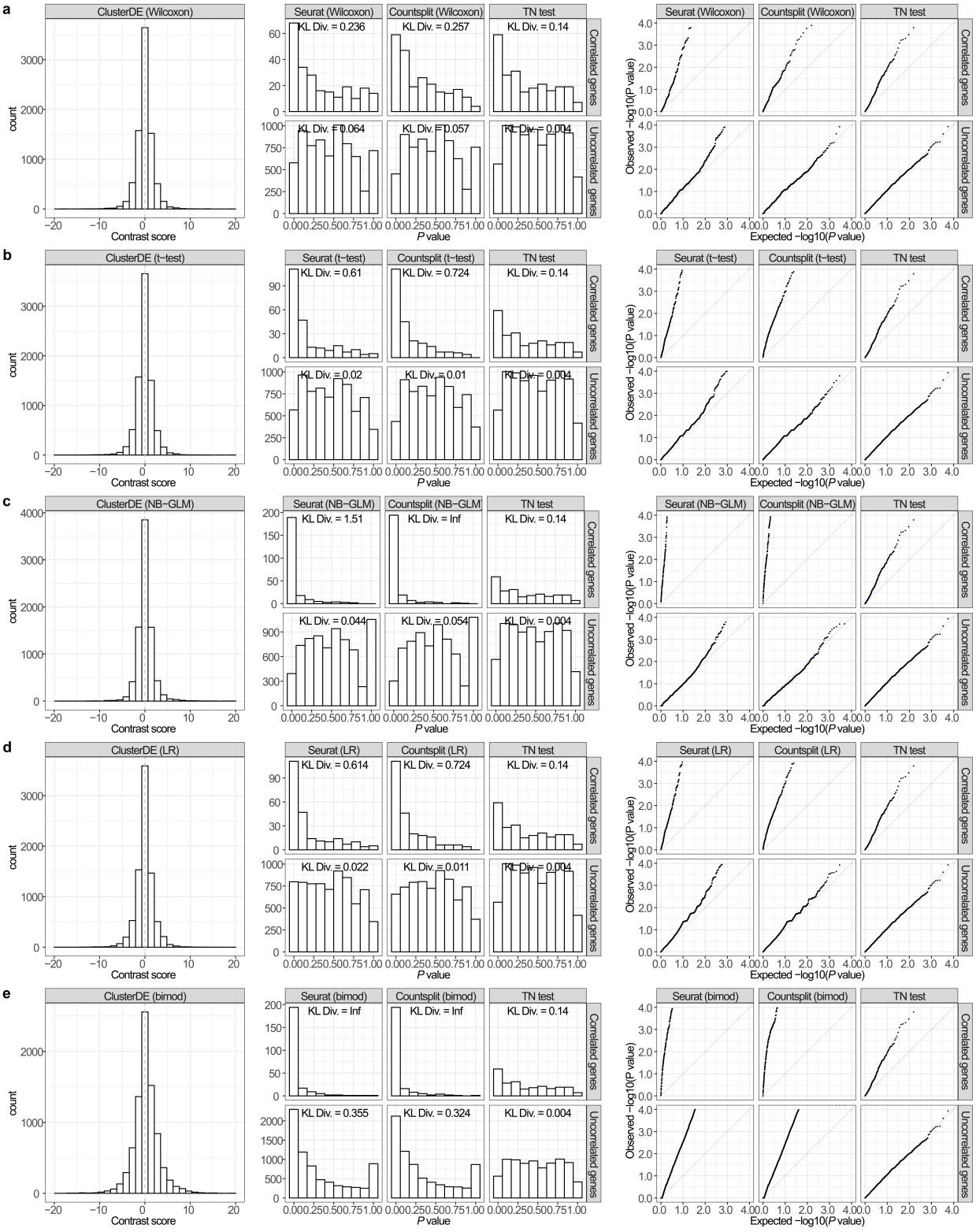


Figure 3.5: Validity checks of the contrast scores of ClusterDE and P values of Seurat, Countsplit, and the TN test on an exemplary one-cell-type dataset, which does not contain any true DE genes by the simulation design (see Simulation designs " Simulation setting with one cell type and zero true DE genes").

The panels (rows) **a**–**e** represent the five DE tests in Seurat used in ClusterDE, Seurat, and Countsplit (see ClusterDE methodology " ClusterDE step 3: DE analysis"); since the TN test has its own DE test, its results are the same in the panels **a**–**e**. The first column shows that the ClusterDE contrast scores of all genes (true non-DE genes) are approximately symmetric around 0, which meets the assumption of ClusterDE for the FDR control. The second column shows the histograms of the P values of the correlated genes (top) and the uncorrelated genes (bottom) from Seurat, Countsplit, and the TN test. A larger empirical Kullback-Leibler divergence (KL div.) between the P value distribution and the theoretical Uniform$[0, 1]$ distribution represents a more severe violation of the P value uniformity assumption. The results show that Countsplit and the TN test have close-to-uniform P values for the uncorrelated genes, but their P values exhibit a severe departure from the uniform distribution for the correlated genes. The third column contains the quantile-quantile plots of the negative log-transformed P values corresponding to the second column.

S3.8.4 Implementation of the TN test and Countsplit

We compared ClusterDE with two existing methods—the TN test [13] and Countsplit [14]—that attempted to address the double-dipping issue in post-clustering DE analysis.

For the TN test, we used the Python module `truncated-normal` (version 0.4). We followed the GitHub tutorial for the implementation (https://github.com/jessemzhang/tn_test/blob/master/experiments/experiments_pbmc3k.ipynb). In the clustering step, we used the same procedure as in ClusterDE step 2. In the DE analysis step, unlike ClusterDE and Countsplit, the TN test has its own DE test, so we did not use any DE tests included in the R package `Seurat` (version 4.2.0).

For Countsplit [14], we used the R package `countsplit` (version 1.0) to split the original count matrix into a training matrix (for clustering) and a test matrix (for DE analysis). In the clustering step, we used the same procedure as in ClusterDE step 2. In the DE analysis step, we used the five DE tests included in the R package `Seurat` (version 4.2.0).

S3.8.5 Alternative strategies for synthetic null generation

Although the model-X knockoffs method was developed for selecting features in a multivariate predictive model (e.g., the Lasso) [15], not for marginal DE tests (where each feature is examined separately), we compared model-X knockoffs to ClusterDE because both methods use the real-data-based negative control idea.

For a direct implementation of the model-X knockoffs method on the post-clustering DE analysis, we used the R package `knockoff` (version 0.3.6) to construct the knockoff data (i.e., the negative control) and used the default `glmnet` method for binary logistic regression (where the cluster labels are considered as the response variable y, and the genes are considered as the features) to select features as DE genes. We test this approach on 50 simulated datasets (due to computational time) with logFC = 2.6 (see Simulation designs " Simulation setting with two cell types and 200 true DE genes") and found that it always selected 0 DE genes (i.e., the power was always 0).

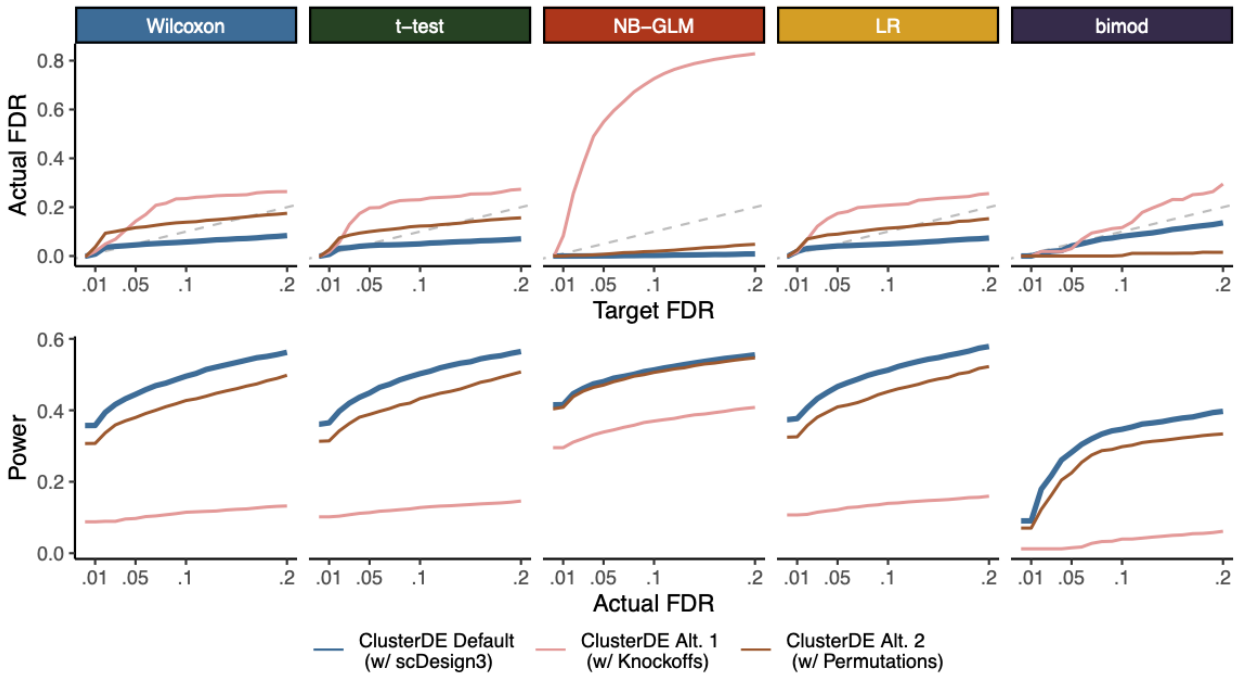


Figure 3.6: The FDRs and power of ClusterDE with three approaches for synthetic null generation: scDesign3 (the default in ClusterDE), the model-X knockoffs, and independent permutations of all genes across cells. Compared with the other two approaches, scDesign3 controls the FDR and yields higher power.

Moreover, we used the knockoff data constructed above and the permuted data (where each gene was independently permuted across all cells) as two alternative synthetic null generation strategies (alternatives to scDesign3) in ClusterDE step 1. Our results on the simulated datasets indicate that scDesign3 led to more solid FDR control and better statistical power than these two alternative strategies for the synthetic null generation (Fig. 3.6).

S3.8.6 Proof of theorem 1

The proof of Theorem 1 is mainly based on the proof in [16], which has also been included below.

For ease of presentation, we introduce the following notations. For $t \in \mathbb{R}$, denote

$$\widehat{G}_m^0(t) = \frac{1}{m_0}\sum_{j\in S_0}\mathbb{1}\left(C_j > t\right), \quad G_m^0(t) = \frac{1}{m_0}\sum_{j\in S_0}\mathbb{P}\left(C_j > t\right),$$
$$\widehat{G}_m^1(t) = \frac{1}{m_1}\sum_{j\in S_1}\mathbb{1}\left(C_j > t\right), \quad \widehat{V}_m^0(t) = \frac{1}{m_0}\sum_{j\in S_0}\mathbb{1}\left(C_j < -t\right).$$

Let $r_m = m_1/m_0$. In addition, denote

$$\mathrm{FDP}_m(t) = \frac{\widehat{G}_m^0(t)}{\widehat{G}_m^0(t) + r_m \widehat{G}_m^1(t)},$$

$$\mathrm{FDP}_m^\dagger(t) = \frac{\widehat{V}_m^0(t)}{\widehat{G}_m^0(t) + r_m \widehat{G}_m^1(t)},$$

$$\overline{\mathrm{FDP}}_m(t) = \frac{G_m^0(t)}{G_m^0(t) + r_m \widehat{G}_m^1(t)}.$$

Lemma 1. *Under Assumption 2, if $m_0 \to \infty$ as $m \to \infty$, we have in probability,*

$$\sup_{t\in\mathbb{R}} \left|\widehat{G}_m^0(t) - G_m^0(t)\right| \longrightarrow 0, \quad \sup_{t\in\mathbb{R}} \left|\widehat{V}_m^0(t) - G_m^0(t)\right| \longrightarrow 0.$$

Proof of Lemma 1. For any $\epsilon \in (0,1)$, denote $-\infty = \alpha_0^m < \alpha_1^m < \cdots < \alpha_{N_\epsilon}^m = \infty$ with $N_\epsilon = \lceil 2/\epsilon \rceil$, such that $G_m^0\left(\alpha_{k-1}^m\right) - G_m^0\left(\alpha_k^m\right) \leqslant \epsilon/2$ for $k = 1, \ldots, N_\epsilon$. By Assumption 2, and such a sequence $\{\alpha_k^m\}$ exists since $G_m^0(t)$ is a continuous function for $t \in \mathbb{R}$. We have

$$\begin{aligned}
\mathbb{P}\left(\sup_{t\in\mathbb{R}} \widehat{G}_m^0(t) - G_m^0(t) > \epsilon\right) &\leqslant \mathbb{P}\left(\bigcup_{k=1}^{N_\epsilon} \sup_{t\in\left[\alpha_{k-1}^m, \alpha_k^m\right)} \widehat{G}_m^0(t) - G_m^0(t) > \epsilon\right) \\
&\leqslant \sum_{k=1}^{N_\epsilon} \mathbb{P}\left(\sup_{t\in\left[\alpha_{k-1}^m, \alpha_k^m\right)} \widehat{G}_m^0(t) - G_m^0(t) > \epsilon\right)
\end{aligned} \tag{S3.1}$$

We note that both $\widehat{G}_m^0(t)$ and $G_m^0(t)$ are monotonically decreasing. Therefore, $\forall k \in \{1, \ldots, N_\epsilon\}$, we have

$$\sup_{t\in\left[\alpha_{k-1}^m, \alpha_k^m\right)} \widehat{G}_m^0(t) - G_m^0(t) \leqslant \widehat{G}_m^0\left(\alpha_{k-1}^m\right) - G_m^0\left(\alpha_k^m\right) \leqslant \widehat{G}_m^0\left(\alpha_{k-1}^m\right) - G_m^0\left(\alpha_{k-1}^m\right) + \epsilon/2.$$

By Equation S3.1, Assumption 2, and the Chebyshev's inequality, it follows that as $m \to \infty$,

$$\mathbb{P}\left(\sup_{t\in\mathbb{R}} \widehat{G}_m^0(t) - G_m^0(t) > \epsilon\right) \leqslant \sum_{k=1}^{N_\epsilon} \mathbb{P}\left(\widehat{G}_m^0\left(\alpha_{k-1}^m\right) - G_m^0\left(\alpha_{k-1}^m\right) > \frac{\epsilon}{2}\right) \leqslant \frac{4cN_\epsilon}{m_0^{2-\alpha}\epsilon^2} \to 0,$$

Similarly, we can show that as $m \to \infty$,

$$\mathbb{P}\left(\inf_{t\in\mathbb{R}} \widehat{G}_m^0(t) - G_m^0(t) < -\epsilon\right) \leqslant \sum_{k=1}^{N_\epsilon} \mathbb{P}\left(\widehat{G}_m^0\left(\alpha_k^m\right) - G_m^0\left(\alpha_k^m\right) < -\frac{\epsilon}{2}\right) \leqslant \frac{4cN_\epsilon}{m_0^{2-\alpha}\epsilon^2} \to 0.$$

This concludes the proof of the first claim in Lemma 2. The second claim follows similarly using the symmetric property of the mirror statistics C_j 's for $j \in S_0$.

Proof of Theorem 1. We first show that for any $\epsilon \in (0, q)$, we have

$$\mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) \geqslant 1 - \epsilon,$$

in which $t_{q-\epsilon} > 0$ satisfying $\mathbb{P}\left(\mathrm{FDP}\left(t_{q-\epsilon}\right) \leqslant q - \epsilon\right) \to 1$. Since the variances of the mirror statistics are upper bounded and also bounded away from 0, by Lemma 1, we have

$$\sup_{0<t\leqslant c} \left|\mathrm{FDP}_m^\dagger(t) - \mathrm{FDP}_m(t)\right| \xrightarrow{m} 0$$

for any constant $c > 0$. By the definition of τ_q, i.e., $\tau_q = \inf\left\{t > 0 : \mathrm{FDP}_m^\dagger(t) \leqslant q\right\}$, we have

$$\begin{aligned}
\mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) &\geqslant \mathbb{P}\left(\mathrm{FDP}_m^\dagger\left(t_{q-\epsilon}\right) \leqslant q\right) \\
&\geqslant \mathbb{P}\left(\left|\mathrm{FDP}_m^\dagger\left(t_{q-\epsilon}\right) - \mathrm{FDP}_m\left(t_{q-\epsilon}\right)\right| \leqslant \epsilon, \mathrm{FDP}\left(t_{q-\epsilon}\right) \leqslant q - \epsilon\right) \\
&\geqslant 1 - \epsilon
\end{aligned}$$

for m large enough. Conditioning on the event $\tau_q \leqslant t_{q-\epsilon}$, we have

$$\begin{aligned}
\limsup_{m\to\infty} \mathbb{E}\left[\mathrm{FDP}_m\left(\tau_q\right)\right] &\leqslant \limsup_{m\to\infty} \mathbb{E}\left[\mathrm{FDP}_m\left(\tau_q\right) \mid \tau_q \leqslant t_{q-\epsilon}\right] \mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) + \epsilon \\
&\leqslant \limsup_{m\to\infty} \mathbb{E}\left[\left|\mathrm{FDP}_m\left(\tau_q\right) - \overline{\mathrm{FDP}}_m\left(\tau_q\right)\right| \mid \tau_q \leqslant t_{q-\epsilon}\right] \mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) \\
&+ \limsup_{m\to\infty} \mathbb{E}\left[\left|\mathrm{FDP}^{\dagger}_m\left(\tau_q\right) - \overline{\mathrm{FDP}}_m\left(\tau_q\right)\right| \mid \tau_q \leqslant t_{q-\epsilon}\right] \mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) \\
&+ \limsup_{m\to\infty} \mathbb{E}\left[\mathrm{FDP}^{\dagger}_m\left(\tau_q\right) \mid \tau_q \leqslant t_{q-\epsilon}\right] \mathbb{P}\left(\tau_q \leqslant t_{q-\epsilon}\right) + \epsilon \\
&\leqslant \limsup_{m\to\infty} \mathbb{E}\left[\sup_{0<t\leqslant t_{q-\epsilon}} \left|\mathrm{FDP}_m(t) - \overline{\mathrm{FDP}}_m(t)\right|\right] \\
&+ \limsup_{m\to\infty} \mathbb{E}\left[\sup_{0<t\leqslant t_{q-\epsilon}} \left|\mathrm{FDP}^{\dagger}_m(t) - \overline{\mathrm{FDP}}_m(t)\right|\right] \\
&+ \limsup_{m\to\infty} \mathbb{E}\left[\mathrm{FDP}^{\dagger}_m\left(\tau_q\right)\right] + \epsilon.
\end{aligned}$$

The first two terms are 0 based on Lemma 2 and the dominated convergence theorem. For the third term, we have $\mathrm{FDP}^{\dagger}_m\left(\tau_q\right) \leqslant q$ by the definition of τ_q. This concludes the proof of Proof of Theorem 1.

S3.8.7 Existing methods do not have FDR control guarantee

In this subsection, we will review and formulate existing methods for clustering + DE analysis in single-cell sequencing data analysis, which may or may not have accounted for double dipping. With the help of method formulation, we then show why all existing methods result in unwanted correlated clustering labels and expression vectors to perform DE analysis on. Finally, a unified theory summarizes the conceptually false discovery rate inflation in those methods.

The naïve method with the double-dipping issue

In the naïve method, it directly uses $\widehat{Z}_1, \ldots, \widehat{Z}_n \in \{0, 1\}$, where $\widehat{Z}_i = g_{\mathbf{Y}}(\mathbf{Y}_i)$, to define two cell groups and then tests each gene j against H^{DD}_{0j} instead of H_{0j}. Due to the existence of gene-gene correlations, it is highly possible that for some non-DE gene j, $Y_j = (Y_{1j}, \ldots, Y_{nj})^{\mathsf{T}}$

and $\widehat{Z} = (\widehat{Z}_1, \ldots, \widehat{Z}_n)^\intercal$ are dependent and correlated.

The Cellsplit method

The Cellsplit method first randomly split the n cells $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n$ into two sets $\mathcal{Y}_{train}$ and $\mathcal{Y}_{test}$, such that $\mathcal{Y}_{train} \cup \mathcal{Y}_{test} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$. Then $g(\cdot) = g_{train}(\cdot)$ is constructed only on $\mathcal{Y}_{train}$ by the user-chosen algorithm, for instance, the Louvain clustering [17]. The cell cluster membership $\widehat{Z}_i = g_{train}(\mathbf{Y}_i) \in \{0, 1\}$ for only gene i's such that $\mathbf{Y}_i \in \mathcal{Y}_{test}$. It is worth noting that $\widehat{Z}_i$ takes $\mathbf{Y}_i$ as the input and thus the dependence between $\widehat{Z}_i$ and $\mathbf{Y}_i$ still exists, although $g_{train}(\cdot) \perp \mathcal{Y}_{test}$.

The TN test [13] is an example of the Cellsplit methods.

Note that sample splitting, or here in our case, the Cellsplit method, is a popular solution to the problems that have the "double dipping issue". However, an intuitive explanation of why it fails here is that the cell cluster membership is a crucial component in constructing our testing hypothesis. Splitting the cells into the training and the testing sets does not help in generating cluster labels that are independent of gene expressions for the cells in the testing set.

The Genesplit method

Following the Cellsplit method, the Genesplit method splits the data by the genes (features). It results in a testing set with only one gene j, which is to be tested, and the original n cells; the remaining $(m-1)$ genes and the n cells construct the training set. The Genesplit method actually works under two certain ideal scenarios.

The first ideal scenario requires two conditions: (1) the oracle clustering algorithm, which is defined below, and (2) the independent gene case where the tested non-DE gene j is independent with any true DE genes.

Definition 1 (Oracle clustering)**.** *The clustering function $g(\cdot)$ is **oracle** in that it only*

correlates with the true DE genes, i.e.,

$$\mathrm{Cov}\left(\widehat{Z}\,,\ Y_j\right) = 0,\ \forall j \notin \mathcal{T}$$

where $\mathcal{T} \subset \{1, \ldots, m\}$ *denotes the set of true DE genes,* $Y_j = (Y_{1j}, \ldots, Y_{nj})^\mathsf{T}$ *is the* j*-th gene's count vector, and* $\widehat{Z} = (\widehat{Z}_1, \ldots, \widehat{Z}_n)^\mathsf{T}$ *denotes the cell cluster membership under current clustering function.*

When the clustering function is *oracle*, it only uses the true DE genes (while unknown). Denote the i-th gene count vector $\mathbf{Y}_i$ restricted to the set of true DE genes $\mathcal{T}$ by $\mathbf{Y}_{i,\mathcal{T}} \in \mathbb{N}_{\geqslant 0}^{|\mathcal{T}|}$. Then $g : \mathbb{N}_{\geqslant 0}^{|\mathcal{T}|} \to \{0, 1\}$, and $g(\mathbf{Y}_i) = g(\mathbf{Y}_{i,\mathcal{T}})$. Since $g(\cdot)$ is constructed from $\mathbf{Y}_{1,\mathcal{T}}, \ldots, \mathbf{Y}_{n,\mathcal{T}}$, it does not use Y_{ij}. Hence, if $\mathbf{Y}_{i,\mathcal{T}}$ is independent of Y_{ij}, we have the cell cluster membership $\widehat{Z}_i = g(\mathbf{Y}_i) = g(\mathbf{Y}_{i,-j})$ independent of Y_{ij}.

The second ideal scenario is when all genes are independent, no matter the clustering function $g(\cdot)$ is oracle or not. In this scenario, the clustering algorithm may use any genes. But the overall independence guarantees that the gene j will not be used to construct $g(\cdot)$ when it is to be tested. Denote the gene count vector $\mathbf{Y}_i$ with gene j removed by $\mathbf{Y}_{i,-j} \in \mathbb{N}_{\geqslant 0}^{m-1}$. Then $g : \mathbb{N}_{\geqslant 0}^{m-1} \to \{0, 1\}$, and $g(\mathbf{Y}_i) = g(\mathbf{Y}_{i,-j})$. Since $g(\cdot)$ does not use Y_{ij}, if $\mathbf{Y}_{i,-j}$ is independent of Y_{ij}, we have $g(\mathbf{Y}_i)$ independent of Y_{ij}. Hence, under the independent gene case, gene split will not make non-DE gene j a false positive.

However, the above two ideal scenarios usually do not hold in real single-cell sequencing data analysis, and the computing efficiency is a great concern.

The Countsplit method

The Countsplit method [14] is a selective inference approach that claims to solve the double dipping issue in the clustering + DE analysis. In Countsplit, each Y_{ij} is independently splitted as Y_{ij}^1 and Y_{ij}^2 such that $Y_{ij} = Y_{ij}^1 + Y_{ij}^2$ and $Y_{ij}^1 \sim \mathrm{Binomial}(Y_{ij}, 0.5)$. The basic idea is to cluster the cells with the first part of counts $[Y_{ij}^1]_{i,j}$, and perform DE analysis on the second part of counts $[Y_{ij}^2]_{i,j}$.

Denote $\mathbf{Y}_i^1 = (Y_{i1}^1, \ldots, Y_{im}^1)^\top$ and $\mathbf{Y}_i^2 = (Y_{i1}^2, \ldots, Y_{im}^2)^\top$. $g(\cdot) = g^1(\cdot)$ is constructed only from $\mathbf{Y}_1^1, \ldots, \mathbf{Y}_n^1$ and does not use Y_{ij}^2. The cell cluster membership $\widehat{Z}_i = g(\mathbf{Y}_i^1) = g^1(\mathbf{Y}_i^1)$. The authors of Countsplit claim that $Y_{ij}^1 \perp Y_{ij}^2$, and thus $\widehat{Z}_i = g(\mathbf{Y}_i^1) = g^1(\mathbf{Y}_i^1) \perp Y_{ij}^2$.

However, the above justification implicitly makes an assumption that $[Y_{ij}]_{i,j}$ are all independent entries, which means that all n cells and all m genes are independent. This assumption does not hold in real scRNA-seq data. Generally, the correlations between Y_{ij} and Y_{ik}, $i = 1, \ldots, n$, lead to the correlations between Y_{ij}^2 and $\widehat{Z}_i = g^1(\mathbf{Y}_i^1) = g^1((Y_{i1}^1, \cdots, Y_{ik}^1, \cdots, Y_{im}^1)^\mathsf{T})$. The dependence and correlation still exist.

Correlations leads to FDR control failure

Lemma 2. *If*

$$\mathrm{Cov}\left(\widehat{Z}\,,\ Y_j\right) > 0, \tag{S3.2}$$

then $H_0^{\mathrm{DD}} : \mathbb{E}[Y_j|\widehat{Z} = 0] = \mathbb{E}[Y_j|\widehat{Z} = 1]$ *may not hold even when* $H_0 : \mathbb{E}[Y_j|Z = 0] = \mathbb{E}[Y_j|Z = 1]$ *holds.*

Proof.

$$\mathbb{E}[Y_j] = \mathbb{E}[\mathbb{E}[Y_j|Z]] = \mathbb{P}(Z = 0)\mathbb{E}[Y_j|Z = 0] + \mathbb{P}(Z = 1)\mathbb{E}[Y_j|Z = 1] \tag{S3.3}$$

$$= \mathbb{E}[\mathbb{E}[Y_j|\widehat{Z}]] = \mathbb{P}(\widehat{Z} = 0)\mathbb{E}[Y_j|\widehat{Z} = 0] + \mathbb{P}(\widehat{Z} = 1)\mathbb{E}[Y_j|\widehat{Z} = 1]. \tag{S3.4}$$

When H_0 holds, assume $\mu := \mu_0 = \mathbb{E}[Y_j|Z = 0] = \mathbb{E}[Y_j|Z = 1] = \mu_1$, then

$$\mathbb{E}[Y_j] = \mathbb{P}(Z = 0) \cdot \mu + \mathbb{P}(Z = 1) \cdot \mu = \mu.$$

(S3.2) implies that

$$
\begin{aligned}
0 < \mathrm{Cov}\left(\widehat{Z},\ Y_j\right) &= \mathbb{E}\left[\widehat{Z}Y_j\right] - \mathbb{E}\left[\widehat{Z}\right]\mathbb{E}\left[Y_j\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{Z}Y_j|\widehat{Z}\right]\right] - \left(\mathbb{P}(\widehat{Z}=0)\cdot 0 + \mathbb{P}(\widehat{Z}=1)\cdot 1\right)\cdot\mu \\
&= \left(\mathbb{P}(\widehat{Z}=0)\mathbb{E}[0\cdot Y_j|\widehat{Z}=0] + \mathbb{P}(\widehat{Z}=1)\mathbb{E}[1\cdot Y_j|\widehat{Z}=1]\right) - \mathbb{P}(\widehat{Z}=1)\cdot\mu \\
&= \mathbb{P}(\widehat{Z}=1)\mathbb{E}[Y_j|\widehat{Z}=1] - \mathbb{P}(\widehat{Z}=1)\cdot\mu \\
&= \mathbb{P}(\widehat{Z}=1)\cdot\left(\mathbb{E}[Y_j|\widehat{Z}=1] - \mu\right)
\end{aligned}
$$

Given $\mathbb{P}(\widehat{Z}=1) > 0$, we have

$$\mathbb{E}[Y_j|\widehat{Z}=1] > \mu \tag{S3.5}$$

According to (S3.4),

$$
\begin{aligned}
\mathbb{E}[Y_j|\widehat{Z}=0] &= \frac{\mathbb{E}[Y_j] - \mathbb{P}(\widehat{Z}=1)\mathbb{E}[Y_j|\widehat{Z}=1]}{\mathbb{P}(\widehat{Z}=0)} \\
&< \frac{\mu\left(1-\mathbb{P}(\widehat{Z}=1)\right)}{\mathbb{P}(\widehat{Z}=0)} = \mu \\
&< \mathbb{E}[Y_j|\widehat{Z}=1].
\end{aligned}
$$

Therefore, H_0^{DD} does not hold when H_0 actually holds. Hence, by testing H_{0j}^{DD}, the double-dipping issue may make gene j a false positive for H_{0j}; that is, at the conceptual level, gene j may be $\mathrm{DEG}^{\mathrm{DD}}$ but not a $\mathrm{DEG}^{\mathrm{true}}$. □

All existing methods generally suffer from the correlation between $\widehat{Z}$ and Y_j for some non-DE gene j. False discoveries are made conceptually, and FDR is thus inflated conceptually. In section Results, it is confirmed that FDR is inflated for all existing methods in real single-cell sequencing data analysis.

S3.8.8 Supplementary figures

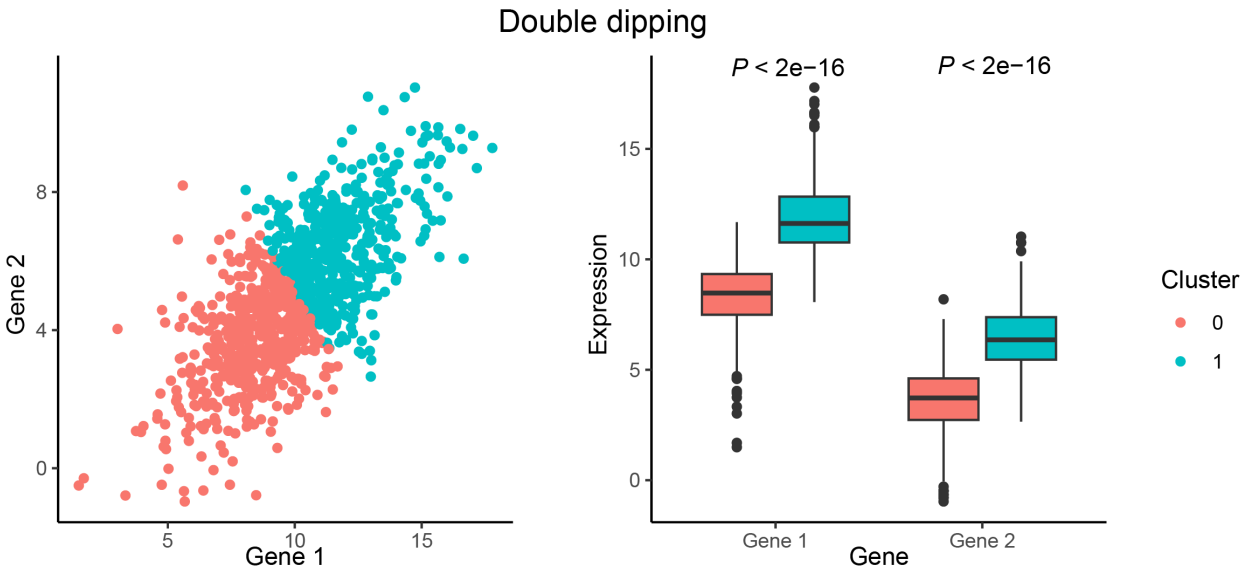
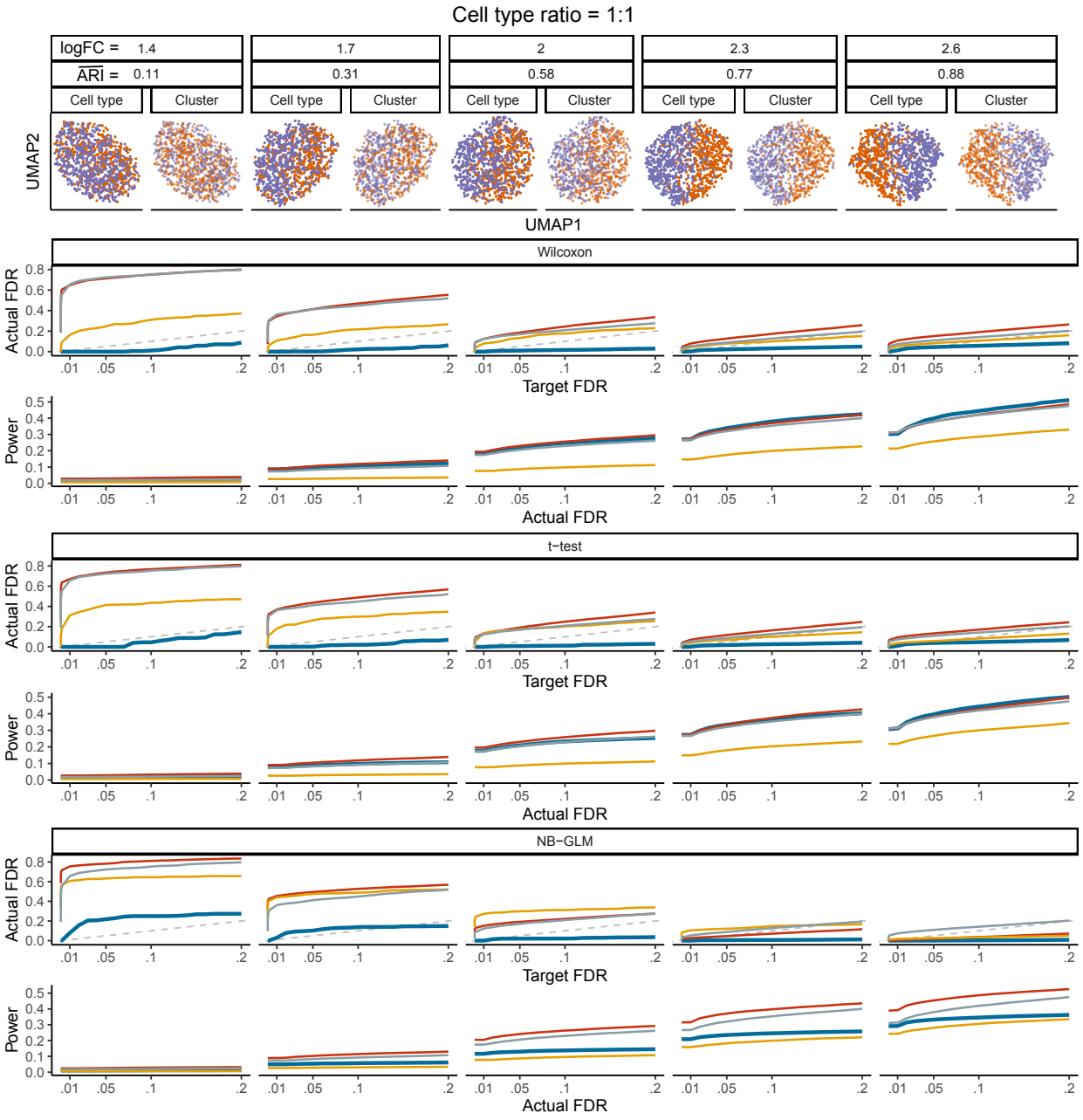


Figure 3.7: A toy example to showcase the double-dipping issue.

The two genes' expressions follow a bivariate Gaussian distribution as the cells come from a homogeneous cell type. However, if we run the K-means clustering to divide the cells into two clusters, the two genes are forced to exhibit different distributions in the two clusters.


Cell type ratio = 1:1
logFC = 1.4
1.7
2
2.3
2.6
ARI = 0.11
0.31
0.58
0.77
0.88
Cell type
Cluster
UMAP2
UMAP1
Wilcoxon
t−test
NB−GLM
Actual FDR
Power
Target FDR
0.0
0.2
0.4
0.6
0.8
0.1
0.3
0.5
.01
.05
.1
.2

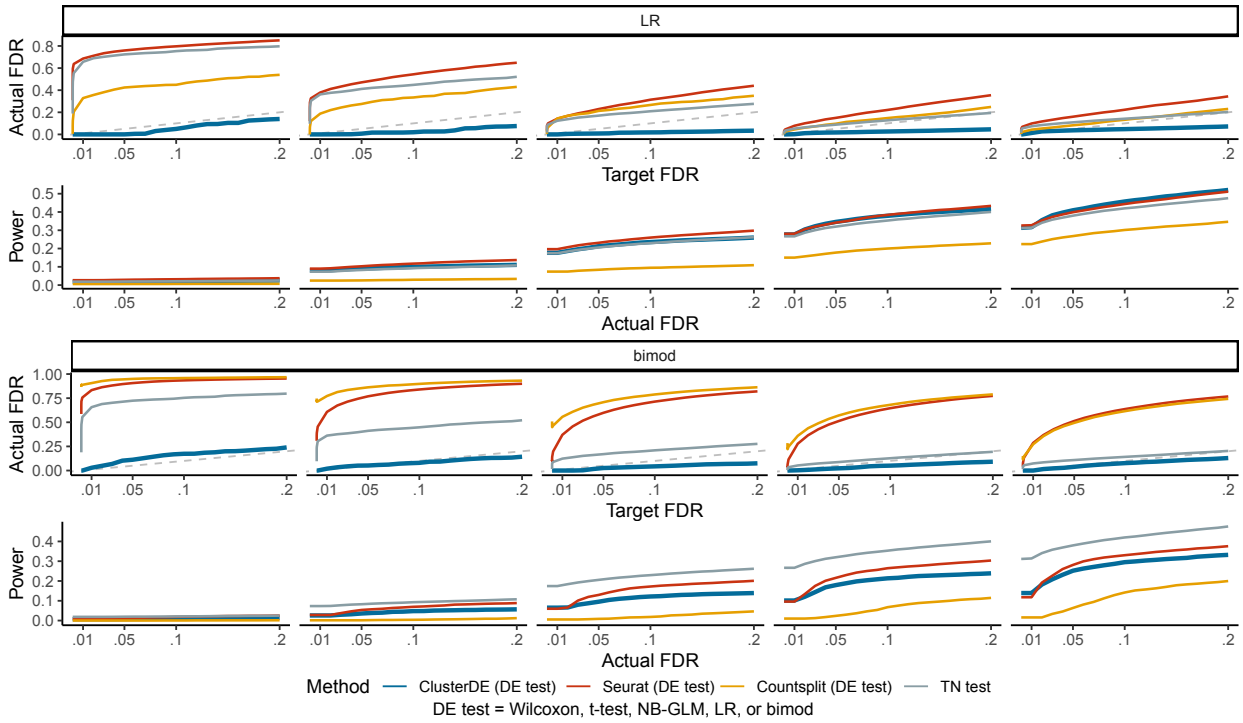
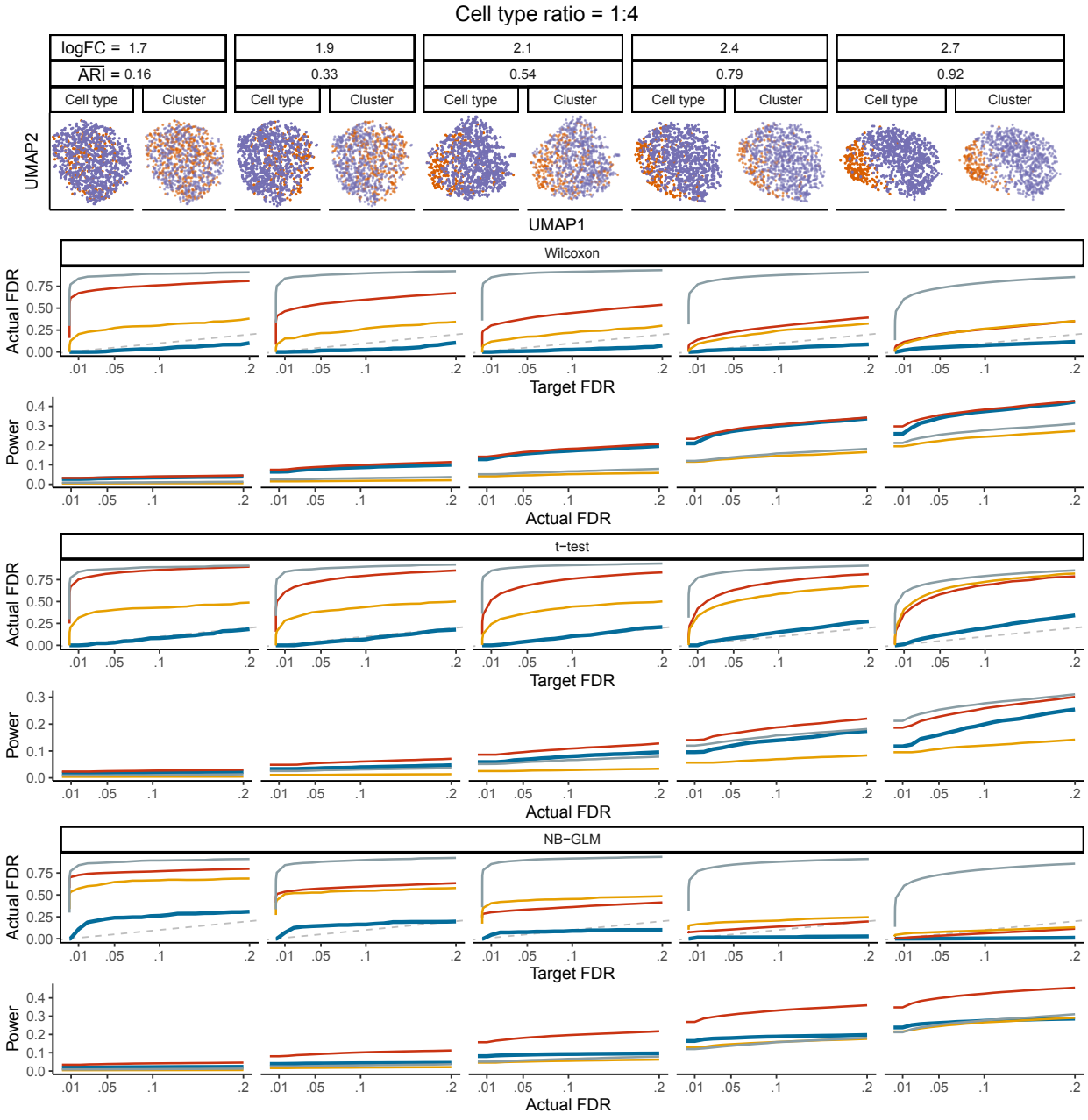


Figure 3.8: The FDRs and power of ClusterDE and the existing methods under various severity levels of double dipping when the two cell types have a size ratio of 1 : 1.

The log fold change (logFC) summarizes the average gene expression difference between the two cell types in simulation (see ClusterDE methodology "Simulation setting with two cell types and 200 true DE genes"). Corresponding to a small logFC, a small adjusted Rand index (ARI) represents a bad agreement between cell clusters and cell types, representing a more severe double-dipping issue. Across various severity levels of double dipping and the five DE tests, ClusterDE controls the FDRs under the target FDR thresholds (diagonal dashed line) and achieves comparable or higher power compared to the existing methods at the same actual FDRs.


Cell type ratio = 1:4
logFC = 1.7
1.9
2.1
2.4
2.7
ARI = 0.16
0.33
0.54
0.79
0.92
Cell type
Cluster
UMAP2
UMAP1
Wilcoxon
t−test
NB−GLM
Actual FDR
Power
Target FDR
0.00
0.25
0.50
0.75
0.0
0.1
0.2
0.3
0.4
.01
.05
.1
.2

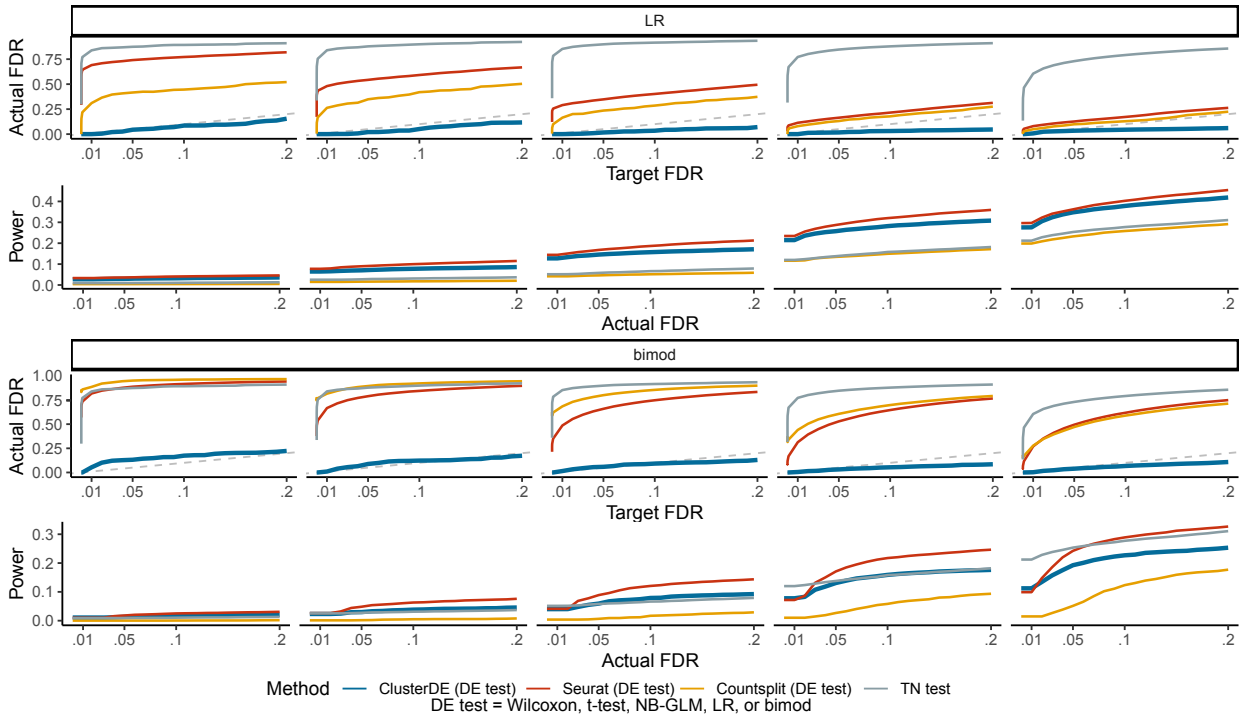
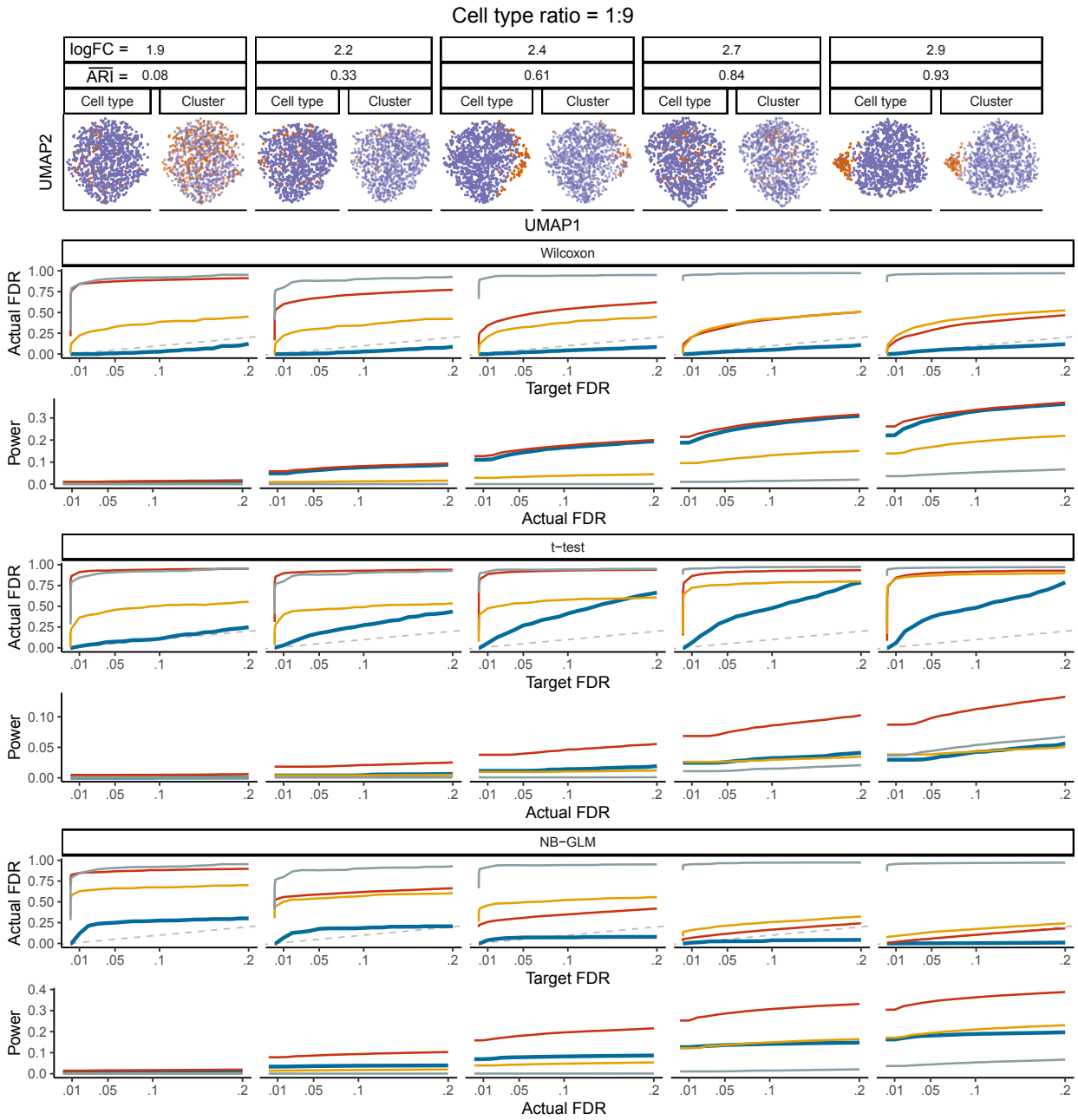


Figure 3.9: The FDRs and power of ClusterDE and the existing methods under various severity levels of double dipping when the two cell types have a size ratio of 1 : 4.

The log fold change (logFC) summarizes the average gene expression difference between the two cell types in simulation (see ClusterDE methodology "Simulation setting with two cell types and 200 true DE genes"). Corresponding to a small logFC, a small adjusted Rand index (ARI) represents a bad agreement between cell clusters and cell types, representing a more severe double-dipping issue. Across various severity levels of double dipping and the five DE tests, ClusterDE controls the FDRs under the target FDR thresholds (diagonal dashed line) and achieves comparable or higher power compared to the existing methods at the same actual FDRs.


Cell type ratio = 1:9
logFC = 1.9
2.2
2.4
2.7
2.9
ARI = 0.08
0.33
0.61
0.84
0.93
Cell type
Cluster
UMAP2
UMAP1
Wilcoxon
t−test
NB−GLM
Actual FDR
Power
Target FDR

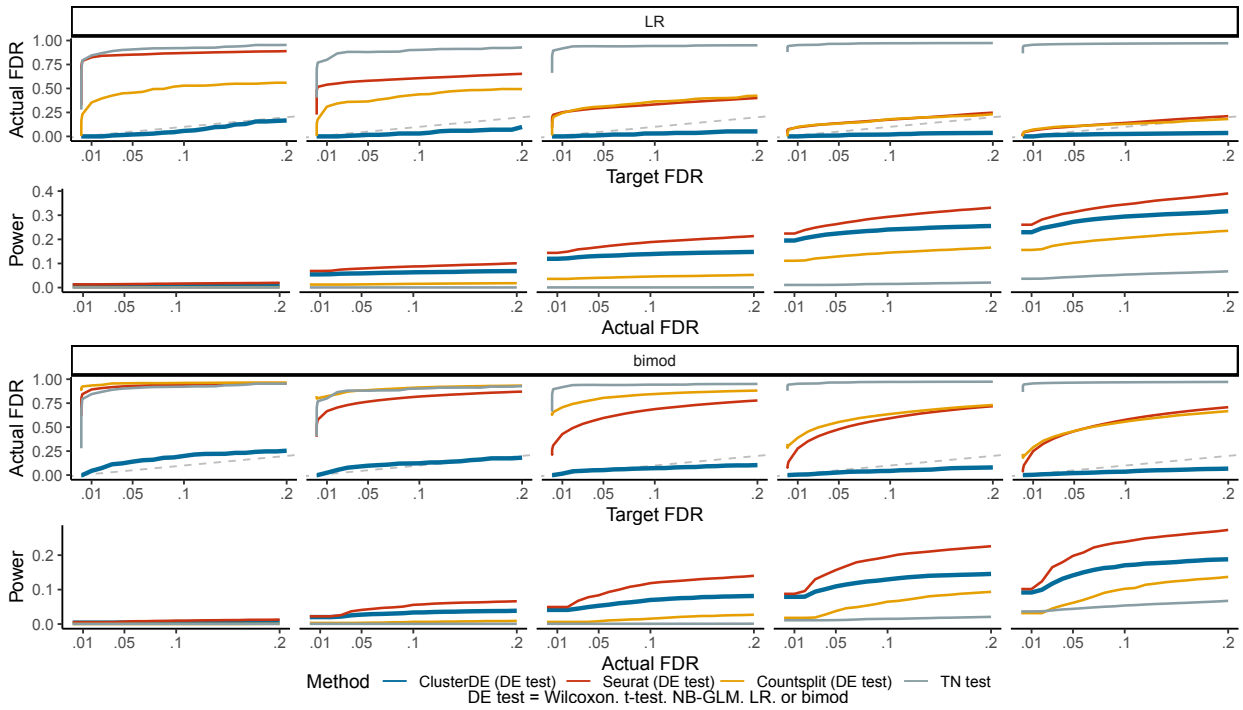


Figure 3.10: The FDRs and power of ClusterDE and the existing methods under various severity levels of double dipping when the two cell types have a size ratio of $1:9$.

The log fold change (logFC) summarizes the average gene expression difference between the two cell types in simulation (see ClusterDE methodology "Simulation setting with two cell types and 200 true DE genes"). Corresponding to a small logFC, a small adjusted Rand index (ARI) represents a bad agreement between cell clusters and cell types, representing a more severe double-dipping issue. Across various severity levels of double dipping and the five DE tests, ClusterDE controls the FDRs under the target FDR thresholds (diagonal dashed line) and achieves comparable or higher power compared to the existing methods at the same actual FDRs.

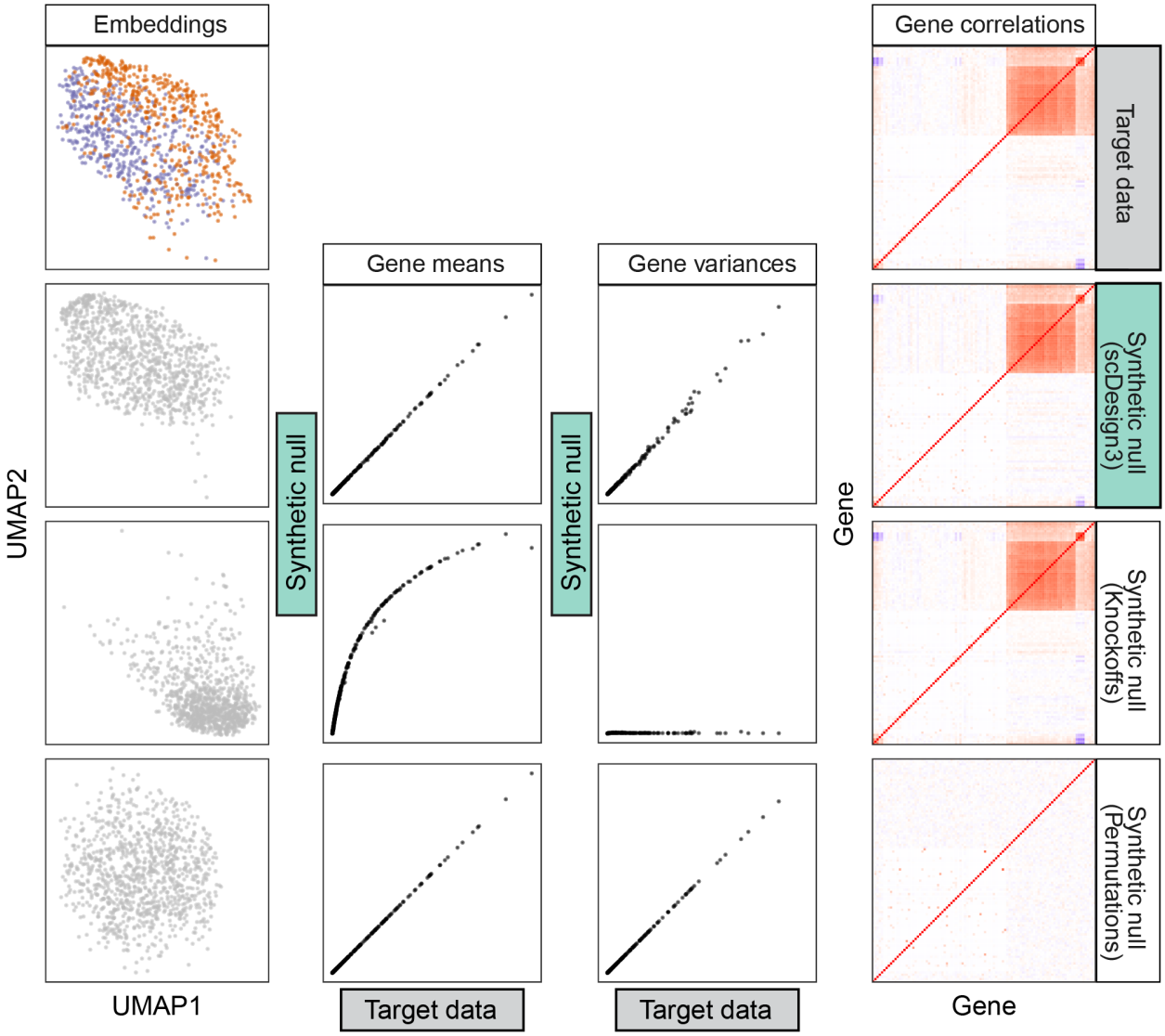


Figure 3.11: When the target data contains cells from two cell types (simulation; see ClusterDE methodology "Simulation setting with one cell type and zero true DE genes"), the synthetic null data generated by ClusterDE (second row) fills the gap between the two cell types but resembles the target data in other visual aspects of UMAP cell embeddings (left), per-gene expression mean and variance statistics (middle), and gene-gene correlations.

In contrast, the synthetic null data generated by knockoffs (third row) and permutations (fourth row) do not resemble the target data. For the synthetic null data generated by knockoffs, it preserves the gene-gene correlations of the target data, but does not preserve per-gene expression mean and variance statistics. For the synthetic null data generated by permutations, it preserves per-gene expression mean and variance statistics of the target data, but does not preserve the gene-gene correlations.

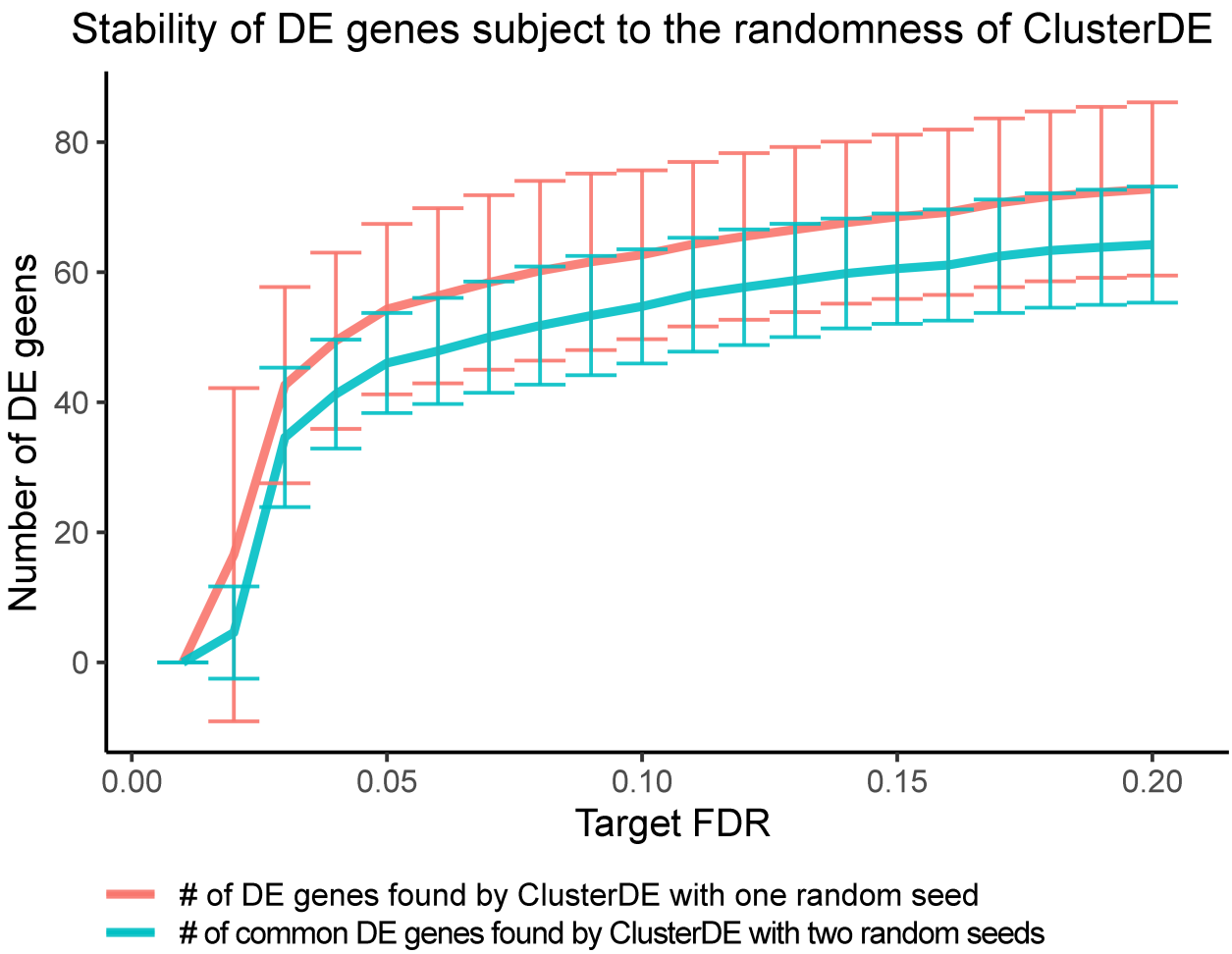
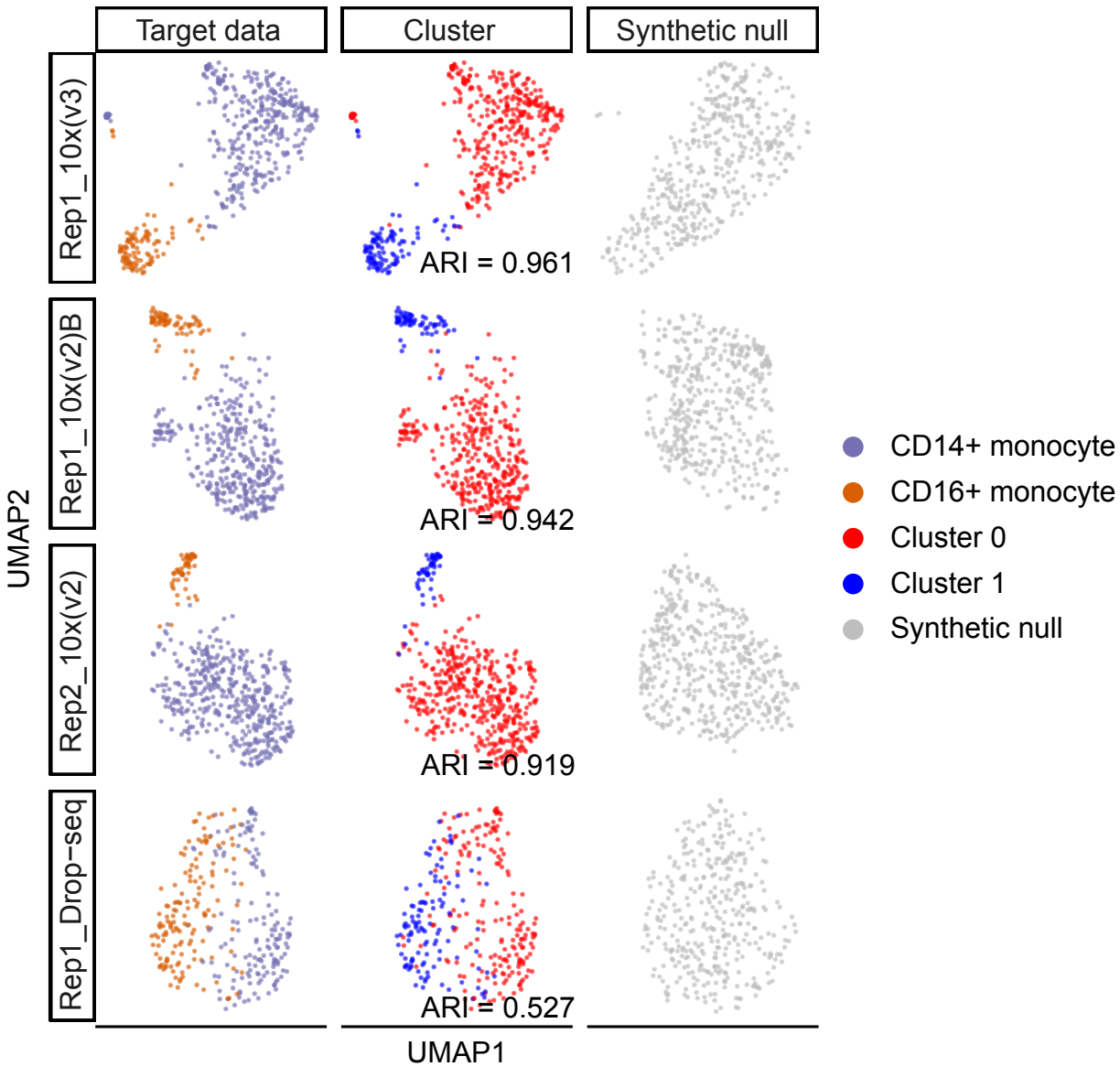


Figure 3.12: Stability of the DE genes identified by Cluster in relation to the randomness of synthetic null generation.

Given one target dataset simulated with two cell types (see ClusterDE methodology "Simulation setting with two cell types and 200 true DE genes"), 50 synthetic null datasets are generated with 50 random seeds, and DE genes are identified by ClusterDE using each synthetic null dataset. The red curve shows the mean and standard deviation (half of the vertical bar height) of the numbers of DE genes identified at each target FDR across the 50 random seeds. The cyan curve shows the mean and standard deviation (half of the vertical bar height) of the numbers of DE genes shared between two random seeds, across $\binom{50}{2}$ pairs of random seeds, at each target FDR. The results show that the DE genes identified by ClusterDE remain relatively stable and robust to the randomness.


Target data
Cluster
Synthetic null
Rep1_10x(v3)
Rep1_10x(v2)B
Rep2_10x(v2)
Rep1_Drop−seq
ARI = 0.961
ARI = 0.942
ARI = 0.919
ARI = 0.527
UMAP2
UMAP1
CD14+ monocyte
CD16+ monocyte
Cluster 0
Cluster 1
Synthetic null

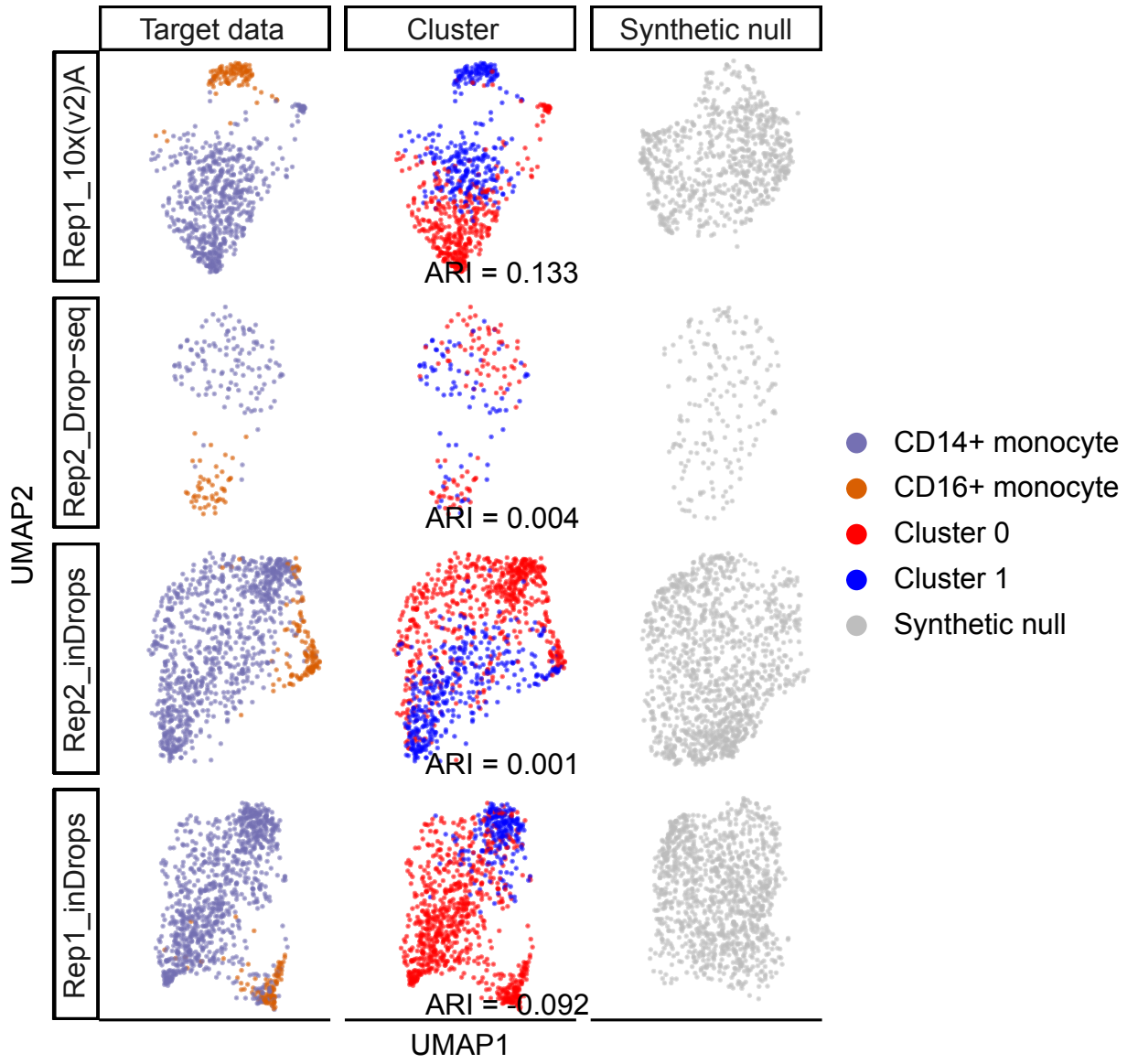


Figure 3.13: UMAP visualizations and Seurat clustering accuracies (ARIs) of the eight PBMC monocyte datasets (ordered by ARIs from high to low).

The first and second columns show the UMAP visualizations of the eight datasets as the target data, with the cells labeled by the monocyte subtypes (the first column) or the clusters (the second column). The third column shows the UMAP visualizations of the synthetic null data corresponding to the eight target datasets. The horizontal dashed line between rows 4 and 5 divides the eight datasets based on the clustering accuracy. It is expected that monocyte-subtype markers are more likely to be identified as post-clustering DE genes from the top four datasets than the bottom four datasets.

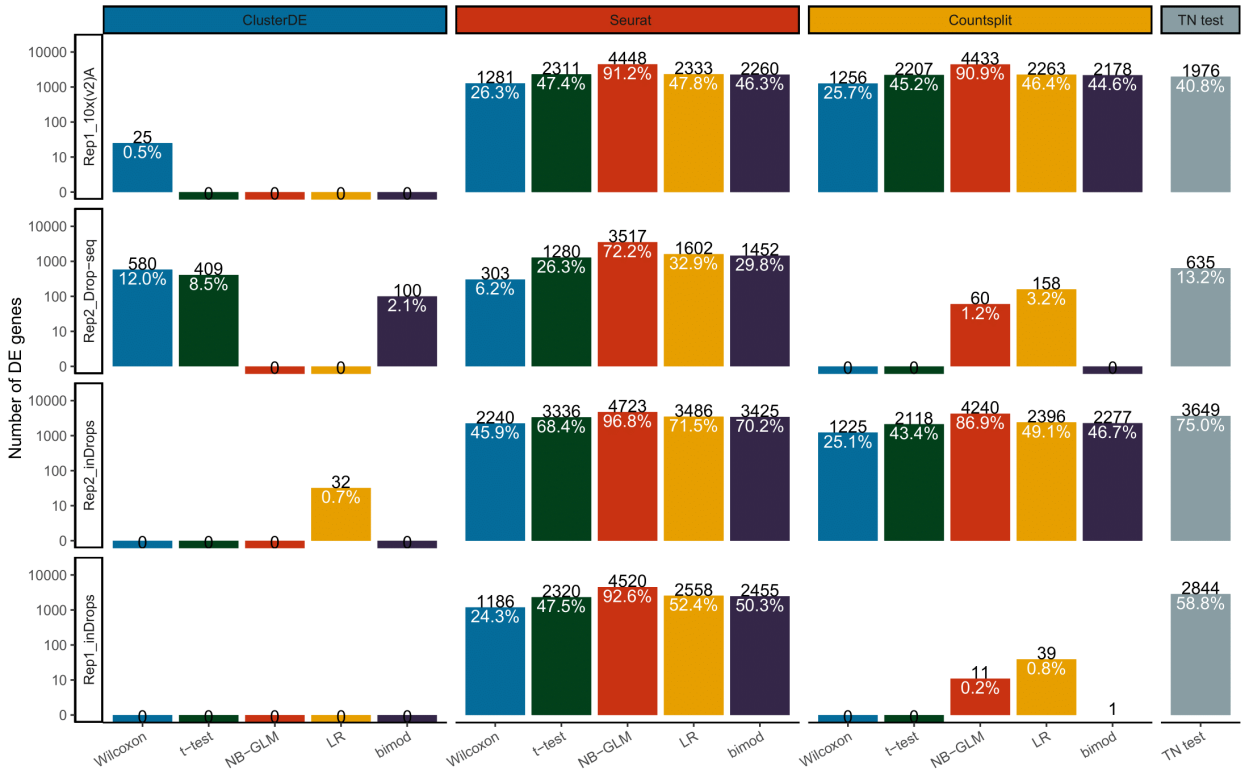


Figure 3.14: ClusterDE avoids false discoveries under double dipping.

Although the datasets contain two monocyte subtypes, the clustering results poorly match the subtype labels (the bottom four datasets in Fig. 3.13), and thus no DE genes should be discovered. The numbers in black and white are the number of DE genes and the proportion of DE genes among all genes, respectively. In most cases, ClusterDE does not find DE genes, as expected.

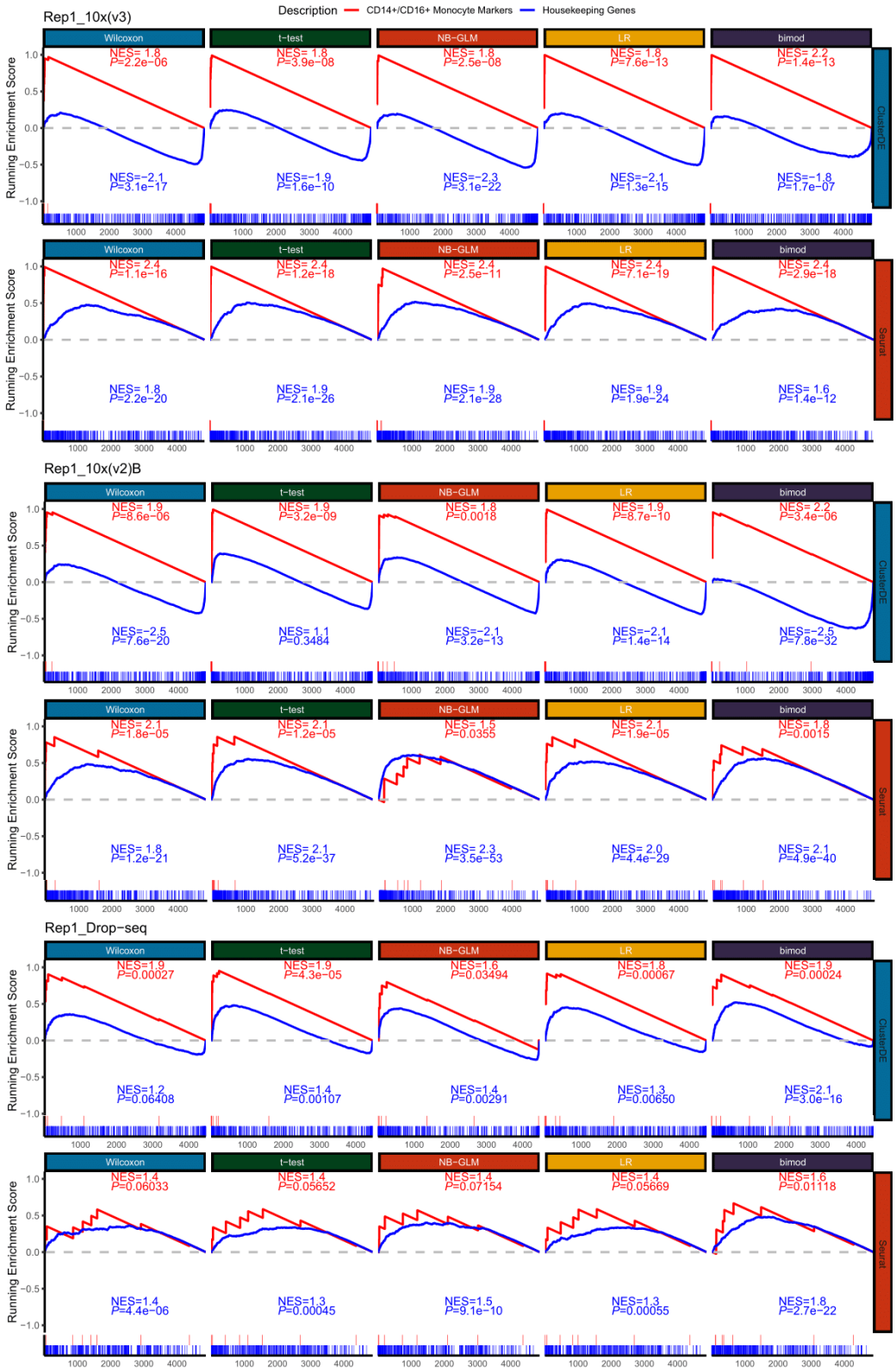

Description
CD14+/CD16+ Monocyte Markers
Housekeeping Genes
Rep1_10x(v3)
Wilcoxon
t-test
NB-GLM
LR
bimod
Running Enrichment Score
ClusterDE
Seurat
Rep1_10x(v2)B
Rep1_Drop-seq

Figure 3.15: Gene set enrichment analysis (GSEA) of the ranked DE gene lists identified by ClusterDE and Seurat with five DE tests from three datasets.

The red lines represent the enrichment of the "$CD14^+/CD16^+$ Monocyte Marker Genes" set, and the blue lines represent the enrichment of the "Housekeeping Genes" set. The short vertical lines at the bottom show the rank distributions of the genes in the two gene sets within each ranked DE gene list. The normalized enrichment score (NES) reflects the direction and magnitude of enrichment, and the P value indicates the significance of enrichment.

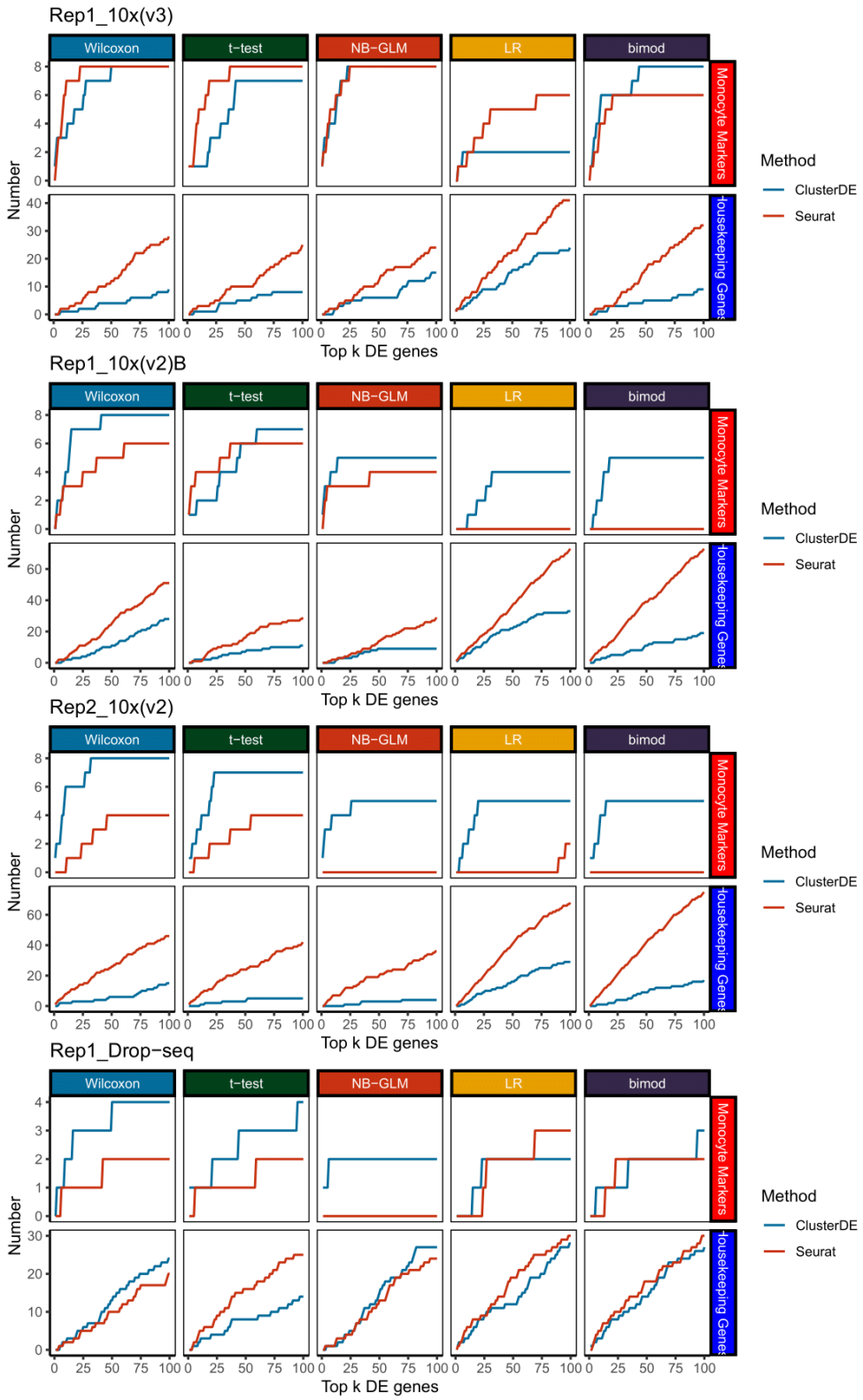

Rep1_10x(v3)
Rep1_10x(v2)B
Rep2_10x(v2)
Rep1_Drop-seq
Wilcoxon
t-test
NB-GLM
LR
bimod
Monocyte Markers
Housekeeping Genes
Number
Top k DE genes
Method
ClusterDE
Seurat

Figure 3.16: Overlaps between monocyte markers/housekeeping genes and the top k DE genes, with k ranging from 1 to 100.

The horiontal axis represents k (for example, $k = 100$ means we select the top 100 DE genes from the DE gene lists). The vertical axis indicates the number of monocyte markers (the top row in each panel) or housekeeping genes (the bottom row in each panel) among the top k DE genes found by each of the five DE tests (columns). In most cases, ClusterDE (blue line) identifies more monocyte subtype markers and fewer housekeeping genes compared to Seurat (red line).

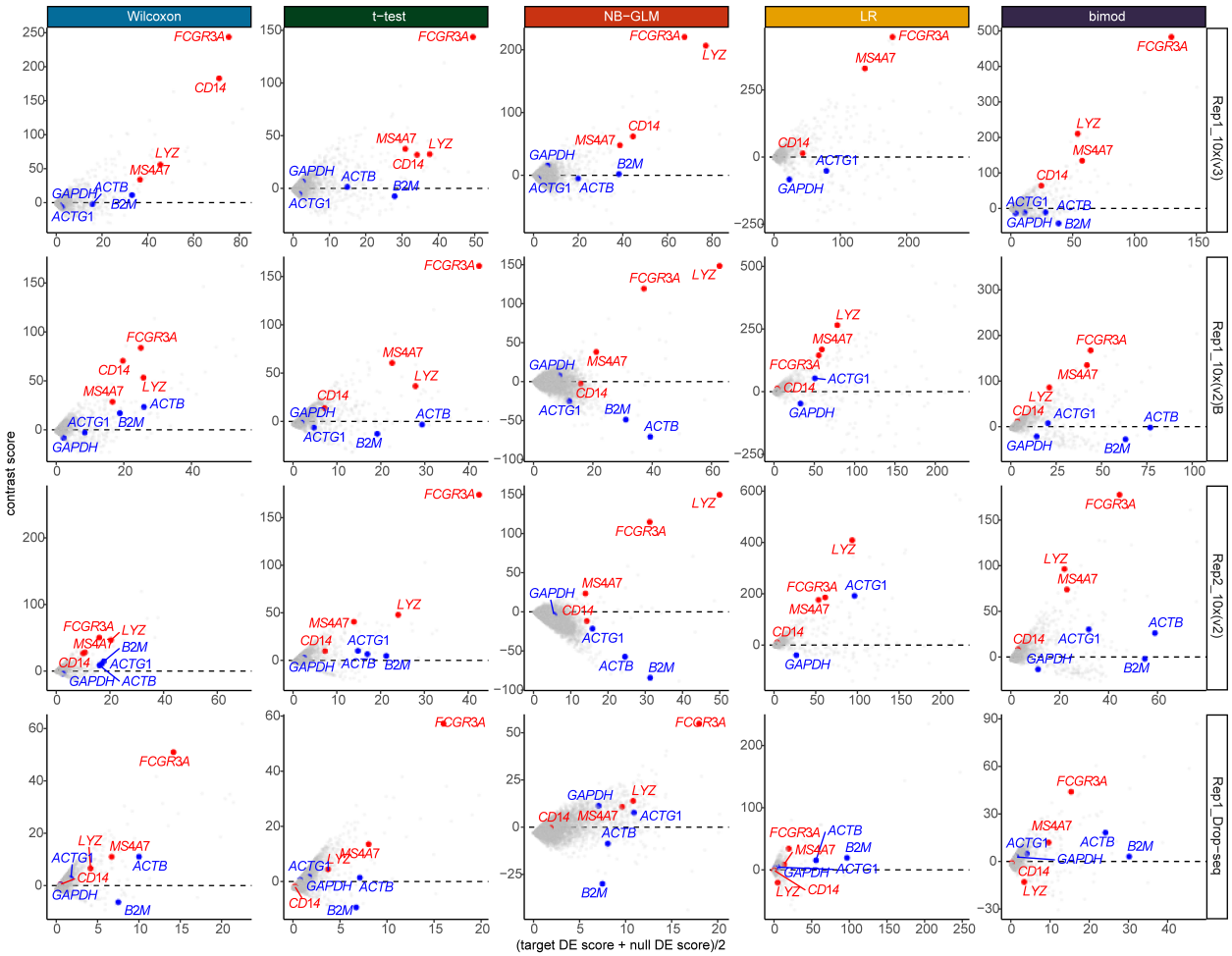


Figure 3.17: The minus-average (MA) plots of ClusterDE contrast scores (target DE score minus null DE score) vs. averages of target DE scores and null DE scores.

The red color labels four well-known $CD14^+/CD16^+$ subtype markers, and the blue color labels four well-known housekeeping genes. The dashed black line indicates the contrast scores of 0. For housekeeping genes, their DE scores are large in both target data and synthetic null data, so their contrast scores are centered around 0. Hence, these housekeeping genes would be ranked top by Seurat (which only examines target DE scores) but not by ClusterDE. On the other hand, the DE scores of subtype markers are much larger in target data than in synthetic null data, so their contrast scores are large and ranked top by ClusterDE.

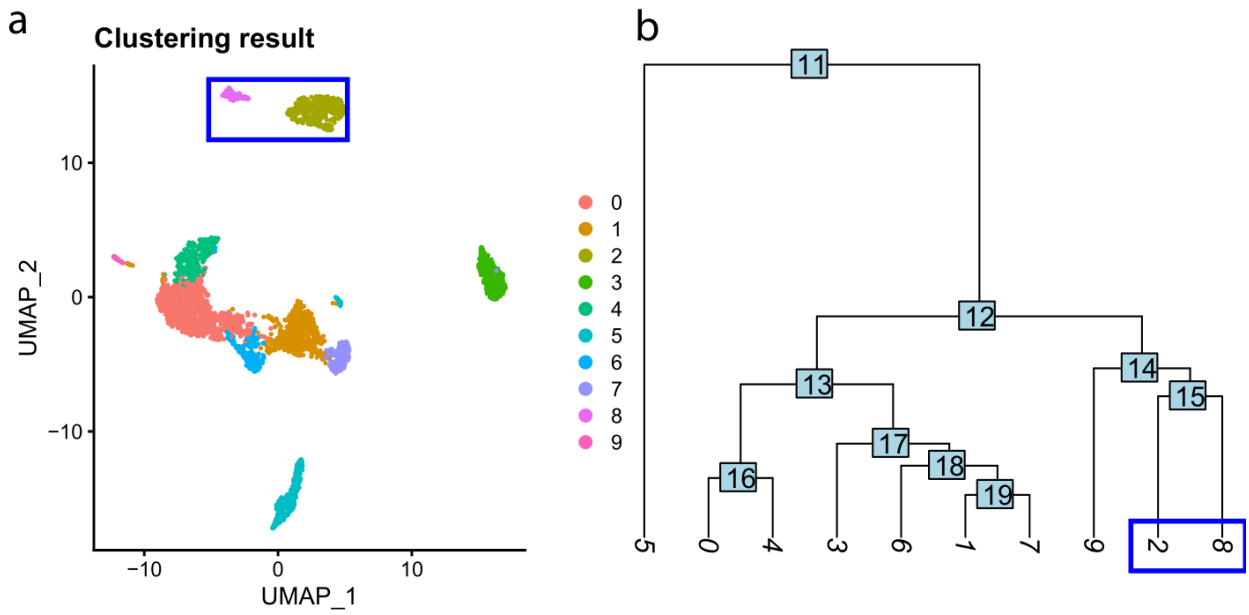


Figure 3.18: A demonstration of using ClusterDE in the presence of multiple cell clusters.

a, A UMAP visualization of Seurat clusters found in the PBMC dataset Rep1 10x(v3). The blue box labels two neighboring clusters, which roughly represent the $CD14^+/CD16^+$ monocytes. **b**, The cluster tree constructed by Seurat, with clusters 2 and 8 corresponding to the two clusters in the blue box in **a**. It is recommended that ClusterDE can help annotate two neighboring cell clusters in UMAP or a cluster tree, based on the more trustworthy post-clustering DE genes identified by ClusterDE. The reason is that neighboring clusters are more likely to be spurious.

CHAPTER 4

Summary and future directions

In this dissertation, two gene selection methods, scPNMF and ClusterDE, have been introduced as two solutions to different tasks in single-cell sequencing data analysis. Below I summarize the two methods and list the future research directions for each one.

4.1 Sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling

In Chapter 2, we proposed scPNMF, an unsupervised gene selection and data projection method for scRNA-seq data. Motivated by the nature of targeted gene filing data, the major goals of scPNMF include selecting a fixed number of informative genes to distinguish cell types and guiding gene selection for targeted gene profiling experiments. Moreover, scPNMF can project a new targeted gene profiling dataset with the selected genes to the low-dimensional space that embeds a reference scRNA-seq dataset. Besides, scPNMF also works as a dimensionality reduction method with good interpretability. Each dimension in the low-dimensional space found by scPNMF can be considered as a new functional "feature" (as a linear combination of correlated and thus functionally related genes). The mutual exclusiveness makes the PNMF bases used in scPNMF advantageous over the PCA bases in terms of removing confounding effects. Therefore, scPNMF has great potential in deciphering cell heterogeneity in single-cell data by working as an interpretable dimensionality reduction method.

A few related questions remain open. The first key question for gene selection is: how many genes should be selected as informative genes to fully capture the biological variations of

interest? In our studies, we observe that the informative gene number being 200 is generally an elbow point for the clustering accuracies for most gene selection methods, including scPNMF (Fig. 2.14). Therefore, 200 genes may be an empirical guideline for sufficiently capturing biological variations in scRNA-seq data. However, it remains challenging to decide the minimum number of informative genes, given that the underlying cell sub-population structure is data-specific and might be complex. We plan to explore this problem in the future with the possible use of information theory.

Second, the idea of selecting the informative genes by leveraging the linear dimension reduction methods can be extended to accommodate other single-cell multi-omics technologies, such as chromatin accessibility landscapes measured by single-cell ATAC-seq [62], or even to integrate data across multi-omics datasets. The extensions can also applied to cases where the mutually exclusive functional groups are conceptually existent, for instance, in determining surrogate variables.

Third, it is worth noting that the multimodality test for basis selection in scPNMF only accounts for discrete cell types but not continuous cell trajectories. Therefore, other strategies are needed to select informative bases to capture biological variations along continuous cell trajectories.

4.2 Post-clustering differential expression methods robust to false-positive inflation caused by double dipping

In Chapter 3, we proposed ClusterDE, which is an effective solution to the double dipping issue in post-clustering DE analysis. Notably, ClusterDE adapts to a wide range of clustering algorithms and DE tests. For post-clustering DE analysis with more than two clusters, we recommend using ClusterDE in a stepwise manner.

ClusterDE is our first attempt to solve the double dipping issue with the help of realistic *in silico* synthetic null data as a negative control, and identify discoveries by contrasting the target data and the synthetic null. Therefore, the concept of synthetic null data (*in silico*

negative control) can be readily extended to other analyses also affected by double dipping, such as multi-batch post-clustering DE analysis, post-pseudotime DE analysis [84], post-spatially variable gene identification, and data integration analysis. As double dipping is almost surely unavoidable in single-cell sequencing data analysis due to the lack of external knowledge, we proposed a general strategy to reduce false discoveries caused by double dipping by setting up synthetic null data and using a contrastive strategy to find more reliable discoveries. The major obstacles to be overcome are how to make an accurate null hypothesis in new tasks, and how to generate realistic in silico negative control data. Some of our group members are already putting effort into these cutting-edge tasks.

Secondly, we will continue to work on the theoretical part of the ClusterDE method. As we have studied the post-clustering DE analysis for a while, a clear definition has been made of the analysis task. We will illustrate why ClusterDE succeeds in controlling FDR in an asymptotic way, and why competing methods conceptually fail to control FDR. We are currently wrapping up this part into a separate manuscript.

Third, improving the power performance while controlling the FDR control is always another future direction of ClusterDE. This aligns with the update of the Clipper [69] method, which was developed by Xinzhou Ge, also an author of the ClusterDE method. We would like to explore alternative formats and combinations of contrast scores. The theoretical evidence also assists this part.

4.3 Combination of scPNMF and ClusterDE for general single-cell sequencing data analysis

In cutting-edge single-cell sequencing data analysis pipelines, such as the R package `Seurat` [9], performing cell clustering with informative gene selection and identifying cell-type marker genes are both essential steps. As these two tasks appear in a sequential manner, questions such as the overlaps between these two types of "interesting" genes, and the influence of the results of one task on the other have already brought up researchers' interest [86]. We are happy to explore these problems with our understanding, knowledge, and experience on

these tasks. Hopefully, it will provide researchers in this field with a more accurate and interpretable tool to decipher gene functions and find proper interesting genes.

Bibliography

[1] Diego Adhemar Jaitin et al. "Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types". In: *Science* 343.6172 (2014), pp. 776–779.

[2] S Steven Potter. "Single-cell RNA sequencing for the study of development, physiology and disease". In: *Nature Reviews Nephrology* 14.8 (2018), pp. 479–492.

[3] Mario L Suvà and Itay Tirosh. "Single-cell RNA sequencing in cancer: lessons learned and emerging challenges". In: *Molecular cell* 75.1 (2019), pp. 7–12.

[4] Arjun Raj et al. "Imaging individual mRNA molecules using multiple singly labeled probes". In: *Nature methods* 5.10 (2008), pp. 877–879.

[5] Jeffrey R Moffitt et al. "High-throughput single-cell gene-expression profiling with multiplexed error-robust fluorescence in situ hybridization". In: *Proceedings of the National Academy of Sciences* 113.39 (2016), pp. 11046–11051.

[6] Fatma Uzbas et al. "BART-Seq: cost-effective massively parallelized targeted sequencing for genomics, transcriptomics, and single-cell analysis". In: *Genome biology* 20.1 (2019), pp. 1–16.

[7] Jamie L Marshall et al. "HyPR-seq: Single-cell quantification of chosen RNAs via hybridization and sequencing of DNA probes". In: *Proceedings of the National Academy of Sciences* 117.52 (2020), pp. 33404–33413.

[8] Mirjana Efremova and Sarah A Teichmann. "Computational methods for single-cell omics across modalities". In: *Nature methods* 17.1 (2020), pp. 14–17.

[9] Seurat. *Analysis, visualization, and integration of spatial datasets with Seurat*. 2021. URL: https://satijalab.org/seurat/articles/spatial_vignette.html.

[10] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19.1 (2018), pp. 1–5.

[11] Olivier Thellin et al. "Housekeeping genes as internal standards: use and limits". In: *Journal of biotechnology* 75.2-3 (1999), pp. 291–295.

[12] Eli Eisenberg and Erez Y Levanon. "Human housekeeping genes, revisited". In: *TRENDS in Genetics* 29.10 (2013), pp. 569–574.

[13] Aravind Subramanian et al. "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". In: *Cell* 171.6 (2017), pp. 1437–1452.

[14] Philip Brennecke et al. "Accounting for technical noise in single-cell RNA-seq experiments". In: *Nature methods* 10.11 (2013), pp. 1093–1095.

[15] Fang Wang et al. "SCMarker: ab initio marker selection for single cell transcriptome profiling". In: *PLoS computational biology* 15.10 (2019), e1007445.

[16] Zhijian Yuan, Zhirong Yang, and Erkki Oja. "Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering". In: *Neural Process. Lett* (2009), pp. 11–13.

[17] Jesse M Zhang, Govinda M Kamath, and N Tse David. "Valid post-clustering differential analysis for single-cell RNA-Seq". In: *Cell systems* 9.4 (2019), pp. 383–392.

[18] Anna Neufeld et al. "Inference after latent variable estimation for single-cell RNA sequencing data". In: *Biostatistics* (Dec. 2022). kxac047. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxac047. eprint: https://academic.oup.com/biostatistics/advance-article-pdf/doi/10.1093/biostatistics/kxac047/47830968/kxac047.pdf. URL: https://doi.org/10.1093/biostatistics/kxac047.

[19] Alexis Vandenbon and Diego Diez. "A clustering-independent method for finding differentially expressed genes in single-cell transcriptome data". In: *Nature communications* 11.1 (2020), p. 4318.

[20] Anna Hendrika Cornelia Vlot, Setareh Maghsudi, and Uwe Ohler. "Cluster-independent marker feature identification from single-cell omics data using SEMITONES". In: *Nucleic Acids Research* 50.18 (2022), e107–e107.

[21] Chanwoo Kim et al. "MarcoPolo: a method to discover differentially expressed genes in single-cell RNA-seq data without depending on prior clustering". In: *Nucleic acids research* 50.12 (2022), e71–e71.

[22] Jiadi Zhu and Youlong Yang. "scMEB: a fast and clustering-independent method for detecting differentially expressed genes in single-cell RNA-seq data". In: *BMC genomics* 24.1 (2023), pp. 1–15.

[23] Alsu Missarova et al. "Sensitive cluster-free differential expression testing." In: *bioRxiv* (2023), pp. 2023–03.

[24] Huidong Chen et al. "SIMBA: SIngle-cell eMBedding Along with features". In: *Nature Methods* (2023), pp. 1–11.

[25] Baolin Liu et al. "An entropy-based metric for assessing the purity of single cell populations". In: *Nature communications* 11.1 (2020), p. 3155.

[26] Jiyuan Fang et al. "Clustering Deviation Index (CDI): a robust and accurate internal measure for evaluating scRNA-seq data clustering". In: *Genome Biology* 23.1 (2022), p. 269.

[27] Wei Vivian Li. "Phitest for analyzing the homogeneity of single-cell populations". In: *Bioinformatics* 38.9 (2022), pp. 2639–2641.

[28] Maria Mircea et al. "Phiclust: a clusterability measure for single-cell transcriptomics reveals phenotypic subpopulations". In: *Genome Biology* 23.1 (2022), pp. 1–24.

[29] I.N. Grabski, K. Street, and R.A. Irizarry. "Significance analysis for clustering with single-cell RNA-sequencing data". In: *Nat Methods* 1.1 (2023), p. 1.

[30] Kenneth D Birnbaum. "Power in numbers: single-cell RNA-seq strategies to dissect complex tissues". In: *Annual review of genetics* 52 (2018), pp. 203–221.

[31] Chenxu Zhu, Sebastian Preissl, and Bing Ren. "Single-cell multimodal omics: the power of many". In: *Nature methods* 17.1 (2020), pp. 11–14.

[32] Christoph Hafemeister and Rahul Satija. "Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression". In: *Genome biology* 20.1 (2019), pp. 1–15.

[33] Tim Stuart et al. "Comprehensive Integration of Single-Cell Data". In: *Cell* 177 (2019), pp. 1888–1902. DOI: 10.1016/j.cell.2019.05.031. URL: https://doi.org/10.1016/j.cell.2019.05.031.

[34] Aaron TL Lun, Karsten Bach, and John C Marioni. "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts". In: *Genome biology* 17.1 (2016), p. 75.

[35] Tallulah S Andrews and Martin Hemberg. "M3Drop: dropout-based feature selection for scRNASeq". In: *Bioinformatics* 35.16 (2019), pp. 2865–2867.

[36] Lan Jiang et al. "GiniClust: detecting rare cell types from single-cell gene expression data with Gini index". In: *Genome biology* 17.1 (2016), p. 144.

[37] Evan Z Macosko et al. "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets". In: *Cell* 161.5 (2015), pp. 1202–1214.

[38] Maayan Baron et al. "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure". In: *Cell systems* 3.4 (2016), pp. 346–360.

[39] Xun Zhu et al. "Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization". In: *PeerJ* 5 (2017), e2888.

[40] Philippe Boileau, Nima S Hejazi, and Sandrine Dudoit. "Exploring high-dimensional biological data with sparse contrastive principal component analysis". In: *Bioinformatics* 36.11 (2020), pp. 3422–3430.

[41] Zhana Duren et al. "Integrative analysis of single-cell genomics data by coupled non-negative matrix factorizations". In: *Proceedings of the National Academy of Sciences* 115.30 (2018), pp. 7723–7728.

[42] Ghislain Durif et al. "Probabilistic count matrix factorization for single cell expression data analysis". In: *Bioinformatics* 35.20 (2019), pp. 4011–4019.

[43] Shuqin Zhang et al. "Dimensionality reduction for single cell RNA sequencing data using constrained robust non-negative matrix factorization". In: *NAR Genomics and Bioinformatics* 2.3 (2020), lqaa064.

[44] Chao Gao and Joshua D Welch. "Iterative Refinement of Cellular Identity from Single-Cell Data Using Online Learning". In: *International Conference on Research in Computational Molecular Biology*. Springer. 2020, pp. 248–250.

[45] Zi Yang and George Michailidis. "A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data". In: *Bioinformatics* 32.1 (2016), pp. 1–8.

[46] Joshua D Welch et al. "Single-cell multi-omic integration compares and contrasts features of brain cell identity". In: *Cell* 177.7 (2019), pp. 1873–1887.

[47] Zhirong Yang and Erkki Oja. "Linear and nonlinear projective nonnegative matrix factorization". In: *IEEE Transactions on Neural Networks* 21.5 (2010), pp. 734–749.

[48] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.

[49] Jean-Philippe Brunet et al. "Metagenes and molecular pattern discovery using matrix factorization". In: *Proceedings of the national academy of sciences* 101.12 (2004), pp. 4164–4169.

[50] Jose Ameijeiras-Alonso, Rosa M Crujeiras, and Alberto Rodrıguez-Casal. "Mode testing, critical bandwidth and excess mass". In: *Test* 28.3 (2019), pp. 900–919.

[51] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature methods* 16.12 (2019), pp. 1289–1296.

[52] Saskia Freytag et al. "Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data". In: *F1000Research* 7 (2018).

[53] Robert D Barber et al. "GAPDH as a housekeeping gene: analysis of GAPDH mRNA expression in a panel of 72 human tissues". In: *Physiological genomics* 21.3 (2005), pp. 389–395.

[54] Nicholas Silver et al. "Selection of housekeeping genes for gene expression studies in human reticulocytes using real-time PCR". In: *BMC molecular biology* 7.1 (2006), p. 33.

[55] Collin M Blakely et al. "Evolution and clinical impact of co-occurring genetic alterations in advanced-stage EGFR-mutant lung cancers". In: *Nature genetics* 49.12 (2017), pp. 1693–1704.

[56] Ben O'leary, Richard S Finn, and Nicholas C Turner. "Treating cancer with selective CDK4/6 inhibitors". In: *Nature reviews Clinical oncology* 13.7 (2016), pp. 417–430.

[57] Carminia Maria Della Corte et al. "Efficacy of continuous EGFR-inhibition and role of Hedgehog in EGFR acquired resistance in human lung cancer cells with activating mutation of EGFR". In: *Oncotarget* 8.14 (2017), p. 23020.

[58] Tianyi Sun et al. "scDesign2: an interpretable simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured". In: *bioRxiv* (2020).

[59] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[60] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.

[61] Florian Buettner et al. "f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq". In: *Genome biology* 18.1 (2017), pp. 1–13.

[62] Sebastian Pott and Jason D Lieb. "Single-cell ATAC-seq: strength in numbers". In: *Genome Biology* 16.1 (2015), pp. 1–4.

[63] Dongyuan Song, Kexin Li, and Jingyi Jessica Li. *scPNMF: sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling*. Version 1. May 2021. DOI: 10.5281/zenodo.4797997. URL: https://doi.org/10.5281/zenodo.4797997.

[64] Ashraful Haque et al. "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications". In: *Genome medicine* 9.1 (2017), pp. 1–12.

[65] Tallulah S Andrews et al. "Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data". In: *Nature protocols* 16.1 (2021), pp. 1–9.

[66] Yuhan Hao et al. "Integrated analysis of multimodal single-cell data". In: *Cell* 184.13 (2021), pp. 3573–3587.

[67] Anna Neufeld et al. "Data thinning for convolution-closed distributions". In: *arXiv preprint arXiv:2301.07276* (2023).

[68] Dongyuan Song et al. "scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics". In: *Nature Biotechnology* (2023), pp. 1–6.

[69] Xinzhou Ge et al. "Clipper: p-value-free FDR control on high-throughput data from two conditions". In: *Genome biology* 22.1 (2021), pp. 1–29.

[70] Valentine Svensson. "Droplet scRNA-seq is not zero-inflated". In: *Nature Biotechnology* 38.2 (2020), pp. 147–150.

[71] Tae Hyun Kim, Xiang Zhou, and Mengjie Chen. "Demystifying "drop-outs" in single-cell UMI data". In: *Genome biology* 21.1 (2020), pp. 1–19.

[72] Ruochen Jiang et al. "Statistics or biology: the zero-inflation controversy about scRNA-seq data". In: *Genome biology* 23.1 (2022), pp. 1–24.

[73] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing statistical hypotheses*. Vol. 3. Springer, 1986.

[74] Tianyi Sun et al. "scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured". In: *Genome biology* 22.1 (2021), p. 163.

[75] Andrew McDavid et al. "Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments". In: *Bioinformatics* 29.4 (2013), pp. 461–467.

[76] Rina Foygel Barber and Emmanuel J Candès. "Controlling the false discovery rate via knockoffs". In: (2015).

[77] Angelo Duò, Mark D Robinson, and Charlotte Soneson. "A systematic performance evaluation of clustering methods for single-cell RNA-seq data". In: *F1000Research* 7 (2018).

[78] Emmanuel Candes et al. "Panning for gold:'model-X'knockoffs for high dimensional controlled variable selection". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 80.3 (2018), pp. 551–577.

[79] Grace XY Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature communications* 8.1 (2017), pp. 1–12.

[80] Luyi Tian et al. "Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments". In: *Nature methods* 16.6 (2019), pp. 479–487.

[81] Jiarui Ding et al. "Systematic comparison of single-cell and single-nucleus RNA-sequencing methods". In: *Nature biotechnology* (2020), pp. 1–10.

[82] Yu Matsuzaki et al. "β2-Microglobulin is an appropriate reference gene for RT-PCR-based gene expression analysis of hematopoietic stem cells". In: *Regenerative Therapy* 1 (2015), pp. 91–97.

[83] Gordon K Smyth and Terry Speed. "Normalization of cDNA microarray data". In: *Methods* 31.4 (2003), pp. 265–273.

[84] Dongyuan Song and Jingyi Jessica Li. "PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data". In: *Genome biology* 22.1 (2021), p. 124.

[85] Dongyuan Song, Kexin Li, and Jingyi Jessica Li. *ClusterDE: a post-clustering differential expression (DE) method robust to false-positive inflation caused by double dipping*. Version 0.9.0. July 2023. DOI: 10.5281/zenodo.8161964. URL: https://doi.org/10.5281/zenodo.8161964.

[86] FionaMoon. *difference between highly variable genes and marker genes*. 2022. URL: https://github.com/satijalab/seurat/issues/5743.