Title

Improving Statistical Rigor in Single-cell and Spatial Omics

Author

Song, Dongyuan

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Improving Statistical Rigor in Single-cell and Spatial Omics

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Bioinformatics

by

Dongyuan Song

2024

ABSTRACT OF THE DISSERTATION

Improving Statistical Rigor in Single-cell and Spatial Omics

by

Dongyuan Song
Doctor of Philosophy in Bioinformatics
University of California, Los Angeles, 2024
Professor Jingyi Li, Chair

The recent technological revolution in single-cell and spatial omics has provided unprecedented multi-modal views of individual cells, transforming our understanding of cell biology in health and disease. Numerous computational methods have been developed to analyze data generated from these technologies. However, the statistical rigor of existing computational methods is often questionable: many computational methods are complicated "black-box" algorithms (e.g., deep-learning-based methods). Therefore, it remains challenging to obtain correct statistical interpretation (e.g., well-calibrated p-values), and to avoid misinterpretation of observed data (e.g., exaggerated false discoveries). Due to the existence of numerous methods, the field crucially needs precise, statistically robust, and interpretable methods. During my Ph.D., I have been focusing on combining statistics and computational biology to provide accurate statistical interpretation to computational analyses in single-cell and spatial omics. This dissertation aims to address this statistical rigor issue through three main themes.

My first theme concentrates on the probabilistic generative models for high-dimensional single-cell and spatial multi-omics data. The realistic simulation of single-cell and spatial multi-omics data plays a critical role in both evaluating the performance of computational tools and facilitating the exploration of experimental designs. However, the complex topology of cells and the high-dimension features pose significant challenges to this endeavor. To overcome this challenge, I developed scDesign3, the first unified framework for realistic in

silico data generation of both single-cell and spatial omics [1].

My second theme focuses on differential expression (DE) tests and false discovery rate (FDR) control based on inferred covariates. Identifying differentially expressed (DE) genes between or between cell states is a crucial task in investigating the underlying molecular mechanisms in cells. However, in single-cell RNA sequencing (scRNA-seq) analysis, the latent cell states are usually inferred from the data (e.g., inferred cell types by clustering or continuous trajectories by pseudotime inference). Therefore, conventional statistical tests can behave incorrectly if we ignore the fact that the covariates are inferred rather than observed. Hence, I developed PseudotimeDE, a robust DE method that accounts for pseudotime inference uncertainty and yields well-calibrated p-values [2]. Separately, post-clustering DE was another related issue that drew our attention. This two-step procedure uses the same data twice: once to define cell clusters as potential cell types, and then to identify DE genes as potential cell-type marker genes. This practice, often known as "double dipping," can lead to the erroneous identification of false-positive cell-type marker genes, particularly when the cell clusters themselves are not well-defined. To overcome this challenge, I proposed ClusterDE, a post-clustering DE method for controlling the FDR of identified DE genes regardless of the clustering quality by using "synthetic null data" [3].

My third theme aims at feature selection and subsampling in large-scale scRNA-seq data. The large number of genes ($\sim$20,000) and increasing number of measured cells ($>$ 1 million) in scRNA-seq datasets remain a challenge for data analysis. A practical solution to this computational bottleneck involves the strategic selection of a subset of cells or genes. We developed scSampler, a fast diversity-preserving cell subsampling inspired by space-filling design in the field of experimental design [4]. scSampler selects a small subset of cells to accurately represent the primary variability in the entire dataset. In addition, we developed scPNMF, an unsupervised gene selection method through matrix factorization [5]. This method effectively selects a significantly smaller subset of genes ($\sim$100) while still achieving robust discrimination in cell-type identification. We developed scGTM, a flexible and interpretable model that captures the trend of gene expression along the pseudotime of cells to select genes with specific expression patterns [6].

The dissertation of Dongyuan Song is approved.

Xinshu Xiao

Roy Wollman

Alexander Hoffmann

Jingyi Li, Committee Chair

University of California, Los Angeles

2024

To my parents and my wife

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This dissertation is the result of not only my efforts but also the invaluable support of my advisor, collaborators, family, and friends. I am truly grateful for the roles they played during my Ph.D. journey at UCLA.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Jingyi Jessica Li. Working with Jessica is the luckiest thing of my entire academic career. Since joining Jessica's group five years ago, Jessica has taught me the basics of doing research and provided generous help in both academics and life. Without her guidance, support, and encouragement, I would not be able to finish so many interesting projects. She will always be my role model in my future career. I wish I could make my unique contributions to science that she would be proud of.

I would also like to thank the members of my dissertation committee, Dr. Xinshu Grace Xiao, Dr. Roy Wollman, and Dr. Alexander Hoffmann, for their constructive feedback and insightful comments on my research. Their expertise and suggestions have greatly enhanced the quality of this research.

I would like to thank all current and previous members of the Junction of Statistics and Biology (JSB) lab for their assistance and discussions. In particular, I would like to thank Dr. Kexin Li and Dr. Xinzhou Ge. They are not only my collaborators in several important projects, but also my best friends from LA to Boston. I would also like to thank Qingyang Wang, Guanao Yan, Dr. Tianyi Sun, Christy Lee, Chris Dong, and Dr. Nan Miles Xi for their contribution in several publications.

Last but not least, I sincerely thank my friends and family for their love and support in this journey. I would like to thank my mother, Quan Hong, and my father, Zhenqian Song, for their unconditional love in my life. I would like to thank my friends, Xuhang Li and Huiya Yang couple, for our friendship from undergraduate to Ph.D. I would like to especially thank my wife, Dr. Xutao Wang, for her company during my difficult time, including my Ph.D. application, the COVID-19 pandemic, and my job search.

VITA

2013–2017	B.S. in *Biological Science*, School of Life Sciences, Fudan University, Shanghai, China.
2017–2019	M.S. in *Computational Biology & Quantitative Genetics*, Harvard T.H. Chan School of Public Health, Boston.
2019–2024	Graduate Student Researcher, *Bioinformatics*, University of California, Los Angeles.

PUBLICATIONS

(* indicates equal contribution. The list includes only those works where I am the lead author during my Ph.D. studies.)

Song, D., Wang, Q., Yan, G., Liu, T., Sun, T., Li, J. J. (2024). scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics.*Nature Biotechnology*, 42(2), 247-252.

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*.

Song, D.*, Xi, N. M.*, Li, J. J., Wang, L. (2022). scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. *Bioinformatics*, 38(11), 3126-3127.

Cui, E. H.*, **Song, D.***, Wong, W. K., Li, J. J. (2022). Single-cell generalized trend model (scGTM): a flexible and interpretable model of gene expression trend along cell pseudotime. *Bioinformatics*, 38(16), 3927-3934.

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.

Song, D., Li, J. J. (2021). PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data. *Genome biology*, 22(1), 124.

CHAPTER 1

Introduction

The single-cell and spatial omics technologies are at the forefront of biotechnology, offering unprecedented insights into complex biological systems. In the history of technology development, the first attempt is single-cell RNA sequencing (scRNA-seq), which measures the whole transcriptomic profiles of each individual cell [7, 8]. At the same time, new technologies are developed for measuring other types of omics at single-cell resolution, such as single-cell chromatin accessibility (e.g., scATAC-seq [9] and sci-ATAC-seq [10]), single-cell DNA methylation [11] and single-cell protein abundance (e.g., single-cell mass cytometry [12]). Moreover, to provide a comprehensive view of cellular function and regulation, single-cell multi-omics technologies are invented to simultaneously measure more than one feature modality, such as SNARE-seq (gene expression plus chromatin accessibility) [13] and CITE-seq (gene expression plus surface protein abundance) [14]. In addition, spatial omics takes this a step further by not only measuring omics features of individual cells but also preserving the spatial locations of these cells within tissues at different levels of resolutions (e.g., 10x Visium [15], Slide-seq [16], Slide-seq V2 [17], and MERFISH [18]).

The rapid development of various experimental technologies has led to the explosion of computational tools; thousands of methods have been developed to address different analytic tasks [19]. Some representative tasks include cell clustering to identify discrete cell types or states [20], trajectory inference to model continuous transitions [21], and differential expression analysis for detecting statistically significant gene changes [22]. One challenge in the current field is the lack of statistical rigor. First, it remains unclear how we should simulate single-cell and spatial omics data to check and benchmark existing computational methods from different statistical perspectives. Second, the statistical interpretation of differential

gene analysis is often questionable, such as invalid p-values and exaggerated false discovery rate (FDR). Lastly, the increasing number of measured cells in newer single-cell datasets greatly increases computational time, making many existing computational methods not scalable.

This dissertation will focus on my contribution to improving the statistical rigor in single-cell and spatial omics during my doctoral studies. We selected three representative publications as the solutions to the above three challenges. We first discussed the use of scDesign3 to generate realistic in silico data with a unified statistical framework. Next, we discuss the use of PseudotimeDE to generate well-calibrated p values of differential expression tests in trajectory analysis. Lastly, we discussed the use of scSampler for large-scale data subsampling to accelerate downstream analysis.

1.1 scDesign3: generation of realistic in silico data for multimodal single-cell and spatial omics

In the single-cell and spatial omics field, computational challenges include method benchmarking, data interpretation, and in silico data generation. To address these challenges, in Chapter 2, we propose an all-in-one statistical simulator, scDesign3, to generate realistic single-cell and spatial omics data, including various cell states, experimental designs, and feature modalities, by learning interpretable parameters from real datasets. Furthermore, using a unified probabilistic model for single-cell and spatial omics data, scDesign3 can infer biologically meaningful parameters, assess the goodness-of-fit of inferred cell clusters, trajectories, and spatial locations, and generate in silico negative and positive controls for benchmarking computational tools.

1.2 PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data

To investigate the molecular mechanisms underlying cell state changes, a crucial analysis is to identify differentially expressed (DE) genes along the pseudotime inferred from single-cell RNA-sequencing data. However, existing methods do not account for pseudotime inference uncertainty, and they have either ill-posed p-values or restrictive models. In Chapter 3, we propose PseudotimeDE, a DE gene identification method that adapts to various pseudotime inference methods, accounts for pseudotime inference uncertainty, and produces well-calibrated p-values. Comprehensive simulations and real-data applications verify that PseudotimeDE outperforms existing methods in false discovery rate control and power.

1.3 scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data

The number of cells measured in single-cell transcriptomic data has grown rapidly in recent years. For such large-scale data, subsampling is a powerful and often necessary tool for exploratory data analysis. However, the easiest random subsampling is not ideal from the perspective of preserving rare cell types. Therefore, diversity-preserving subsampling is required for a fast exploration of cell types in a large-scale dataset. In Chapter 4, we propose scSampler, an algorithm for fast diversity-preserving subsampling of single-cell transcriptomic data. Using simulated and real data, we show that scSampler consistently outperforms existing subsampling methods in terms of both the computational time and the Hausdorff distance between the full and subsampled datasets.

1.4 Summary

During my doctoral study, I developed the aforementioned computational methods that aim to improve the statistical rigor and computational speed in single-cell and spatial omics. The details of these projects will be described in Chapter 2–4 of this dissertation.

CHAPTER 2

scDesign3: generation of realistic in silico data for multimodal single-cell and spatial omics

2.1 Introduction

Single-cell and spatial omics technologies have provided unprecedented multi-modal views of individual cells. As the earliest single-cell technologies, single-cell RNA-seq (scRNA-seq) enabled the measurement of transcriptome-wide gene expression levels and the discovery of novel cell types and continuous cell trajectories [7, 23]. Later, other single-cell omics technologies were developed to measure additional molecular feature modalities, including single-cell chromatin accessibility (e.g., scATAC-seq [9] and sci-ATAC-seq [10]), single-cell DNA methylation [11], and single-cell protein abundance (e.g., single-cell mass cytometry [12]). More recently, single-cell multi-omics technologies were invented to simultaneously measure more than one feature modality, such as SNARE-seq (gene expression and chromatin accessibility) [13] and CITE-seq (gene expression and surface protein abundance) [14]. In parallel to single-cell omics, spatial transcriptomics technologies were advanced to profile gene expression levels with spatial location information of cell neighborhoods (i.e., multi-cell resolution; e.g., 10x Visium [15] and Slide-seq [16]), individual cells (i.e., single-cell resolution; e.g., Slide-seqV2 [17]), or sub-cellular components (i.e., sub-cellular resolution; e.g., MERFISH [18]).

Thousands of computational methods have been developed to analyze single-cell and spatial omics data for various tasks [24], making method benchmarking a pressing challenge for method developers and users. Fair benchmarking relies on comprehensive evaluation metrics that reflect real data analytical goals; however, meaningful metrics usually require

ground truths that are rarely available in real data. (For example, most real datasets contain "cell types" obtained by cell clustering and manual annotation without external validation; using such "cell types" as ground truths would biasedly favor the clustering method used in the original study.) Therefore, fair benchmarking demands in silico data that contain ground truths and mimic real data, calling for realistic simulators.

The demand for realistic simulators motivated two recent benchmark studies, in which 12 and 16 scRNA-seq simulators were evaluated [25, 26]. Due to the complexity of scRNA-seq data, these benchmarked simulators all require training on real scRNA-seq data, and they are more realistic than the de novo simulators that use no real data but generate synthetic data from theoretical models [26].

Although the benchmark studies found that the simulators scDesign2 [27], ZINB-WaVE [28], and muscat [29] can generate realistic scRNA-seq data from discrete cell types [25, 26], few simulators can generate realistic scRNA-seq data from continuous cell trajectories by mimicking real data [26, 30–33]. Moreover, realistic simulators are lacking for single-cell omics other than scRNA-seq, not to mention single-cell multi-omics and spatial transcriptomics. (To our knowledge, simATAC is the only scATAC-seq simulator that learns from real data, but it can only generate discrete cell types [34].) Hence, a large gap exists between the diverse benchmarking needs and the limited functionalities of existing simulators.

To fill in the gap, we introduce scDesign3, a realistic and most versatile simulator to date. As Fig. 2.1a shows, scDesign3 can generate realistic synthetic data from diverse settings, including cell latent structures (discrete cell types and continuous cell trajectories), feature modalities (e.g., gene expression, chromatin accessibility, methylation, protein abundance, and multi-omics), spatial locations, and experimental designs (batches and conditions). Note that the predecessor scDesign2 is a special case of scDesign3 for generating scRNA-seq data from discrete cell types; a detailed comparison of scDesign3 with the previous two versions is in Table 2.1. To our knowledge, scDesign3 offers the first probabilistic model that unifies the generation and inference for single-cell and spatial omics data. Equipped with interpretable parameters and a model likelihood, scDesign3 is beyond a versatile simulator and has unique advantages for generating customized in silico data, which can serve as negative and positive

controls for computational analysis, and for assessing the goodness-of-fit of inferred cell clusters, trajectories, and spatial locations in an unsupervised way (Fig. 2.2a).

2.2 scDesign3 methodology

2.2.1 Mathematical notations of scDesign3's training data

The training data of scDesign3 contain three matrices: a cell-by-feature matrix (e.g., features are genes or chromatin regions), a cell-by-state-covariate matrix (e.g., cell-state covariates include the cell type, pseudotime, or spatial coordinate), and an optional cell-by-design-covariate matrix (e.g., design covariates include the batch or condition).

Mathematically, first, we denote by $\mathbf{Y} = [Y_{ij}] \in \mathbb{R}^{n \times m}$ the cell-by-feature matrix with n cells as rows, m features as columns, and Y_{ij} as the measurement of feature j in cell i. For single-cell sequencing data, $\mathbf{Y}$ is often a count matrix (i.e., $\mathbf{Y} \in \mathbb{N}^{n \times m}$, with Y_{ij} indicating the read or unique molecular identifier (UMI) count of feature j in cell i); then the sequencing depth (i.e., the total number of reads or UMIs) is $N = \sum_{i=1}^{n} \sum_{j=1}^{m} Y_{ij}$.

Second, we denote by $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n \times p}$ the cell-by-state-covariate matrix with n cells as rows and p cell-state covariates as columns. In $\mathbf{X}$, the i-th row $\mathbf{x}_i \in \mathbb{R}^p$ is cell i's state covariate vector. Typical cell-state covariates include the cell type ($p = 1$ categorical variable), the cell pseudotime in p lineage trajectories (p continuous variables), and the 2- or 3-dimensional cell spatial locations ($p = 2$ or 3 continuous variables).

Third, we denote by $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_n]^\mathsf{T} \in \mathbb{R}^{n \times q}$ the cell-by-design-covariate matrix with n cells as rows and q design covariates as columns. In $\mathbf{Z}$, the i-th row $\mathbf{z}_i \in \mathbb{R}^q$ is cell i's design covariate vector. Example design covariates are categorical variables such as the batch and condition. Note that $\mathbf{Z}$ is optional: it is not required if cells are from a single condition and measured in a single batch. To simplify the discussion, in the following text, we write $\mathbf{Z} = [\mathbf{b}, \mathbf{c}]$, where $\mathbf{b} = (b_1, \ldots, b_n)^\mathsf{T}$ has $b_i \in \{1, \ldots, B\}$ representing cell i's batch, and $\mathbf{c} = (c_1, \ldots, c_n)^\mathsf{T}$ has $c_i \in \{1, \ldots, C\}$ representing cell i's condition.

2.2.2 Modeling features' marginal distributions

For each feature $j = 1, \ldots, m$ in every cell $i = 1, \ldots, n$, the measurement Y_{ij}—conditional on cell i's state covariates $\mathbf{x_i}$ and design covariates $\mathbf{z}_i = (b_i, c_i)^\mathsf{T}$—is assumed to follow a distribution $F_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i \; ; \; \mu_{ij}, \sigma_{ij}, p_{ij})$, which is specified as the generalized additive model for location, scale and shape (GAMLSS) [35] (i.e., the distribution family F_j depends on feature j only, but the parameters μ_{ij}, σ_{ij}, and p_{ij} depend on both feature j and cell i):

$$\begin{cases} Y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i & \overset{\text{ind}}{\sim} F_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i \; ; \; \mu_{ij}, \sigma_{ij}, p_{ij}) \\ \theta_j(\mu_{ij}) & = \alpha_{j0} + \alpha_{jb_i} + \alpha_{jc_i} + f_{jc_i}(\mathbf{x}_i) \\ \log(\sigma_{ij}) & = \beta_{j0} + \beta_{jb_i} + \beta_{jc_i} + g_{jc_i}(\mathbf{x}_i) \\ \text{logit}(p_{ij}) & = \gamma_{j0} + \gamma_{jb_i} + \gamma_{jc_i} + h_{jc_i}(\mathbf{x}_i) \end{cases}, \tag{1}$$

where $\theta_j(\cdot)$ denotes feature j's specific link function of the mean parameter μ_{ij}, depending on F_j (Table 2.3); σ_{ij} denotes the scale parameter (e.g., standard deviation or dispersion); p_{ij} denotes the zero-inflation proportion parameter. Note that μ_{ij}, σ_{ij}, and p_{ij} do not always co-exist, depending on the form of F_j (Table 2.3). To ensure model identifiability, for $j = 1, \ldots, m$, we set $\alpha_{jb_i} = \beta_{jb_i} = \gamma_{jb_i} = 0$ when $b_i = 1$ and $\alpha_{jc_i} = \beta_{jc_i} = \gamma_{jc_i} = 0$ when $c_i = 1$.

$\theta_j(\mu_{ij})$ is assumed to have feature j's specific intercept α_{j0}, batch b_i's effect α_{jb_i} (specific to feature j), condition c_i's effect α_{jc_i} (specific to feature j), and cell-state covariates $\mathbf{x}_i$'s effect $f_{jc_i}(\mathbf{x}_i)$ (specific to feature j and condition c_i).

$\log(\sigma_{ij})$ is assumed to have feature j's specific intercept β_{j0}, batch b_i's effect β_{jb_i} (specific to feature j), condition c_i's effect β_{jc_i} (specific to feature j), and cell-state covariates $\mathbf{x}_i$'s effect $g_{jc_i}(\mathbf{x}_i)$ (specific to feature j and condition c_i).

$\text{logit}(p_{ij})$ is assumed to have feature j's specific intercept γ_{j0}, batch b_i's effect γ_{jb_i} (specific to feature j), condition c_i's effect γ_{jc_i} (specific to feature j), and cell-state covariates $\mathbf{x}_i$'s effect $h_{jc_i}(\mathbf{x}_i)$ (specific to feature j and condition c_i).

For $\theta_j(\mu_{ij})$, $\log(\sigma_{ij})$, and $\text{logit}(p_{ij})$, the interaction effects are considered between the

condition and cell-state covariates, but not between the batch and cell-state covariates. This modeling choice is made based on empirical observations and the simplicity preference [36].

Note that if only the mean parameter μ_{ij} is assumed to depend on the state covariates $\mathbf{x}_i$, batch b_i, and condition c_i, then the GAMLSS degenerates to a generalized additive model (GAM) [37].

Depending on the modality of feature j (e.g., a gene's UMI count), scDesign3 specifies F_j to be one of the six distributions: Gaussian (Normal), Bernoulli, Poisson, Negative Binomial (NB), Zero-inflated Poisson (ZIP), and Zero-inflated Negative Binomial (ZINB); see Table 2.3 for the specifications. Different specifications of F_j correspond to different link functions $\theta_j(\cdot)$ and parameters; see Table 2.3 for the details.

Depending on cell i's cell-state covariates $\mathbf{x}_i$, scDesign3 specifies the functions $f_{jc_i}(\cdot)$, $g_{jc_i}(\cdot)$, and $h_{jc_i}(\cdot)$ in the corresponding forms. See Table S4 for the details. Below are the three typical forms of $f_{jc_i}(\cdot)$.

(1) When the cell-state covariate is the cell type (out of a total of K_C cell types) and $\mathbf{X} = (x_1, \ldots, x_n)^\mathsf{T}$ is a 1-column matrix with $x_i \in \{1, \ldots, K_C\}$,

$$f_{jc_i}(x_i) = \alpha_{jc_i x_i} \,,$$

which corresponds to cell-type x_i's effect on feature j in condition c_i. Note that for identifiability, $\alpha_{jc_i x_i} = 0$ if $c_i = 1$ or $x_i = 1$.

(2) When the cell-state covariates are the cell pseudotimes in p lineage trajectories, i.e., $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T}$ with x_{il} indicating cell i's pseudotime in the l-th lineage trajectory,

$$f_{jc_i}(\mathbf{x}_i) = \sum_{l=1}^{p} \sum_{k=1}^{K} b_{jc_i lk}(x_{il}) \beta_{jc_i lk} \,,$$

where $\sum_{k=1}^{K} b_{jc_i lk}(\cdot)\beta_{jc_i lk}$ is a cubic spline function for pseudotime in the l-th lineage. This formulation means that feature j under condition c_i has a specific smooth pattern in lineage l. The exact choice K, the dimension of the basis governing the flexibility of f_{jc_i}, is not critical as long as K is not too small (because automatic penalization would be used in the

estimation of f_{jc_i} by the R package `mgcv`, which is used in the R package `gamlss`; see [37]); we set $K = 10$ as default; K cannot be larger than the number of data points.

(3) When the cell-state covariates are 2-dimensional spatial locations, i.e., $\mathbf{x}_i = (x_{i1}, x_{i2})^\mathsf{T}$ indicating cell i's 2-dimensional spatial coordinates,

$$f_{jc_i}(\mathbf{x}_i) = f_{jc_i}^{\mathrm{GP}}(x_{i1}, x_{i2}, K)\,,$$

a low-rank Gaussian process smoother described in [37, 38], where K is the dimension of the basis governing the flexibility of f_{jc_i}. This formulation means that feature j under condition c_i has a smooth 2-dimensional function (i.e., a surface). The exact choice K is not critical as long as K is large (because automatic penalization would be used in the estimation of f_{jc_i} by the R package `mgcv`, which is used in the R package `gamlss`; see [37]); we set $K = 400$ as default; K cannot be larger than the number of data points.

The distribution of $(Y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)$ in Equation (1) is fitted by the function `gamlss()` in the R package `gamlss` (version 5.4-3) or the function `gam()` in the R package `mgcv` (version 1.8-40). The fitted distribution is denoted as $\hat{F}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$, $i = 1, \ldots, n$; $j = 1, \ldots, m$.

2.2.3 Modeling features' joint distribution

For cell $i = 1, \ldots, n$, we denote its measurements of the m features as a random vector $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{im})^\mathsf{T}$, whose joint distribution—conditional on cell i's state covariates $\mathbf{x}_i$ and design covariates $\mathbf{z}_i$—is denoted as $F(\cdot \mid \mathbf{x}_i, \mathbf{z}_i) : \mathbb{R}^m \to [0, 1]$. Section 2.2.2 specifies $F_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$, the distribution of $(Y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)$, $j = 1, \ldots, m$. In scDesign3, the joint cumulative distribution function (CDF) $F(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is modeled from the marginal CDFs $F_1(\cdot \mid \mathbf{x}_i, \mathbf{z}_i), \ldots, F_m(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ using the copula $C(\cdot \mid \mathbf{x}_i, \mathbf{z}_i) : [0, 1]^m \to [0, 1]$:

$$F(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{z}_i) = C\left(F_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, F_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \mid \mathbf{x}_i, \mathbf{z}_i\right)\,,$$

where $\mathbf{y}_i = (y_{i1}, \ldots, y_{im})^\mathsf{T}$ is a realization of $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{im})^\mathsf{T}$.

The copula $C(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ can be (1) the Gaussian copula or (2) the vine copula, specified

below.

The Gaussian copula is defined as

$$
\begin{aligned}
& C\left(F_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, F_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \mid \mathbf{x}_i, \mathbf{z}_i\right) \\
= \; & \Phi_m\left(\Phi^{-1}(F_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i)), \cdots, \Phi^{-1}(F_m(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i)); \; \mathbf{R}(\mathbf{x}_i, \mathbf{z}_i)\right) ,
\end{aligned}
$$

where Φ^{-1} denotes the inverse of the CDF of the standard Gaussian distribution, $\Phi_m(\cdot; \mathbf{R}(\mathbf{x}_i, \mathbf{z}_i))$ denotes the CDF of an m-dimensional Gaussian distribution with a zero mean vector and a covariance matrix equal to the correlation matrix $\mathbf{R}(\mathbf{x}_i, \mathbf{z}_i)$.

An issue with the Gaussian copula is that the likelihood calculation is not straightforward in the high-dimensional case when m is large and the sample correlation matrix $\hat{\mathbf{R}}(\mathbf{x}_i, \mathbf{z}_i)$, as an estimator of $\mathbf{R}(\mathbf{x}_i, \mathbf{z}_i)$, is not invertible. Then, the likelihood cannot be computed based on $\hat{\mathbf{R}}(\mathbf{x}_i, \mathbf{z}_i)$. To address this issue, we consider the vine copula.

The vine copula is a way to "decompose" a high-dimensional copula into a sequence of bivariate copulas, in which every pair of features is modeled as a bivariate Gaussian distribution. In short, the vine copula provides a regular vine (R-vine) structure that uses conditioning to sequentially decompose an m-dimensional copula into a sequence of bivariate copulas; then the m-dimensional copula density function is approximated by the product of the bivariate copula density functions [39]. The vine copula is advantageous to the Gaussian copula because it enables the likelihood calculation in the high-dimensional case.

To estimate $C(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ as either the Gaussian or vine copula, we use the plug-in approach that takes the estimated $\hat{F}_1(\cdot \mid \mathbf{x}_i, \mathbf{z}_i), \ldots, \hat{F}_m(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ from Section 2.2.2. Specifically, when $\hat{F}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is a continuous distribution, each observed y_{ij} is transformed as $u_{ij} = \hat{F}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)$. When $\hat{F}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is a discrete distribution with the support on non-negative integers (e.g., the Poisson distribution), $u_{1j}, \ldots, u_{nj}$ follow a discrete distribution. Since the Gaussian and vine copulas assume that features follow continuous distributions, we use the

distributional transformation as in [27]:

$$u_{ij} = (1 - v_{ij})\hat{F}_j(y_{ij} - 1 \mid \mathbf{x}_i, \mathbf{z}_i) + v_{ij}\hat{F}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)\,,\ y_{ij} = 1, 2, \ldots\,,$$

where v_{ij}'s are sampled independently from Uniform$[0, 1]$, $i = 1, \ldots, n$; $j = 1, \ldots, m$. To unify and simplify our notations, we write $u_{ij} = \tilde{F}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)$, where $\tilde{F}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is the CDF of a continuous distribution.

Then $C(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is estimated from $\mathbf{u}_1, \ldots, \mathbf{u}_n$, where $\mathbf{u}_i = (u_{i1}, \ldots, u_{im})^\mathsf{T}$. For the Gaussian copula, we use the function `cora()` in the R package `Rfast` (version 2.0.6); specifically, $\hat{\mathbf{R}}(\mathbf{x}_i, \mathbf{z}_i)$ is the sample correlation matrix of $\{\Phi^{-1}(\mathbf{u}_j) : (\mathbf{x}_j, \mathbf{z}_j)$ is in a subset of $(\mathbf{x}_i, \mathbf{z}_i)\}$, where $\Phi^{-1}(\mathbf{u}_i) = (\Phi^{-1}(u_{i1}), \ldots, \Phi^{-1}(u_{im}))^\mathsf{T}$. For the vine copula, we use the function `vinecop()` in R package `rvinecoplib` (version 0.6.2.1.1).

Then the estimated joint distribution $\hat{F}(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is

$$\hat{F}(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{z}_i) = \hat{C}\left(\tilde{F}_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, \tilde{F}_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \;\middle|\; \mathbf{x}_i, \mathbf{z}_i\right). \tag{2}$$

2.2.4 Model likelihood, AIC, and BIC

Given Equation (2), the estimated probability density function of cell i's m-dimensional feature vector $\mathbf{y}_i$, conditional on the cell-state covariates $\mathbf{x}_i$ and the design covariates $\mathbf{z}_i$, is

$$\hat{f}(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{z}_i) = \hat{c}\left(\tilde{F}_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, \tilde{F}_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \;\middle|\; \mathbf{x}_i, \mathbf{z}_i\right) \prod_{j=1}^{m} \tilde{f}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)\,,$$

where $\hat{c}(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is the probability density function of $\hat{C}(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$, and $\tilde{f}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$ is the probability density function of $\tilde{F}_j(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$. Hence, the log-likelihood is

$$\begin{aligned}
\ell &= \sum_{i=1}^{n} \log \hat{f}(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{z}_i) \\
&= \sum_{i=1}^{n} \log \hat{c}\left(\tilde{F}_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, \tilde{F}_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \;\middle|\; \mathbf{x}_i, \mathbf{z}_i \right) + \sum_{i=1}^{n} \sum_{j=1}^{m} \log \tilde{f}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i) \\
&= \ell^{\text{Copula}} + \ell^{\text{Marginal}},
\end{aligned}$$

so the log-likelihood ℓ can be written as the sum of a copula log-likelihood

$$\ell^{\text{Copula}} = \sum_{i=1}^{n} \log \hat{c}\left(\tilde{F}_1(y_{i1} \mid \mathbf{x}_i, \mathbf{z}_i), \cdots, \tilde{F}_m(y_{im} \mid \mathbf{x}_i, \mathbf{z}_i) \;\middle|\; \mathbf{x}_i, \mathbf{z}_i \right)$$

and a marginal log-likelihood

$$\ell^{\text{Marginal}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \log \tilde{f}_j(y_{ij} \mid \mathbf{x}_i, \mathbf{z}_i)\,.$$

Given k model parameters and n cells (i.e., the sample size n is the number of cells), the Akaike information criterion (AIC) and Bayesian information criterion (BIC) are

$$\begin{aligned}
\text{AIC} &= 2k - 2\ell\,; \\
\text{BIC} &= 2k \log(n) - 2\ell\,,
\end{aligned}$$

so smaller AIC and BIC values indicate better goodness-of-fit of a model to data.

Because of the likelihood decomposition, the AIC and BIC are also decomposable

$$\begin{aligned}
\text{AIC} &= \text{AIC}^{\text{Copula}} + \text{AIC}^{\text{Marginal}}\,; \\
\text{BIC} &= \text{BIC}^{\text{Copula}} + \text{BIC}^{\text{Marginal}}\,,
\end{aligned}$$

where $\text{AIC}^{\text{Copula}}$ and $\text{BIC}^{\text{Copula}}$ only include the number of parameters in $\hat{c}(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$, and $\text{AIC}^{\text{Marginal}}$ and $\text{BIC}^{\text{Marginal}}$ only include the total number of parameters in $\tilde{f}_1(\cdot \mid \mathbf{x}_i, \mathbf{z}_i), \ldots, \tilde{f}_m(\cdot \mid \mathbf{x}_i, \mathbf{z}_i)$.

2.2.5 Synthetic data generation by scDesign3

To generate a synthetic cell-by-feature matrix $\mathbf{Y}' \in \mathbb{R}^{n' \times m}$, which contains n' synthetic cells and the same m features as in the training data, scDesign3 allows the specification of a cell-by-state-covariate matrix $\mathbf{X}' \in \mathbb{R}^{n' \times p}$ and an optional cell-by-design-covariate matrix $\mathbf{Z}' \in \mathbb{N}^{n' \times q}$ (depending on whether the training data have $\mathbf{Z}$) for the n' synthetic cells. Note that $\mathbf{X}'$ and $\mathbf{Z}'$ can be specified by users, generated by resampling the rows of $\mathbf{X}$ and $\mathbf{Z}$, or sampled from some generative models of the rows of $\mathbf{X}$ and $\mathbf{Z}$.

Given $\mathbf{X}$, $\mathbf{Z}$, and the fitted distributions in Sections 2.2.2 and 2.2.3, scDesign3 samples n' synthetic cells in the following steps.

First, for each synthetic cell i', given its cell-state covariates $\mathbf{x}_{i'}$ and design covariates $\mathbf{z}_{i'}$, we independently sample an m-dimensional vector (with values in $[0, 1]$) from the m-dimensional copula estimated in Section 2.2.3:

$$(U_{i'1}, \ldots, U_{i'm})^\mathsf{T} \sim \hat{C}(\cdot \mid \mathbf{x}_{i'}, \mathbf{z}_{i'}), \; i' = 1, \ldots, n'.$$

Second, based on the m features' fitted marginal distributions in Section 2.2.2, we calculate the conditional distribution of $Y_{i'j}$, the measurement of feature j in synthetic cell i', given the synthetic cell's cell-state covariates $\mathbf{x}_{i'}$ and design covariates $\mathbf{z}_{i'} = (b_{i'}, c_{i'})^\mathsf{T}$, where $b_{i'} \in \{1, \ldots, B\}$ and $c_{i'} \in \{1, \ldots, C\}$:

$$Y_{i'j} \mid \mathbf{x}_{i'}, \mathbf{z}_{i'} \sim \hat{F}_j(\cdot \mid \mathbf{x}_{i'}, \mathbf{z}_{i'}) = F_j(\cdot \mid \mathbf{x}_{i'}, \mathbf{z}_{i'} \; ; \; \hat{\mu}_{i'j}, \hat{\sigma}_{i'j}, \hat{p}_{i'j}),$$

where

$$\begin{cases} \theta(\hat{\mu}_{i'j}) & = \hat{\alpha}_{j0} + \hat{\alpha}_{jb_{i'}} + \hat{\alpha}_{jc_{i'}} + \hat{f}_{jc_{i'}}(\mathbf{x}_{i'}), \\ \log(\hat{\sigma}_{i'j}) & = \hat{\beta}_{j0} + \hat{\beta}_{jb_{i'}} + \hat{\beta}_{jc_{i'}} + \hat{g}_{jc_{i'}}(\mathbf{x}_{i'}), \\ \mathrm{logit}(\hat{p}_{i'j}) & = \hat{\gamma}_{j0} + \hat{\gamma}_{jb_{i'}} + \hat{\gamma}_{jc_{i'}} + \hat{h}_{jc_{i'}}(\mathbf{x}_{i'}). \end{cases}$$

Note that $\hat{\mu}_{i'j}$, $\hat{\sigma}_{i'j}$, and $\hat{p}_{i'j}$ may not be all required, depending on the form of F_j (Table 2.3).

Then the m-dimensional feature vector of synthetic cell i' is $(Y_{i'1}, \ldots, Y_{i'm})^{\mathsf{T}}$, where

$$Y_{i'j} = \hat{F}_j^{-1}(U_{i'j} \mid \mathbf{x}_{i'}, \mathbf{z}_{i'}) \,, \; j = 1, \ldots, m \,.$$

Thanks to the parametric form of $\hat{F}_j(\cdot \mid \mathbf{x}_{i'}, \mathbf{z}_{i'})$, users can generate the synthetic data in their demand by modifying the parameters. For instance, if users want the expected sequencing depth of $\mathbf{Y}'$ to change from N (the sequencing depth of $\mathbf{Y}$) to N', they can scale the mean parameter:

$$Y_{i'j} \mid \mathbf{x}_{i'}, \mathbf{z}_{i'} \sim F_j \left(\cdot \;\middle|\; \mathbf{x}_{i'}, \mathbf{z}_{i'} \;;\; \frac{N'}{N} \hat{\mu}_{i'j}, \, \hat{\sigma}_{i'j}, \, \hat{p}_{i'j} \right) .$$

If users want to remove the batch effects, they can set

$$\hat{\alpha}_{jb_{i'}} = \hat{\beta}_{jb_{i'}} = \hat{\gamma}_{jb_{i'}} = 0 \,,$$

for all $i' = 1, \ldots, n'$; $j = 1, \ldots, m$.

If users want to remove the condition effects, they can set

$$\begin{aligned} \hat{\alpha}_{jc_{i'}} = \hat{\beta}_{jc_{i'}} = \hat{\gamma}_{jc_{i'}} &= 0 \,; \\ \hat{f}_{jc_{i'}}(\cdot) &= \hat{f}_{j1}(\cdot) \,; \\ \hat{g}_{jc_{i'}}(\cdot) &= \hat{g}_{j1}(\cdot) \,; \\ \hat{h}_{jc_{i'}}(\cdot) &= \hat{h}_{j1}(\cdot) \,, \end{aligned}$$

for all $i' = 1, \ldots, n'$; $j = 1, \ldots, m$.

2.2.6 The comparison of scDesign, scDesign2, and scDesign3

Table 2.1 lists a detailed comparison of scDesign3 with the previous two versions scDesign [40] and scDesign2 [27]. Note that scDesign2 is a special case of scDesign3 for generating scRNA-seq data from discrete cell types.

2.3 Results

2.3.1 scDesign3 functionality 1: simulation

We verified scDesign3 as a realistic and versatile simulator in four exemplar settings where existing simulators have gaps: (1) scRNA-seq data of continuous cell trajectories, (2) spatial transcriptomics data, (3) single-cell epigenomics data, and (4) single-cell multi-omics data (Fig. 2.1). Under each setting, we show that the synthetic data of scDesign3 resemble the test data (i.e., left-out real data unused for training), confirming that the scDesign3 model fits well but does not overfit the training data.

In the first setting about continuous cell trajectories, scDesign3 mimics three scRNA-seq datasets containing single or bifurcating cell trajectories (datasets EMBRYO, MARROW, and PANCREAS in Table 2.2). Fig. 2.1b–c and Figs. 2.3–2.5c–d show that scDesign3 generates realistic synthetic cells that resemble left-out real cells, as evidenced by high values ($\geqslant 1.75$) of mLISI (mean Local Inverse Simpson's Index), which indicates the degree of similarity between synthetic and real cells and has a lower bound of 1 and a perfect value of 2 [41]. Moreover, scDesign3 preserves eight gene- and cell-specific characteristics, including gene expression mean and variance, gene detection frequency, cell library size, cell-cell distance, cell detection frequency, cell-cell correlation, and, in particular, gene-gene correlation (Figs. 2.3–2.5a–b). Since no existing simulators can generate cells in continuous trajectories by learning from real data, we benchmarked scDesign3 against ZINB-WaVE, muscat, and SPARSIM—three top-performing simulators for generating discrete cell types in previous benchmark studies [25, 26]—and a deep-learning-based simulator scGAN [42]. The results show that scDesign3 outperforms these four simulators in generating more realistic synthetic cells (by achieving higher mLISI values) and in better preserving the eight gene- and cell-specific characteristics, especially cell-cell distances and gene-gene correlations (Fig. 2.1b–c and Figs. 2.3–2.5). In addition, scDesign3 can output the pseudotime truths of synthetic cells for benchmarking purposes, a functionality unavailable in existing simulators to our knowledge.

In the second setting about spatial transcriptomics, scDesign3 emulates four spatial tran-

scriptomics datasets generated by the 10x Visium and Slide-seq technologies (datasets VISIUM, SLIDE, OVARIAN, and ACINAR in Table 2.2). First, Fig. 2.1d–e and Fig. 2.6 show that scDesign3 recapitulates the expression patterns of spatially variable genes (by achieving high correlations between the corresponding synthetic and real spatial patterns). Second, Figs. 2.7–2.10a–b show that scDesign3 preserves the eight gene- and cell-specific characteristics. Third, Figs. 2.7–2.10c–d use PCA and UMAP embeddings to confirm that the synthetic data of scDesign3 resemble the test data (mLISI values $\geqslant 1.87$). Fourth, scDesign3 mimics spatial transcriptomics data so that each of three prediction algorithms (gradient boosting machine, random forest, and support vector machine) has highly consistent prediction errors (average Pearson correlation > 0.99) between the models trained on real data and scDesign3 synthetic data separately (Fig. 2.11); moreover, the scDesign3 model can fit complex spatial patterns in less-structured tissues such as cancer tissues (Fig. 2.12). Notably, in these examples, scDesign3 generates spatial transcriptomics data from spatial locations without cell type annotations (i.e., scDesign3-spatial; see Section 2.2.2). Figs. 2.7–2.10 show that these synthetic data of scDesign3 are similarly realistic compared to the synthetic data scDesign3 generates under an ideal scenario where annotated cell types are available (i.e., scDesign3-ideal; see Section 2.2.2). These results confirm scDesign3's ability to recapitulate cell heterogeneity without needing cell type annotations. Moreover, by fitting a model for spatial transcriptomics data, scDesign3 can estimate a smooth function for every gene's expected expression levels at spatial locations, a functionality unachievable by existing scRNA-seq simulators.

In addition, when trained on a pair of scRNA-seq data and multi-cell-resolution spatial transcriptomics data (where each spot contains multiple cells), scDesign3 can generate realistic multi-cell-resolution spatial transcriptomics data with cell-type proportions specified at each spot (i.e., ground truths) (Fig. 2.1f; Fig. 2.13a). Using this functionality to benchmark cell-type deconvolution algorithms for spatial transcriptomics data, we found that CARD [43] and RCTD [44] outperformed SPOTlight [45] in estimating the absolute cell-type proportions, though the three algorithms performed similarly well in estimating each cell type's relative proportions within a tissue slice (Fig. 2.13b).

In the third setting about single-cell epigenomics, scDesign3 resembles two single-cell chromatin accessibility datasets profiled by the 10x scATAC-seq and sci-ATAC-seq protocols (datasets ATAC and SCIATAC in Table 2.2). For both protocols, scDesign3 generates realistic synthetic cells (with each cell represented as a vector of genomic regions' read counts) despite the higher sparsity of single-cell ATAC-seq data compared to scRNA-seq data (Fig. 2.1g; Fig. 2.1h left; Figs. 2.14-2.15). Moreover, coupled with our newly proposed read simulator scReadSim [46], scDesign3 extends the simulation of synthetic cells from the count level to the read level, unblocking its application for benchmarking read-level bioinformatics tools (Fig. 2.1h right).

In the fourth setting about single-cell multi-omics, scDesign3 mimics a CITE-seq dataset (dataset CITE in Table 2.2) and simulates a multi-omics dataset from separately measured RNA expression and DNA methylation modalities (dataset SCGEM in Table 2.2). First, scDesign3 resembles the CITE-seq dataset by simultaneously simulating the expression levels of 1000 highly variable genes and 10 surface proteins. Fig. 2.1i shows that the RNA and protein expression levels of three exemplary surface proteins are highly consistent between the synthetic data of scDesign3 and the test data. Moreover, scDesign3 recapitulates the correlations between the RNA and protein expression levels of the 10 surface proteins (Fig. 2.16b). Second, scDesign3 simulates a single-cell multi-omics dataset with joint RNA expression and DNA methylation modalities by learning from (1) two single-omics datasets measuring the two modalities separately (Fig. 2.1j left) and (2) joint low-dimensional embeddings of the two single-omics datasets. This synthetic multi-omics dataset preserves the cell trajectory in the two single-omics datasets (Fig. 2.1j right). The functionality to generate multi-omics data from single-omics data allows scDesign3 to benchmark the computational methods that integrate modalities from unmatched cells [47].

2.3.2 scDesign3 functionality 2: interpretation

Providing the first universal probabilistic model for single-cell and spatial omics data, scDesign3 has broad applications beyond generating realistic synthetic data. We summarize the

prominent applications of the scDesign3 model in three aspects: model parameters, model selection, and model alteration (Fig. 2.2a).

First, the scDesign3 model has an interpretable parametric structure consisting of genes' marginal distributional parameters and pairwise gene correlations. In addition to being interpretable, the scDesign3 model is flexible to incorporate cell covariates (such as cell type, pseudotime, and spatial locations) via the use of generalized additive models (see Section 2.2.2), making the scDesign3 model fit well to various single-cell and spatial omics data—a property confirmed by scDesign3's realistic simulation in the aforementioned four settings (Fig. 2.1). The combined interpretability and flexibility enables scDesign3 to estimate the possibly non-linear relationship between every gene's mean expression and cell covariates, thus allowing statistical inference of gene expression changes between cell types, along cell trajectories (Fig. 2.2b), and across spatial locations (Fig. 2.2c).

Besides inferring every gene's expression characteristics, scDesign3 also estimates pairwise gene correlations conditional on cell covariates, thus providing insights into the possible gene regulatory relationships within each cell type, at a cell differentiation time, or in a spatial region. Specifically, scDesign3 estimates gene correlations by two statistical techniques, Gaussian copula and vine copula, which have complementary advantages (see Section 2.2.3): Gaussian copula is fast to fit but only outputs a gene correlation matrix; vine copula is slow to fit but outputs a hierarchical gene correlation network (a "vine" with the top layer indicating the most highly correlated genes, i.e., "hub genes") and thus more interpretable.

As an example application to a dataset containing four human peripheral blood mononuclear cell (PBMC) types (ZHENGMIX4 in Table 2.2), Fig 2.2d shows that Gaussian copula reveals similar gene correlation matrices for similar cell types (regulatory T cells vs. naive cytotoxic T cells) and distinct gene correlation matrices for distinct cell types (CD14+ monocytes vs. naive cytotoxic T cells). Moreover, vine copula discovers canonical cell-type marker genes as hub genes: *LYZ* for CD14+ monocytes and *CD79A* for B cells.

Second, scDesign3 outputs the model likelihood, enabling likelihood-based model selection criteria such as Akaike information criterion (AIC) and Bayesian information criterion

(BIC). This model selection functionality allows scDesign3 to evaluate the "goodness-of-fit" of a model to data and to compare competing models with the same types of cell covariates. A noteworthy application of this functionality is to evaluate how well does an inferred latent variable (e.g., cell cluster assignment, cell pseudotime, and cell spatial location) describe data, thus enabling us to evaluate inferred cell clusters, pseudotimes, and spatial locations from the goodness-of-fit perspective in the absence of ground truths or external knowledge. Although scDesign3 AIC and BIC rely on the scDesign3 model and do not represent the ground truth, we demonstrate that scDesign3 AIC and BIC are useful "unsupervised" criteria for assessing how well do inferred cell clusters, pseudotimes, and spatial locations agree with data under the scDesign3 model.

For cell clustering, we benchmarked scDesign3 BIC against the "supervised" adjusted Rand index (ARI), which requires true cell cluster labels, and a newly proposed unsupervised criterion, clustering deviation index (CDI) [48], on eight datasets with known cell types in a published benchmark study [49]. The results show that scDesign3 BIC has good agreement with ARI (mean Spearman correlation < -0.7) and has better or similar performance compared to CDI's performance on six out of the eight datasets (Fig. 2.17b).

For pseudotime inference, scDesign3 BIC is strongly correlated (mean Spearman correlation < -0.7) with the "supervised" R^2, which measures the consistency between the true and inferred (or perturbed) pseudotimes, on multiple synthetic datasets with true pseudotimes (Fig. 2.2e top; Fig. 2.17a). Further, scDesign3 BIC agrees with UMAP visualization: compared to TSCAN and Monocle3, the pseudotime inferred by Slingshot has the best (smallest) BIC and best agrees with the low-dimensional representation of the cell manifold (Fig. 2.2e bottom).

For spatial location inference, we benchmarked scDesign3 AIC against the mean cosine similarity (a supervised metric that measures the similarity between inferred and true spatial locations) using 2 sets of inferred spatial locations and 10 sets of perturbed spatial locations. We find scDesign3 AIC and the mean cosine similarity negatively correlated (mean Spearman correlation $\leqslant -0.7$) on two spatial transcriptomics datasets MOUSE-CORTEX and MOUSE-VISUAL (Table 2.2), suggesting that scDesign3 AIC is an effective assessment criterion of

spatial locations' goodness-of-fit (Fig. 2.17c). Note that scDesign3 AIC outperforms BIC in this case, possibly due to the reason that genes' spatial patterns are complex and thus need complex models.

Third, scDesign3 has a model alteration functionality enabled by its transparent probabilistic modeling and interpretable parameters: given the scDesign3 model parameters estimated on real data, users can alter the model parameters to reflect a hypothesis (i.e., a hypothetical truth) and generate the corresponding synthetic data that bear real data characteristics. Hence, users can flexibly generate synthetic data with varying ground truths for comprehensive benchmarking of computational methods.

We argue that this functionality is a vital advantage scDesign3 has over deep-learning based simulators [42], which cannot be easily altered to reflect a specific hypothesis. We demonstrate how to use this model alteration functionality in three examples. In the first example, scDesign3 generates synthetic data with different cell-type-specific condition effects (Fig. 2.2f). In the real data (CONDITION in Table 2.2), gene *IFI6*'s expression is up-regulated after stimulation in both CD16+ monocytes and B cells (Fig. 2.2f top-left). With scDesign3's fitted model, users can alter *IFI6*'s mean parameters to make *IFI6*'s expression up-regulated by stimulation in both cell types (Fig. 2.2f top-right), unchanged by stimulation in both cell types (Fig. 2.2f bottom-left), or up-regulated by stimulation in CD16+ monocytes only (Fig. 2.2f bottom-right). In the second example, scDesign3 generates synthetic datasets with or without batch effects (Fig. 2.2g). Trained on a real dataset (BATCH in Table 2.2) containing two batches with batch effects (Fig. 2.2g left), scDesign3's model, if without alteration, can generate synthetic data retaining the batch effects (Fig. 2.2g middle), or it can have the batch parameter altered to generate synthetic data without batch effects (Fig. 2.2g right). In the third example, scDesign3 generates synthetic data under two hypotheses: the null hypothesis (H_0) that only one cell type exists and the alternative hypothesis (H_1) that two cell types exist (Fig. 2.2h). Given a real dataset (ZHENGMIX4 in Table 2.2) containing two cell types (Fig. 2.2h left), the scDesign3 model can be fitted in two ways: under H_1, the model is fitted using the cell type information (Fig. 2.2h middle); under H_0, the model is fitted by assuming all cells are of one type (Fig. 2.2h right). The two fitted

models can generate the corresponding synthetic data under H_1 and H_0. Particularly, the synthetic data under H_0 can serve as the negative control for benchmarking computational pipelines that use cell clustering to identify the possible existence of cell types.

2.4 Discussion

In summary, scDesign3 accommodates various cell statuses, diverse omics modalities, and complex experimental designs. Although the scDesign3 model should not be treated as the true model, its interpretable parameters precede functionalities besides data simulation. First, scDesign3 model parameters offer a comprehensive interpretation of real data. Second, scDesign3 allows likelihood-based model selection to assess the goodness-of-fit of inferred cell clusters, trajectories, and spatial locations. Of course, this unsupervised model-based assessment cannot replace supervised metrics or compare models with different types of cell latent structures (e.g., cell clusters vs. trajectories). Third, scDesign3 can generate synthetic data under specific hypotheses by having its model parameters altered.

2.5 Code and Data Availability

The scDesign3 package is available at https://github.com/SONGDONGYUAN1994/scDesign3. The comprehensive tutorials are available at https://songdongyuan1994.github.io/scDesign3/docs/index.html. In the tutorials, we described the input and output formats, model parameters, and exemplary datasets for each functionality of scDesign3. The source code for reproducing the results is available in the Zenodo repository at https://doi.org/10.5281/zenodo.7110761. All datasets used in the study are publicly available. Table S2 lists the datasets from 17 published studies (sources included). The pre-processed datasets are available in the Zenodo repository at https://doi.org/10.5281/zenodo.7110761.

2.6 Acknowledgments

This chapter is based on my joint work with other collaborators, especially Qingyang Wang, and my Ph.D. advisor Dr. Jingyi Jessica Li.

2.7 Figures

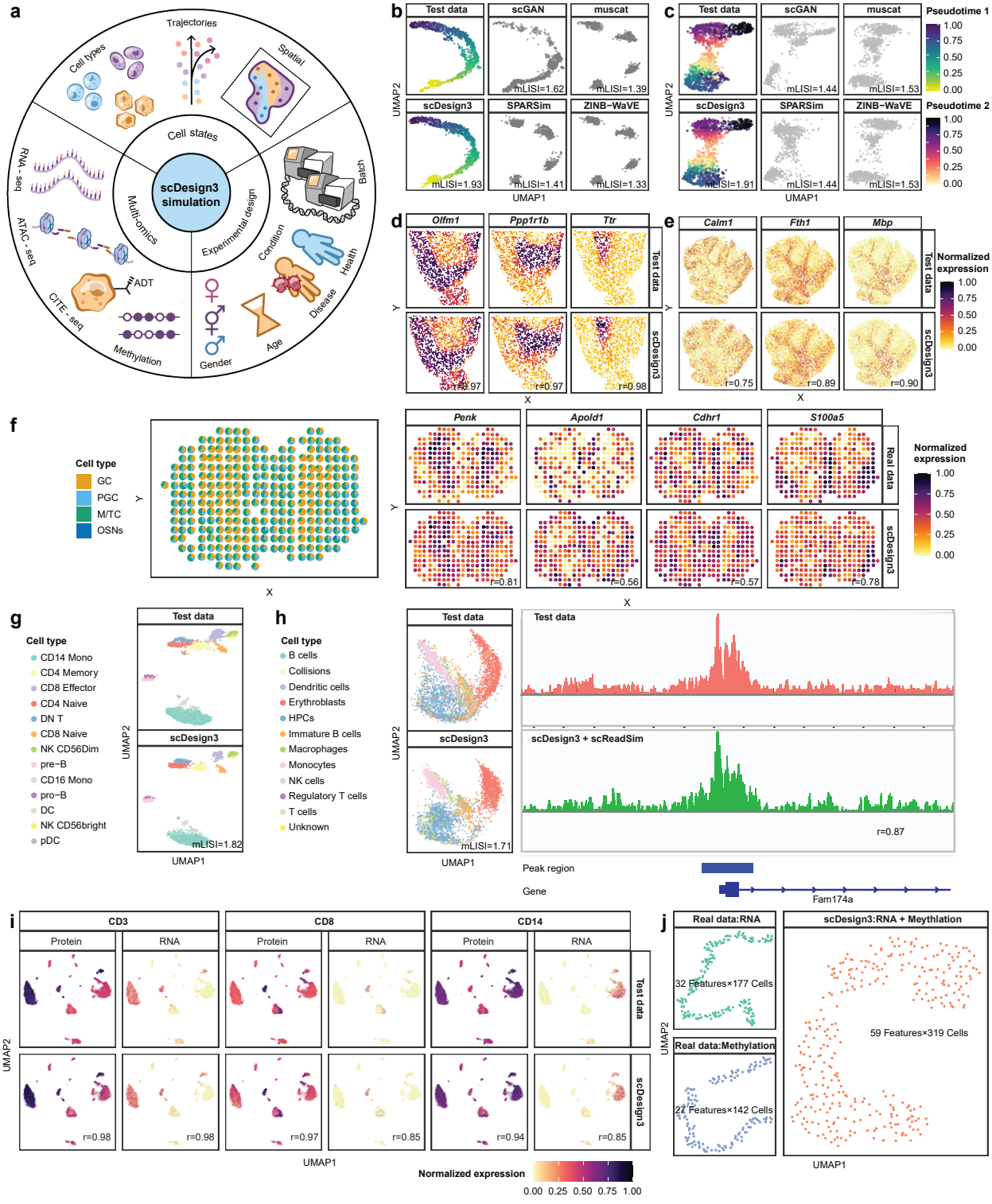


Figure 2.1: scDesign3 generates realistic synthetic data of diverse single-cell and spatial omics technologies.

a, An overview of scDesign3's simulation functionalities: cell states (e.g., discrete types, continuous trajectories, and spatial locations); multi-omics modalities (e.g., RNA-seq, ATAC-seq, and CITE-seq); experimental designs (e.g., batches and conditions). **b**–**c**, scDesign3 outperformed existing simulators scGAN, muscat, SPARSim, and ZINB-WaVE in simulating scRNA-seq datasets with a single trajectory (b) and bifurcating trajectories (c). Larger mLISI values represent better resemblance between synthetic data and test data. **d**–**e**, scDesign3 simulated realistic gene expression patterns in spatial transcriptomics datasets measured by 10x Visium (d) and Slide-seq (e). Large Pearson correlation coefficients (r) represent similar spatial patterns in synthetic and test data. **f**, using paired scRNA-seq data and spatial transcriptomics data (MOB-SC and MOB-SP in Table 2.2) as input, we defined the "ground-truth" cell-type proportions at each spot (left). Each color represents a cell type. With the cell-type proportions, scDesign3 generated synthetic spatial transcriptomics data in which every spot is a mixture of synthetic single cells, given the spot's cell-type proportions. The four cell-type marker genes exhibit similar spatial expression patterns in real data (right top) and synthetic data (right bottom). Large r values represent similar expression patterns in synthetic and test data. **g**, scDesign3 simulated a realistic scATAC-seq dataset at the count level. **h**, scDesign3 simulated a realistic sci-ATAC-seq dataset at both the count level (left: UMAP visualizations of real and synthetic cells based on peak counts) and the read level when coupled with scReadSim [46] (right: pseudobulk read coverages). **i**, scDesign3 simulated realistic CITE-seq data. Four genes' protein and RNA abundances are shown on the cell UMAP embeddings in test data (top) and synthetic data (bottom). Large r values represent similar expression patterns in synthetic and test data. **j**, scDesign3 generated a multi-omics (RNA expression + DNA methylation) dataset (right) by learning from two real single-omics datasets with RNA expression or DNA methylation only (left). The synthetic data preserved the linear cell topology.

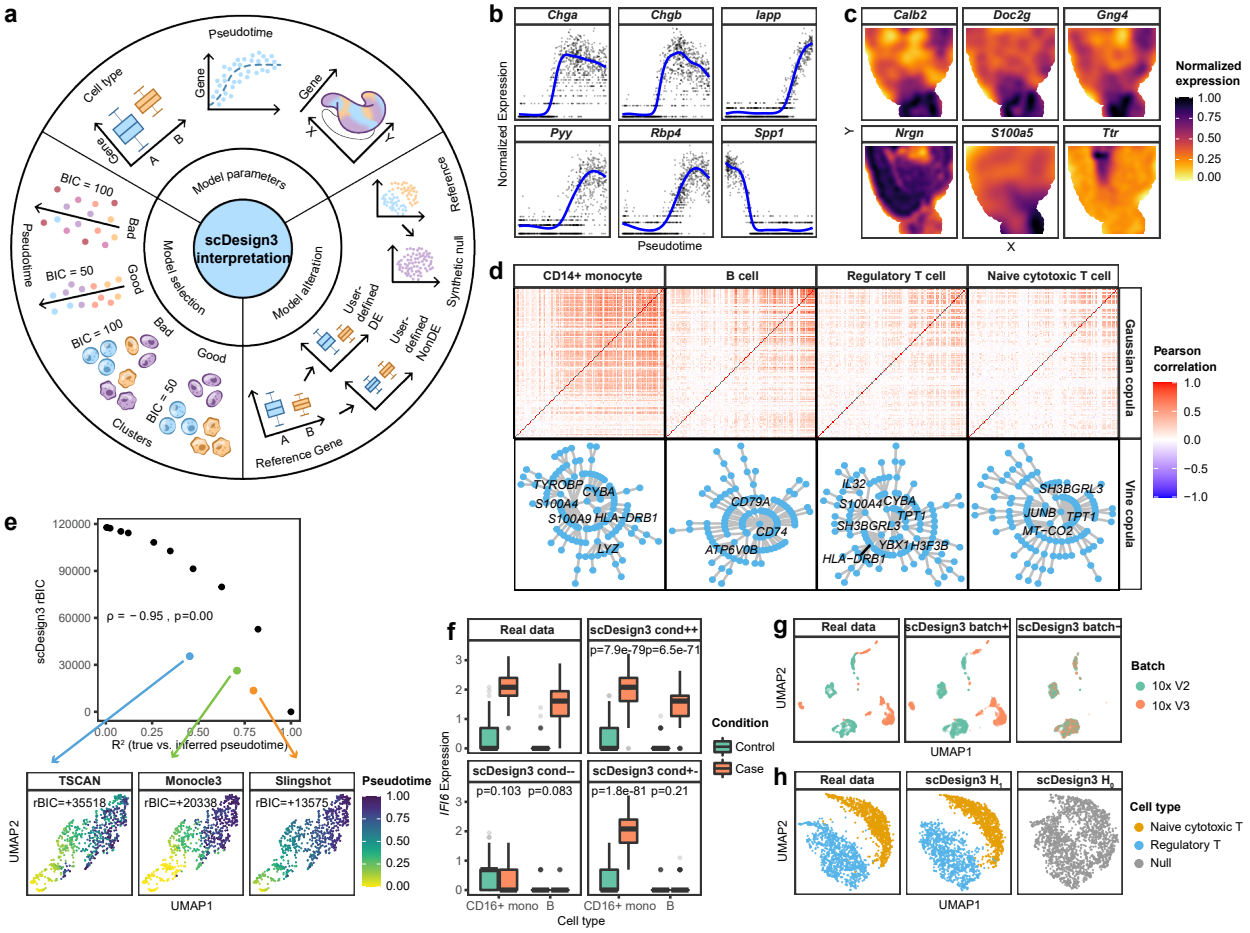


Figure 2.2: scDesign3 enables comprehensive interpretation of real data.

a, Summary of scDesign3's interpretation functionalities. **b**, scDesign3 estimated six genes' expression trends along cell pseudotime (PANCREAS in Table 2.2). **c**, scDesign3 estimated six genes' spatial expression trends (VISIUM in Table 2.2). **d**, scDesign3 estimated cell-type-specific gene correlations (ZHENGMIX4 in Table 2.2): correlation matrices by Gaussian copula (top); vine representations by vine copula (bottom), with genes in the first layer (roughly the genes strongly correlated) labeled. **e**, scDesign3's unsupervised assessment of goodness-of-fit. On synthetic scRNA-seq data with true pseudotimes (based on EMBYRO in Table 2.2), scDesign3 BIC and R^2 were evaluated on inferred pseudotimes of TSCAN (blue), Monocle3 (green), and Slingshot (orange), with perturbed true pseudotimes (black) as reference. Top: relative BIC (rBIC = BIC minus the smallest BIC) vs. R^2; the p-value (p) is from the one-sided test of Spearman's rank correlation ρ ($H_0 : \rho = 0$; $H_1 : \rho < 0$). Bottom: UMAP visualization of the three methods' inferred pseudotimes. **f**, In the CONDITION dataset (Table 2.2), gene *IFI6* was up-regulated in both CD16+ monocytes and B cells from control (green) to stimulation (red). scDesign3 simulated data where *IFI6* was up-regulated in both cell types (cond++), unchanged in both cell types (cond), or up-regulated in CD16+ monocytes only (cond+). The box center lines, bounds, and whiskers denote the medians, first and third quartiles, and minimum and maximum values within 1.5× the interquartile range of the box limits, respectively (the control and stimulation conditions have $n_{\text{control}} = 1772$ and $n_{\text{stimulation}} = 2188$ cells, respectively). The p-values (p) are from the two-sided Wilcoxon rank-sum test. **g**, The BATCH dataset (Table 2.2) contains two batches (left). scDesign3 preserved the batch effects in synthetic data generation (batch+) or generated synthetic data without batch effects (batch−). **h**, The ZHENGMIX4 dataset (Table 2.2) contains two cell types (left). scDesign3 resembled the real data under the alternative hypothesis (H_1 : two cell types existed) (middle) or generated synthetic data under the null hypothesis (H_0: one cell type existed) (right).

2.8 Supplementary materials

2.8.1 Supplementary figures

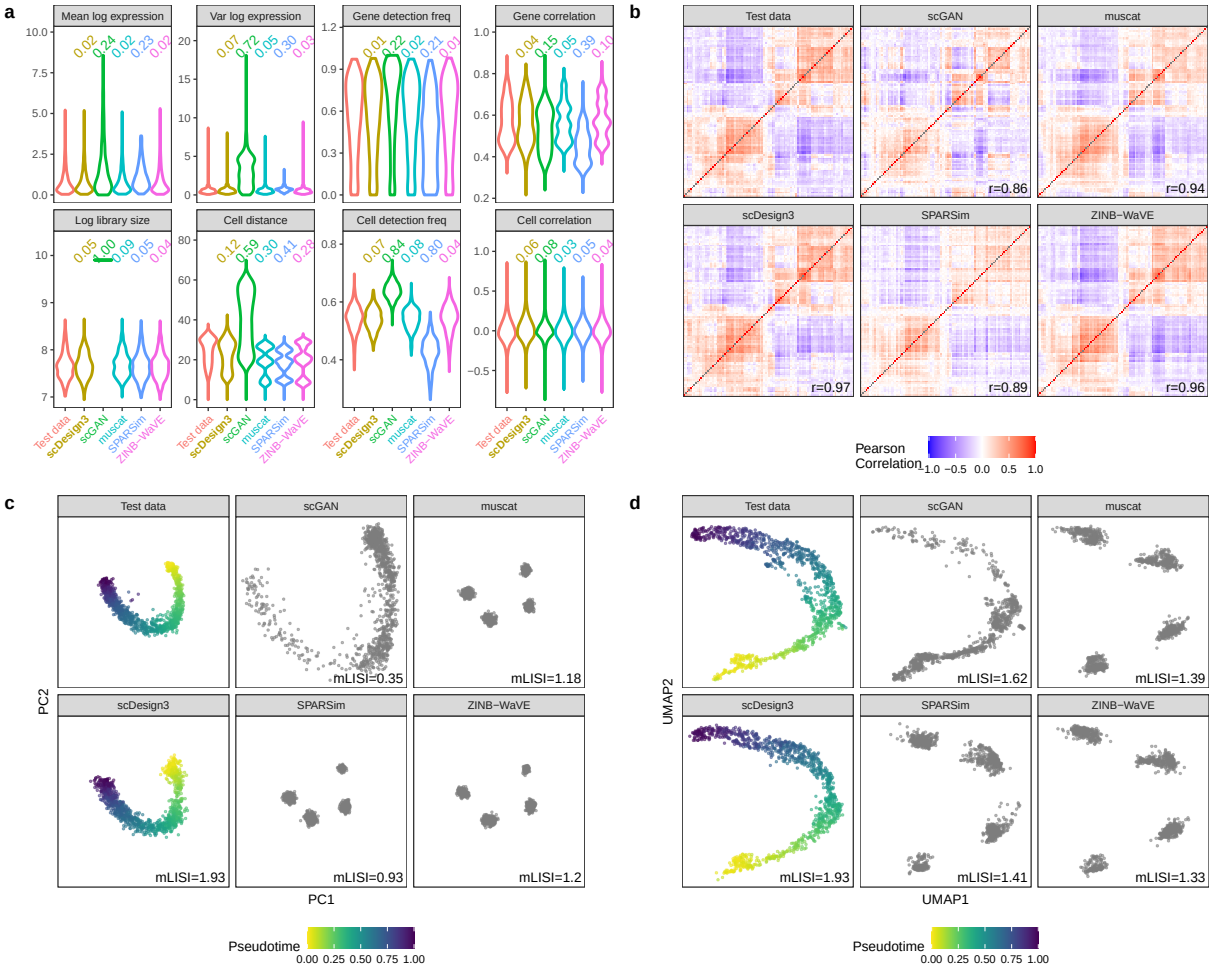


Figure 2.3: Benchmarking scDesign3 against four existing scRNA-seq simulators (scGAN, muscat, SPARSim, and ZINB-WaVE) for generating scRNA-seq data from a single trajectory (mouse pancreatic endocrinogenesis).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 and the four simulators. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3 and the four simulators. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3 and the four simulators. The color labels each cell's pseudotime value; note that only the synthetic data by scDesign3 outputs the pseudotime truths. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3 and the four simulators.

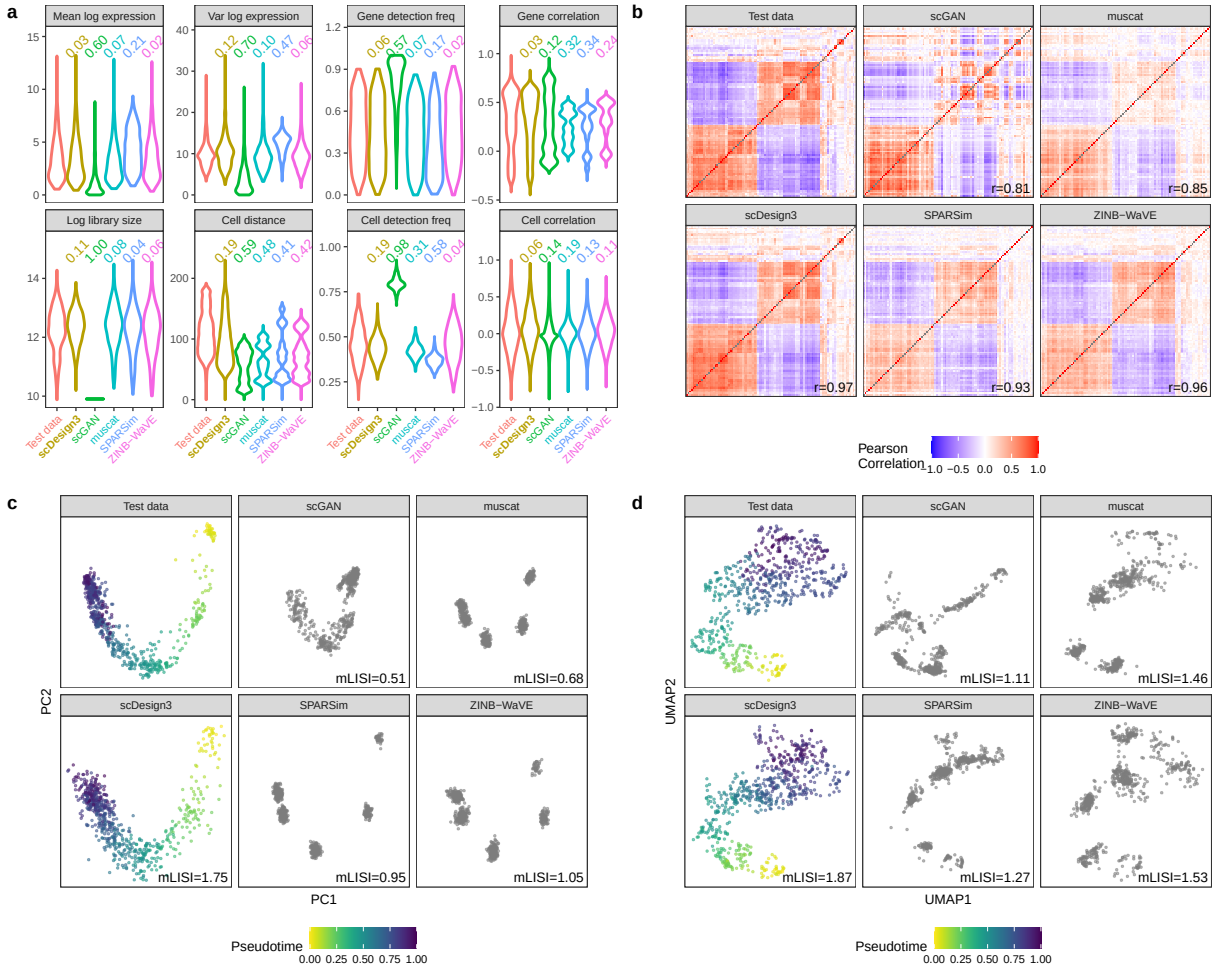


Figure 2.4: Benchmarking scDesign3 against four existing scRNA-seq simulators (scGAN, muscat, SPARSim, and ZINB-WaVE) for generating scRNA-seq data from a single trajectory (human preimplantation embryos).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 and the four simulators. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3 and the four simulators. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3 and the four simulators. The color labels each cell's pseudotime value; note that only the synthetic data by scDesign3 outputs the pseudotime truths. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3 and the four simulators.

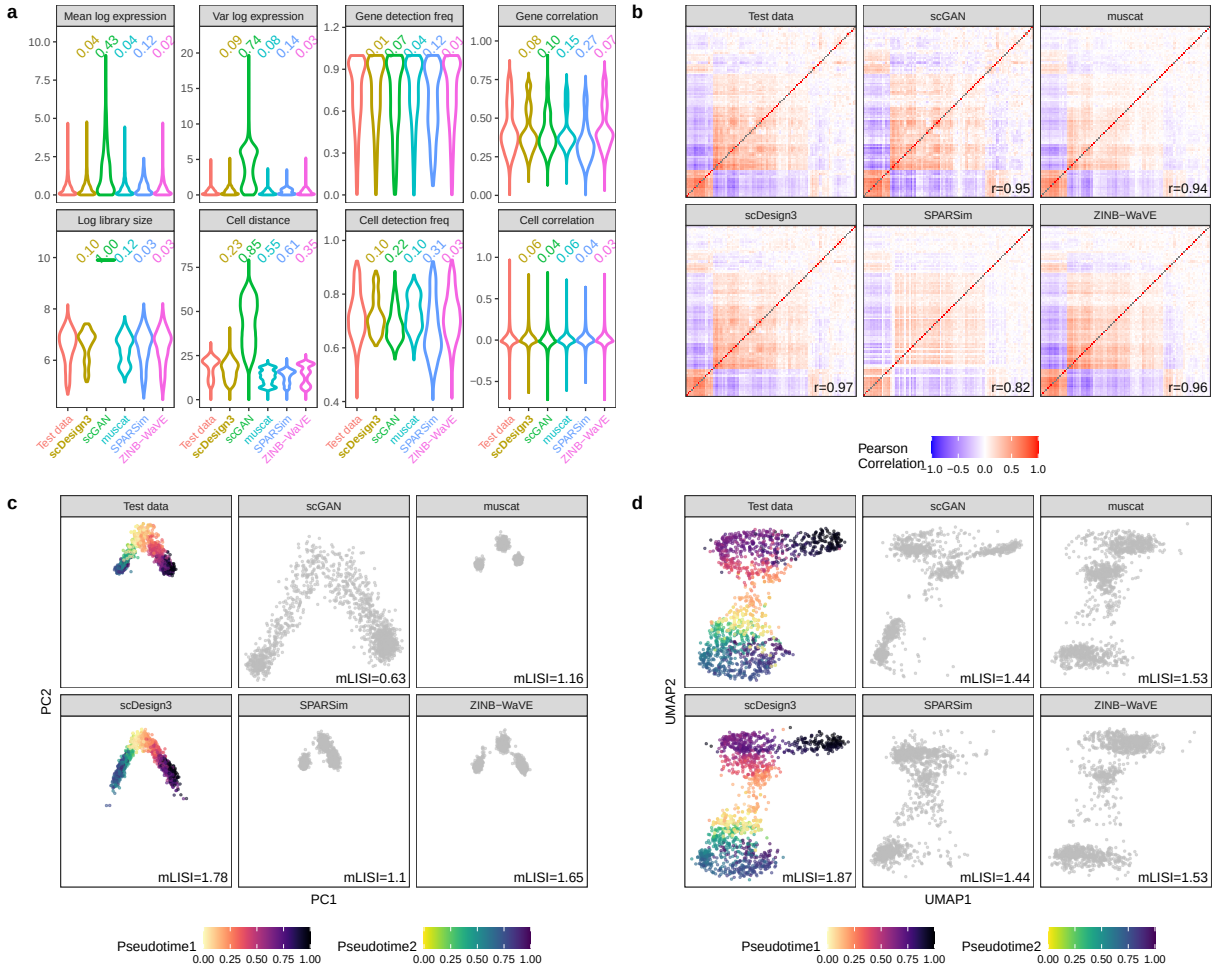


Figure 2.5: Benchmarking scDesign3 against four existing scRNA-seq simulators (scGAN, muscat, SPARSim, and ZINB-WaVE) for generating scRNA-seq data from bifurcating trajectories (myeloid progenitors in mouse bone marrow).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 and the four simulators. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3 and the four simulators. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3 and the four simulators. The color labels each cell's pseudotime value; note that only the synthetic data by scDesign3 outputs the pseudotime truths. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3 and the four simulators.

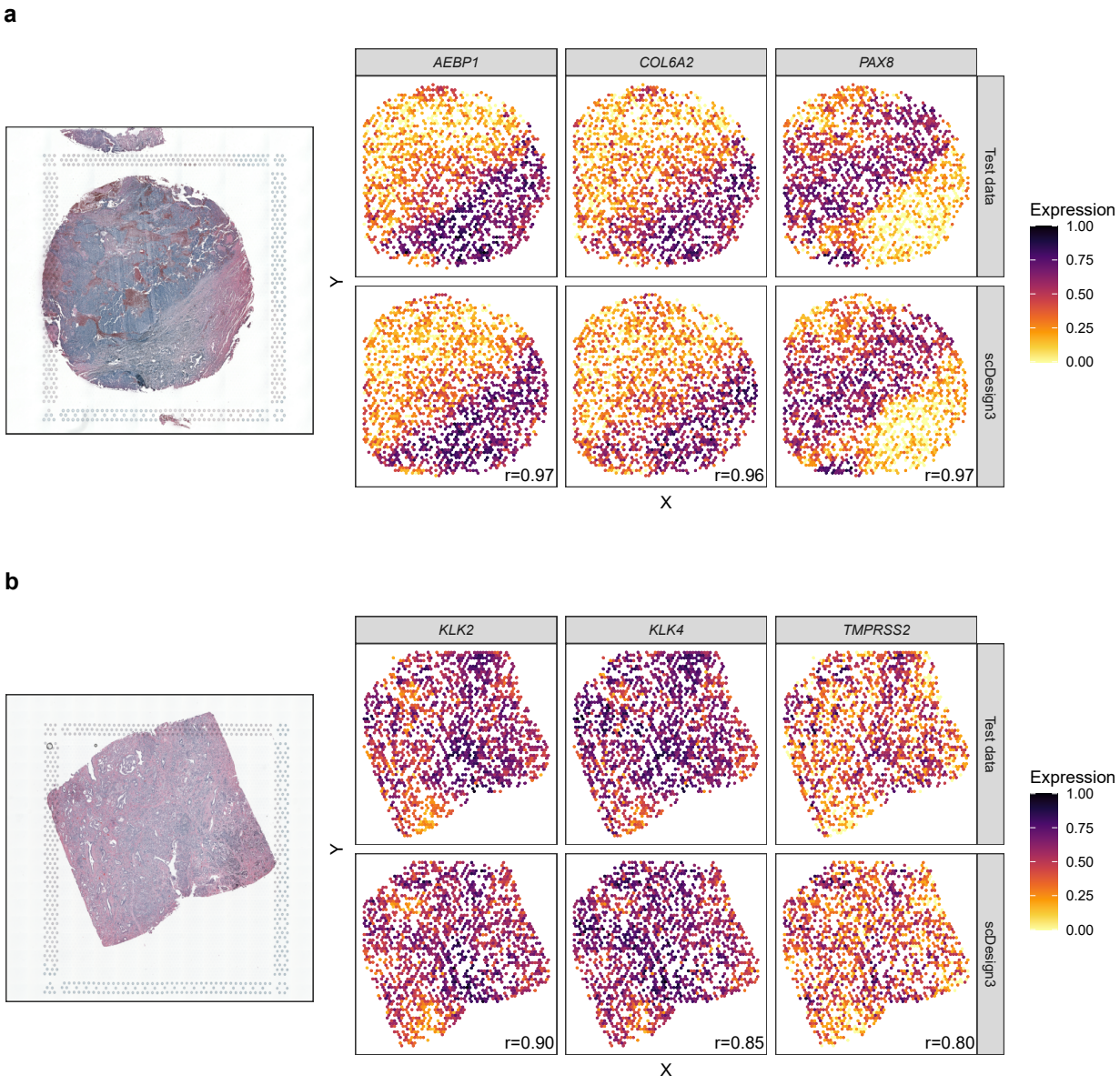


Figure 2.6: scDesign3 simulates realistic gene expression patterns for cancer transcriptomics datasets.

Human ovarian cancer (a) and human prostate cancer, acinar cell carcinoma (b). The tissue samples are measured with both H&E (hematoxylin and eosin stain, left) and spatial transcriptomics (right, three cancer-related genes). Large Pearson correlation coefficients (r) represent similar spatial patterns in synthetic and test data.

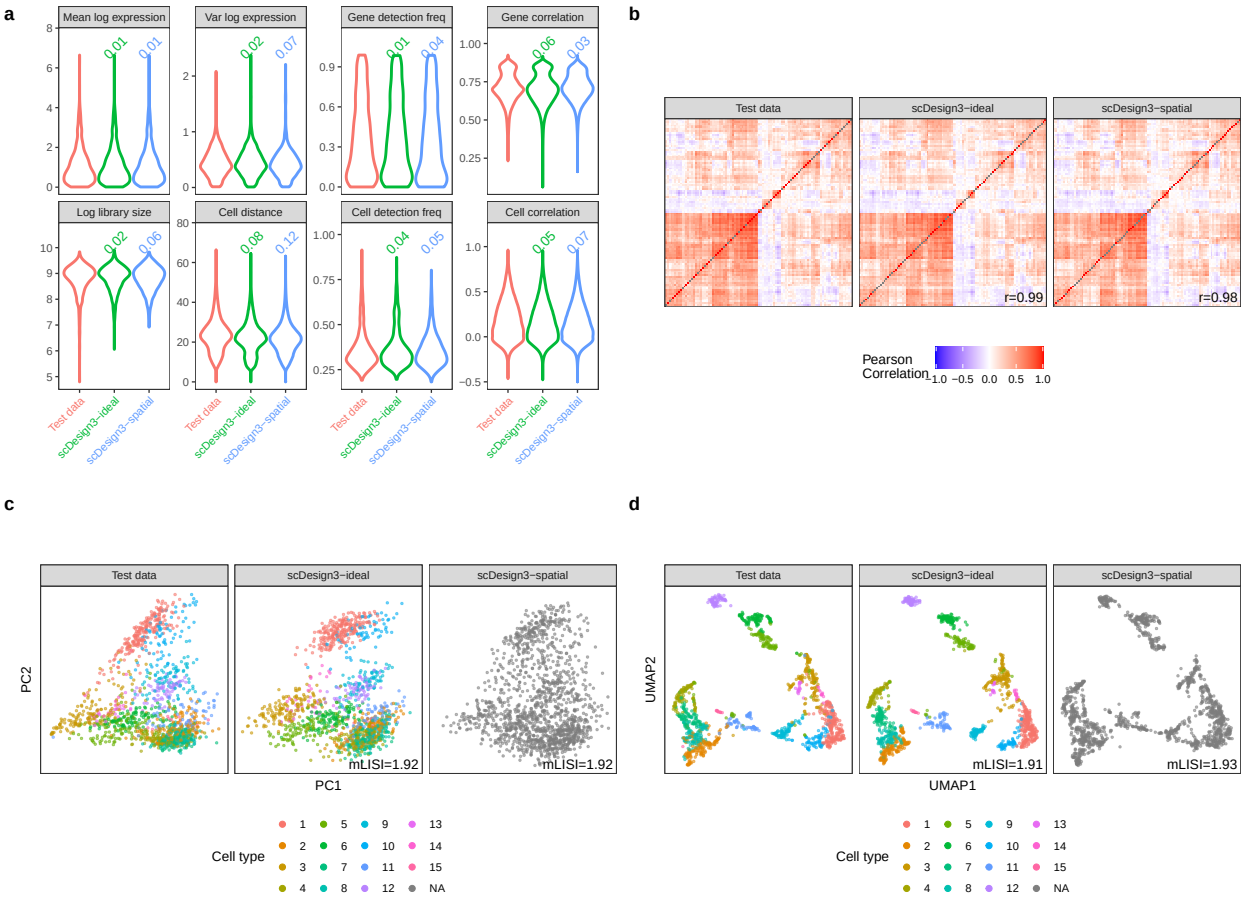


Figure 2.7: scDesign3 simulates 10x Visium spatial transcriptomics data (sagital mouse brain slices).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels (scDesign3-ideal) and spatial locations (scDesign3-spatial), respectively. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The color labels each cell's cell type (cluster). Since the scDesgin3-spatial data only uses spatial locations, it does not rely on cell types. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. In summary, scDesign3 realistically simulates 10x Visium data based on spatial locations without needing cell type annotations.

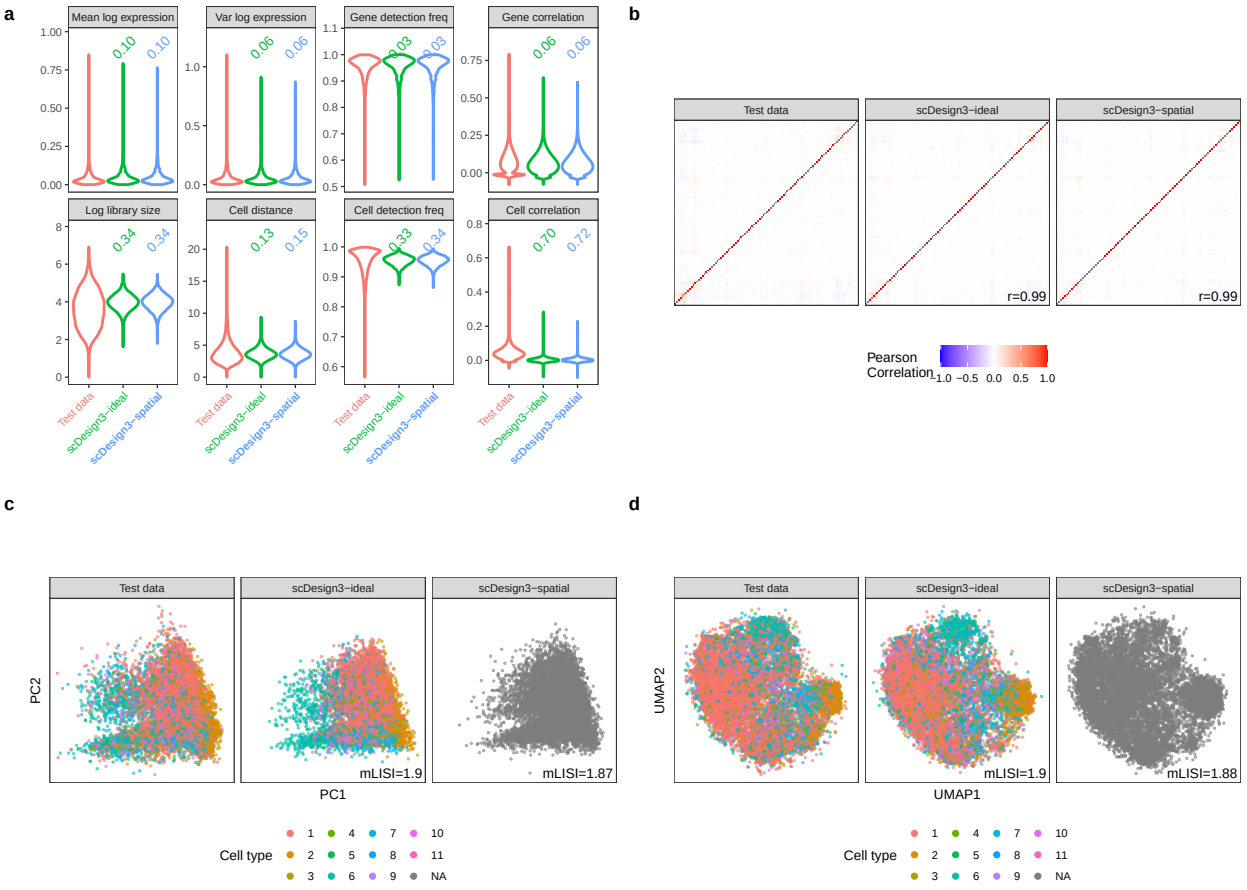


Figure 2.8: scDesign3 simulates Slide-seq spatial transcriptomics data (coronal cerebellum).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels (scDesign3-ideal) and spatial locations (scDesign3-spatial), respectively. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The color labels each cell's cell type (cluster). Since scDesgin3-spatial only uses spatial locations, it does not rely on cell types. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. In summary, scDesign3 realistically simulates Slide-seq data based on spatial locations without needing cell type annotations.

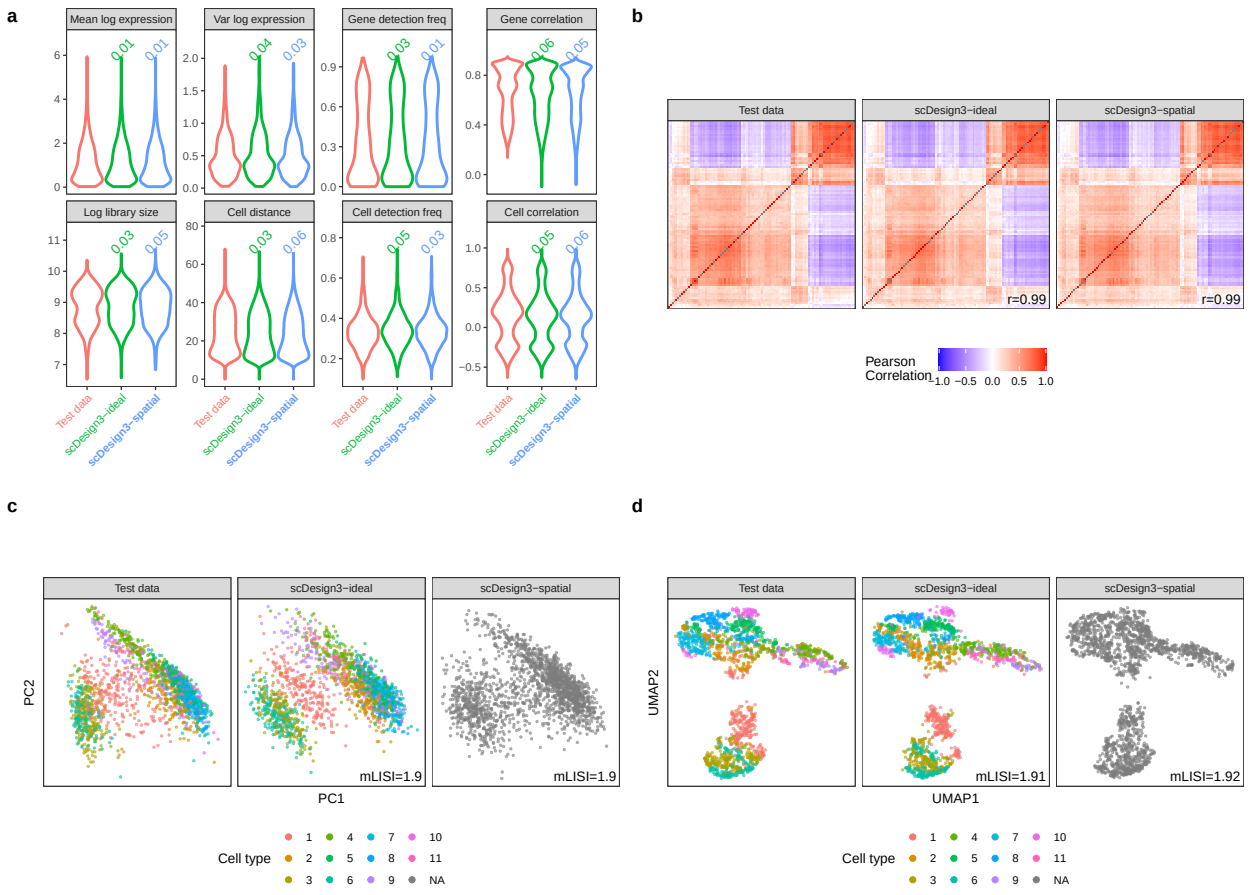


Figure 2.9: scDesign3 simulates 10x Visium cancer spatial transcriptomics data (human ovarian cancer).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels (scDesign3-ideal) and spatial locations (scDesign3-spatial), respectively. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The color labels each cell's cell type (cluster). Since the scDesgin3-spatial data only uses spatial locations, it does not rely on cell types. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. In summary, scDesign3 realistically simulates 10x Visium data based on spatial locations without needing cell type annotations.

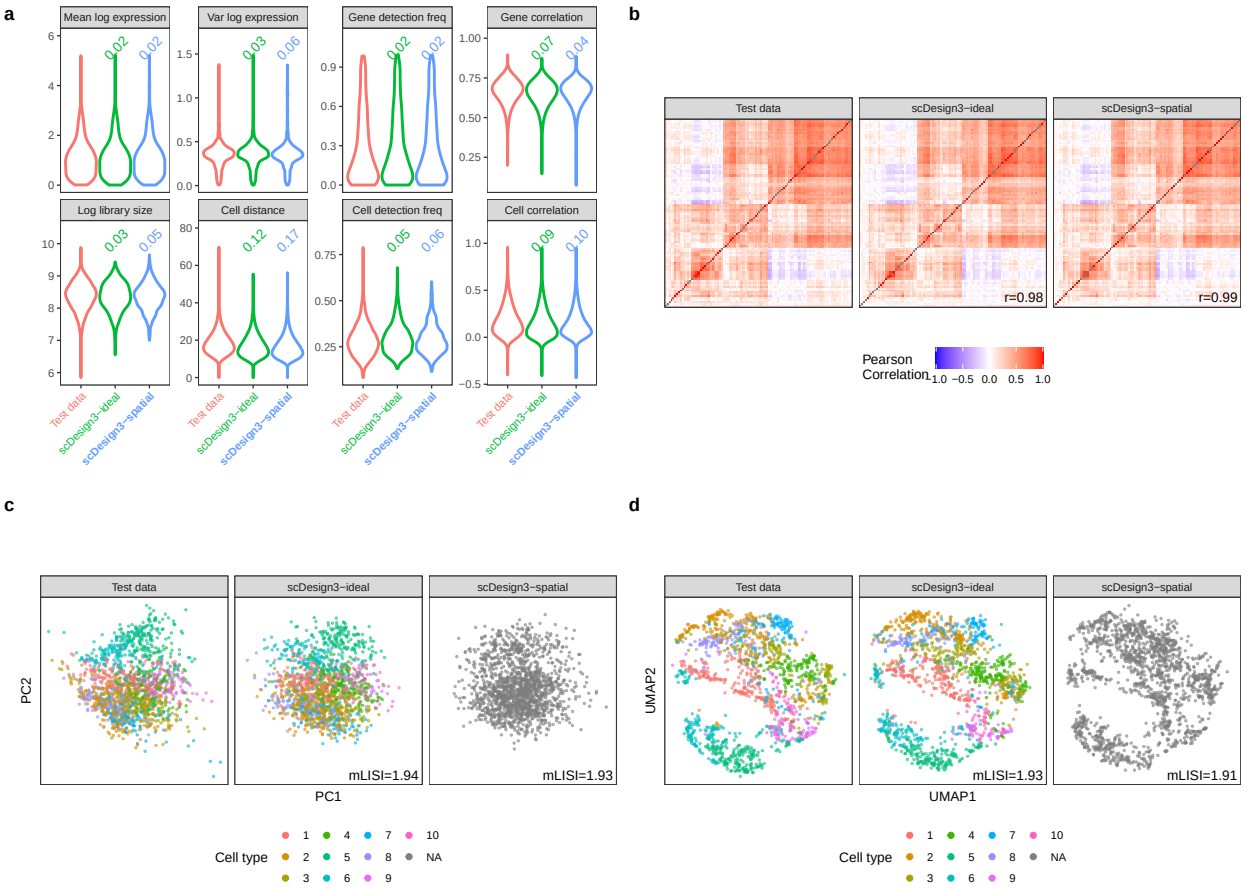


Figure 2.10: scDesign3 simulates 10x Visium cancer spatial transcriptomics data (human prostate cancer, acinar cell carcinoma).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels (scDesign3-ideal) and spatial locations (scDesign3-spatial), respectively. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene-gene correlation matrices (showing top 100 highly expressed genes) in the test data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. The color labels each cell's cell type (cluster). Since the scDesgin3-spatial data only uses spatial locations, it does not rely on cell types. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3-ideal and scDesign3-spatial. In summary, scDesign3 realistically simulates 10x Visium data based on spatial locations without needing cell type annotations.

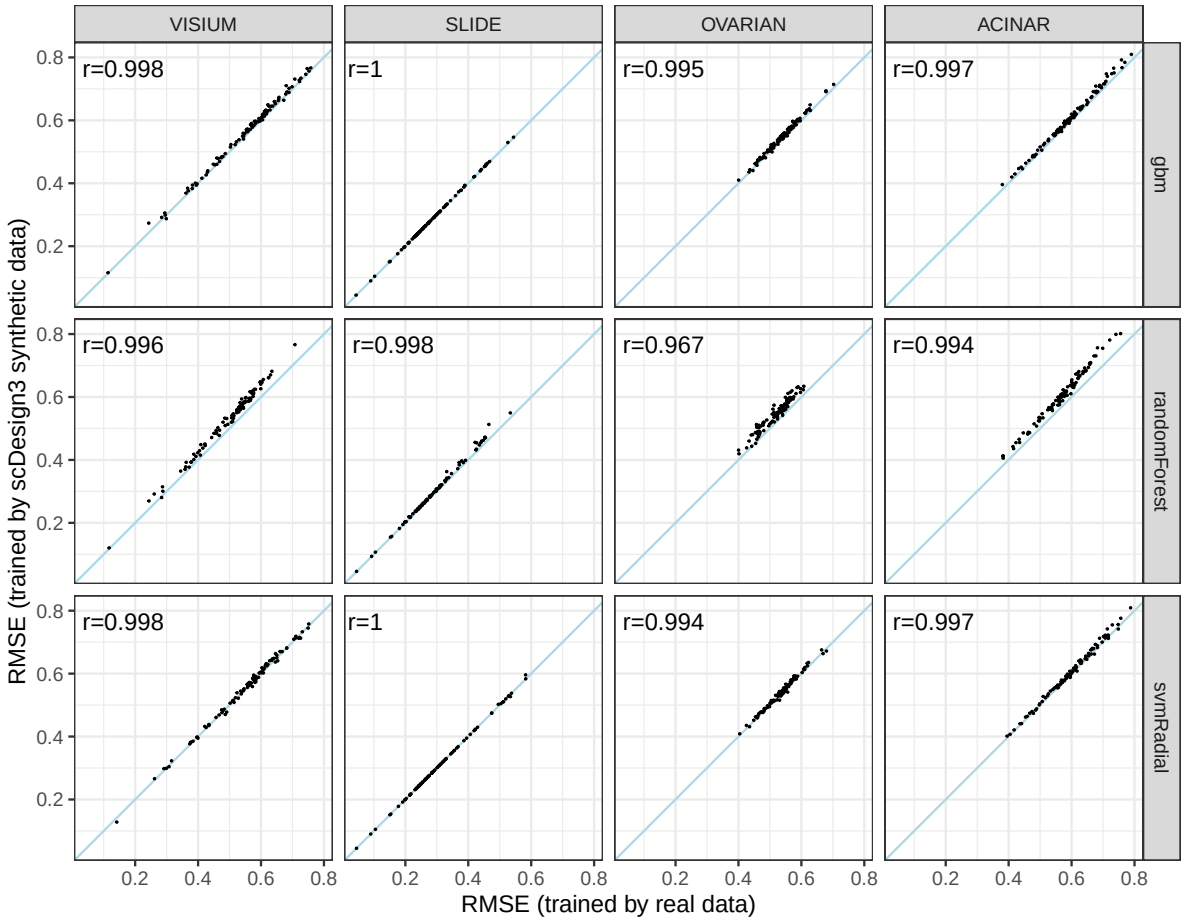


Figure 2.11: scDesign3 mimics spatial transcriptomics data so that prediction algorithms have similar prediction performance when trained on real data or scDesign3 synthetic data.

In detail, we first split each of four spatial transcriptomics datasets (VISIUM, SLIDE, OVARIAN, and ACINAR) into two datasets (training and testing) by randomly splitting the spatial locations into two halves. Second, we use each of the four training datasets to fit scDesign3 and generate the corresponding synthetic dataset. Third, on each pair of training dataset and synthetic dataset (among a total of four pairs), we train each of three prediction algorithms (gbm: gradient boosting machine; randomForest: random forest; svmRadial: support vector machine with the radial kernel) to predict each gene's expression at a spatial location (input: spatial location; output: the gene's log(count+1) expression level at the location), obtaining a pair of prediction models for each gene. Fourth, we apply each pair of prediction models to the corresponding testing dataset and calculate each model's root-mean-squared error (RMSE) for predicting each gene, obtaining a pair of RMSEs. As a result, in each panel, we plot the RMSEs for each prediction algorithm (row) and dataset (column), with each dot in the panel representing a gene. We observe that all genes' RMSEs are highly similar, reflecting that scDesign3 synthetic data well mimic real data.

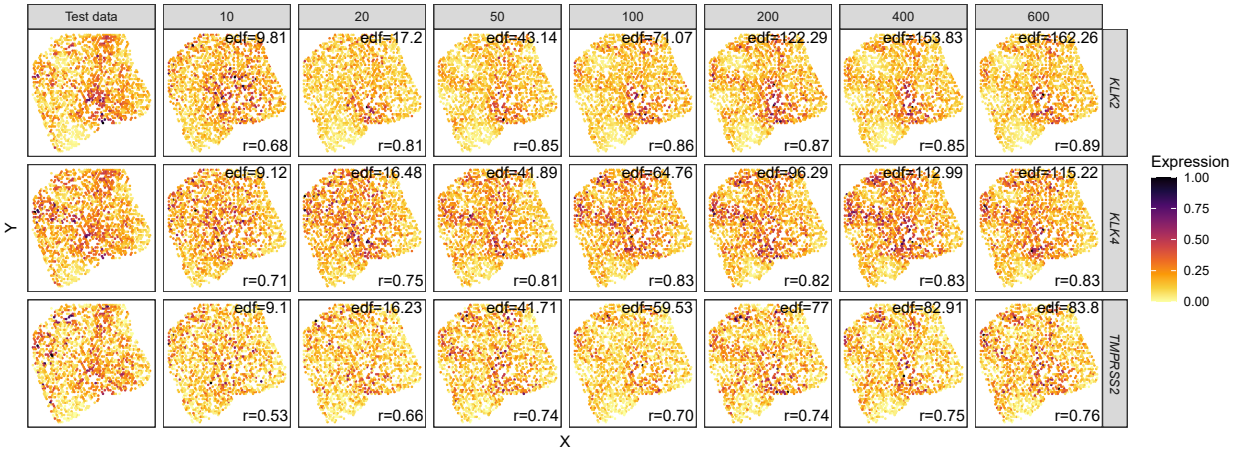


Figure 2.12: The effect of K on simulating spatial transcriptomics data.

The rows represent three cancer-related genes; column 1 represents real test data; columns 2- 8 represent scDesign3 synthetic data generated using varying input basis numbers K. A large Pearson correlation coefficient (r) represents similar spatial patterns in synthetic and test data. The effective degrees of freedom (edf) represent the wiggliness of the fitted surface. With a larger K, scDesign3 is able to fit more complex patterns. The overfitting issue is accounted for by the automatic smoothness estimation [37]: when K is sufficiently large, edf (model complexity) and r (model goodness-of-fit) both become stable.

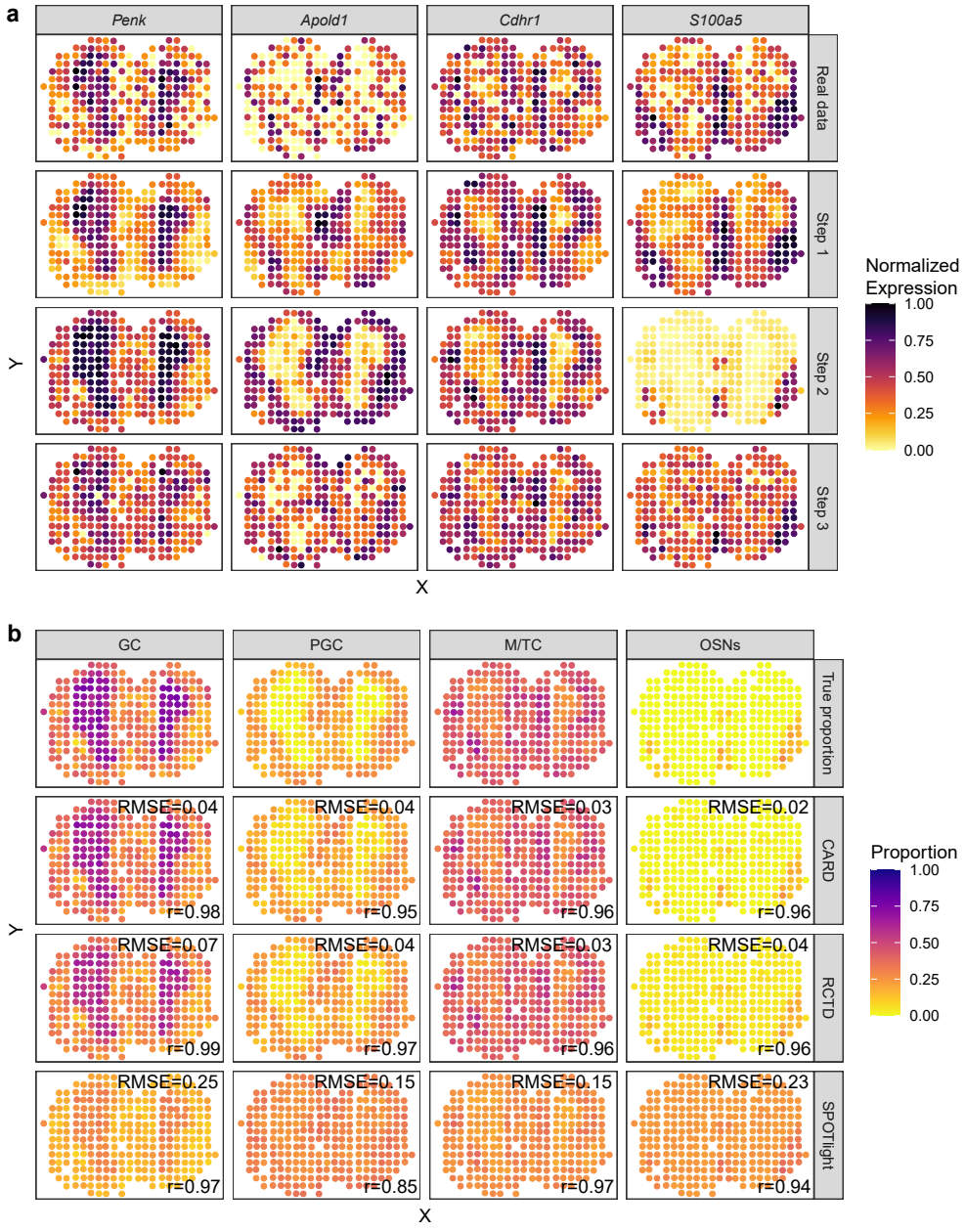


Figure 2.13: scDesign3 simulates spot-resolution spatial transcriptomics data for benchmarking cell-type deconvolution methods.

a, the scDesign3 spot simulation mimics the real data well by showing similar expression patterns for the four cell-type marker genes. **b**, Using scDesign3 synthetic data, we benchmark three spatial deconvolution methods (CARD [44], RCTD [43], and SPOTlight [45]). For each of four cell types (columns), we use two metrics—Pearson correlation (r) and root-mean-square error (RMSE)—to compare the estimated proportions by each deconvolution method (rows 2–4) to the true proportions (top row). Large r values represent similar spatial patterns of proportions, while small RMSE values represent similar values of proportions. Although all three methods well capture the spatial patterns of each cell type's proportions (evidenced by large r values), CARD and RCTD outperform SPOTlight by estimating cell-type proportions more accurately (evidenced by smaller RMSE values).

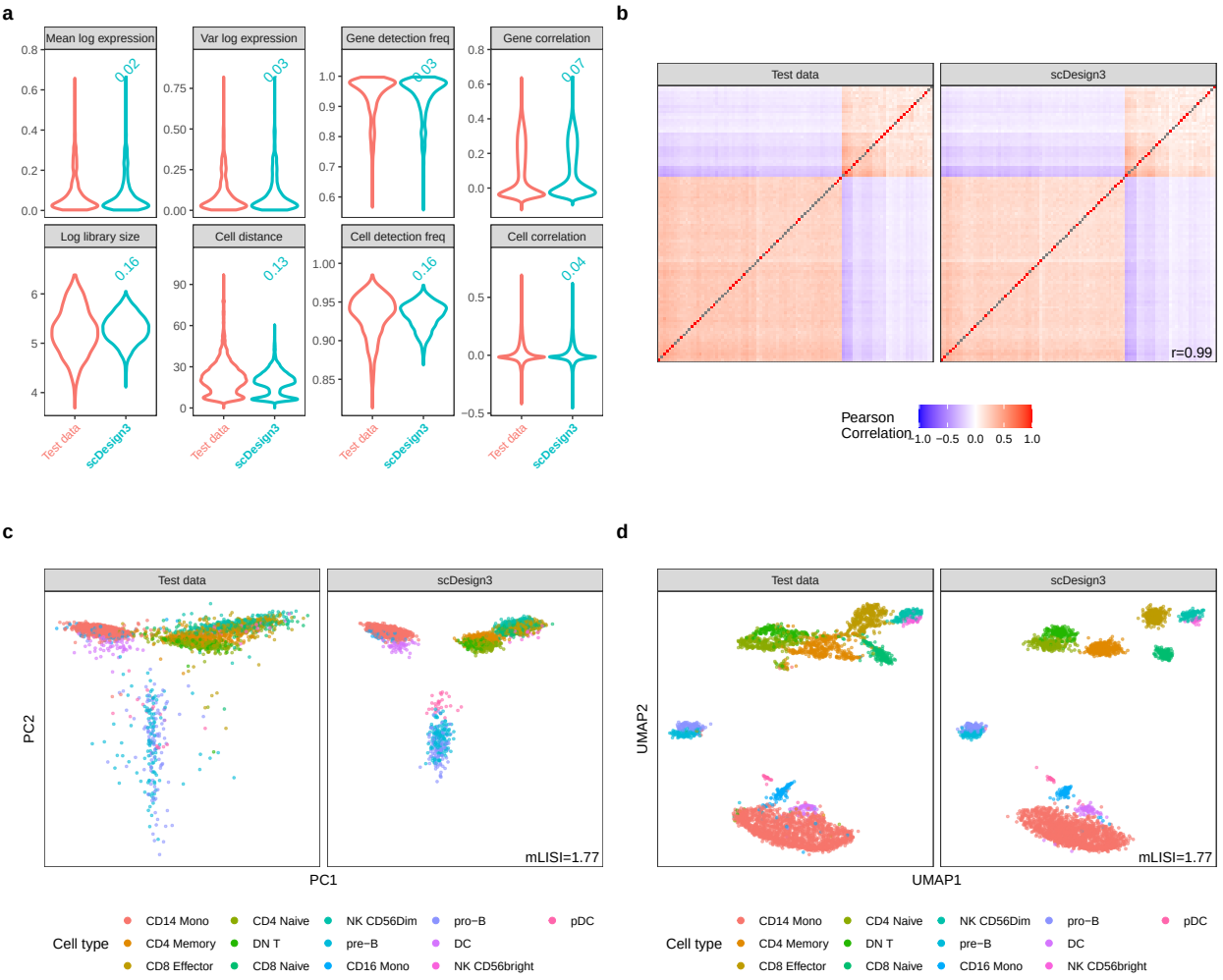


Figure 2.14: scDesign3 simulates scATAC-seq data (human PBMCs).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the peak-peak correlation matrices in the test data and the synthetic data generated by scDesign3. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3. The color labels each cell's cell type. An mLISI value close to 2 means that the synthetic data resemble the test data well in the low-dimensional space. **d**, UMAP visualization of the test data and the synthetic data generated by scDesign3.

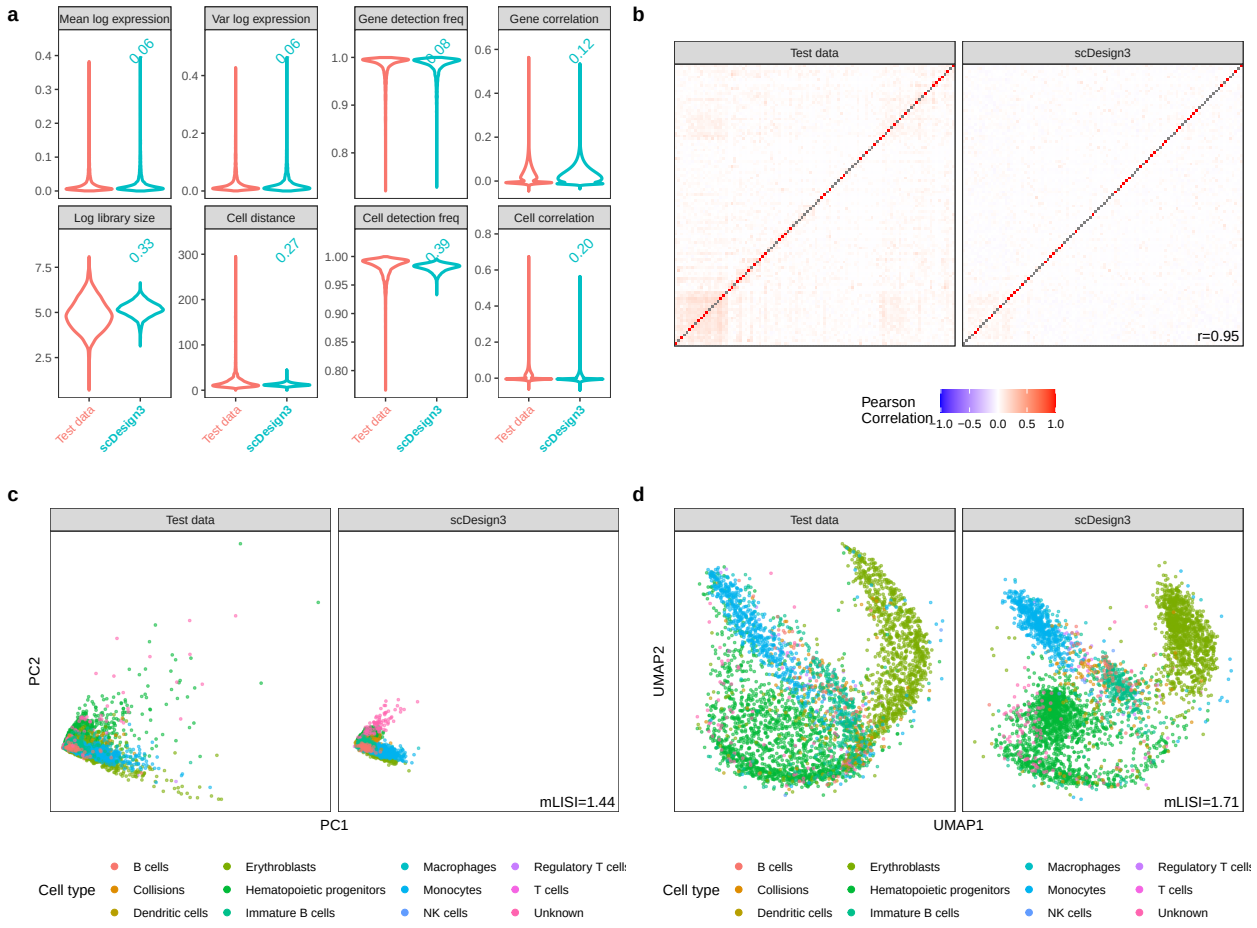


Figure 2.15: scDesign3 simulates sci-ATAC-seq data (mouse bone marrow).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3 using cell type labels. Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution.**b**, Heatmaps of the peak-peak correlation matrices in the test data and the synthetic data generated by scDesign3. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3. The color labels each cell's cell type. An mLISI value close to 2 means that the synthetic data resemble the test data well in the low-dimensional space. **d**, UMAP visualization of the test data and the synthetic data generated by scDesign3.

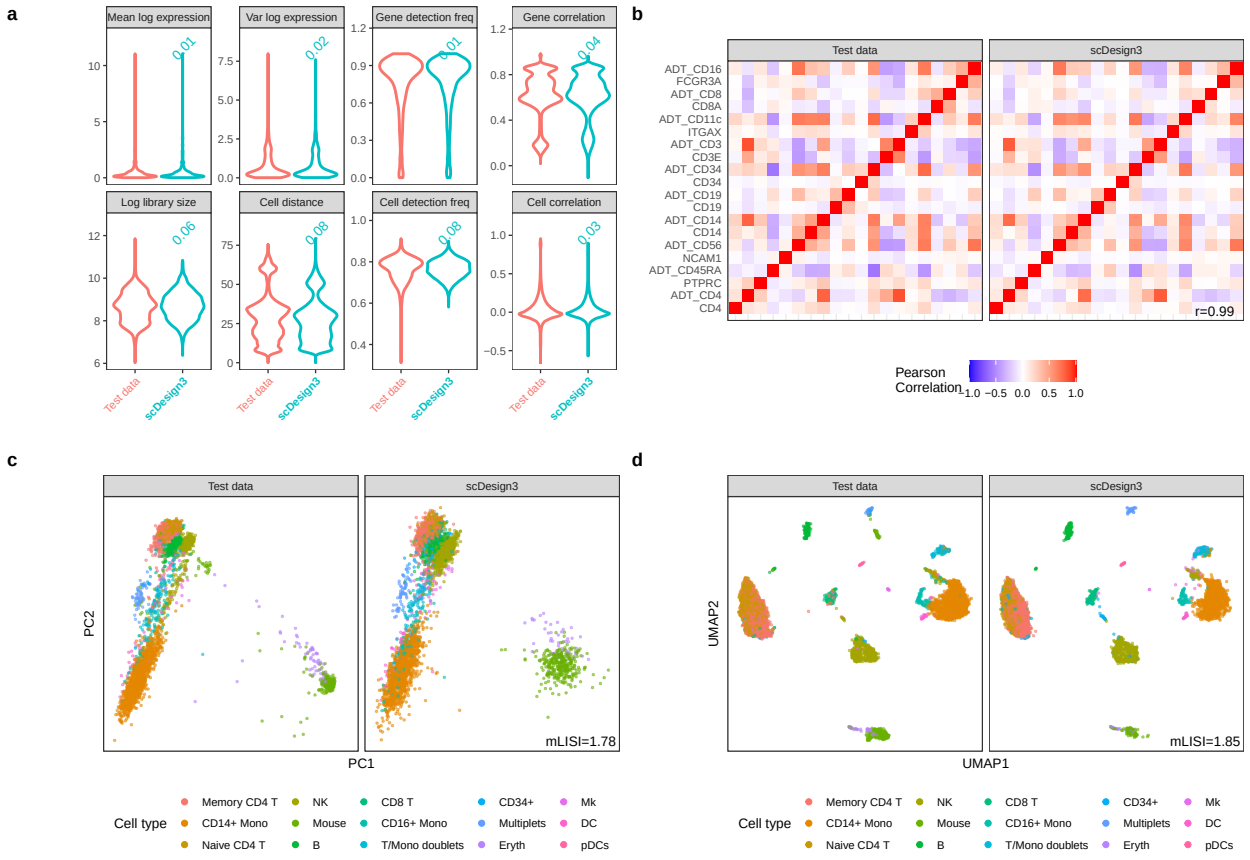


Figure 2.16: scDesign3 simulates CITE-seq data (human PBMCs).

a, Distributions of eight summary statistics in the test data and the synthetic data generated by scDesign3. The CITE-seq dataset simultaneously measures each cell's gene expression and surface protein abundance by Antibody-Derived Tags (ADTs). Each number on top of a violin plot (the distribution of a summary statistic in a synthetic dataset) is the Kolmogorov-Smirnov (KS) distance between the synthetic data distribution (indicated by that violin plot) and the test data distribution. A smaller number indicates better agreement between the synthetic data and the test data in terms of that summary statistic's distribution. **b**, Heatmaps of the gene and protein correlation matrices (10 proteins with names starting with "ADT" and their corresponding genes) from test data and the synthetic data generated by scDesign3. The Pearson's correlation coefficient r measures the similarity between two correlation matrices, one from the test data and the other from the synthetic data. scDesign3 recapitulates the correlations between the RNA and protein expression levels of the 10 surface proteins. **c**, PCA visualization (top two PCs) of the test data and the synthetic data generated by scDesign3. The color labels each cell's cell type. An mLISI value close to 2 means that the synthetic data resemble the real data well in the low-dimensional space. **d**, UMAP visualization of the real data and the synthetic data generated by scDesign3.

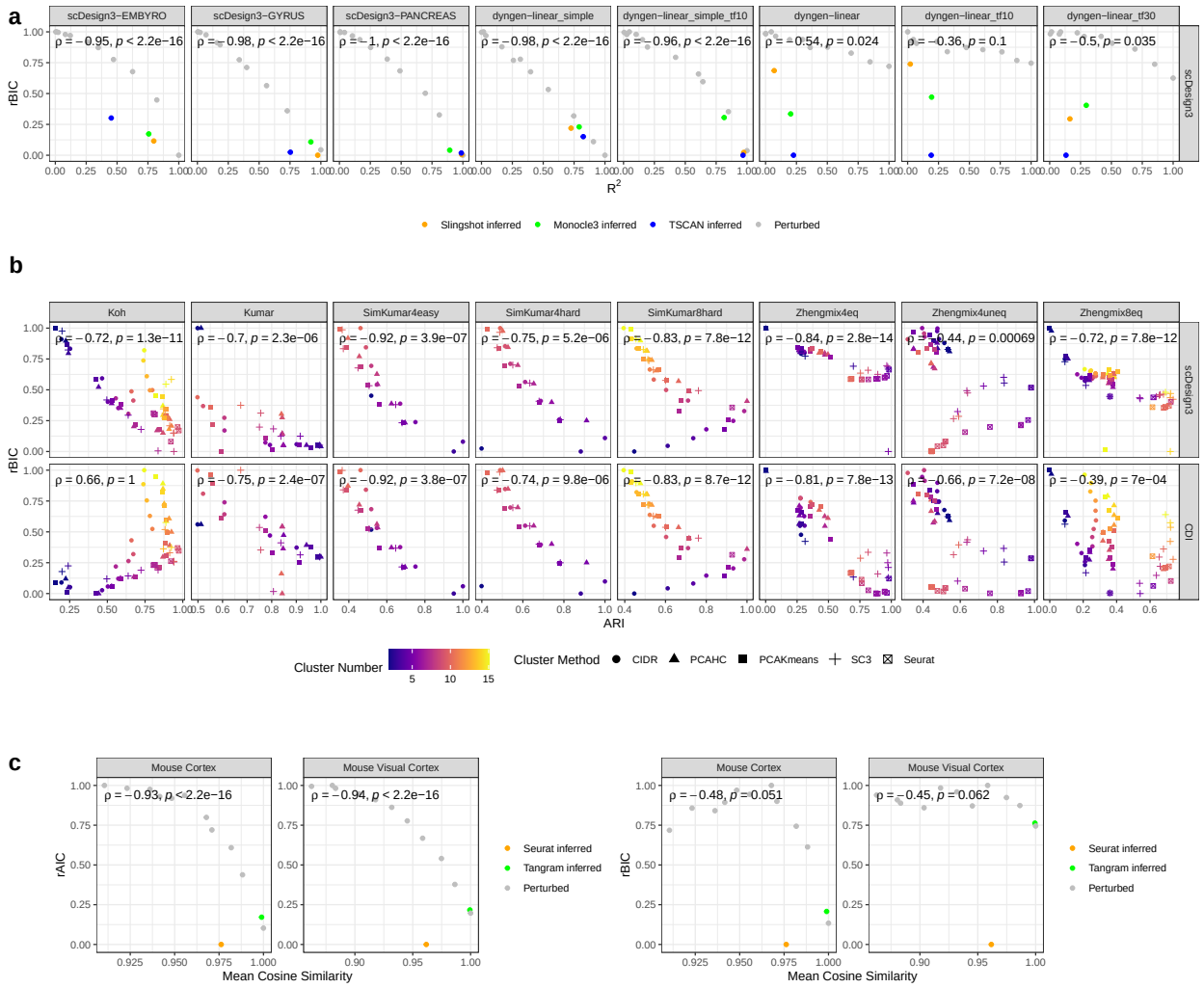


Figure 2.17: scDesign3 provides an unsupervised quantification of the goodness-of-fit of pseudotime, clusters, and inferred locations.

For visual clarity, we plot the relative BIC/AIC (rBIC/rAIC) by re-scaling scDesign3's marginal BIC/AIC to $[0, 1]$. **a**, The scDesign3 rBIC (unsupervised) is negatively correlated with the R^2 (supervised) between the perturbed pseudotime plus three inferred pseudotime and the true pseudotime in each of the eight datasets. The true pseudotime is the ground truth used for generating the synthetic data. **b**, Comparison of scDesign3 rBIC and Clustering Deviation Index (CDI) rBIC. The scDesign3 rBIC (unsupervised) negatively correlates with the ARI (supervised). The scDesign3 rBIC has better or similar performance than CDI's performance on six out of the eight datasets. The color scale shows the number of clusters, and the shapes represent clustering algorithms. **c**, The scDesign3 rAIC (unsupervised) is negatively correlated with the mean cosine similarity (supervised) between the perturbed locations plus two inferred locations, and the true locations in each of the two spatial datasets. The true locations are the ground truth used for generating the semi-synthetic data. Due to the high complexity of spatial patterns, the AIC outperforms BIC since it less penalizes the model complexity.

2.8.2 Supplementary tables

Table 2.1: Comparison of scDesign, scDesign2, and scDesign3

PropertyVersion	scDesign	scDesign2	**scDesign3**
Cell covariate	Cell types[1]	Cell types	Cell types **Cell trajectories** **Spatial locations**
Feature type	RNA expr.[2]	RNA expr.	RNA expr. **Chromatin accessibility** **Protein expr.** **DNA methylation**
Feature correlation	N/A	Gaussian copula	Gaussian copula **Vine copula**
Multi-modality	N/A	N/A	**Multi-omics**[3] **Multiple omics**[4]
Experimental design	N/A	N/A	**Multiple conditions** **Multiple batches**
Feature distribution	Gamma-Normal mixture	Poisson, ZIP NB, ZINB	Poisson, ZIP NB, ZINB **Bernoulli, Normal**
Feature mean function	Step function[5]	Step function	Step function, **1D smooth function**[6] **2D smooth surface**[7]
Model selection	N/A	N/A	**AIC, BIC**

Unique properties of scDesign3 are highlighted in **boldface**.

[1]: scDesign allows cell types to be connected by artificial paths.
[2]: The acronym "expr." stands for "expression."
[3]: "Multi-omics" means a cell is measured with multiple modalities.
[4]: "Multiple omics" means a cell is measured with only one modality and more than one modality is measured on different cells; in this case, scDesign3 requires all cells to be aligned to a common latent space by an integration method.
[5]: Step function is a function of cell type and outputs a constant for each cell type.
[6]: 1D smooth function is a function of cell pseudotime and is modeled by the spline.
[7]: 2D smooth surface is a function of 2D cell spatial location and is modeled by the Gaussian process.

Table 2.2: Real datasets used in scDesign3

Dataset name	Protocol	Cell-state covariates	Design covariates	Feature number (m)	Cell or spot number (n)	Description	Ref
ACINAR	10x Visium	spatial location	N/A	1000 genes	3043	human prostate cancer, acinar cell carcinoma	[50]
ATAC	10x scATAC-seq	cell type	N/A	1133 peaks	7034	human PBMCs	[51]
BATCH	10x scRNA-seq (V2/V3)	cell type	two batches	1000 genes	6276	human PBMCs	[52]
CITE	CITE-seq	cell type	N/A	1000 genes + 10 proteins	8617	human CBMCs	[14]
EMBYRO	scRNA-seq	cell pseudotime in one trajectory	N/A	1000 genes	1289	human preimplantation embryos	[53]
IFNB	10x scRNA-seq	cell type	two conditions	1000 genes	13999	IFNB-stimulated/control PBMCs	[54]
MARROW	MARS-seq	cell pseudotimes in two trajectories	N/A	1000 genes	2660	myeloid progenitors in mouse bone marrow	[55]
MOB-SC	10x Chromium	cell type	N/A/	182 genes	12640	mouse olfactory bulb	[56]
MOB-SP	spatial transcriptomics	spatial location	N/A	182 genes	278	mouse olfactory bulb	[57]
MOUSE-CORTEX	seqFISH+	spatial location	N/A	10000 genes	524	mouse cortex	[58]
MOUSE-VISUAL	STARmap	spatial location	N/A	1020 genes	1549	mouse visual cortex	[58]
OVARIAN	10x Visium	spatial location	N/A	1000 genes	3455	human ovarian cancer	[50]
PANCREAS	10x scRNA-seq	cell pseudotime in one trajectory	N/A	1000 genes	2087	mouse pancreatic endocrinogenesis	[59]
SCGEM-METH	scGEM	2D UMAP locations[1]	N/A	27 methylation loci	142	human foreskin fibroblast reprogramming to iPS	[60]
SCGEM-RNA	scGEM	2D UMAP locations	N/A	32 genes	177	same as SCGEM-METH	[60]
SCIATAC	sci-ATAC-seq	cell type	N/A	3836 peaks	4025	mouse bone marrow	[61]
SLIDE	Slide-seq	spatial location	N/A	1000 genes	23372	coronal cerebellum	[62]
VISIUM	10x Visium	spatial location	N/A	1000 genes	2096	a sagittal mouse brain slice	[63]
ZHENGMIX4	10x scRNA-seq	cell type	N/A	1556 genes	3555	human PBMCs	[8]

Table 2.3: Choices of feature j's marginal distribution F_j

Distribution	Parameters	Probability density function (PDF) or Probability mass function (PMF)	Link function	Applicable data type
Gaussian	μ: mean σ: standard deviation	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2};\ x \in \mathbb{R}$	$\theta_j(\mu) = \mu$	Normalized data (e.g., log-transformed count data)
Bernoulli	μ: mean	$f(x) = \mu^x (1-\mu)^{1-x};\ x \in \{0, 1\}$	$\theta_j(\mu) = \log \frac{\mu}{1-\mu}$	Binary data (e.g., DNA methylation)
Poisson	μ: mean	$f(x) = \frac{\mu^x e^{-\mu}}{x!};\ x \in \{0, 1, 2, \ldots\}$	$\theta_j(\mu) = \log \mu$	Count data without over-dispersion
Negative Binomial	μ: mean σ: dispersion	$f(x) = \frac{\Gamma(x+\frac{1}{\sigma})}{\Gamma(\frac{1}{\sigma})\Gamma(x+1)} \left(\frac{1}{1+\sigma\mu}\right)^{\frac{1}{\sigma}} \left(\frac{\sigma\mu}{1+\sigma\mu}\right)^x;\quad x \in \{0, 1, 2, \ldots\}$	$\theta_j(\mu) = \log \mu$	Count data with over-dispersion (e.g., UMI-based scRNA-seq; spatial transcriptomics; protein abundance)
Zero-inflated Poisson	μ: mean p: zero-inflation proportion	$f(x) = \begin{cases} p + (1-p)e^{-\mu};\ x = 0 \\ \frac{(1-p)\mu^x e^{-\mu}}{x!};\ x = 1, 2, 3, \ldots \end{cases}$	$\theta_j(\mu) = \log \mu$	Poisson count data with excess zeros (e.g., scATAC-seq)
Zero-inflated Negative Binomial	μ: mean σ: dispersion p: zero-inflation proportion	$f(x) = \begin{cases} p + (1-p)(1+\sigma\mu)^{-\frac{1}{\sigma}};\ x = 0 \\ \frac{(1-p)\Gamma(x+\frac{1}{\sigma})}{\Gamma(\frac{1}{\sigma})\Gamma(x+1)} \left(\frac{1}{1+\sigma\mu}\right)^{\frac{1}{\sigma}} \left(\frac{\sigma\mu}{1+\sigma\mu}\right)^x; \\ x = 1, 2, 3, \ldots \end{cases}$	$\theta_j(\mu) = \log \mu$	Negative binomial count data with excess zeros (e.g., full-length non-UMI-based scRNA-seq)

Table 2.4: Forms of the functions $f_{jc_i}(\cdot)$, $g_{jc_i}(\cdot)$, and $h_{jc_i}(\cdot)$ of cell-state covariates

Covariate type	Covariate form	Function form	Explanation	Geometric meaning
Discrete cell type	$x_i \in \{1, \ldots, K_C\}$	$f_{jc_i}(x_i) = \alpha_{jc_i x_i}$	Cell type x_i has the effect $\alpha_{jc_i x_i}$; for identifiability, $\alpha_{jc_i x_i} = 0$ if $x_i = 1$	One intercept for each cell type
Continuous pseudotime in one lineage	$x_i \in [0, \infty)$	$f_{jc_i}(x_i) = \sum_{k=1}^{K} b_{jc_i k}(x_i)\beta_{jc_i k}$	$b_{jc_i k}(\cdot)$ is a basis function of cubic spline; K is the dimension of the basis	A curve along the pseudotime
Continuous pseudotimes in p lineages	$\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T} \in [0, \infty)^p$	$f_{jc_i}(\mathbf{x}_i) = \sum_{l=1}^{p} \sum_{k=1}^{K} b_{jc_i lk}(x_{il})\beta_{jc_i lk}$	$b_{jc_i lk}(\cdot)$ is a basis function of cubic spline; K is the the dimension of the basis (default $K = 10$)	One curve along each lineage
Spatial location	$\mathbf{x}_i = (x_{i1}, x_{i2})^\mathsf{T} \in \mathbb{R}^2$	$f_{jc_i}(\mathbf{x}_i) = f_{jc_i}^{\mathrm{GP}}(x_{i1}, x_{i2}, K)$	$f_{jc_i}^{\mathrm{GP}}(\cdot, \cdot, K)$ is a Gaussian process smoother [37]; K is the dimension of the basis (default $K = 400$)	A smooth surface

Table 2.5: Comparison of scDesign3 and four other simulators for generating scRNA-seq data of discrete cell types (performance metrics were averaged from datasets PANCREAS, EMBYRO, and MARROW)

Metric	scDesign3	scGAN	SPARSim	muscat	ZINB-WaVE
PCA mLISI	1.82[1]	0.50	0.99	1.01	1.30
UMAP mLISI	1.89	1.39	1.37	1.46	1.46
KS distance of mean log expr.[2]	0.03	0.42	0.19	0.04	0.02
KS distance of var log expr.	0.09	0.72	0.30	0.08	0.04
KS distance of gene detection freq.[3]	0.03	0.29	0.17	0.04	0.01
KS distance of gene corr.[4]	0.05	0.12	0.33	0.17	0.14
KS distance of log library size	0.09	1.00	0.04	0.10	0.04
KS distance of cell distance	0.18	0.68	0.48	0.44	0.35
KS distance of cell detection freq.	0.12	0.68	0.56	0.16	0.04
KS distance of cell corr.	0.06	0.09	0.11	0.09	0.06
Corr. of corr. matrices	0.97	0.87	0.88	0.91	0.96

[1]: The underlines highlight the best result(s) of each metric.
[2]: The acronym "expr." stands for "expression."
[3]: The acronym "freq." stands for "frequency."
[4]: The acronym "corr." stands for "correlation."

CHAPTER 3

PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data

3.1 Introduction

In recent years, single-cell RNA-sequencing (scRNA-seq) technologies have undergone rapid development to dissect transcriptomic heterogeneity and to discover cell types or states in complex tissues [20, 64]. Embracing the capacity to measure transcriptomes of numerous cells simultaneously, scRNA-seq provides a powerful means to capture continuous cell-state transition across cells, and it has been used to study key cellular processes such as immune response [65] and cell development [66]. For example, a study of human fibroblasts identified distinct fibroblast subtypes responsible for mediating inflammation or tissue damage in arthritis [67]; a study of maternal-fetal interface tissue revealed new cell states and the importance of this tissue in maternal immune tolerance of paternal antigens [68]; a study of thymic development elucidated new principles of naïve T cell repertoire formation [69].

Pseudotime inference, also known as trajectory inference, is one of the most thriving scRNA-seq data analysis topics. The concept of "pseudotime" was first proposed in 2014 [70], and since then, more than 40 pseudotime inference methods have been developed [21]. Pseudotime inference aims to infer the ordering of cells along a lineage based on the cells' gene expression profiles measured by scRNA-seq, and the inferential target is "pseudotime," a time-like variable indicating the relative position a cell takes in a lineage. By establishing a temporal dimension in a static scRNA-seq dataset, pseudotime inference allows the probing

of individual genes' expression dynamics along with continuous cell-state changes. If a gene's mean expression changes along pseudotime, the gene is referred to as differentially expressed (DE) and is likely to play an important role in the underlying cellular process that gives rise to the pseudotime. Identifying DE genes is the most crucial analysis after pseudotime inference because genes are the most fundamental functional units for understanding biological mechanisms.

Several methods have been developed to identify DE genes along inferred cell pseudotime. Popular pseudotime inference methods—TSCAN [71], Slingshot [72], Monocle [70], and Monocle2 [73]—include a built-in functionality for identifying DE genes after pseudotime inference. Their common approach is to use the generalized additive model (GAM) [74–76] to fit each gene's expression level in a cell as a smooth-curve function of the cell's inferred pseudotime. However, these built-in methods for DE gene identification are restricted as an add-on and downstream step of the pseudotime inference method in the same software package, and they cannot take external, user-provided pseudotime as input. Therefore, if users would like to use a new pseudotime inference method, they cannot use these built-in DE methods.

To our knowledge when we finished the project, only two DE gene identification methods can take any user-provided pseudotime. The first and state-of-the-art one is tradeSeq, which uses the negative binomial generalized additive model (NB-GAM) to model the relationship between each gene's expression in a cell and the cell's pseudotime [77]. Its p-value calculation is based on a chi-squared distribution, an inaccurate approximation to the null distribution. As a result, its p-values lack the correct probability interpretation. This issue is noted in the tradeSeq paper: "Rather than attaching strong probabilistic interpretations to the p-values (which, as in most RNA-seq applications, would involve a variety of hard-to-verify assumptions and would not necessarily add much value to the analysis), we view the p-values simply as useful numerical summaries for ranking the genes for further inspection." Hence, the uncalibrated p-values of tradeSeq cannot be used for p-value-based statistical procedures such as the type I error control and the false discovery rate (FDR) control. The second method is Monocle3, better known as a pseudotime inference method [23], yet it also allows DE

gene identification based on user-provided cell covariates via regression analysis. For clarity, we refer to the pseudotime inference and differential expression functionalities in Monocle3 as "Monocle3-PI" and "Monocle3-DE," respectively. (Note that by "Monocle3-DE," we mean the "regression analysis `fit_models()`," not the "graph-autocorrelation analysis `graph_test()`," in the Monocle3 R package; only the former works for user-provided pseudotime.) Monocle3-DE uses the generalized linear model (GLM) to identify DE genes for a user-provided covariate, e.g., pseudotime. However, GLM is more restrictive than GAM in that GLM assumes the logarithmic transformation of a gene's expected read count in a cell is a strictly linear function of the cell's pseudotime, while this assumption does not hold for many genes [78]. Hence, Monocle3-DE would miss those complex relationships between gene expression and pseudotime that do not satisfy its GLM assumption. In other words, Monocle3-DE's restrictive GLM assumption impairs its power in identifying DE genes.

Besides the scRNA-seq methods we mentioned above, there are methods developed for identifying physical-time-varying DE genes from bulk RNA-seq time-course data. Among those methods, the ones allowing for continuous time can in principle be used to identify DE genes along pseudotime. Two examples of such methods are NBAMSeq [79] and ImpulseDE2 [80]. NBAMSeq is similar to tradeSeq in the use of NB-GAM, but it uses the Bayesian shrinkage method in DESeq2 [81] to estimate gene variances, while tradeSeq does not. ImpulseDE2 [80], a method favorably rated in a benchmark study for bulk RNA-seq data [82], models gene differential expression by a unique "impulse" model. A later study modified ImpulseDE2 to identify DE genes along pseudotime from scRNA-seq data [77]. However, the performance of NBAMSeq and ImpulseDE2 on scRNA-seq data lacks benchmarking. Loosely related, many methods can identify DE genes between discrete cell clusters, groups, or conditions [22, 83–86]; however, these methods are inapplicable to finding DE genes along continuous pseudotime.

More importantly, the existing methods that identify DE genes along pseudotime have a common limitation: they ignore the uncertainty of inferred cell pseudotime, which they consider as one fixed value per cell. This issue arises from the fact that most pseudotime inference methods only return point estimates of cell pseudotime without uncertainty quan-

tification (i.e., every cell only receives an inferred pseudotime without a standard error), with few exceptions [87, 88]. Hence, downstream DE gene identification methods treat these point estimates as fixed and ignore their uncertainty. However, this ignorance of uncertainty would result in invalid p-values, leading to either failed FDR control or power loss. This critical problem has been noted in several pseudotime inference method papers [71, 72, 87] and in the tradeSeq paper [77], yet it remains an open challenge to our knowledge.

Motivated by the ill-posed p-value issue of existing pseudotime-based differential expression methods, we propose PseudotimeDE, the first method that accommodates user-provided pseudotime inference methods, takes into account the random nature of inferred pseudotime, and outputs well-calibrated p-values. PseudotimeDE uses subsampling to estimate pseudotime inference uncertainty and propagates the uncertainty to its statistical test for DE gene identification. As the most notable advantage of PseudotimeDE over existing methods, PseudotimeDE's well-calibrated p-values ensures the reliability of FDR control and other downstream analyses, as well as avoiding unnecessary power loss due to overly-conservative p-values.

3.2 PseudotimeDE methodology

3.2.1 Mathmatical notations of PseudotimeDE

We denote by $\mathbf{Y} = (Y_{ij})$ an $n \times m$ gene expression count matrix, whose rows and columns correspond to n cells and m genes, respectively; that is, Y_{ij} is the read count of gene j in cell i. Taking $\mathbf{Y}$ as input, a pseudotime inference method would return a pseudotime vector $\mathbf{T} = (T_1, \ldots, T_i, \ldots, T_n)^\mathsf{T}$, where $T_i \in [0, 1]$ denotes the normalized inferred pseudotime of cell i (i.e., the cells with the smallest and largest pseudotime have $T_i = 0$ and 1, respectively; normalization is used for visualization simplicity). Note that T_i is a random variable due to the random-sampling nature of the n cells and the possible uncertainty introduced by the pseudotime inference method.

3.2.2 Uncertainty estimation

To estimate the uncertainty of pseudotime $\mathbf{T}$, we subsample 80% cells (rows) in $\mathbf{Y}$ for B times. Although there are some theoretical results about the optimal subsample size [89], they do not apply to our problem setting. Hence, we simply choose 80% because it is widely used [90, 91], similar to the popularity of 5-fold cross validation in machine learning [92]. Simulation results also supports that 80% is a reasonable choice, and PseudotimeDE is robust to various subsampling proportions (Fig. 3.28). It is worth noting that the bootstrap technique is inapplicable for our problem because it leads to repeated sampling of the same cell, causing issues with some pseudotime inference methods such as Monocle2. If the cells have pre-defined groups (i.e., cell types), we use the stratified sampling by first subsampling 80% cells within each group and then combining these within-group subsamples into one subsample. By default, we set $B = 1000$. For each subsample $\mathbf{Y}^b = (Y^b_{ij})$, an $n' \times m$ matrix where $n' = \lfloor .8n \rfloor$, we perform pseudotime inference with the same parameters used for the original dataset $\mathbf{Y}$. As a result, we obtain B subsample-based realizations of pseudotime $\mathbf{T}$: $\left\{\mathbf{T}^1, \cdots, \mathbf{T}^b, \cdots, \mathbf{T}^B\right\}$, where $\mathbf{T}^b \in [0, 1]^{n'}$, and each cell appears in approximately 80% of these B realizations. Note that we have to apply pseudotime inference to each subsample

before permutation to account for pseudotime inference uncertainty; otherwise, if each subsample's pseudotime is just a subsample of all cells' pseudotime, we are essentially treating all cells' pseudotime as fixed, and the uncertainty in pseudotime inference is ignored. Here is the mathematical explanation. Given that we have n cells with inferred pseudotime as $T_1, \ldots, T_n$, if we use direct subsampling, then in the b-th subsampling, the subsampled n' cells' pseudotime is just a size-n' subsample of $\{T_1, \ldots, T_n\}$. Instead, in PseudotimeDE, the subsampled n' cells' inferred pseudotime $T_1^b, \ldots, T_{n'}^b$ may be n' values that are not in $T_1, \ldots, T_n$. In other words, the uncertainty in pseudotime inference is reflected in $T_1^b, \ldots, T_{n'}^b$.

3.2.3 PseudotimeDE model

We use the negative binomial–generalized additive model (NB-GAM) as the baseline model to describe the relationship between every gene's expression in a cell and the cell's pseudotime. For gene j ($j = 1, \ldots, m$), its expression Y_{ij} in cell i and the pseudotime T_i of cell i ($i = 1, \ldots, n$) are assumed to follow

$$\begin{cases} Y_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j)\,, \\ \log(\mu_{ij}) = \beta_{j0} + f_j(T_i)\,, \end{cases}$$

where $\mathrm{NB}(\mu_{ij}, \phi_j)$ denotes the negative binomial distribution with mean μ_{ij} and dispersion ϕ_j, and $f_j(T_i) = \sum_{k=1}^{K} b_k(T_i)\beta_{jk}$ is a cubic spline function. The number of knots k is predefined as 6 and usually has little effect on results [93]. For gene j, PseudotimeDE fits the NB-GAM to $(Y_{1j}, \ldots, Y_{nj})^\mathsf{T}$ and $\mathbf{T} = (T_1, \ldots, T_n)^\mathsf{T}$ using the R package `mgcv` (version 1.8.31), which estimates model parameters by penalized-restricted maximum likelihood estimation.

To account for excess zeros in scRNA-seq data that may not be explained by the NB-GAM, we introduce a hidden variable Z_{ij} to indicate the "dropout" event of gene j in cell i, and the resulting model is called the zero-inflated negative binomial–generalized additive

model (ZINB-GAM):

$$\begin{cases} Z_{ij} \sim \mathrm{Ber}(p_{ij})\,, \\ Y_{ij}|Z_{ij} \sim Z_{ij} \cdot \mathrm{NB}(\mu_{ij}, \phi_j) + (1 - Z_{ij}) \cdot 0\,, \\ \log(\mu_{ij}) = \beta_{j0} + f_j(T_i)\,, \\ \mathrm{logit}(p_{ij}) = \alpha_{j0} + \alpha_{j1} \log(\mu_{ij})\,. \end{cases}$$

For gene j, PseudotimeDE fits the ZINB-GAM to $(Y_{1j}, \ldots, Y_{nj})^\mathsf{T}$ and $\mathbf{T} = (T_1, \ldots, T_n)^\mathsf{T}$ using the expectation-maximization (EM) algorithm, which is partially based on R package `zigam` [94]. To use PseudotimeDE, users can specify whether to use the ZINB-GAM or NB-GAM. If users do not provide a specification, PseudotimeDE will automatically choose between the two models for each gene by the Akaike information criterion (AIC). By default, PseudotimeDE uses NB-GAM unless the AIC of ZINB-GAM exceeds the AIC of NB-GAM by at least 10, a threshold suggested by [95].

3.2.4 Statistical test and p-value calculation

To test if gene j is DE along cell pseudotime, PseudotimeDE defines the null and alternative hypotheses as

$$H_0 : f_j(\cdot) = \mathbf{0} \quad \text{vs.} \quad H_1 : f_j(\cdot) \neq \mathbf{0}$$

We denote the estimate of $(f_j(T_1), \ldots, f_j(T_n))^\mathsf{T}$ by $\hat{\boldsymbol{f}}_j$, whose estimated covariance matrix (of dimensions $n \times n$) is denoted by $\hat{\mathbf{V}}_{f_j}$. Then the test statistic is

$$S_j = \hat{\boldsymbol{f}}_j^\mathsf{T} \hat{\mathbf{V}}_{f_j}^{r-} \hat{\boldsymbol{f}}_j\,,$$

where $\hat{\mathbf{V}}_{f_j}^{r-}$ is the rank-r pseudoinverse of $\hat{\mathbf{V}}_{f_j}$, where r is determined in the way described in [96]. When the T_i's are fixed, the asymptotic null distribution of S_j is described in [96], and the p-value can be calculated by the R package `mgcv`.

A key novelty of PseudotimeDE is its accounting for the uncertainty of inferred pseudo-

time. When the T_i's are random, the asymptotic null distribution of S_j given that T_i's are fixed [96] and the p-value calculation in the R package `mgcv` no longer apply. To address this issue and estimate the null distribution, PseudotimeDE uses the following permutation procedure: (1) PseudotimeDE randomly permutes each subsample-based realization $\mathbf{T}^b = (T_1^b, \ldots, T_{n'}^b)^\mathsf{T}$ into $\mathbf{T}^{*b} = (T_1^{*b}, \ldots, T_{n'}^{*b})^\mathsf{T}$; (2) PseudotimeDE fits the above model to $(Y_{1j}^b, \ldots, Y_{n'j}^b)^\mathsf{T}$ and $\mathbf{T}^{*b}$, and calculates the test statistic S_j's value as s_j^b using the R package `mgcv`; (3) PseudotimeDE performs (1) and (2) for $b = 1, \ldots, B$ and collects the resulting $\{s_j^1, \ldots, s_j^B\}$ as the null values of the test statistic S_j.

Then PseudotimeDE estimates the null distribution of S_j in two ways. Based on the estimated null distribution in either way and the observed test statistic value s_j, which is calculated from the original dataset by the R package `mgcv`, PseudotimeDE calculates a p-value for gene j.

1. **Empirical estimate**. PseudotimeDE uses the empirical distribution of $\{s_j^1, \ldots, s_j^B\}$ as the estimated null distribution. Following the suggestion in [97], PseudotimeDE calculates the p-value of gene j as

$$p_j^{\text{emp}} = \frac{\sum_{b=1}^{B} \mathbb{I}(s_j^b \geqslant s_j) + 1}{B + 1},$$

   where $\mathbb{I}(\cdot)$ is the indicator function. We refer to this p-value as the "empirical p-value."

2. **Parametric estimate**. The resolution of p_j^{emp} depends on the number of permutations B, because the smallest value p_j^{emp} may take is $1/(B+1)$. Although users often cannot afford a too large B due to limited computational resources, they still desire a high resolution of p-values to control the FDR to a small value (e.g., 5%) when the number of tests (i.e., the number of genes in DE gene identification) is large. To increase the resolution of p-values, PseudotimeDE fits a parametric distribution to $\{s_j^1, \ldots, s_j^B\}$ and uses the fitted distribution as the estimated null distribution. Driven by the empirical distribution of $\{s_j^1, \ldots, s_j^B\}$, PseudotimeDE considers two parametric distributions: (1) a gamma distribution $\Gamma(\alpha, \beta)$ with $\alpha, \beta > 0$ and (2) a two-component gamma mixture

model $\gamma\Gamma(\alpha_1, \beta_1) + (1-\gamma)\Gamma(\alpha_2, \beta_2)$ with $0 < \gamma < 1$ and $\alpha_1, \beta_1, \alpha_2, \beta_2 > 0$. After fitting both distributions to $\{s_j^1, \ldots, s_j^B\}$ using the maximum likelihood estimation (gamma distribution fit by the R package `fitdistrplus` (version 1.0.14) [98] and gamma mixture model fit by the R package `mixtools` (version 5.4.5) [99]), PseudotimeDE chooses between the two fitted distributions by performing the likelihood ratio test (LRT) with 3 degrees of freedom (i.e., difference in the numbers of parameters between the two distributions). If the LRT p-value is less or equal than 0.01, PseudotimeDE uses the fitted two-component gamma mixture model as the parametric estimate of the null distribution of S_j; otherwise, PseudotimeDE uses the fitted gamma distribution. The Anderson-Darling goodness-of-fit test verifies that such a parametric approach fits the empirical distributions well (Fig. 3.29). Denoting the cumulative distribution function of the parametrically estimated null distribution by $\hat{F}_j(\cdot)$, PseudotimeDE calculates the p-value of gene j as

$$p_j^{\text{param}} = 1 - \hat{F}_j(s_j)\,,$$

where is referred to as the "parametric p-value."

PseudotimeDE outputs both p_j^{emp} and p_j^{param} for gene j, $j = 1, \ldots, m$. Empirical evidence shows that parametric p-values agree with empirical p-values well across the $[0, 1]$ interval (Fig. 3.27). All the findings in the Results section are based on $p_1^{\text{param}}, \ldots, p_m^{\text{param}}$ due to their higher resolution.

3.3 Results

3.3.1 Overview of the PseudotimeDE method

The statistical method of PseudotimeDE consists of four major steps: subsampling, pseudotime inference, model fitting, and hypothesis testing (Fig. 3.1). The first two steps are performed at the cell level and include all informative genes (whose selection depends on the pseudotime inference method, e.g., Slingshot and Monocle3-PI), while the last two steps are performed on every gene that is potentially DE.

1. In the subsampling step, PseudotimeDE subsamples 80% of cells from the original dataset to capture the uncertainty of pseudotime inference, the same technique as used in [21, 72, 100].

2. In the pseudotime inference step, PseudotimeDE applies a user-specified pseudotime inference method to the original dataset and each subsample, so that every cell receives its inferred pseudotime in the original dataset and all the subsamples that include it. To construct null cases where genes are non-DE for later hypothesis testing, PseudotimeDE permutes the inferred pseudotime in each subsample, independent of other subsamples.

3. In the model fitting step, PseudotimeDE fits NB-GAM or zero-inflated negative binomial GAM (ZINB-GAM) to every gene in the original dataset to obtain a test statistic that indicates the effect size of the inferred pseudotime on the gene's expression.

4. In the hypothesis testing step, for every gene, Pseudotime fits the same model used for the original dataset to the permuted subsamples to obtain approximate null values of the gene's test statistic (the null values are approximate because the subsamples do not have the same number of cells as in the original dataset). To save the number of subsamples needed and to improve the p-value resolution, Pseudotime fits a Gamma distribution or a mixture of two Gamma distributions to these null values. It subsequently uses the fitted parametric distribution as the approximate null distribution of the test statistic. Finally, PseudotimeDE calculates a right-tail p-value for the gene from the gene's test statistic in the original dataset and the approximate null distribution.

3.3.2 Simulations verify that pseudotimeDE outperforms existing methods in the validity of p-values and the identification power

We use a widely-used simulator `dyntoy` [21, 77] to generate four synthetic scRNA-seq datasets, among which three are single-lineage datasets with low-, medium- and high-dispersion levels, and the other is a bifurcation dataset. Since the single-lineage high-dispersion dataset

best resembles the real scRNA-seq data (Fig. 3.9-3.10), we use it as our primary case. We apply two pseudotime inference methods—Slingshot and Monocle3-PI—to each synthetic dataset to infer cell pseudotime.

First, we find that PseudotimeDE successfully captures the underlying uncertainty of inferred pseudotime. The first layer—"linear uncertainty"—reflects the randomness of inferred cell pseudotime within a cell lineage (Fig. 3.2a & c). Fig. 3.2b & d show the distributions of individual cells' inferred pseudotime by Slingshot and Monocle3-PI, respectively, across 1000 subsampled datasets, confirming that linear uncertainty is specific to pseudotime inference methods. Between the two methods, Monocle3-PI demonstrates greater linear uncertainty. The second layer—"topology uncertainty"—reflects the randomness of lineage construction. The synthetic bifurcation dataset contains two cell lineages. Slingshot correctly constructs the bifurcation topology from the original dataset and the 1000 subsampled datasets. While Monocle3-PI captures the bifurcation topology from the original dataset (Fig. 3.2e), it fails to capture the topology from over 50% of subsamples (Fig. 3.2f shows randomly picked 10 subsamples), demonstrating its greater topology uncertainty than Slingshot's.

After confirming pseudotime inference uncertainty, we benchmark PseudotimeDE against four DE gene identification methods: tradeSeq, Monocle3-DE, NBAMSeq, and ImpulseDE2. The first two methods, tradeSeq and Monocle3-DE, are the state-of-the-art for scRNA-seq data analysis and thus serve as the main competitors of PseudotimeDE. In our benchmark, we first evaluate these methods in terms of the validity of their p-values, which should be uniformly distributed between 0 and 1 under the null hypothesis (i.e., a gene is not DE). Our results show that, among the five methods, PseudotimeDE generates the best-calibrated p-values that follow the expected uniform distribution most closely (Fig. 3.3a & f and Figs. 3.11–3.13a & f). Among the existing four methods, only Monocle3-DE provides roughly calibrated p-values, while tradeSeq, NBAMSeq, and ImpulseDE2 output p-values that are much deviated from the expected uniform distribution. This observation is confirmed by the Kolmogorov-Smirnov test, which evaluates how closely p-values follow the uniform distribution. Since the identification of DE genes relies on a small p-value cutoff, the smaller p-values are more important than the larger ones. Hence, we re-plot the p-values

on $-\log_{10}$ scale to closely examine the calibration of small p-values (Fig. 3.3b & g and Figs. 3.11–3.13b & g). Again, PseudotimeDE returns the best-calibrated p-values, while the other four methods generate overly small p-values that would inflate false discoveries. This is reflected in our results: at a target 5% FDR threshold, PseudotimeDE leads to the best FDR control among all methods (Fig. 3.3c & h and Figs. 3.11–3.13c & h).

Next, we compare these methods in terms of their ability to distinguish DE genes from non-DE genes, ability measured by the area under the receiver operating characteristic curve (AUROC) values (Fig. 3.3d & i and Figs. 3.11–3.13d & i). PseudotimeDE achieves the highest AUROC values. Among the other four methods, tradeSeq and NBAMSeq have slightly lower AUROC values than PseudotimeDE's, and Monocle3-DE and ImpulseDE2 have much lower AUROC values than the other three methods'. The reason is that PseudotimeDE, tradeSeq, and NBAMSeq all use the flexible model NB-GAM, while Monocle3-DE and ImpulseDE2 use much more restrictive models, which limit their power.

Realizing that the ill-calibrated p-values of the existing four methods invalidate their FDR control, we compare all five methods in terms of their power under an actual 5% false discovery proportion (FDP, defined as the proportion of false discoveries among the discoveries in one synthetic dataset) instead of the nominal 5% FDR. Our results show that PseudotimeDE achieves the highest power on all datasets except for the bifurcation dataset, where PseudotimeDE has slightly lower power than tradeSeq's (Fig. 3.3e & j and Figs. 3.11–3.13e & j). These results demonstrate the high power of PseudotimeDE and its effective FDR control, which is lacking in existing methods. In summary, our simulation results verify that PseudotimeDE outperforms existing methods in terms of generating well-calibrated p-values, which are essential for FDR control, and identifying DE genes with high power. Notably, the two bulk RNA-seq methods, NBAMSeq and ImpulseDE2, yield worse results than the three scRNA-seq methods do. Hence, we only focus on the scRNA-seq methods in the following three real data applications.

3.3.2.1 Real data example 1: dendritic cells stimulated with lipopolysaccharide

In the first application, we compare PseudotimeDE with tradeSeq and Monocle3-DE on a dataset of mouse dendritic cells (DCs) after stimulation with lipopolysaccharide (LPS, a component of gram-negative bacteria) [101]. In this dataset, gene expression changes are expected to be associated with the immune response process. We first apply Slingshot and Monocle3-PI to this dataset to infer cell pseudotime, and then we input the inferred pseudotime into PseudotimeDE, tradeSeq, and Monocle3-DE for DE gene identification. Consistent with our simulation results, the p-values of tradeSeq are ill-calibrated: their bimodal distributions indicate that they do not follow the uniform distribution under the null hypothesis; instead, many of them are inflated, and this inflation would lead to power loss in DE gene identification (Fig. 3.4a & e). Indeed, at a nominal Benjamini-Hochberg (BH) adjusted p-value $\leqslant 0.01$ threshold (which corresponds to controlling the FDR $\leqslant 1\%$ when p-values are valid), tradeSeq identifies the smallest number of DE genes, while PseudotimeDE identifies the most DE genes, followed by Monocle3-DE. Notably, most of the DE genes identified by tradeSeq are also identified by PseudotimeDE (Fig. 3.4b & f), a result consistent with the over-conservativeness of tradeSeq due to its inflated p-values. Unlike tradeSeq, Monocle3-DE does not exhibit the inflated p-value issue; however, it uses a more restrictive model than PseudotimeDE and tradeSeq do. Hence, we use functional analyses to investigate whether Monocle3-DE misses certain DE genes due to its restrictive modeling. We also investigate whether the additional DE genes found by PseudotimeDE but missed by tradeSeq or Monocle3-DE are biologically meaningful.

Our first strategy is to perform gene ontology (GO) analysis on the DE genes identified by each method and compare the enriched GO terms. We find that more GO terms are enriched (with enrichment p-values < 0.01) in the DE genes identified by PseudotimeDE (Fig. 3.14a & c), and that the PseudotimeDE-specific GO terms are related to immune responses (Fig. 3.14b & d). However, comparing enriched GO terms does not directly reflect the difference of DE genes identified by different methods. Hence, our second strategy is to probe the functions of the DE genes that are uniquely identified by one method in pairwise comparisons

of PseudotimeDE vs. tradeSeq and PseudotimeDE vs. Monocle3-DE. We first perform GO analysis on each set of uniquely identified DE genes. For a fair comparison of two methods, we remove the overlapping DE genes found by both methods from the background gene list in GO analysis. Our results show that many more GO terms are enriched (with enrichment p-values < 0.01) in Pseudotime-specific DE genes than in tradeSeq- or Monocle3-DE-specific DE genes (Fig. 3.4c & g). Moreover, many of those PseudotimeDE-specific GO terms are directly related to the immune responses of DCs to LPS stimulation, including the GO terms "cellular response to lipopolysaccharide" and "defense response to Gram-negative bacterium" (Fig. 3.4d & h). To focus more on immune responses, we next perform enrichment analysis using the immunologic signatures (C7) in the Molecular Signatures Database (MSigDB) [102]. Our results show that only PseudotimeDE-specific DE genes have enriched MSigDB C7 terms (with BH adjusted p values < 0.01), while tradeSeq- and Monocle3-DE-specific DE genes have almost no enrichment (Fig. 3.14a & c). More importantly, many enriched terms in PseudotimeDE-specific DE genes were found by previous studies of DCs stimulated with LPS (see examples in Fig. 3.14b & d); this is direct evidence that supports the validity of PseudotimeDE-specific DE genes. For illustration purpose, we visualize the expression levels of some known and novel DE genes identified by PseudotimeDE using UMAP, and clear DE patterns are observed (Fig. 3.16–3.17). In conclusion, our functional analyses verify that PseudotimeDE identifies biologically meaningful DE genes missed by tradeSeq and Monocle3-DE, confirming that PseudotimeDE has high power in addition to its well-calibrated p-values.

3.3.2.2 Real data example 2: pancreatic beta cell maturation

In the second application, we compare PseudotimeDE with tradeSeq and Monocle3-DE on a dataset of mouse beta cell maturation process [103]. We first apply Slingshot and Monocle3-PI to this dataset to infer cell pseudotime, and then we input the inferred pseudotime into PseudotimeDE, tradeSeq, and Monocle3-DE for DE gene identification. Consistent with previous results, the p-values of tradeSeq follow a bimodal distribution, suggesting that many of them are incorrectly inflated (Fig. 3.5a & f). At the nominal BH-adjusted p-value

$\leqslant 0.01$ level, PseudotimeDE identifies the second most DE genes, fewer than Monocle3-DE's identified DE genes and much more than tradeSeq's (Fig. 3.5b & g). As the numbers of identified DE genes cannot reflect these methods' performance, we use three approaches to evaluate the DE genes identified by each method.

We first perform GO analysis on each set of uniquely identified DE genes, using the same pairwise comparisons of PseudotimeDE vs. tradeSeq and PseudotimeDE vs. Monocle3-DE as for the LPS-dendritic data. Our results show that more GO terms are enriched (with enrichment p-values < 0.01) in PseudotimeDE-specific DE genes than in tradeSeq- or Monocle3-DE-specific DE genes (Fig. 3.5c & h). Moreover, many of those PseudotimeDE-specific GO terms are directly related to pancreatic beta cell development, e.g., "positive/negative regulation of Notch signaling pathway" [104] and "endocrine pancreas development" (Fig. 3.5c & h). As a complementary result, we also perform GO analysis on the DE genes identified by each method. We find that the GO terms, which are only enriched in the DE genes identified by PseudotimeDE, are related to beta cell development and thus more biologically meaningful than the GO terms that are only enriched in the DE genes identified by tradeSeq or Monocle3-DE (Fig. 3.18b & d).

Second, we utilize the DE genes identified from bulk RNA-seq data in the original paper [103] to evaluate the DE gene rankings established by PseudotimeDE, tradeSeq, and Monocle3-DE from scRNA-seq data. Taking the bulk DE genes as a gene set, we perform the gene-set enrichment analysis (GSEA) [102] on all genes' $-\log_{10} p$-values output by PseudotimeDE, tradeSeq, and Monocle3-DE. Among the three methods, PseudotimeDE leads to the highest normalized enrichment score (NES), suggesting that the bulk DE genes are most enriched in the top-ranked DE genes found by PseudotimeDE.

Third, we examine a highly credible DE gene *Slc39a10* [103, 105] and a verified non-DE gene *Sst* [103] as representative examples. For *Slc39a10*, both PseudotimeDE and Monocle3-DE yield small p-values ($< 10^{-6}$), while tradeSeq outputs a p-value > 0.1 and thus misses it (Fig. 3.5e & g). For *Sst*, PseudotimeDE yields the largest p-value (> 0.001), while tradeSeq and Monocle3-DE yield extremely small p-values ($< 10^{-10}$) and thus mistaken it as a DE gene. Hence, PseudotimeDE has the best performance on these two representative genes.

For illustration purpose, we visualize the expression levels of some known and novel DE genes identified by PseudotimeDE using UMAP, and clear DE patterns are observed (Figs. 3.19-3.20).

3.3.2.3 Real data example 3: bone marrow differentiation

In the third application, we compare PseudotimeDE with tradeSeq and Monocle3-DE on a dataset of mouse bone marrow differentiation [55]. We apply Slingshot with UMAP for dimensionality reduction to infer cell pseudotime as described in the tradeSeq paper [77]. Slingshot constructs the reported bifurcation topology (in the tradeSeq paper) on the original dataset (Fig. 3.6a), but it infers trifurcation topology, instead of bifurcation topology, on 40% of subsamples (Fig. 3.6b shows randomly picked ten subsamples). Note that the third lineage consisting of the cell type megakaryocyte (MK) was reported in the Monocle2 paper (ref. [73]), suggesting the observed topology uncertainty may be biologically meaningful.

For a fair comparison, we only make PseudotimeDE use the subsamples with inferred bifurcation topology, because both tradeSeq and Monocle3-DE use the inferred bifurcation topology from the original data to identify DE genes. Consistent with previous results, the tradeSeq p-values follow a bimodal distribution that is unexpected for well-calibrated p-values. At a nominal BH-adjusted p-value $\leqslant 0.01$ threshold, the three methods identify highly similar DE genes (Fig. 3.6e & g). For instance, PseudotimeDE and tradeSeq share about 80% of their identified DE genes (Jaccard index). From the few method-specific DE genes, functional analyses cannot indicate which method performs better. Therefore, we use GSEA instead to evaluate methods' p-values. Surprisingly, although the three methods identify highly similar DE genes, their p-values lead to vastly different GSEA results. At the $q < 0.25$ level, PseudotimeDE and Monocle3-DE yield hundreds of enriched gene sets, while tradeSeq only yields a few or no enriched gene sets (Fig. 3.6f & h). This result indicates that, besides the ranking of p-values, the nominal values of p-values are also crucial for downstream analysis such as GSEA. Hence, the well-calibrated p-values make PseudotimeDE superior to existing methods for DE gene identification and downstream analyses.

3.3.2.4 Real data example 4: natural killer T cell subtypes

In the fourth application, we compare PseudotimeDE with tradeSeq and Monocle3-DE on a dataset of natural killer T cell (NKT cell) subtypes [106]. We apply Slingshot with PCA for dimensionality reduction to infer cell pseudotime and construct the trifurcation topology (Fig. 3.7a) reported in the original study. We apply the three DE methods to identify DE genes in each of the three lineages. Consistent with the previous results, the p-values of tradeSeq follow a bimodal distribution, suggesting that many of them are incorrectly inflated (Fig. 3.7b).

For validation purpose, we utilize the lineage-specific DE genes identified from bulk RNA-seq data in the original study [106] to evaluate the DE gene rankings established by PseudotimeDE, tradeSeq, and Monocle3-DE from scRNA-seq data. Specifically, we perform the GSEA using the bulk DE gene sets in the same way as for the pancreatic beta cell maturation dataset. The GSEA shows that PseudotimeDE's p-values best agree with the lineage-specific DE genes from bulk data and thus most distinguish the three lineages. For example, for the NKT1 lineage, PseudotimeDE's small p-values are exclusively enriched in the "NKT1 bulk" gene set, while tradeSeq and Monocle3-DE have small p-values enriched in at least two lineage-specific DE gene sets (Fig. 3.7c). This result confirms that, compared with the DE genes identified by the other two DE methods, the top DE genes identified by PseudotimeDE are more biologically meaningful.

3.3.2.5 Real data example 5: cell cycle phases

In the fifth application, we compare PseudotimeDE with tradeSeq and Monocle3-DE on a dataset of human induced pluripotent stem cells (iPSCs) measured with cell cycle phases (FUCCI labels) [107]. The original study has reported 101 cyclic genes whose expression levels have large proportions of variance explained (PVE) explained by cells' FUCCI labels [107]; that is, cells' FUCCI labels are regarded as the predictor, a gene's expression levels in the same cells are regarded as the response, and a PVE is calculated from a nonparametric smoothing fit; hence, the larger the PVE, the better the gene's expression levels can be

predicted by the cell cycle phases. The original study has also developed an R package `peco` to infer cell cycle phases from scRNA-seq data.

In our study, we first construct a benchmark dataset by treating the 101 cyclic genes as true DE genes and using the same genes with expression levels randomly shuffled across cells as the true non-DE genes; hence, our positive and negative sets both contain 101 genes. Then we apply the R package `peco` to this dataset to infer each cell's cycle phase, which is equivalent to pseudotime; that is, we use `peco` as the pseudotime inference method. Finally, we apply the three DE methods.

Our results show that, for the true non-DE genes, only PseudotimeDE generates valid p-values that approximately follow the Uniform$[0, 1]$ distribution (Fig. 3.8a & c). For the true DE genes, PseudotimeDE's ($-\log_{10}$ transformed) p-values, one per gene, have the highest correlation with these genes' PVE, indicating that PseudotimeDE successfully identifies the top DE genes as those with the strongest cyclic trends (Fig. 3.8b). PseudotimeDE also yields successful FDR control, the highest AUROC value, and the highest power, among the three DE methods (Fig. 3.8d, e and f). Therefore, we conclude that PseudotimeDE outperforms tradeSeq and Monocle3-DE in identifying cell cycle-related genes from this iPSC scRNA-seq dataset.

3.4 Discussion

We propose a statistical method PseudotimeDE to identify DE genes along inferred cell pseudotime. PseudotimeDE focuses on generating well-calibrated p-values while taking into account the randomness of inferred pseudotime. To achieve these goals, PseudotimeDE first uses subsampling to estimate the uncertainty of pseudotime. Second, PseudotimeDE fits the NB-GAM or ZINB-GAM to both the original dataset and the permuted subsampled datasets to calculate the test statistic and its approximate null values. Next, PseudotimeDE fits a parametric distribution to estimate the approximate null distribution of the test statistic. Finally, PseudotimeDE calculates p-values with a high resolution. PseudotimeDE is flexible to accommodate cell pseudotime inferred in a standard format by any method. Its use of

NB-GAM and ZINB-GAM allows it to capture diverse gene expression dynamics and to accommodate undesirable zero inflation in data.

Comprehensive studies on simulated and real data confirm that PseudotimeDE yields better FDR control and higher power than four existing methods (tradeSeq, Monocle3-DE, NBAMSeq, and ImpulseDE2) do. On simulation data, PseudotimeDE generates well-calibrated p-values that follow the uniform distribution under the null hypothesis, while existing methods except Monocle3-DE have p-values violating the uniform assumption. Well-calibrated p-values guarantee the valid FDR control of PseudotimeDE. Moreover, thanks to its use of flexible models NB-GAM and ZINB-GAM, PseudotimeDE has higher power than Monocle3-DE, which uses a more restrictive model GLM and thus has less power. PseudotimeDE also outperforms the other three methods–tradeSeq, NBAMSeq, and ImpulseDE2—that generate ill-calibrated p-values in terms of power. On three real scRNA-seq datasets, the DE genes uniquely identified by PseudotimeDE embrace better biological interpretability revealed by functional analyses, and the p-values of PseudotimeDE lead to more significant GSEA results.

An interesting and open question is what pseudotime inference method works the best with PseudotimeDE. While we observe that PseudotimeDE has higher power with Slingshot than with Monocle3-PI in simulation studies, we realize that the reason may be associated with the simulation design (e.g., the lineage structures), and thus we cannot draw a conclusion from this observation. Due to the diversity of biological systems and the complexity of pseudotime inference [21], we decide to leave the choice of pseudotime inference methods open to users, and this is the advantage of PseudotimeDE being flexible to accommodate inferred pseudotime by any methods. In practice, we encourage users to try popular pseudotime inference methods and use PseudotimeDE as a downstream step to identify DE genes, so that they can analyze the identification results and decide which pseudotime inference method is more appropriate for their dataset.

The zero inflation, or "dropout" issue, remains perplexing and controversial in the single-cell field [108–112]. The controversy is regarding whether excess zeros that cannot be explained by Poisson or negative binomial distributions are biological meaningful or not. Facing

this controversy, we provide two models in PseudotimeDE: NB-GAM and ZINB-GAM, the former treating excess zeros as biologically meaningful and the latter not. Specifically, the negative binomial distribution in NB-GAM is fitted to all gene expression counts including excess zeros, while the fitting of the negative distribution in ZINB-GAM excludes excess zeros, which ZINB-GAM implicitly treats as non-biological zeros. PseudotimeDE allows the choice between the two models to be user specified or data driven. From our data analysis, we realize that the choice often requires biological knowledge of the dataset to be analyzed. Specifically, on the LPS-dendritic cell dataset and pancreatic beta cell maturation dataset, we observe that ZINB-GAM leads to power loss: some potential DE genes cannot be identified by ZINB-GAM because zero counts contain useful information (Figs. 3.23-3.25). Our observation is consistent with another recent study [110], whose authors observed that "zero-inflated models lead to higher false-negative rates than identical non-zero-inflated models." Hence, our real data analysis results are based on NB-GAM. However, realizing the complexity of biological systems and scRNA-seq protocols, we leave the choice between NB-GAM and ZINB-GAM as an option for users of PseudotimeDE, and we encourage users to plot their known DE genes as in Figs. 3.23-3.25 to decide which of NB-GAM and ZINB-GAM better captures the gene expression dynamics of their interest.

The current implementation of PseudotimeDE is restricted to identifying the DE genes that have expression changes within a cell lineage. While methods including GPfates [113], Monocle2 BEAM [114], and tradeSeq can test whether a gene's expression change is associated with a branching event leading to two lineages, they do not consider the uncertainty of lineage inference. How to account for such topology uncertainty is a challenging open question, as we have seen in Figs. 3.2f and 3.6b that the inferred lineage may vary from a bifurcation topology to a trifurcation topology on different subsets of cells. A possible direction is to use the selective inference [115, 116], and we will leave the investigation of this question to future research. Due to this topology uncertainty issue, PseudotimeDE is most suitable for single-cell gene expression data that contain only one cell lineage (including cyclic data) or a small number of well separated cell lineages (e.g., bifurcation). The reason is that these data can maintain stable inferred cell topology after subsampling, an

essential requirement of PseudotimeDE. That said, PseudotimeDE is not designed for data with many equivocal cell lineages or a complex cell hierarchy, the data that cannot maintain stable inferred cell topology across subsamples, because for such data, it is difficult to find one-to-one matches between cell lineages inferred from a subsample and those inferred from the original data. Then, a practical solution for such data is to first define a cell lineage of interest and then apply PseudotimeDE to only the cells assigned to this lineage.

There are other open questions to be explored. An important question is: when do we want to identify DE genes along pseudotime? As we have shown in Section 3.3, inferred pseudotime can be highly uncertain. As biologists often sequence cells at multiple physical time points if they want to investigate a biological process, a straightforward analysis is to identify the DE genes that have expression changes across the physical time points. Then we have two definitions of DE genes: the genes whose expression changes across pseudotime vs. physical time. Understanding which definition is more biologically relevant is an open question. Another question is whether it is possible to integrate pseudotime with physical time to identify biologically relevant DE genes. Answering either question requires a statistical formulation that is directly connected to a biological question.

Another question is how to explore gene-gene correlations along cell pseudotime. Current DE methods only detect marginal gene expression changes but ignore gene-gene correlations. It remains unclear whether gene-gene correlations are stable or varying along cell pseudotime. Hence, a new statistical method to detect gene-gene correlation changes along inferred pseudotime may offer new biological insights into gene co-expression and regulation at the single-cell resolution.

3.5 Code and data availability

The R package PseudotimeDE is available at https://github.com/SONGDONGYUAN1994/PseudotimeDE. The tutorials of PseudotimeDE are available at https://songdongyuan1994.github.io/PseudotimeDE/docs/index.html. The source code and data for reproducing the results are available at: http://doi.org/10.5281/zenodo.8161964. The pre-processed

datasets are available at https://figshare.com/articles/dataset/PseudotimeDE_datasets/23596764.

3.6 Acknowledgments

We would like to thank Dr. Kelly Street for his suggestions on the usage of Slingshot and uncertainty estimate. We would like to thank Mr. Huy Nguyen for his assistance in simulation.

3.7 Figures

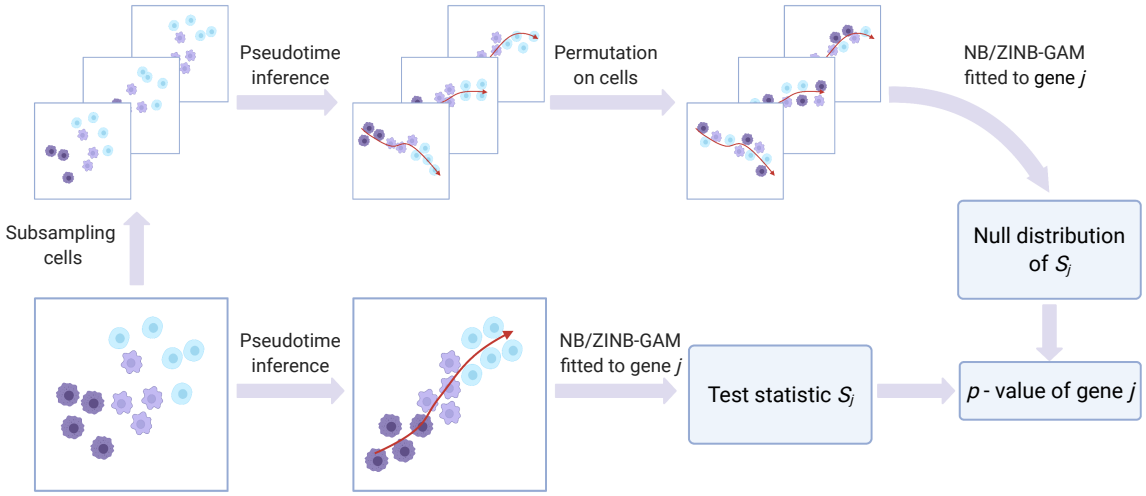


Figure 3.1: An illustration of the PseudotimeDE method.

The core of PseudotimeDE is to obtain a valid null distribution of the DE gene test statistic S_j. To achieve that, PseudotimeDE subsamples 80% cells from the original scRNA-seq data. Then on each subsample, PseudotimeDE performs pseudotime inference (using a user-specified method such as Slingshot and Monocle3-PI) and permutes the inferred pseudotime across cells. Next, PseudotimeDE fits a model (NB-GAM or ZINB-GAM) to the permuted subsamples to obtain the values of S_j under the null hypothesis and uses these values to approximate the null distribution of S_j. In parallel, PseudotimeDE fits the same model to the original dataset and calculates the observed value of S_j. Finally, PseudotimeDE derives the p-value from the observed value and the null distribution of S_j. Detail is described in Methods.

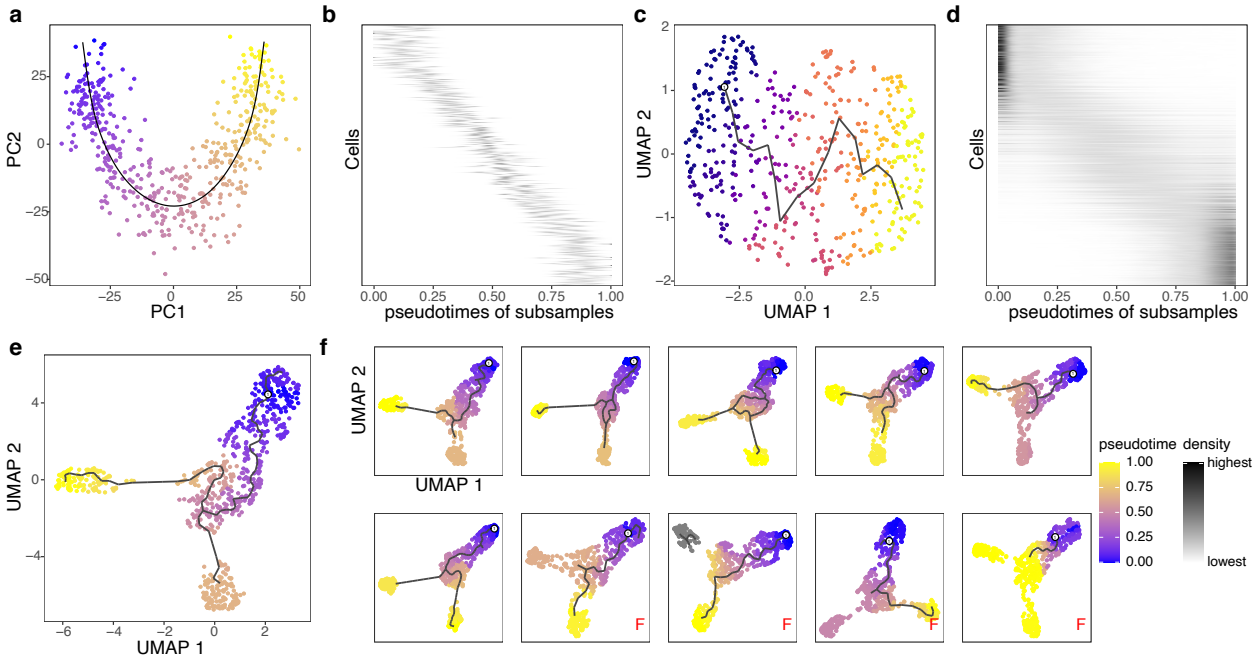


Figure 3.2: PseudotimeDE captures the uncertainty in pseudotime inference.

(a) Visualization of synthetic single-lineage cells marked with inferred pseudotime by Slingshot (using PCA). The black curve denotes the inferred lineage. **(b)** The distributions of individual cells' inferred pseudotime by Slingshot across subsamples. In the vertical axis, cells are ordered by their true time in the lineage used in simulation; for every cell (a vertical coordinate), black dots have horizontal coordinates corresponding to the cell's inferred pseudotime in the subsamples that include the cell. The more horizontally spread out the black dots, the greater uncertainty the pseudotime inference has. **(c)** Visualization of synthetic single-lineage cells marked with inferred pseudotime by Monocle3-PI (using UMAP). The black curve denotes the inferred lineage. Compared with (a), the inferred lineage is more wiggling. **(d)** The distributions of individual cells' inferred pseudotime by Monocle3-PI across subsamples. Compared with (b), the uncertainty in pseudotime inference is greater. **(e)** Visualization of synthetic bifurcating cells marked with inferred pseudotime by Monocle3-PI (using UMAP). Monocle3-PI recovers the bifurcation topology. **(f)** Visualization of ten subsamples of the cells in (e), marked with inferred pseudotime by Monocle3-PI (using UMAP) on each subsample. Four out of the ten subsamples do not have the bifurcation topology correctly inferred (labeled with red "F"), revealing the uncertainty in pseudotime inference by Monocle3-PI. In panels (a), (c), (e) and (f), inferred pseudotime is represented by a color scale from 0 (the earliest pseudotime) to 1 (the latest pseudotime).

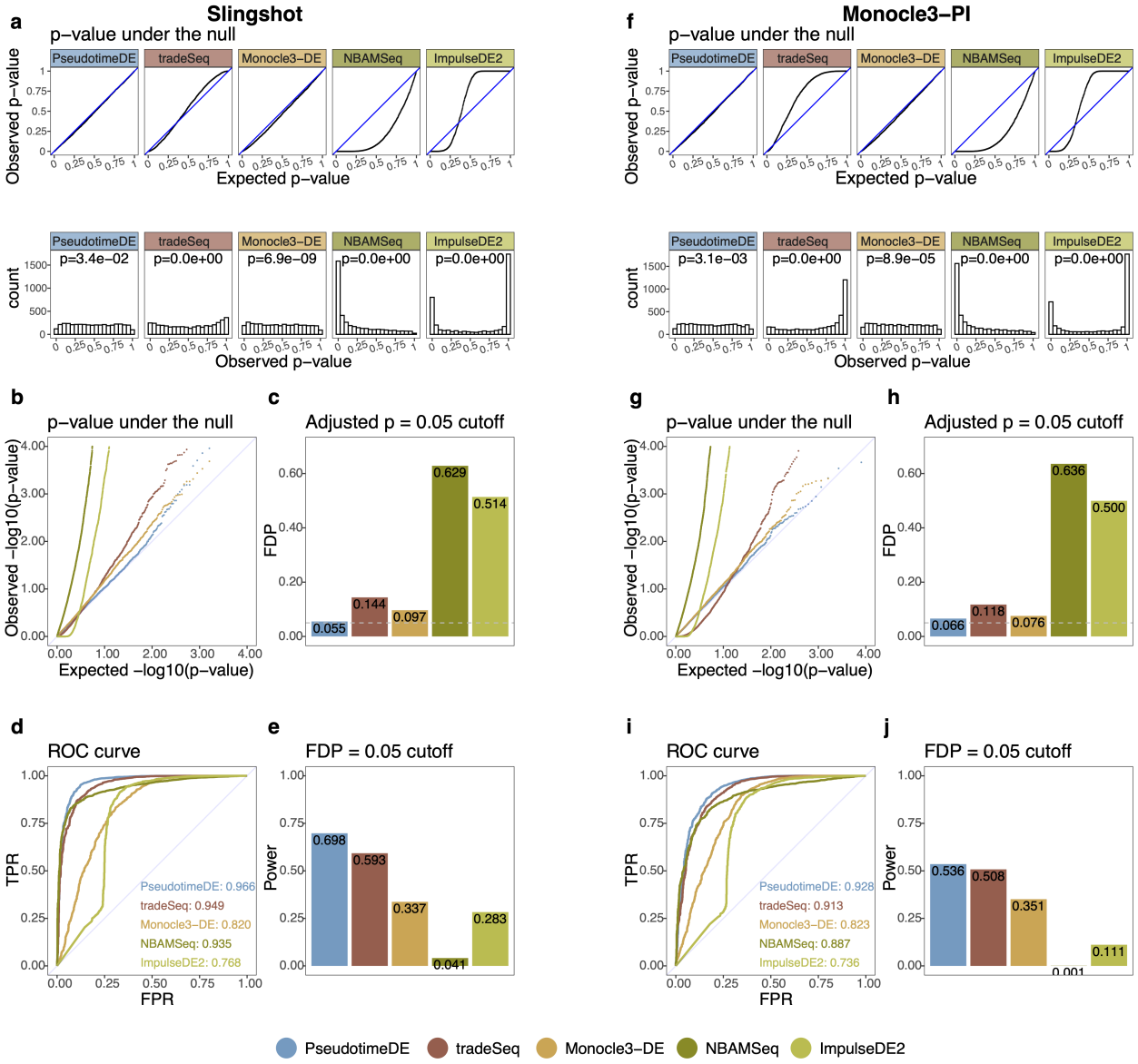
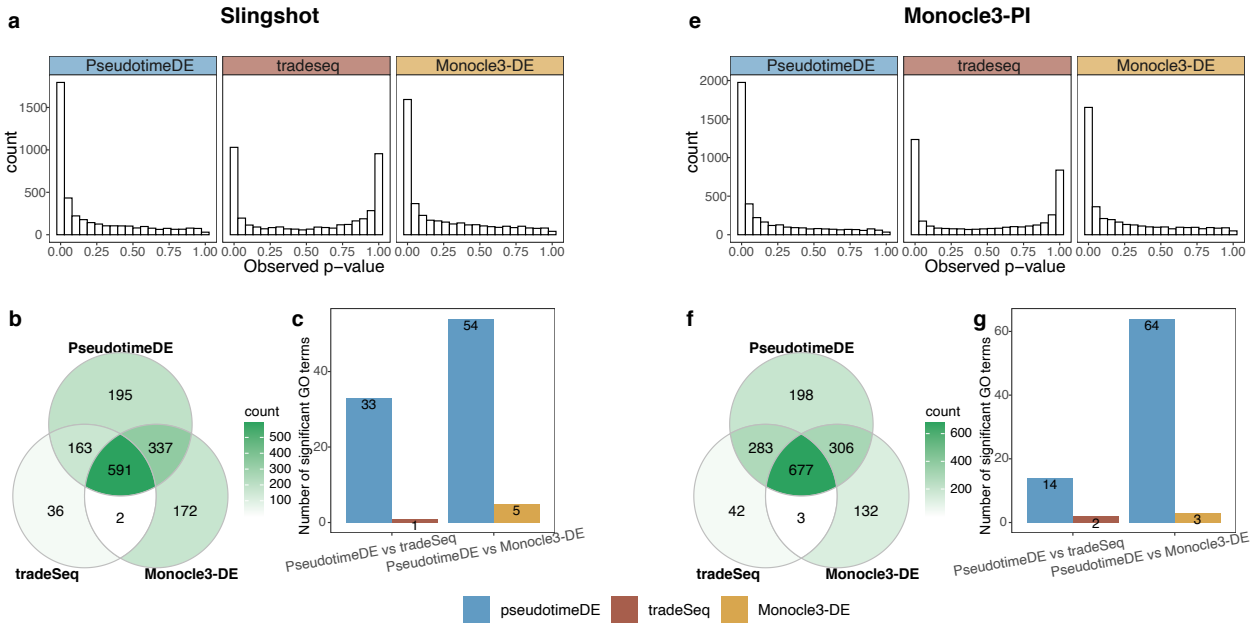


Figure 3.3: PseudotimeDE outperforms four state-of-the-art methods (tradeSeq, Monocle3-DE, NBAMSeq, and ImpulseDE2) for identifying DE genes along cell pseudotime.

Left panels (a)–(e) are based on pseudotime inferred by Slingshot; right panels (f)–(j) are based on pseudotime inferred by Monocle3-PI. **(a) & (f)** Distributions of non-DE genes' observed p-values by five DE methods with inferred pseudotime. Top: quantile-quantile plots that compare the empirical quantiles of the observed p-values against the expected quantiles of the Uniform$[0, 1]$ distribution. Bottom: histograms of the observed p-values. The p-values shown on top of histograms are from the Kolmogorov-Smirnov test under the null hypothesis that the distribution is Uniform$[0, 1]$. The larger the p-value, the more uniform the distribution is. Among the five DE methods, PseudotimeDE's observed p-values follow most closely the expected Uniform$[0, 1]$ distribution. **(b) & (g)** Quantile-quantile plots of the same p-values as in (a) and (f) on the negative $\log_{10}$ scale. PseudotimeDE returns better-calibrated small p-values than the other four methods do. **(c) & (h)** FDPs of the five DE methods with the target FDR 0.05 (BH adjusted-$p \leqslant 0.05$). PseudotimeDE yields the FDP closest to 0.05. **(d) & (i)** ROC curves and AUROC values of the five DE methods. PseudotimeDE achieves the highest AUROC. **(e) & (j)** Power of the five DE methods under the FDP = 0.05 cutoff. PseudotimeDE achieves the highest power.



d

PseudotimeDE vs tradeSeq

ID	description	p
GO:0050729	positive regulation of inflammatory response	0.00424
GO:0071222	cellular responsee to lipopolysaccharide	0.00609

PseudotimeDE vs Monocle3-DE

ID	description	p
GO:0006954	inflammatory responsee	0.00032
GO:0050729	positive regulation of inflammatory response	0.00479
GO:0006955	immune responsee	0.00541
GO:0042742	defense responsee to bacterium	0.00573

h

PseudotimeDE vs tradeSeq

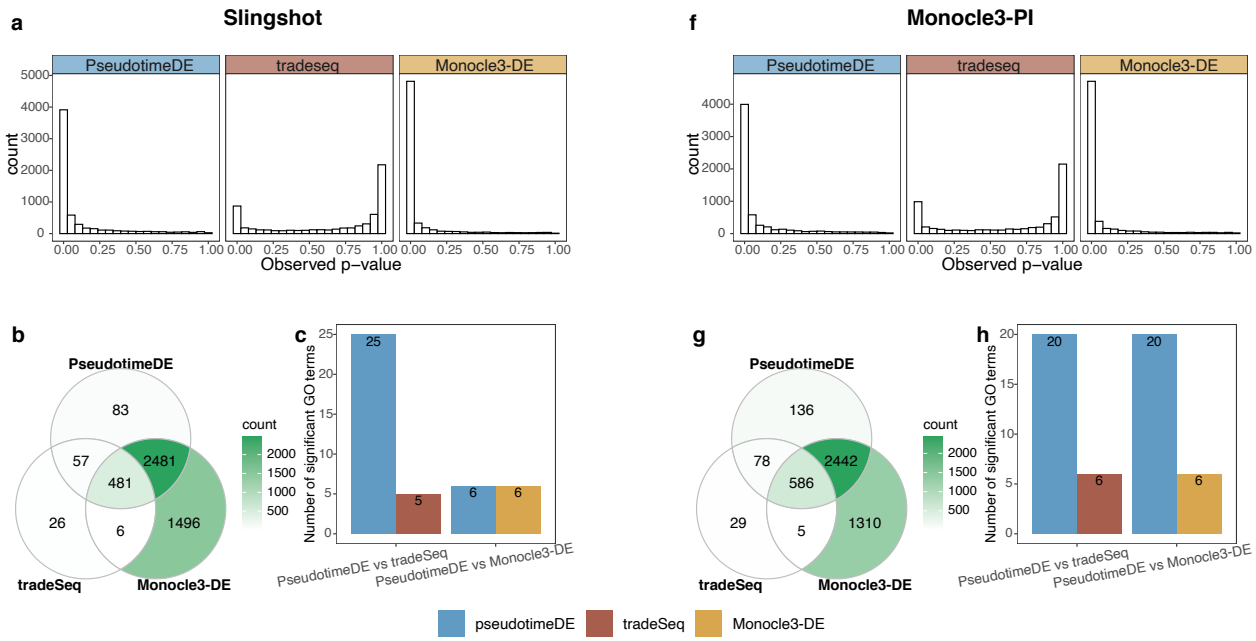
ID	description	p
GO:0002250	adaptive immune response	0.00812
GO:0002673	regulation of acute inflammatory response	0.00955

PseudotimeDE vs Monocle3-DE

ID	description	p
GO:0045087	innate immune response	0.00016
GO:0009617	response to bacterium	0.00039
GO:0032496	response to lipopolysaccharide	0.00049
GO:0006955	immune response	0.00054
GO:0061844	antimicrobial humoral immune response	0.00060
GO:0071222	cellular response to lipopolysaccharide	0.00075
GO:0002250	adaptive immune response	0.00179
GO:0042742	defense response to bacterium	0.00391
GO:0050829	defense response to Gram−negative bacterium	0.00418

Figure 3.4: Application of PseudotimeDE, tradeSeq, and Monocle3-DE to the LPS-dendritic cell dataset.

Left panels (a)–(d) are based on pseudotime inferred by Slingshot; right panels (e)–(h) are based on pseudotime inferred by Monocle3-PI. **(a) & (e)** Histograms of all genes' p-values by the three DE methods. The bimodal distributions of tradeSeq's p-values suggest a violation of the requirement that p-values follow the Uniform$[0, 1]$ distribution under the null hypothesis. **(b) & (f)** Venn plots showing the overlaps of the significant DE genes (BH adjusted-$p \leqslant 0.01$) identified by the three DE methods. PseudotimeDE's DE genes nearly include tradeSeq's. **(c) & (g)** Numbers of GO terms enriched ($p < 0.01$) in the significant DE genes specifically found by PseudotimeDE or tradeSeq/Monocle3-DE in pairwise comparisons between PseudotimeDE and tradeSeq/Monocle3-DE in (b) & (f). Many more GO terms are enriched in the PseudotimeDE-specific DE genes than in the tradeSeq- or Monocle3-DE-specific ones. **(d) & (h)** Example GO terms enriched in the Pseudotime-specific DE genes in (c) & (g). Many of these terms are related to LPS, immune process, and defense to bacterium.



d

PseudotimeDE vs tradeSeq

ID	description	p
GO:0045747	positive regulation of Notch signaling pathway	0.00162
GO:0006486	protein glycosylation	0.00221
GO:0045746	negative regulation of Notch signaling pathway	0.00326
GO:0009791	post−embryonic development	0.00478
GO:0031018	endocrine pancreas development	0.00490
GO:0098609	cell−cell adhesion	0.00975

PseudotimeDE vs Monocle3-DE

ID	description	p
GO:0035773	insulin secretion involved in cellular response	0.0056

i

PseudotimeDE vs tradeSeq

ID	description	p
GO:0009791	post−embryonic development	0.00082
GO:0098609	cell−cell adhesion	0.00351

PseudotimeDE vs Monocle3-DE

ID	description	p
GO:0046503	glycerolipid catabolic process	0.00076
GO:0006639	acylglycerol metabolic process	0.00489
GO:0001953	negative regulation of cell−matrix adhesion	0.00623

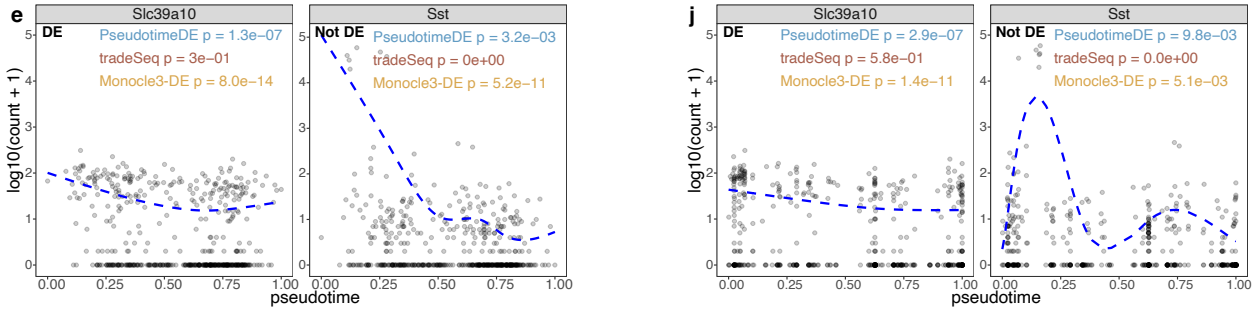


Figure 3.5: Application of PseudotimeDE, tradeSeq, and Monocle3-DE to the pancreatic beta cell maturation dataset.

Left panels (a)–(e) are based on pseudotime inferred by Slingshot; right panels (f)–(j) are based on pseudotime inferred by Monocle3-PI. **(a) & (f)** Histograms of all genes' p-values by the three DE methods. The bimodal distributions of tradeSeq's p-values suggest a violation of the requirement that p-values follow the Uniform$[0, 1]$ distribution under the null hypothesis. **(b) & (g)** Venn plots showing the overlaps of the significant DE genes (BH adjusted-$p \leqslant 0.01$) identified by the three DE methods. PseudotimeDE's DE genes nearly include tradeSeq's. **(c) & (h)** Numbers of GO terms enriched ($p < 0.01$) in the significant DE genes specifically found by PseudotimeDE or tradeSeq/Monocle3-DE in pairwise comparisons between PseudotimeDE and tradeSeq/Monocle3-DE in (b) & (g). Many more GO terms are enriched in the PseudotimeDE-specific DE genes than in the tradeSeq- or Monocle3-DE-specific ones. **(d) & (i)** Example GO terms enriched in the Pseudotime-specific DE genes in (c) & (h). Many of these terms are related to related to insulin, beta cell regulation, and pancreas development. **(e) & (j)** Two examples genes: *Slc39a10* (DE) and *Sst* (non-DE). For *Slc39a10*, both PseudotimeDE and Monocle3-DE yield small p-values ($p < 1e-6$), while tradeSeq does not ($p > 0.1$). For *Sst*, PseudotimeDE yields larger p-values than tradeSeq and Monocle3-DE do. Dashed blue lines are the fitted curves by NB-GAM.

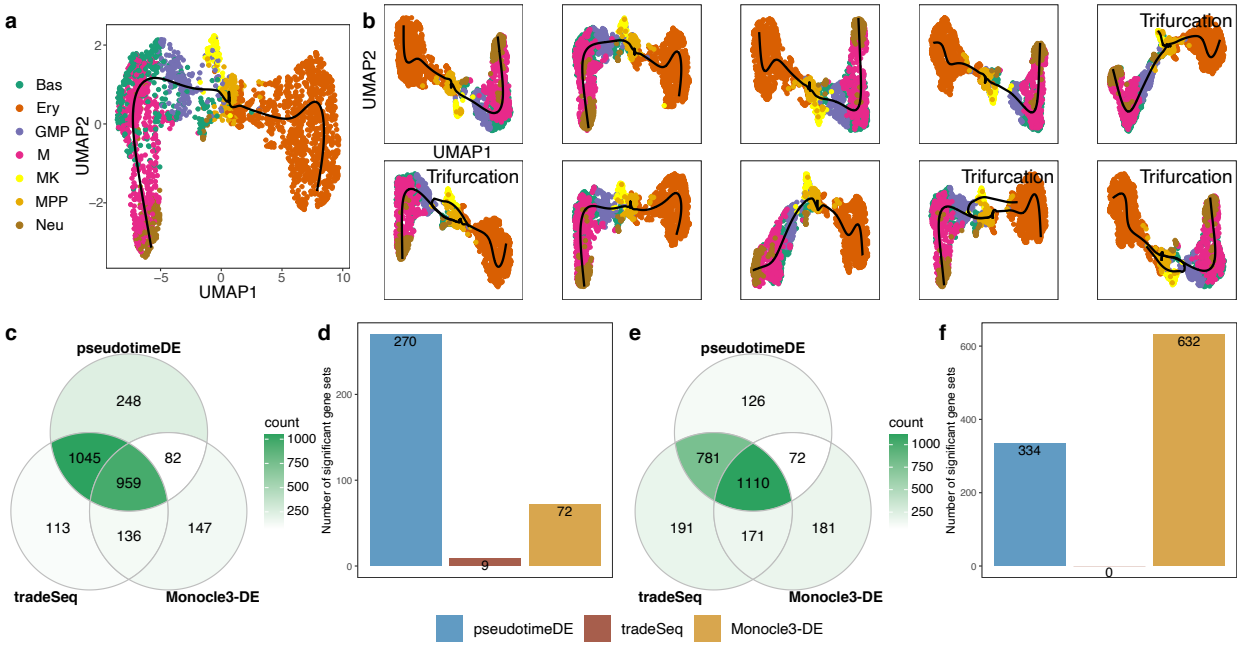


Figure 3.6: Application of PseudotimeDE, tradeSeq, and Monocle3-DE to the mouse bone marrow dataset.

(a) UMAP visualization and inferred pseudotime by Slingshot. Pre-defined cell types are marked by colors. Slingshot returns a bifurcation topology, denoted as lineage 1 (left) and lineage 2 (right). **(b)** UMAP visualization and inferred pseudotime by Slingshot on ten random subsamples. Four out of ten subsamples do not yield bifurcation topology but trifurcation topology, where the third lineage mainly contains the cell type "MK" and was reported in [73]. **(c)** Histograms of all genes' p-values calculated by the three DE methods in the first lineage. **(d)** Histograms of all genes' p-values calculated by the three DE methods in the second lineage. **(e)** Venn plot showing the overlaps of the significant DE genes (BH adjusted-$p \leqslant 0.01$) identified by the three DE methods in lineage 1. PseudotimeDE and tradeSeq share 77.6% (Jaccard index) DE genes. **(f)** Numbers of enriched gene sets ($q < 0.25$) by GSEA using the p-values in lineage 1 by the three DE methods. Although the DE genes are similar in (e), PseudotimeDE yields 270 enriched gene sets, while tradeSeq only yields 9. **(g)** Venn plot showing the overlaps of the significant DE genes (BH adjusted-$p \leqslant 0.01$) identified by the three DE methods in lineage 2. Similar to lineage 1 in (g), PseudotimeDE and tradeSeq share 77.2% (Jaccard index) DE genes. **(h)** Numbers of enriched gene sets ($q < 0.25$) by GSEA using the p-values in lineage 2 by the three DE methods. PseudotimeDE and Monocle3-DE yield hundreds of enriched gene sets, while tradeSeq does not yield any enriched gene sets.

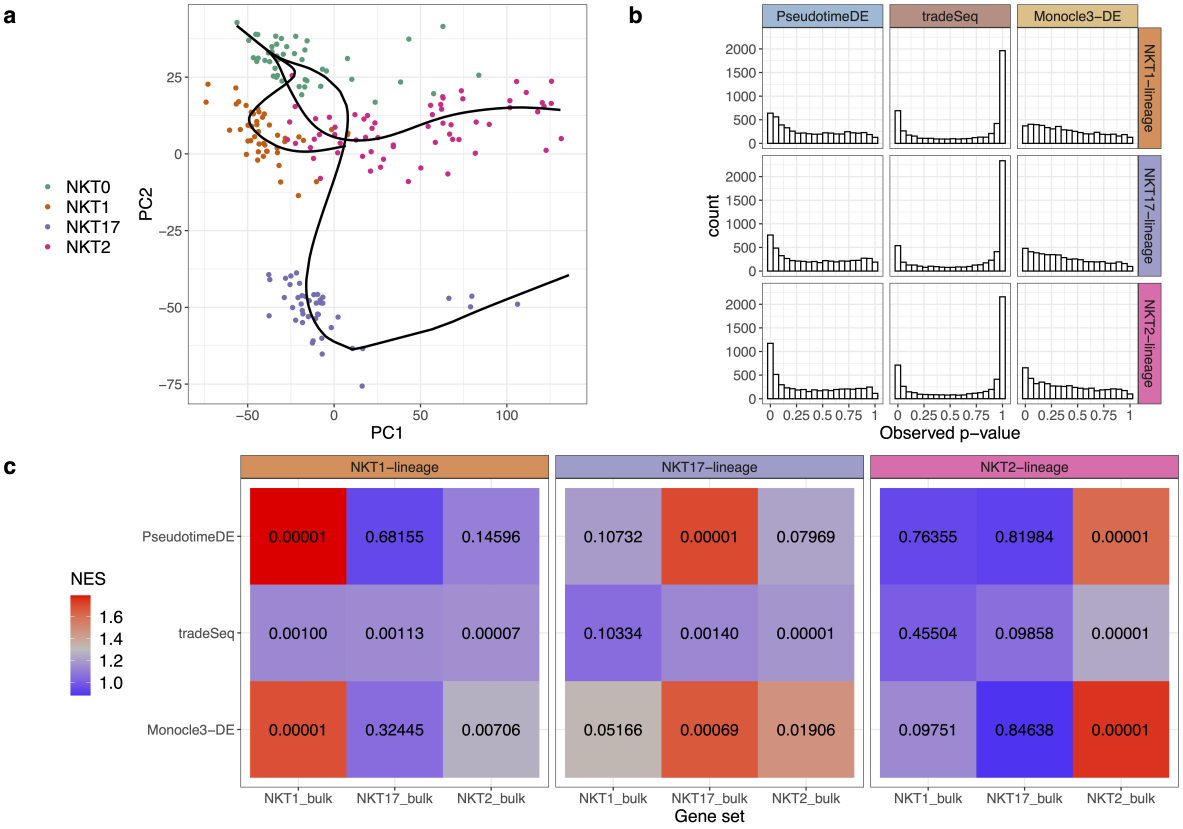


Figure 3.7: Application of PseudotimeDE, tradeSeq, and Monocle3-DE to the natural killer T cell dataset.

(a) PCA visualization and inferred pseudotime by Slingshot. Pre-defined NKT subtypes are marked by colors. Slingshot returns a trifurcation topology, where the three lineages are NKT0 to NKT1, NKT0 to NKT17, and NKT0 to NKT2. **(b)** Histograms of all genes' p-values in the three lineages calculated by the three DE methods. **(c)** Heatmaps of normalized enrichment scores (NESs, marked by colors) and their corresponding p-values (in numbers) from the GSEA. Each NES value and its corresponding p-value are calculated for each DE method and each lineage, based on the p-values of a DE method for a lineage and that lineage's DE genes found from bulk RNA-seq data, denoted by "NKT1 bulk", "NKT17 bulk," or "NKT2 bulk" [106]. Note that among the three DE methods, PseudotimeDE outputs p-values that best agree with the lineage-specific DE genes from bulk data and thus most distinguish the three lineages. For instance, for the NKT1 lineage, PseudotimeDE's small p-values are enriched in the "NKT1 bulk" gene set only, while tradeSeq and Monocle3-DE have small p-values enriched in at least two lineage-specific DE gene sets.

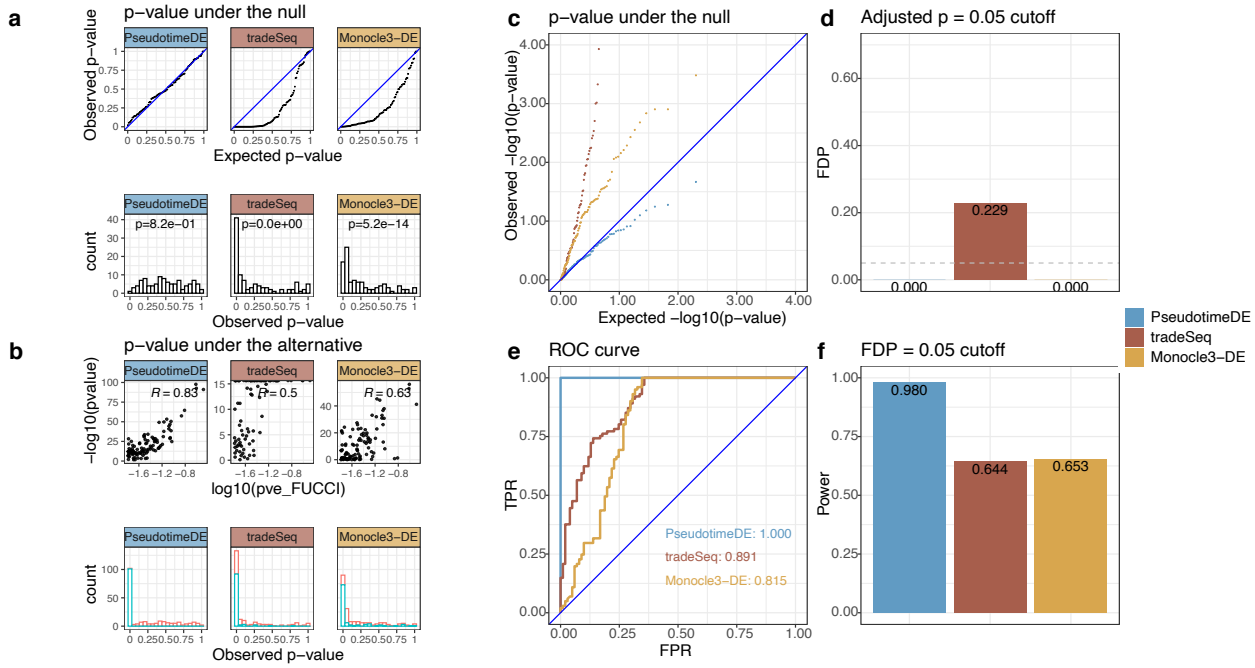


Figure 3.8: Application of PseudotimeDE, tradeSeq, and Monocle3-DE to the cell cycle phase dataset.

(a) Distributions of non-DE genes' p-values by three DE methods with inferred pseudotime. Top: quantile-quantile plots that compare the empirical quantiles of non-DE genes' p-values against the expected quantiles of the Uniform$[0, 1]$ distribution. Bottom: histograms of non-DE genes' p-values. The p-values shown on top of histograms are from the Kolmogorov-Smirnov test under the null hypothesis that the distribution is Uniform$[0, 1]$. The larger the p-value, the more uniform the distribution is. Among the three DE methods, PseudotimeDE's p-values follow most closely the expected Uniform$[0, 1]$ distribution. **(b)** Distributions of DE genes' p-values by three DE methods with inferred pseudotime. Top: scatter plots of DE genes' p-values against the proportions of variance explained (PVE), which measure the strengths of genes' inferred cyclic trends in the original study [107]. PseudotimeDE's p-values ($-\log_{10}$ transformed) have the highest correlation with the PVE, indicating that PseudotimeDE identifies the genes with the strongest cyclic trends as the top DE genes. Bottom: histograms of all genes' p-values. Blue and red colors represent the p-values of DE genes and non-DE genes (same as in (a) bottom), respectively. PseudotimeDE yields the best separation of the two gene groups' p-values. **(c)** Quantile-quantile plots of the same p-values as in (a) on the negative $\log_{10}$ scale. PseudotimeDE returns the best-calibrated p-values. **(d)** FDPs of the three DE methods with the target FDR 0.05 (BH adjusted-$p \leqslant 0.05$). **(e)** ROC curves and AUROC values of the three DE methods. PseudotimeDE achieves the highest AUROC. **(f)** Power of the three DE methods under the FDP $= 0.05$ cutoff. PseudotimeDE achieves the highest power.

3.8 Supplementary materials

3.8.1 Pseudotime inference methods

We apply two state-of-the-art methods, Slingshot and Monocle3-PI, to inferring the cell pseudotime of each dataset. For single-lineage data, we specify the start cluster in Slingshot and the start node in Monocle3-PI. For bifurcation/trifurcation data, we specify the start cluster/node and the end clusters/nodes in Slingshot/Monocle3-PI. By default, the dimensionality reduction methods used for pseudotime inference are PCA and UMAP for Slingshot and Monocle3-PI, respectively. The R Bioconductor package `slingshot` (version 1.4.0) and the R package `monocle3` (version 0.2.0) are used.

3.8.2 DE analysis methods

We compare PseudotimeDE with four existing methods for identifying DE genes along pseudotime/time-course from scRNA-seq data (tradeSeq and Monocle3-DE) or bulk RNA-seq data (ImpulseDE2 and NBAMSeq). All these methods take a count matrix $\mathbf{Y}$ and a pseudotime vector $\mathbf{T}$ as input, and they return a p-value for each gene. For tradeSeq, we use the functions `fitGAM` and `associationTest` (https://statomics.github.io/tradeSeq/articles/tradeSeq.html). The number of knots parameter K in tradeSeq is chosen by 100 random genes based on the tradeSeq vignette. For Monocle3-DE, we use the function `fit_models` (https://cole-trapnell-lab.github.io/monocle3/docs/differential/). Since ImpulseDE2 cannot be applied to scRNA-seq data directly, we follow the modified implementation of ImpulseDE2 in the tradeSeq paper (https://github.com/statOmics/tradeSeqPaper). The R Bioconductor packages `tradeSeq` (version 1.3.15), `monocle3` (version 0.2.0), `ImpulseDE2` (version 1.10.0), and `NBAMSeq` (version 1.10.0) are used.

3.8.3 Functional (gene ontology and gene-set enrichment) analyses

We use the R package `topGO` (version 2.38.1) [117] to perform the gene-ontology (GO) enrichment analysis on identified DE genes. We use the R package `clusterProfiler` (version 3.14.3) [118] to perform the gene-set enrichment analysis (GSEA) analysis on ranked gene lists, where genes in each list are ranked by their ranking sores defined as $-\log_{10}$ transformed p-values (the gene with the smallest p-value is ranked the top); p-values that are exactly zeros are replaced by one-tenth of the smallest non-zero p-value. If unspecified, the GO terms are "biological process (BP)" terms.

3.8.4 Simulation study

We use the R package `dyntoy` (0.9.9) to generate single-lineage data and bifurcation data. For single-lineage data, we generate three datasets with increasing dispersion levels (low dispersion, medium dispersion, and high dispersion). Each single-lineage dataset consists of 500 cells and 5000 genes (with 20% as DE genes). For bifurcation data, we use the medium dispersion level. The bifurcation dataset consists of 750 cells and 5000 genes (with 20% as DE genes).

3.8.5 Case studies

LPS-dendritic cell dataset: this Smart-seq dataset contains primary mouse dendritic cells (DCs) stimulated with lipopolysaccharide (LPS) [101], available at Gene Expression Omnibus (GEO) under accession ID GSE45719. In our analysis, we use the the cells from 1h, 2h, 4h, and 6h in the pre-processed data from the study that benchmarked pseudotime inference methods [21]. After the genes with $> 90\%$ zeros are removed, the final dataset consists of 4016 genes and 390 cells, which are expected to be in a single lineage. When applying tradeSeq, we use the recommended ZINB-WaVE [28] + tradeSeq procedure to account for potential zero-inflation. The R Bioconductor package `zinbwave` (version 1.8.0) is used.

Pancreatic beta cell maturation dataset: this Smart-seq2 dataset measures the maturation process of mouse pancreatic beta cells [103], available at GEO under accession ID GSE87375. We use the cells from cell type "beta" in the pre-processed data from the study that benchmarked pseudotime inference methods [21]. After the genes with $> 90\%$ zeros are removed, the final dataset consists of 6121 genes and 497 cells, which are expected to be in a single lineage. When applying tradeSeq, we use the recommended ZINB-WaVE + tradeSeq procedure to account for potential zero-inflation. The R Bioconductor package `zinbwave` (version 1.8.0) is used.

Mouse bone marrow dataset: this MARS-seq dataset contains myeloid progenitors in mouse bone marrow [55], available at GEO under accession ID GSE72859. We use the pre-processed data provided by the tradeSeq vignette. After the genes with $> 90\%$ zeros are removed, the final dataset consists of 3004 genes and 2660 cells. We follow the procedure of combining UMAP and Slingshot to infer pseudotime as described in tradeSeq paper [77]

Natural killer T cell dataset: this Smart-seq2 dataset measures four natural killer T cell (NKT cell) subtypes in mouse [106], available at GEO under accession ID GSE74597. We use the pre-processed data from the study that benchmarked pseudotime inference methods [21]. After the genes with $> 90\%$ zeros are removed, the final dataset consists of 5270 genes and 197 cells, which are expected to have three lineages. We use PCA + Slingshot to infer the pseudotime. When applying tradeSeq, we use the recommended ZINB-WaVE + tradeSeq procedure to account for potential zero-inflation. The R Bioconductor package `zinbwave` (version 1.8.0) is used.

Cell cycle phase dataset: this Fluidigm protocol dataset measures human induced pluripotent stem cells (iPSCs) [107]. The iPSCs are FUCCI-expressing so that their cell cycle phases can be tracked. The authors also developed an R package `peco` for predicting cell cycle phases from single-cell gene expression data. We use the example dataset provided by `peco`, which consists of 101 known cell cycle-related genes (DE genes). To construct null cases, we randomly shuffle the 101 DE genes' expression levels across cells to create 101 non-DE genes. The final dataset consists of 202 genes and 888 cells. We use the R package `peco` (version 1.1.21) to infer cell cycle phases.

3.8.6 Supplementary figures

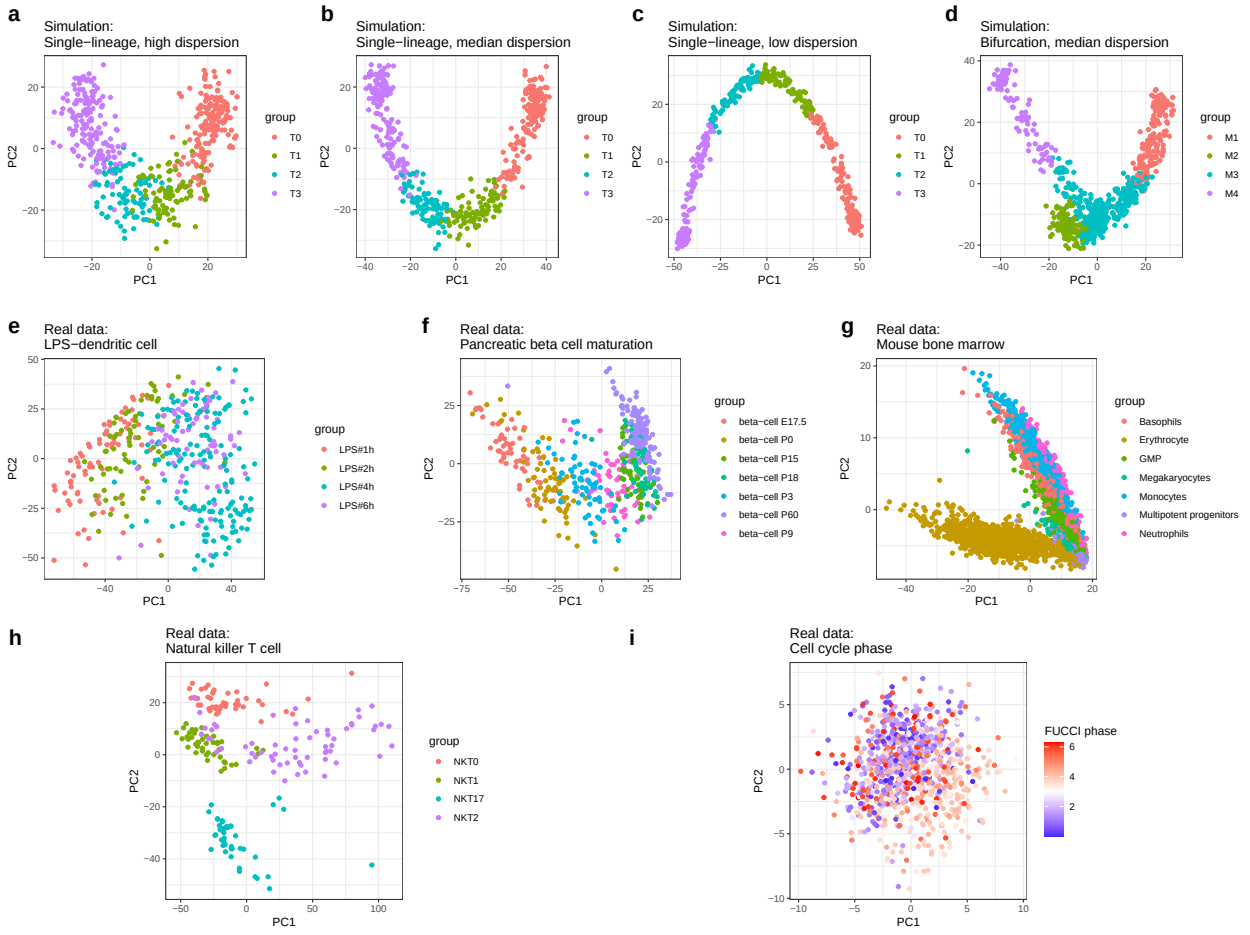


Figure 3.9: PCA visualization of datasets.

PCA visualization of all synthetic and real datasets used in this paper. Panels (a)–(d) are synthetic datasets, where the groups correspond to time points. Panels (e)–(h) are real datasets, where the groups correspond to time points in (e) & (f) or annotated cell types in (g) & (h). Panel (i) is a real dataset with external cell cycle information, where the FUCCI phase indicates experimentally measured cell cycle phase.

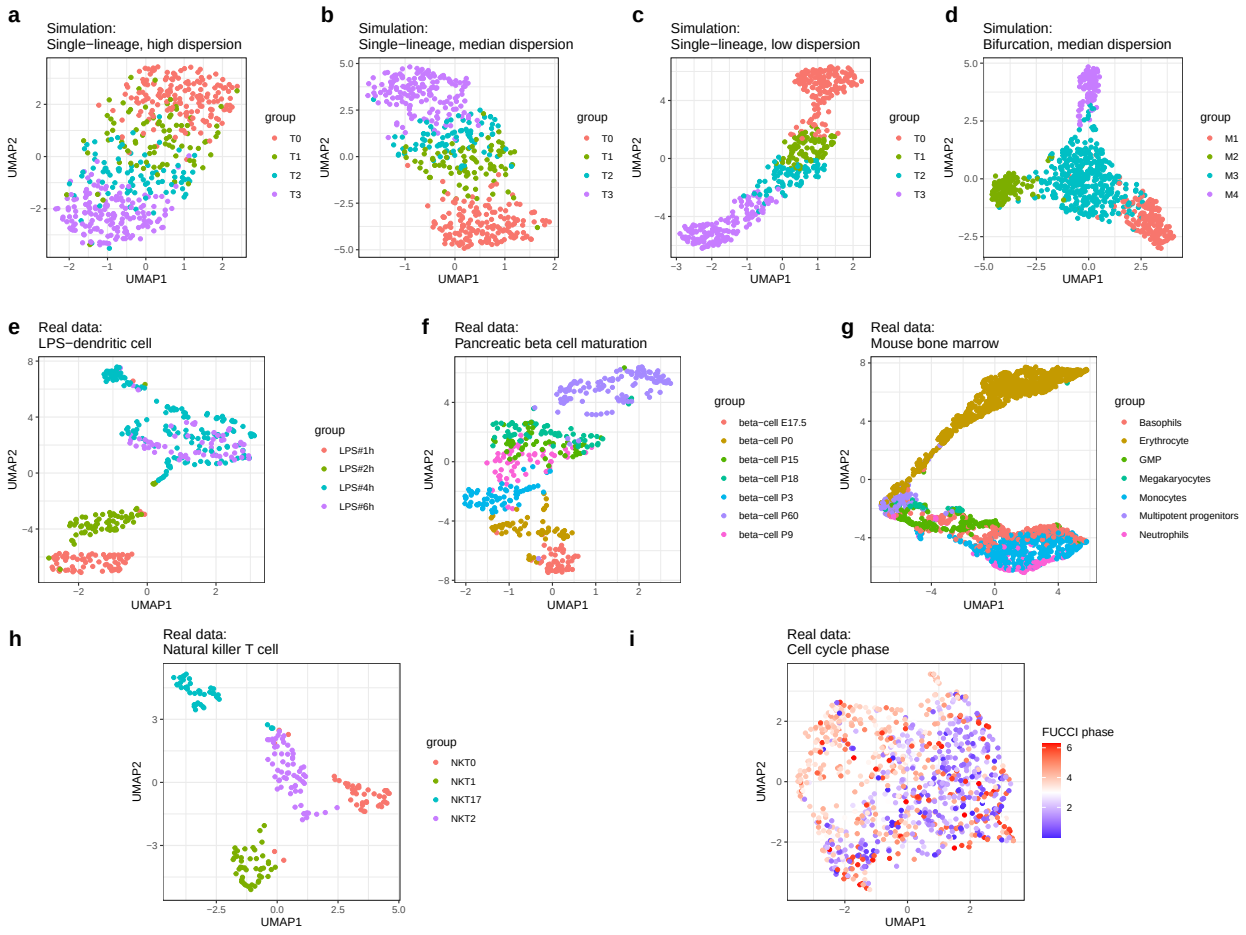


Figure 3.10: UMAP visualization of datasets.

UMAP visualization of all synthetic and real datasets used in this paper. Panels (a)–(d) are synthetic datasets, where the groups correspond to time points. Panels (e)–(h) are real datasets, where the groups correspond to time points in (e) & (f) or annotated cell types in (g) & (h). Panel (i) is a real dataset with external cell cycle information, where the FUCCI phase indicates experimentally measured cell cycle phase.

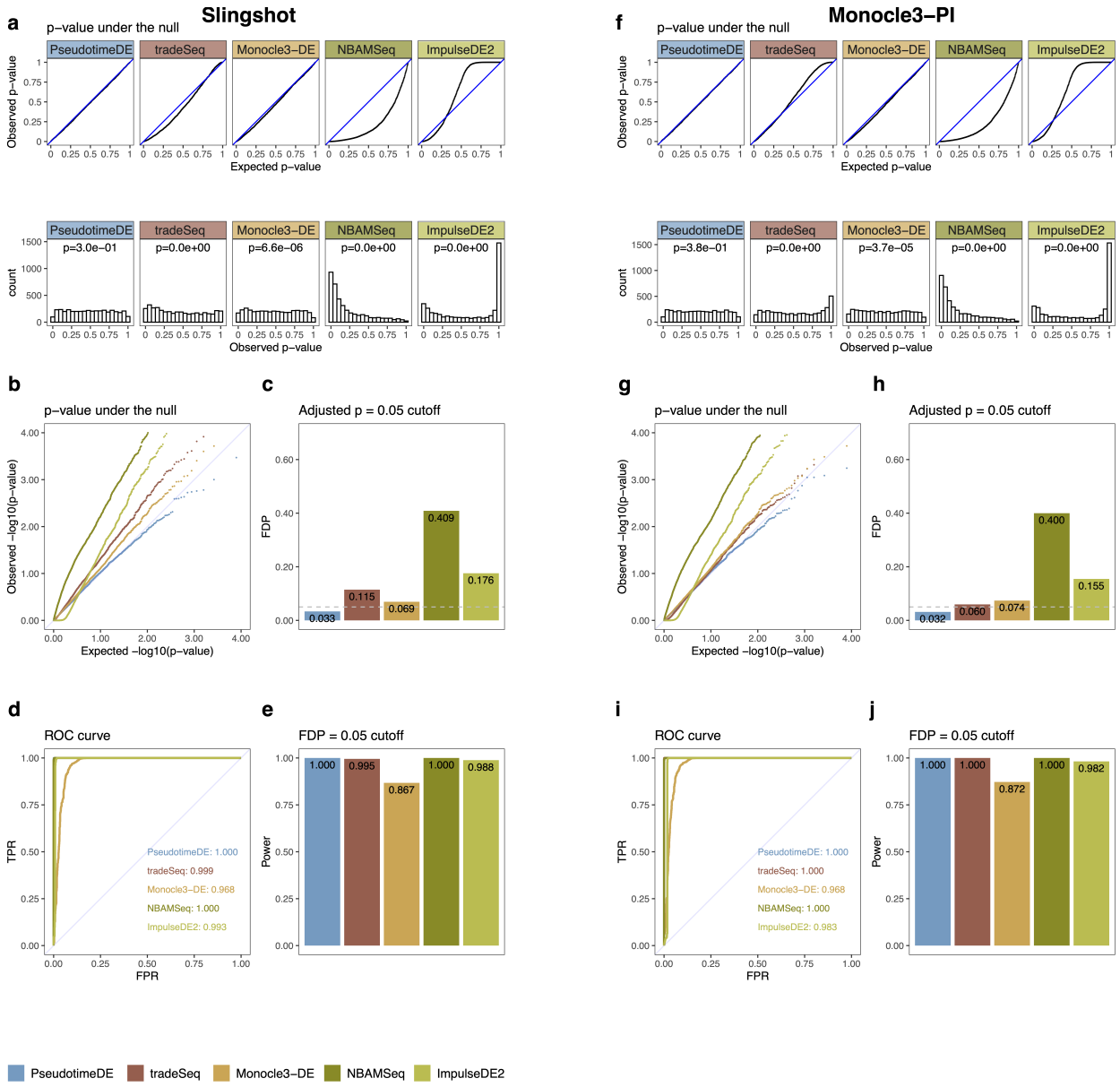


Figure 3.11: Comparison of five methods (PseudotimeDE, tradeSeq, Monocle3-DE, NBAMSeq, ImpulseDE2) for identifying DE genes along cell pseudotime on synthetic single-lineage data with low dispersion.

Left panels (a)–(e) are based on pseudotime inferred by Slingshot; right panels (f)–(j) are based on pseudotime inferred by Monocle3-PI. **(a) & (f)** Distributions of non-DE genes' observed p-values by five DE methods with inferred pseudotime. Top: quantile-quantile plots that compare the empirical quantiles of the observed p-values against the expected quantiles of the Uniform$[0, 1]$ distribution. Bottom: histograms of the observed p-values. The p-values shown on top of histograms are from the Kolmogorov-Smirnov test under the null hypothesis that the distribution is Uniform$[0, 1]$. The larger the p-value, the more uniform the distribution is. Among the five DE methods, PseudotimeDE's observed p-values follow most closely the expected Uniform$[0, 1]$ distribution. **(b) & (g)** Quantile-quantile plots of the same p-values as in (a) and (f) on the negative $\log_{10}$ scale. PseudotimeDE returns better-calibrated small p-values than the other four methods do. **(c) & (h)** FDPs of the five DE methods with the target FDR 0.05 (BH adjusted-$p \leqslant 0.05$). PseudotimeDE yields the FDP below 0.05 while other methods do not. **(d) & (i)** ROC curves and AUROC values of the five DE methods. Since the dispersion of data is extremely low, all methods achieves high AUROC. **(e) & (j)** Power of the five DE methods under the FDP = 0.05 cutoff. Due to the same low dispersion reason, all methods achieves high power.

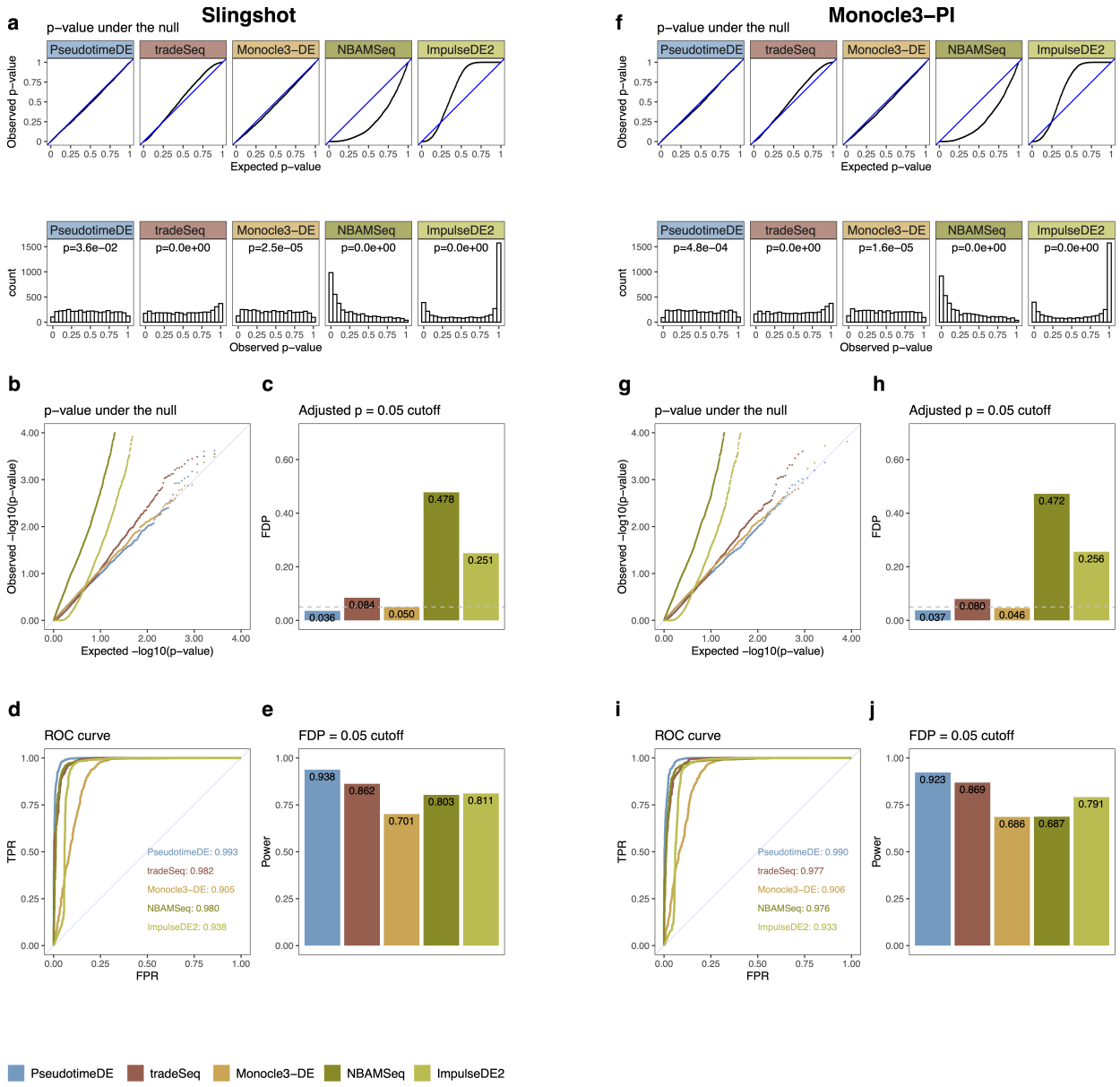


Figure 3.12: Comparison of five methods (PseudotimeDE, tradeSeq, Monocle3-DE, NBAMSeq, ImpulseDE2) for identifying DE genes along cell pseudotime on synthetic single-lineage data with median dispersion.

Left panels (a)–(e) are based on pseudotime inferred by Slingshot; right panels (f)–(j) are based on pseudotime inferred by Monocle3-PI. **(a) & (f)** Distributions of non-DE genes' observed p-values by five DE methods with inferred pseudotime. Top: quantile-quantile plots that compare the empirical quantiles of the observed p-values against the expected quantiles of the Uniform$[0, 1]$ distribution. Bottom: histograms of the observed p-values. The p-values shown on top of histograms are from the Kolmogorov-Smirnov test under the null hypothesis that the distribution is Uniform$[0, 1]$. The larger the p-value, the more uniform the distribution is. Among the five DE methods, PseudotimeDE's observed p-values follow most closely the expected Uniform$[0, 1]$ distribution. **(b) & (g)** Quantile-quantile plots of the same p-values as in (a) and (f) on the negative $\log_{10}$ scale. PseudotimeDE returns better-calibrated small p-values than the other four methods do. **(c) & (h)** FDPs of the five DE methods with the target FDR 0.05 (BH adjusted-$p \leqslant 0.05$).PseudotimeDE yields the FDP below 0.05 while other methods do not. **(d) & (i)** ROC curves and AUROC values of the five DE methods. PseudotimeDE achieves the highest AUROC. **(e) & (j)** Power of the five DE methods under the FDP = 0.05 cutoff. PseudotimeDE achieves the highest power.

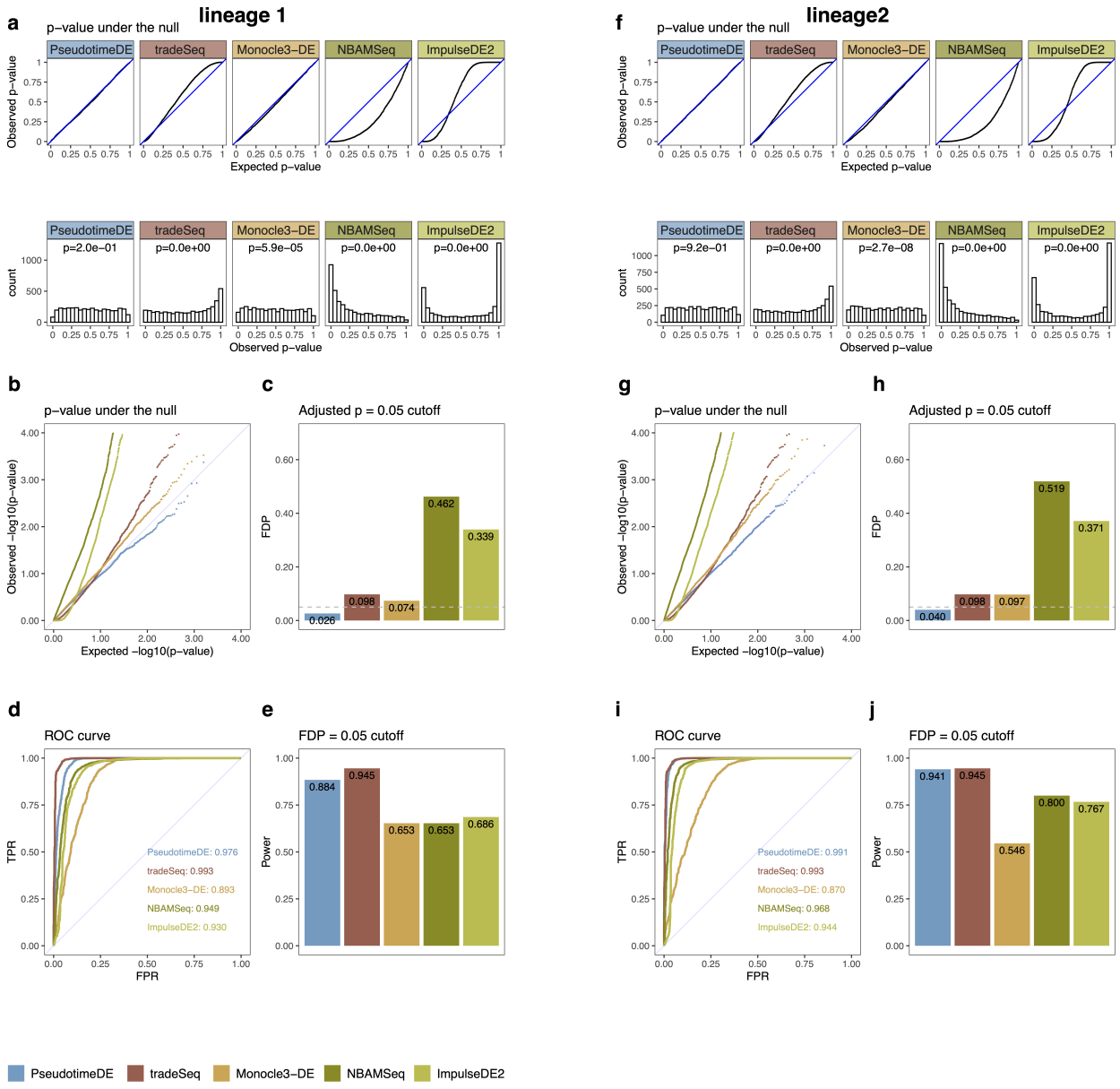


Figure 3.13: Comparison of five methods (PseudotimeDE, tradeSeq, Monocle3-DE, NBAMSeq, ImpulseDE2) for identifying DE genes along cell pseudotime on synthetic bifurcation data.

Pseudotime is inferred by Slingshot. Left panels (a)–(e) are based on lineage 1 of two lineages in bifurcation data; right panels (f)–(j) are based on lineage 2. **(a) & (f)** Distributions of non-DE genes' observed p-values by five DE methods with inferred pseudotime. Top: quantile-quantile plots that compare the empirical quantiles of the observed p-values against the expected quantiles of the Uniform$[0, 1]$ distribution. Bottom: histograms of the observed p-values. The p-values shown on top of histograms are from the Kolmogorov-Smirnov test under the null hypothesis that the distribution is Uniform$[0, 1]$. The larger the p-value, the more uniform the distribution is. Among the five DE methods, PseudotimeDE's observed p-values follow most closely the expected Uniform$[0, 1]$ distribution. **(b) & (g)** Quantile-quantile plots of the same p-values as in (a) and (f) on the negative $\log_{10}$ scale. PseudotimeDE returns better-calibrated small p-values than the other four methods do. **(c) & (h)** FDPs of the five DE methods with the target FDR 0.05 (BH adjusted-$p \leqslant 0.05$). PseudotimeDE yields the FDP below 0.05 while other methods do not. **(d) & (i)** ROC curves and AUROC values of the five DE methods. PseudotimeDE achieves the second highest AUROC which is close to the highest value. **(e) & (j)** Power of the five DE methods under the FDP = 0.05 cutoff. PseudotimeDE achieves the second highest power which is close to the highest value.

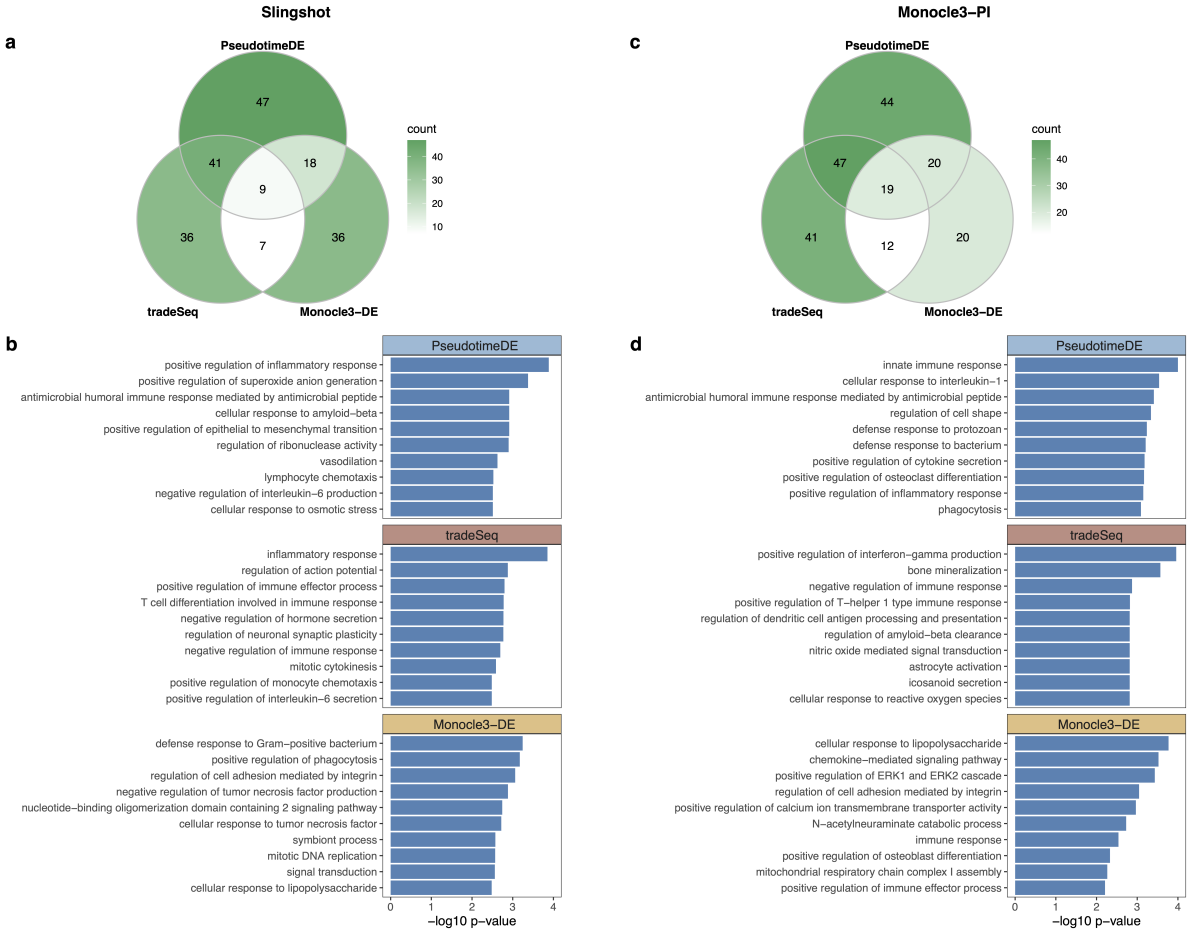
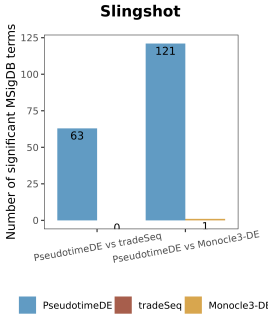


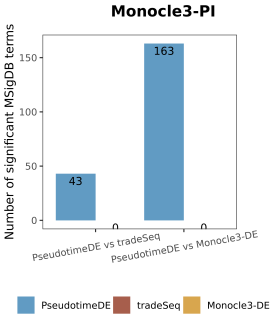
Figure 3.14: GO analysis of DE genes identified in the LPS-dendritic cell dataset.

Left panels (a)–(b) are based on pseudotime inferred by Slingshot; right panels (c)–(d) are based on pseudotime inferred by Monocle3-PI. **(a) & (c)** Numbers of GO terms enriched ($p < 0.01$) in the significant DE genes found by each method in Fig. 3.4 (b) & (f). **(b) & (d)** Top 10 enriched GO terms for each DE method.

a



c



b

PseudotimeDE vs tradeSeq

description	p	p.adj
GSE2706_UNSTIM_VS_2H_LPS_AND_R848_DC_DN	1.2e-05	0.0024
GSE2706_2H_VS_8H_R848_AND_LPS_STIM_DC_DN	2.6e-05	0.0036
GSE14000_4H_VS_16H_LPS_DC_TRANSLATED_RNA_UP	6.4e-05	0.0062

PseudotimeDE vs Monocle3-DE

description	p	p.adj
GSE2706_UNSTIM_VS_2H_LPS_AND_R848_DC_DN	4.1e-07	0.00011
GSE2706_R848_VS_R848_AND_LPS_2H_STIM_DC_DN	3.0e-06	0.00040
GSE14000_UNSTIM_VS_4H_LPS_DC_DN	4.2e-06	0.00051
GSE14000_UNSTIM_VS_4H_LPS_DC_TRANSLATED_RNA_DN	7.1e-06	0.00072
GSE2706_R848_VS_LPS_2H_STIM_DC_DN	5.6e-05	0.00338

d

PseudotimeDE vs tradeSeq

description	p	p.adj
GSE14000_UNSTIM_VS_16H_LPS_DC_TRANSLATED_RNA_UP	4.8e-05	0.0058

PseudotimeDE vs Monocle3-DE

description	p	p.adj
GSE2706_UNSTIM_VS_2H_LPS_DC_DN	7.8e-08	3.1e-05
GSE14000_UNSTIM_VS_4H_LPS_DC_DN	9.9e-07	1.6e-04
GSE2706_UNSTIM_VS_2H_LPS_AND_R848_DC_DN	9.9e-07	1.6e-04
GSE7219_UNSTIM_VS_LPS_AND_ANTI_CD40_STIM_NIK_NFKB2_KO_DC_DN	3.4e-05	2.0e-03
GSE3982_CTRL_VS_LPS_48H_DC_DN	2.4e-04	7.5e-03

Figure 3.15: MSigDB over-representation analysis of DE genes identified in the LPS-dendritic cell dataset.

Left panels (a)–(b) are based on pseudotime inferred by Slingshot; right panels (c)–(d) are based on pseudotime inferred by Monocle3-PI. **(a) & (c)** Numbers of MSigDB terms enriched (BH adjusted-$p < 0.01$) in the significant DE genes specifically found by PseudotimeDE or tradeSeq/Monocle3-DE in pairwise comparisons between PseudotimeDE and tradeSeq/Monocle3-DE in Fig. 3.4 (b) & (f). **(b) & (d)** Example MSigDB terms enriched in the Pseudotime-specific DE genes in (a) & (b). The explanation of terms can be found from MSigDB. All listed terms are related to the response of dendritic cells (DC) stimulated by LPS.

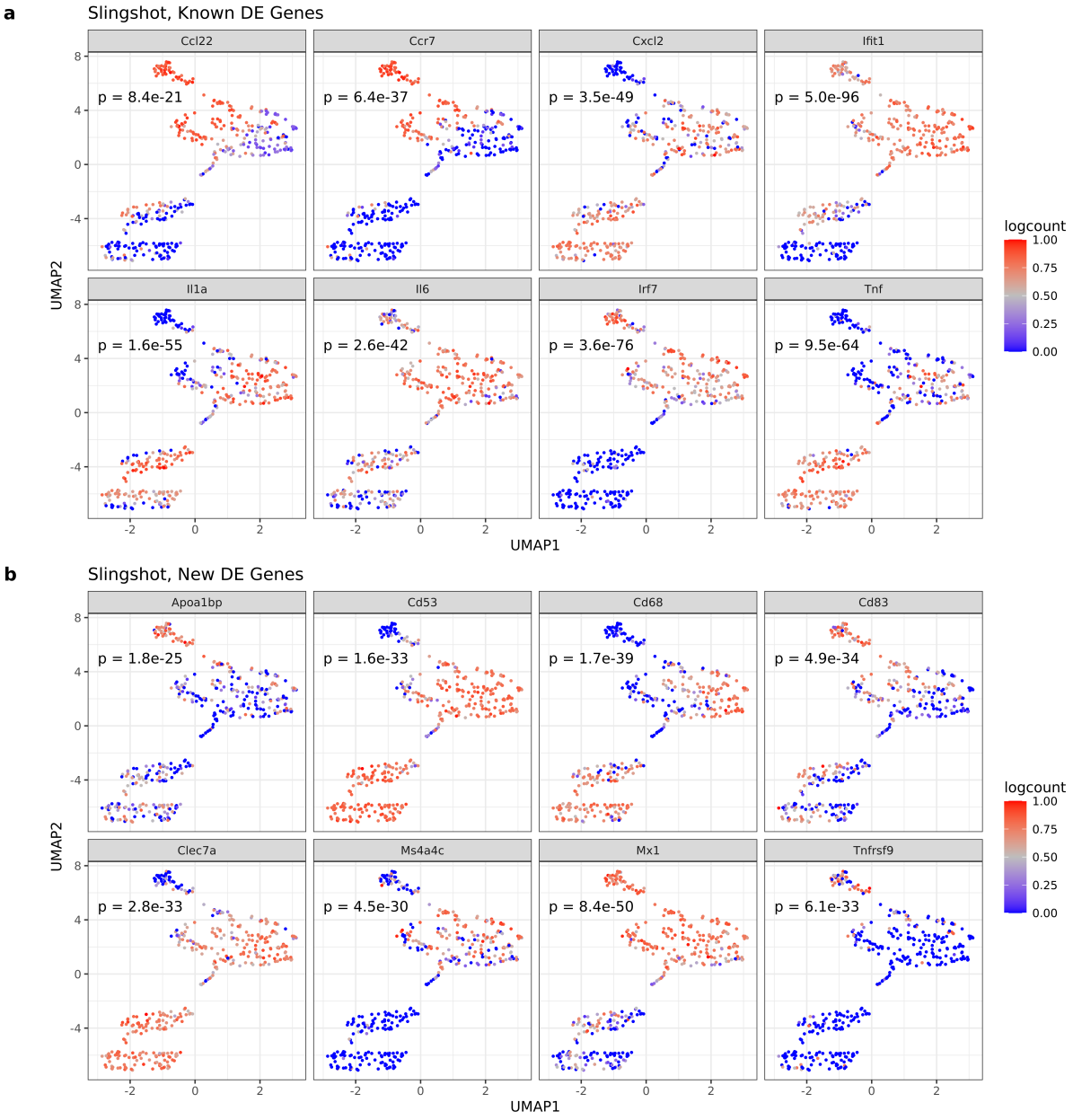


Figure 3.16: UMAP visualization of example DE genes identified by PseudotimeDE, using Slingshot as the pseudotime inference method, in the LPS-dendritic cell dataset.

p-values returned by PseudotimeDE are reported for all the 16 example genes. **(a)** Examples of known DE genes, which have been reported as highly confident DE genes in the original study [101]. **(b)** Examples of new DE genes, which are identified by PseudotimeDE but found as non-DE by either tradeSeq or Monocle3-DE.

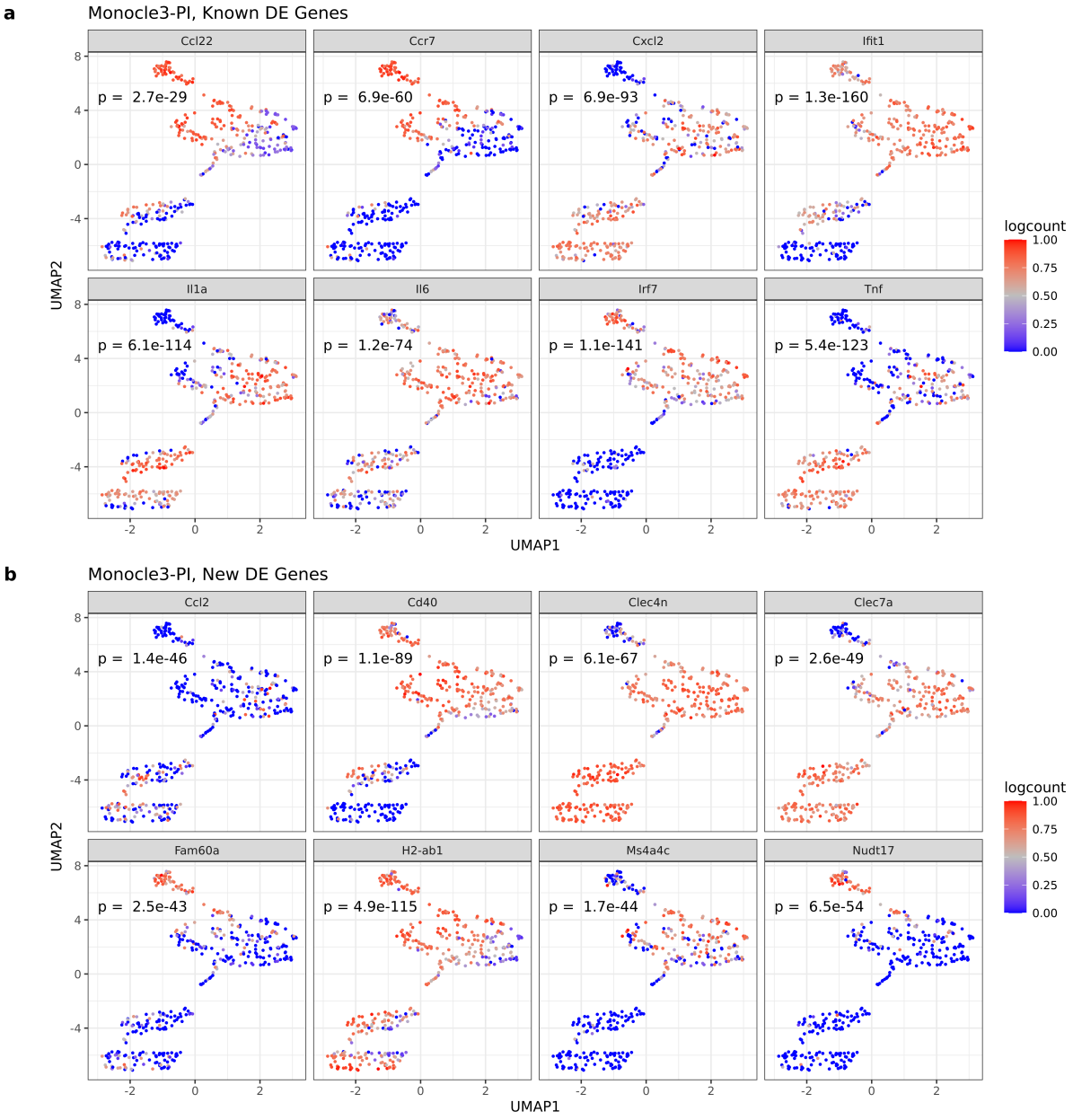


Figure 3.17: UMAP visualization of example DE genes identified by PseudotimeDE, using Monocle3-PI as the pseudotime inference method, in the LPS-dendritic cell dataset.

p-values returned by PseudotimeDE are reported for all the 16 example genes. **(a)** Examples of known DE genes, which have been reported as highly confident DE genes in the original study [101]. **(b)** Examples of new DE genes, which are identified by PseudotimeDE but found as non-DE by either tradeSeq or Monocle3-DE.

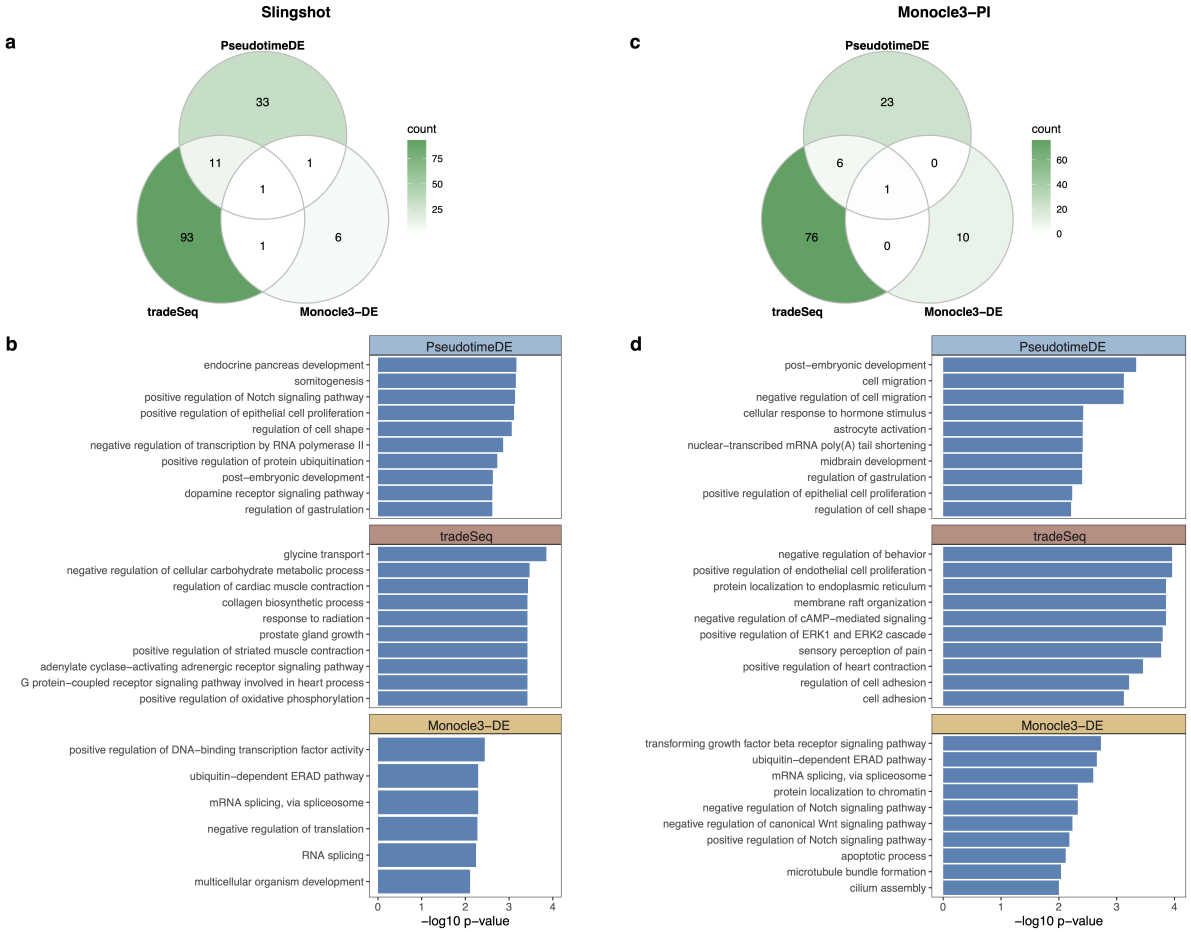


Figure 3.18: GO analysis of DE genes identified in the pancreatic beta cell maturation dataset.

Left panels (a)–(b) are based on pseudotime inferred by Slingshot; right panels (c)–(d) are based on pseudotime inferred by Monocle3-PI. **(a) & (c)** Numbers of GO terms enriched ($p < 0.01$) in the significant DE genes found by each method in Fig. 3.5 (b) & (f). **(b) & (d)** Top 10 significant GO terms from each DE method.

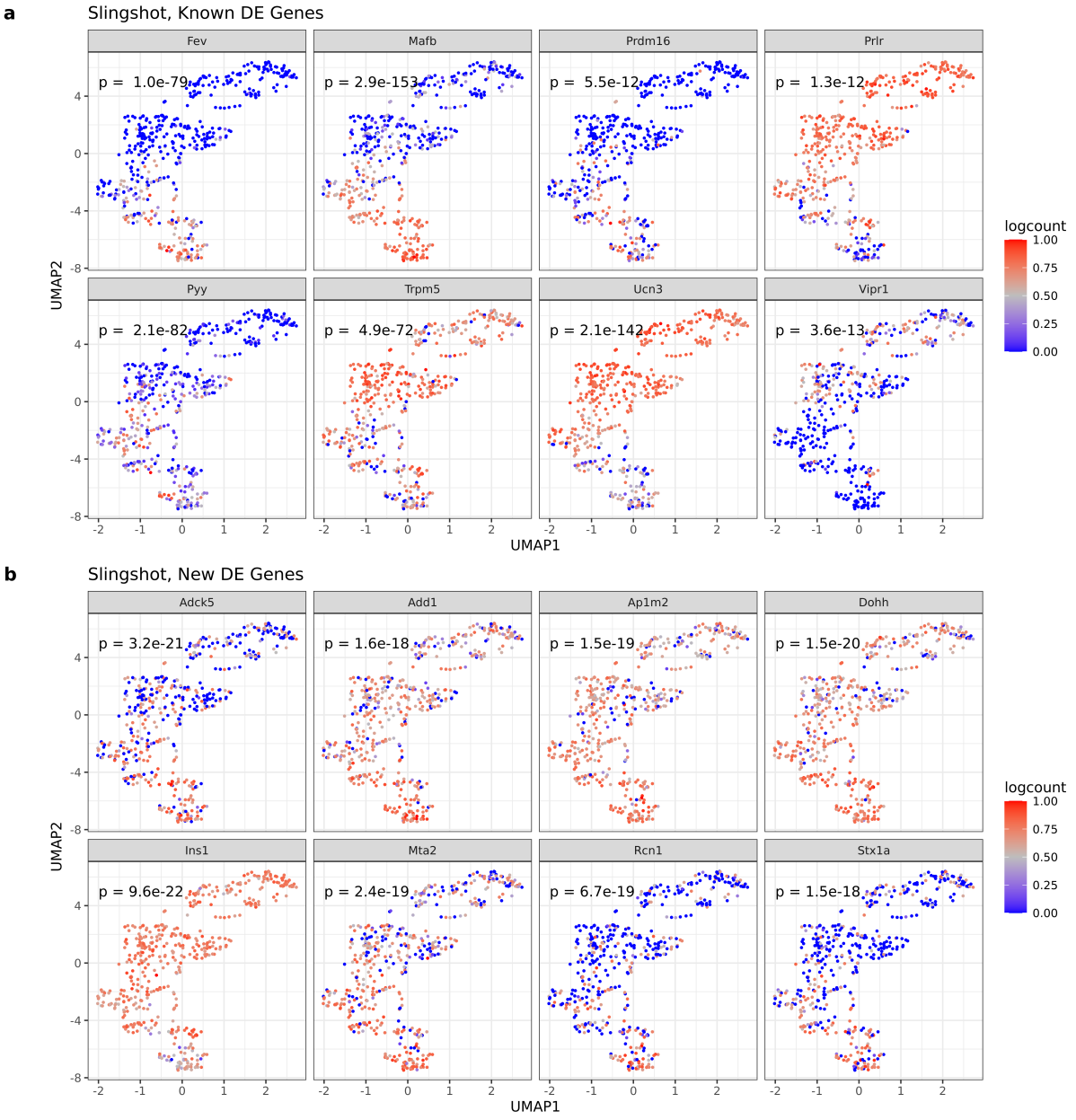


Figure 3.19: UMAP visualization of example DE genes identified by PseudotimeDE, using Slingshot as the pseudotime inference method, in the pancreatic beta cell maturation cell dataset.

p-values returned by PseudotimeDE are reported for all the 16 example genes. **(a)** Examples of known DE genes, which have been reported as highly confident DE genes in the original study [103]. **(b)** Examples of new DE genes, which are identified by PseudotimeDE but found as non-DE by either tradeSeq or Monocle3-DE.

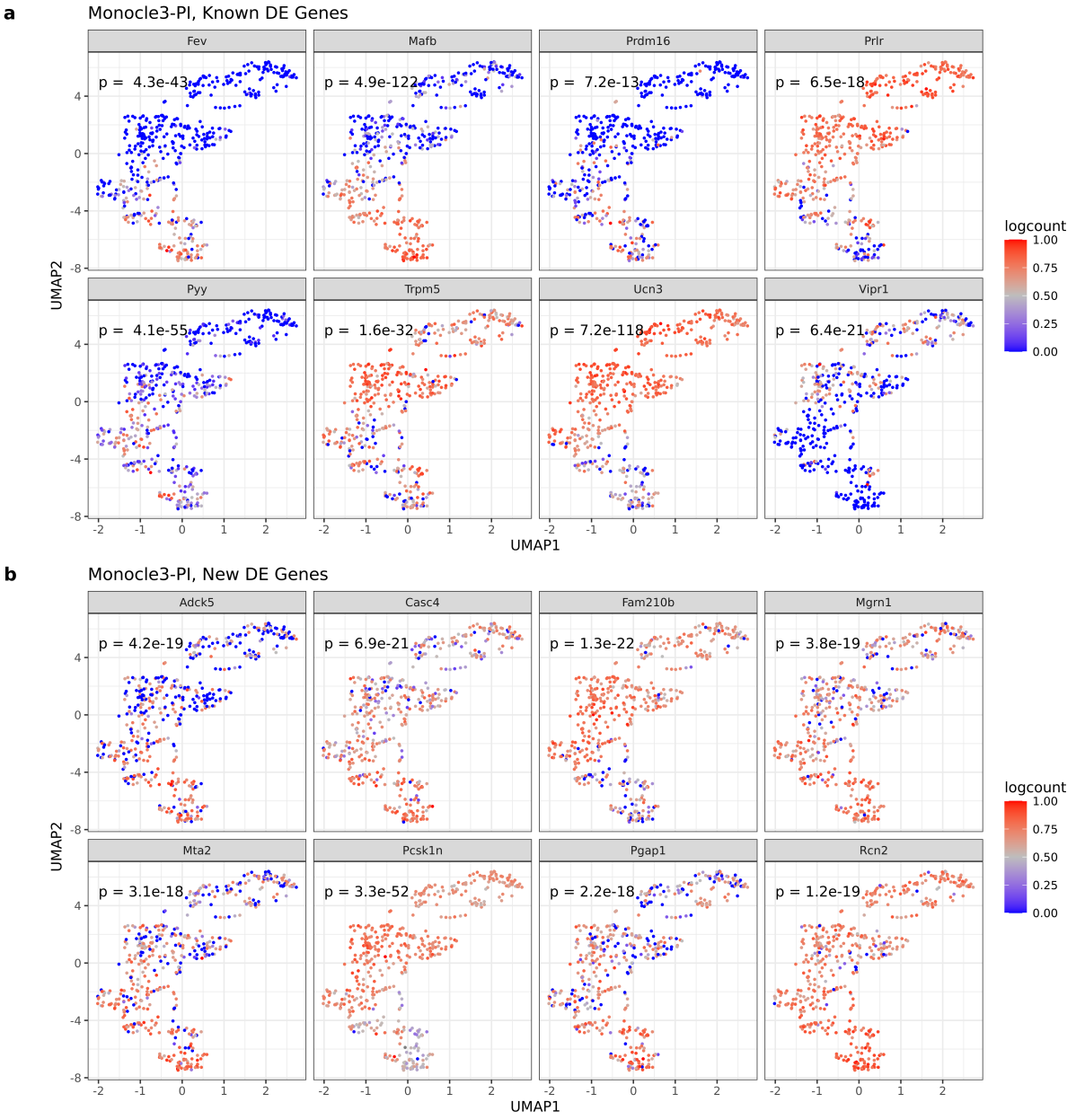


Figure 3.20: UMAP visualization of example DE genes identified by PseudotimeDE, using Monocle3-PI as the pseudotime inference method, in the pancreatic beta cell maturation cell dataset.

p-values returned by PseudotimeDE are reported for all the 16 example genes. **(a)** Examples of known DE genes, which have been reported as highly confident DE genes in the original study [103]. **(b)** Examples of new DE genes, which are identified by PseudotimeDE but found as non-DE by either tradeSeq or Monocle3-DE.

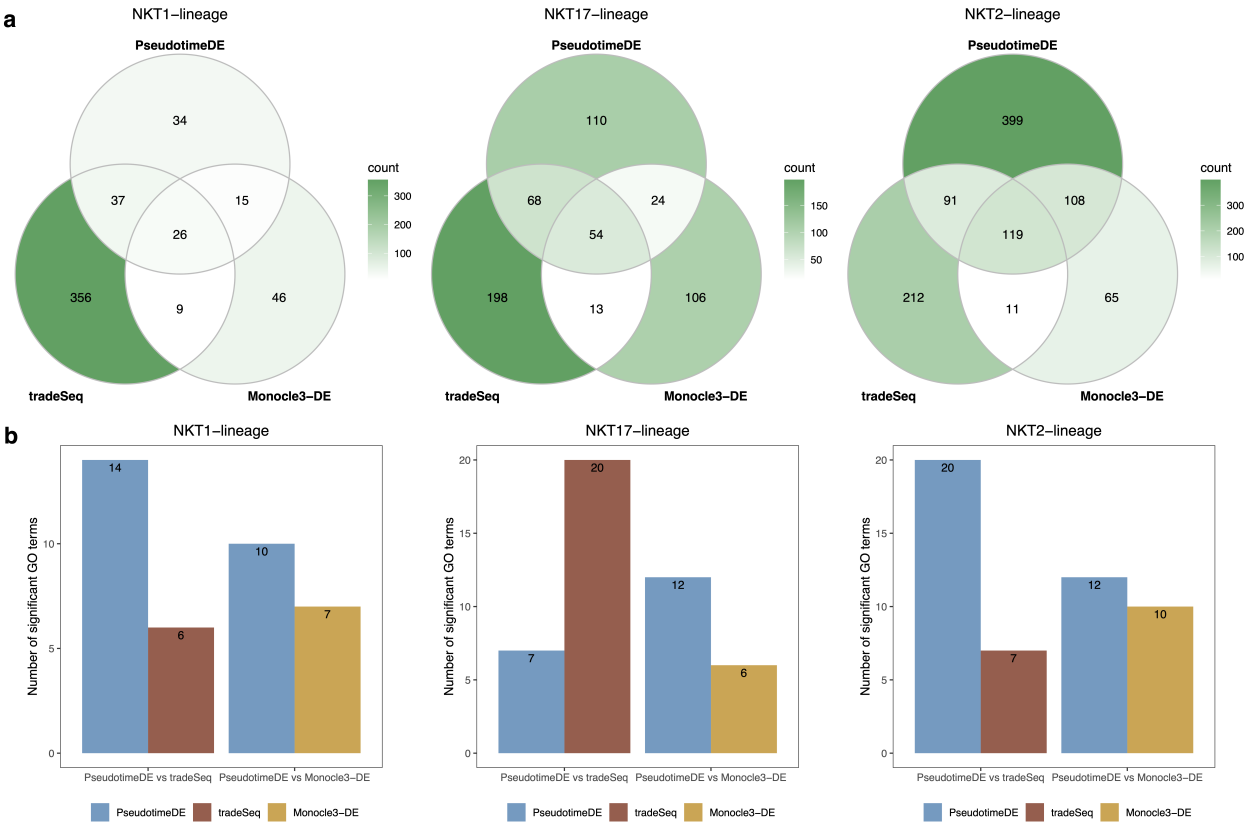


Figure 3.21: DE genes identified in the natural killer T cell dataset.

(a) Venn plots showing the overlaps of the significant DE genes (BH adjusted-$p \leqslant 0.05$) identified by the three DE methods. **(b)** Numbers of GO terms enriched ($p < 0.05$) in the significant DE genes specifically found by PseudotimeDE or tradeSeq/Monocle3-DE in pairwise comparisons between PseudotimeDE and tradeSeq/Monocle3-DE in (a).

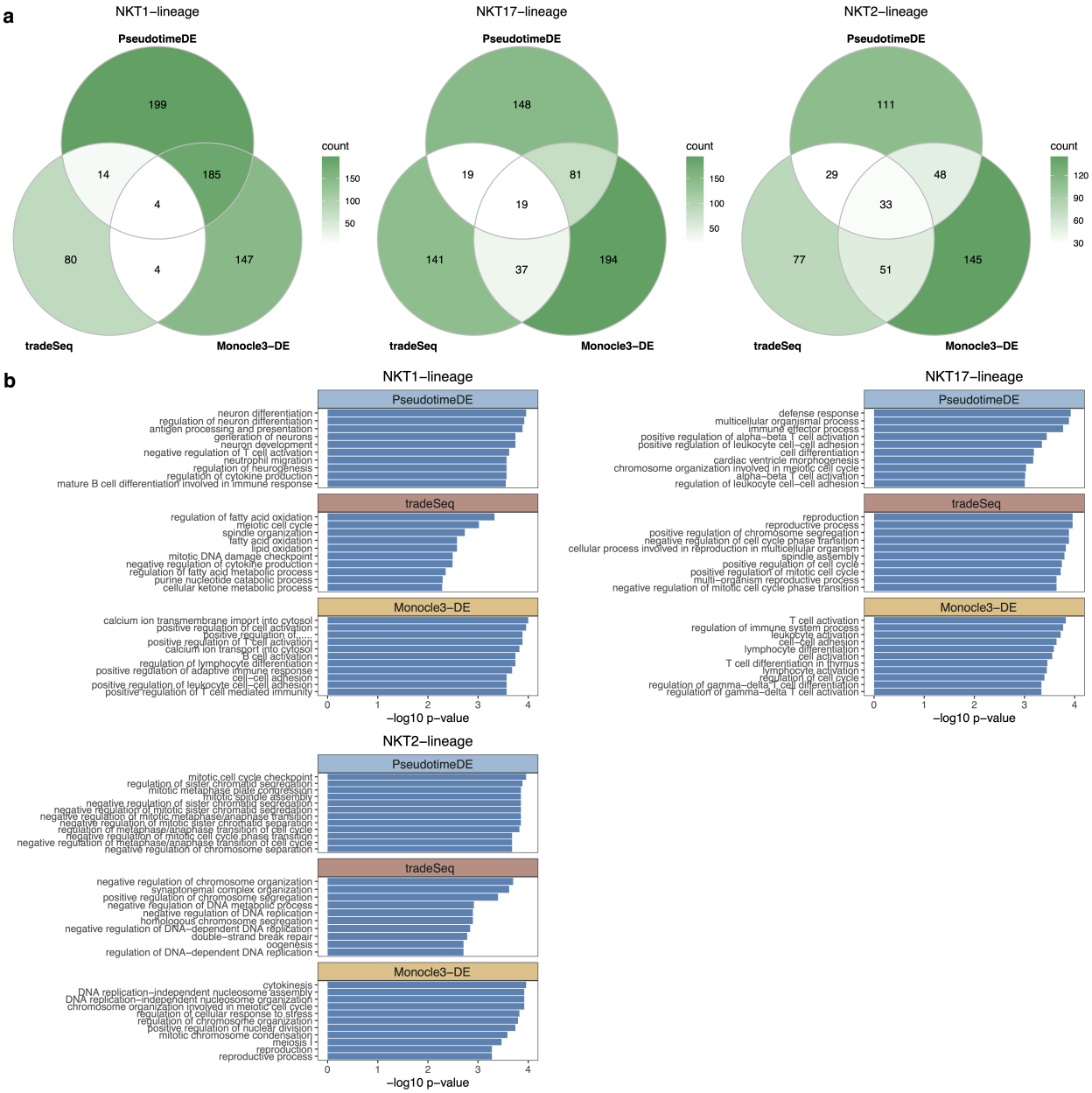


Figure 3.22: GO analysis of DE genes identified in the natural killer T cell dataset.

(a) Numbers of GO terms enriched ($p < 0.05$) in the significant DE genes found by each method. **(b)** Top 10 enriched GO terms for each DE method.

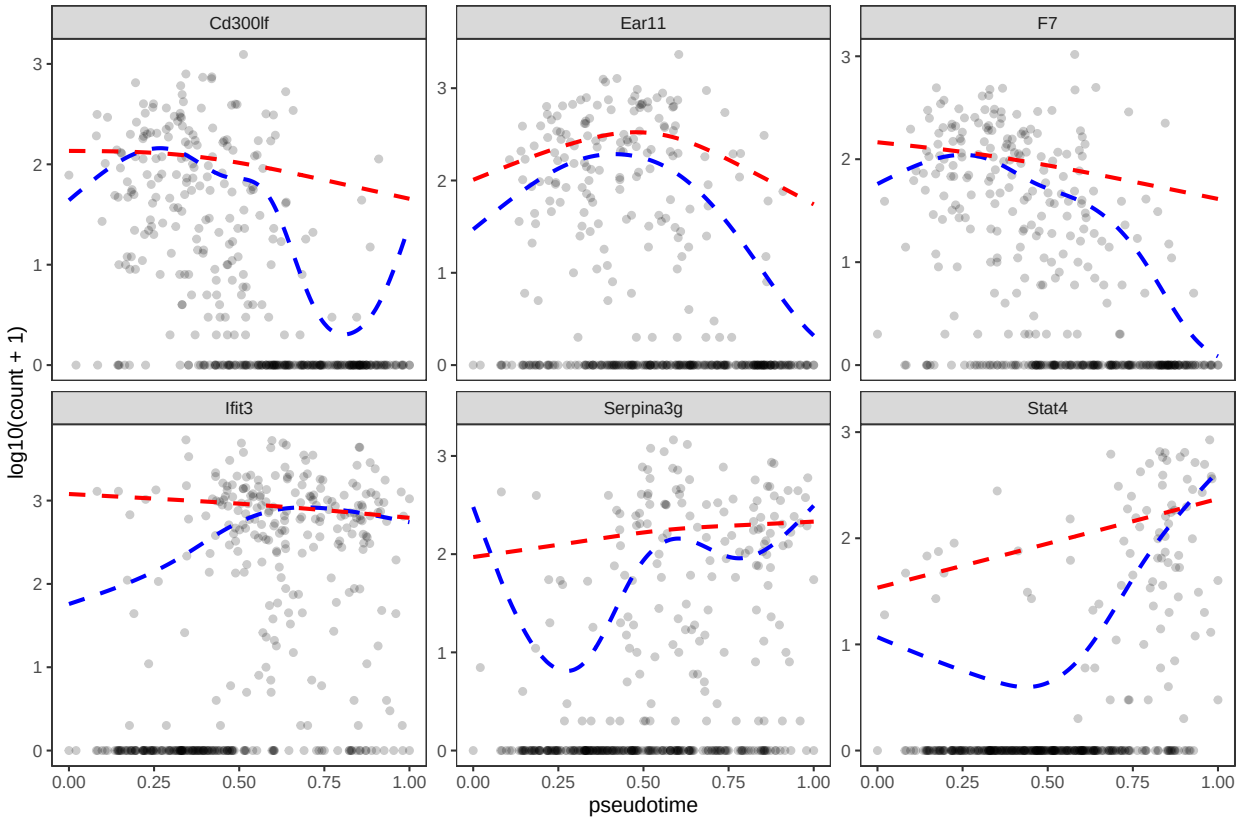


Figure 3.23: Comparison of NB-GAM and ZINB-GAM on the LPS-dendritic cell dataset with Slingshot pseudotime.

Example fitted results of NB-GAM / ZINB-GAM on six genes from the LPS-dendritic cell dataset with pseudotime inferred by Slingshot. NB-GAM yields small p-values ($p < 1e-10$) and ZINB-GAM yields large p-values ($p > 0.01$). Dashed blue lines and red lines are the fitted curves by NB-GAM and ZINB-GAM, respectively.

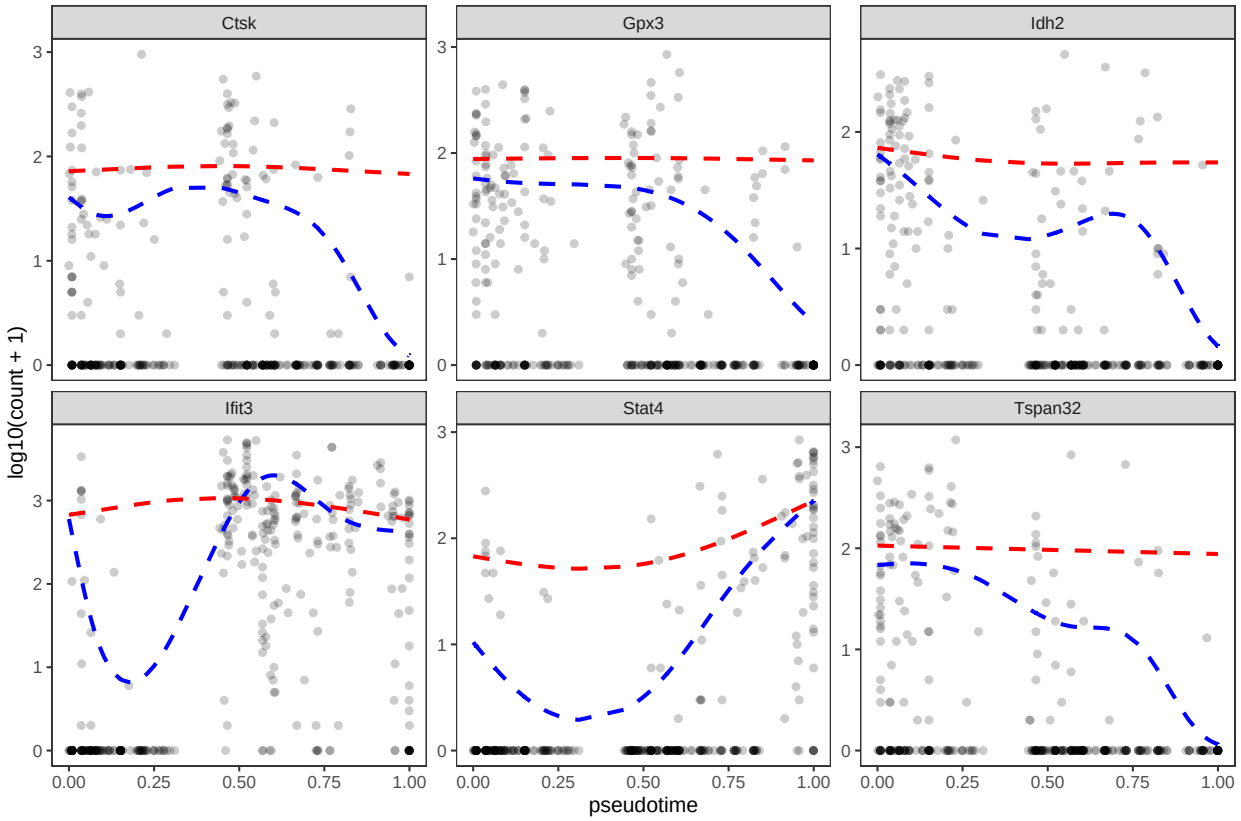


Figure 3.24: Comparison of NB-GAM and ZINB-GAM on the LPS-dendritic cell dataset with Monocle3-PI pseudotime.

Example fitted results of NB-GAM / ZINB-GAM on six genes from the LPS-dendritic cell dataset with pseudotime inferred by Monocle3-PI. NB-GAM yields small p-values ($p < 1e-10$) and ZINB-GAM yields large p-values ($p > 0.01$). Dashed blue lines and red lines are the fitted curves by NB-GAM and ZINB-GAM, respectively.

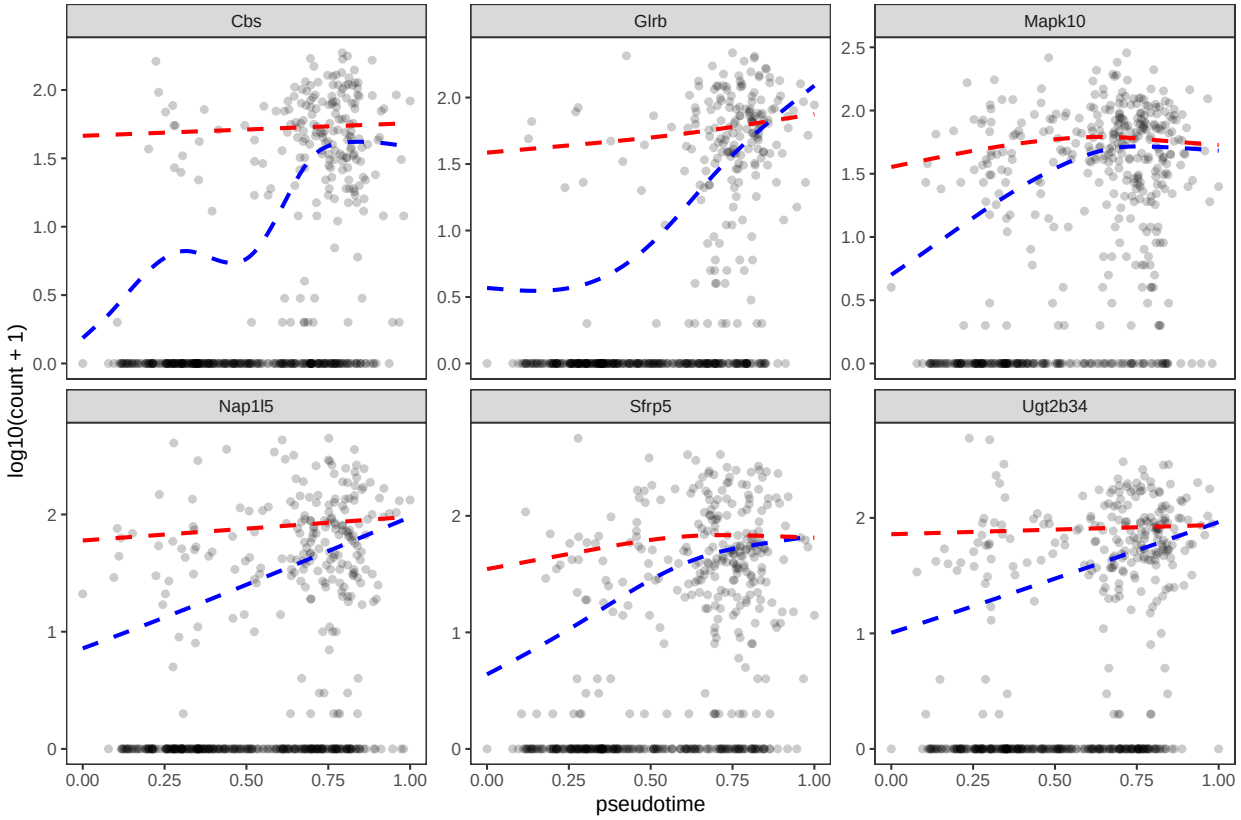


Figure 3.25: Comparison of NB-GAM and ZINB-GAM on the pancreatic beta cell maturation dataset with Slingshot pseudotime.

Example fitted results of NB-GAM / ZINB-GAM on six genes fromthe pancreatic beta cell maturation dataset with pseudotime inferred by Slingshot. NB-GAM yields small p-values ($p < 1e-10$) and ZINB-GAM yields large p-values ($p > 0.01$). Dashed blue lines and red lines are the fitted curves by NB-GAM and ZINB-GAM, respectively.

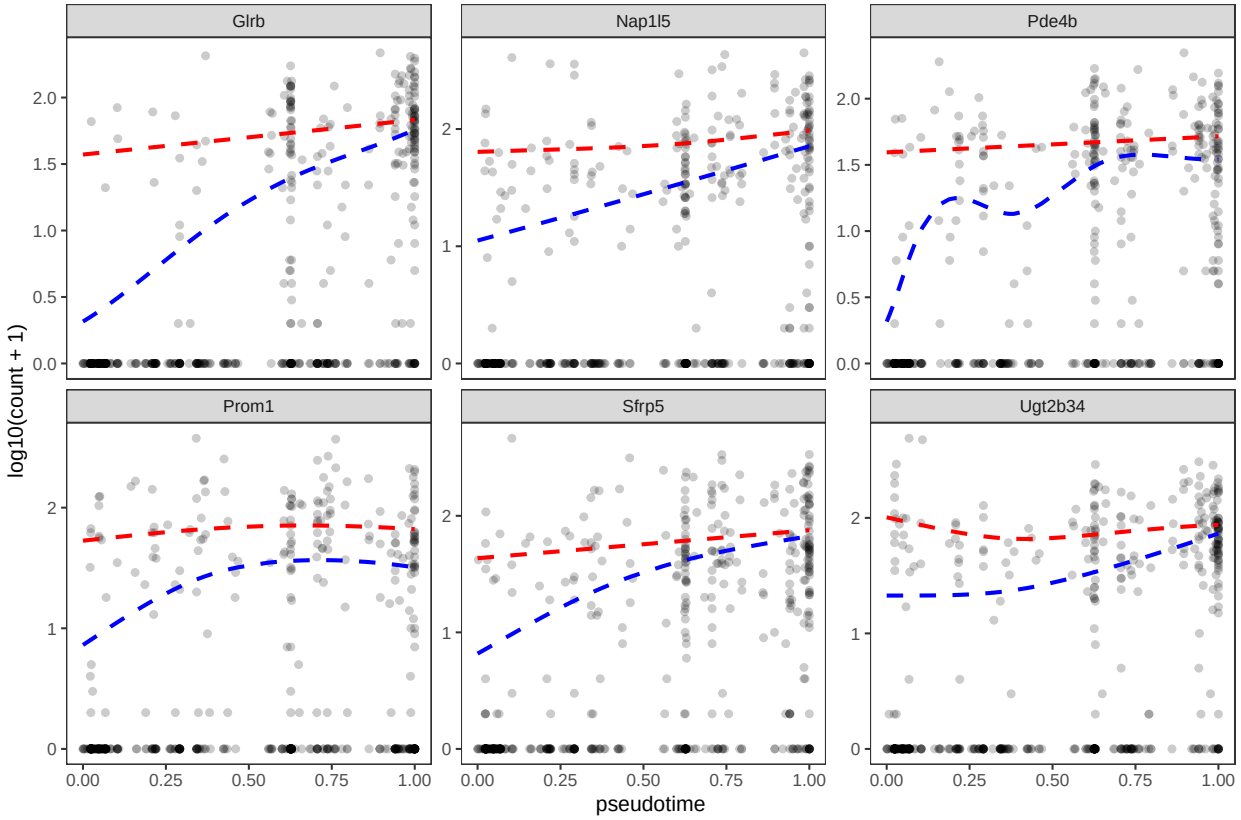


Figure 3.26: Comparison of NB-GAM and ZINB-GAM on the pancreatic beta cell maturation cell dataset with Slingshot pseudotime.

Example fitted results of NB-GAM / ZINB-GAM on six genes from the pancreatic beta cell maturation with pseudotime inferred by Monocle3-PI. NB-GAM yields small p-values ($p < 1e-10$) and ZINB-GAM yields large p-values ($p > 0.01$). Dashed blue lines and red lines are the fitted curves by NB-GAM and ZINB-GAM, respectively.

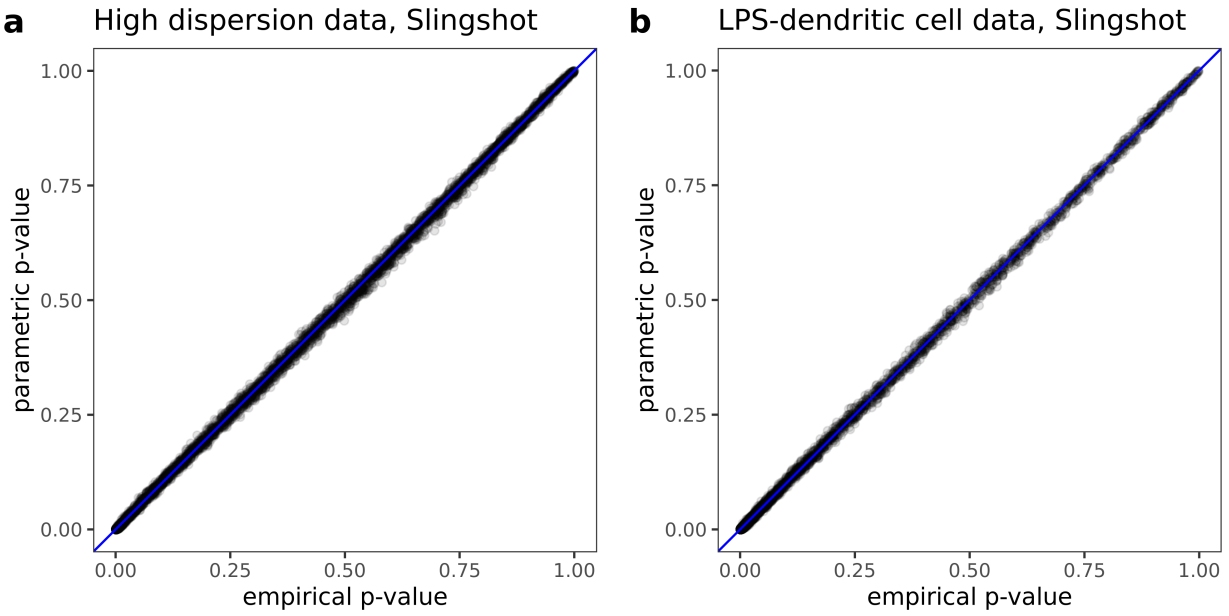


Figure 3.27: Comparison of empirical p-value and parametric p-value.

Scatter plot of empirical p-values and parametric p-values. **(a)** Scatter plot based on synthetic high dispersion dataset and pseudotime inferred by Slingshot. **(b)** Scatter plot based on LPS-dendritic cell dataset and pseudotime inferred by Slingshot. The parametric p-values are perfectly correlated with empirical p-values, suggesting that the parametric model well captures the estimated null distribution of test statistics.

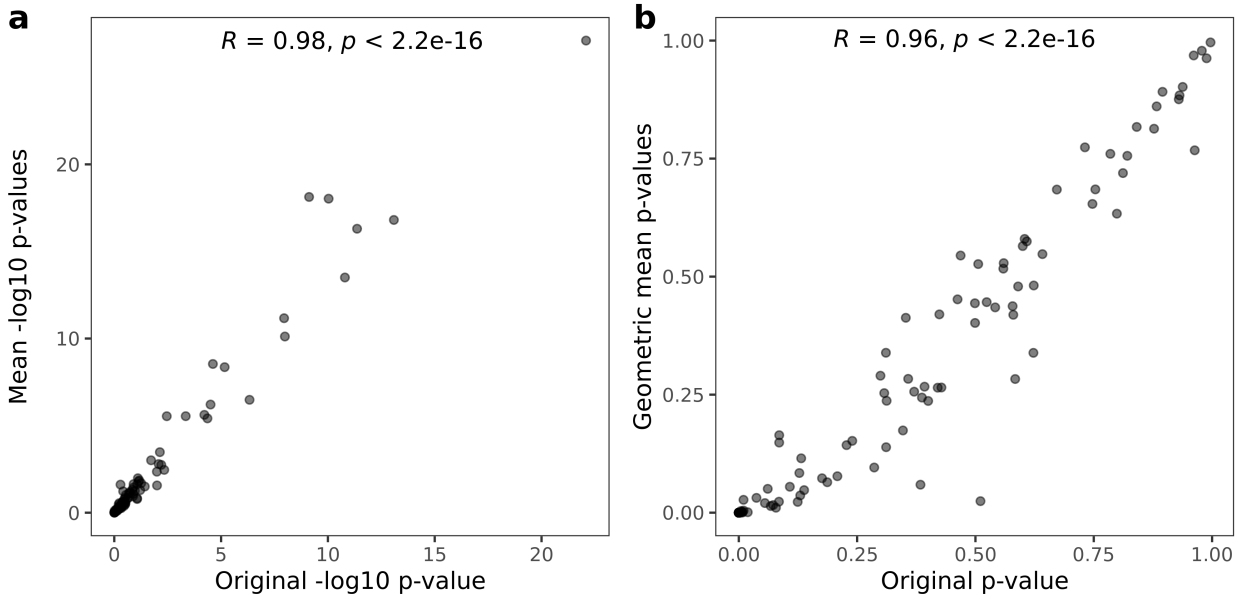


Figure 3.28: Comparison of p-values using 1000 subsamples and p-values using 100 subsamples.

The p-values are based on based on synthetic high dispersion dataset and pseudotime inferred by Slingshot. **(a)** Scatter plot of the original p-value using 1000 subsamples on negative $-\log_{10}$ scale and the mean of 50 p-values using 100 subsamples on negative $-\log_{10}$ scale. The strict linearity (Pearson correlation coefficient $R = 0.98$) suggests that using 100 subsamples yield similar p-values as using 1000 subsamples on log scale. **(b)** Scatter plot of the original p-value using 1000 subsamples and the geometric mean of 50 p-values using 100 subsamples. The strict linearity (Pearson correlation coefficient $R = 0.96$) suggests that using 100 subsamples yield similar p-values as using 1000 subsamples on raw scale.

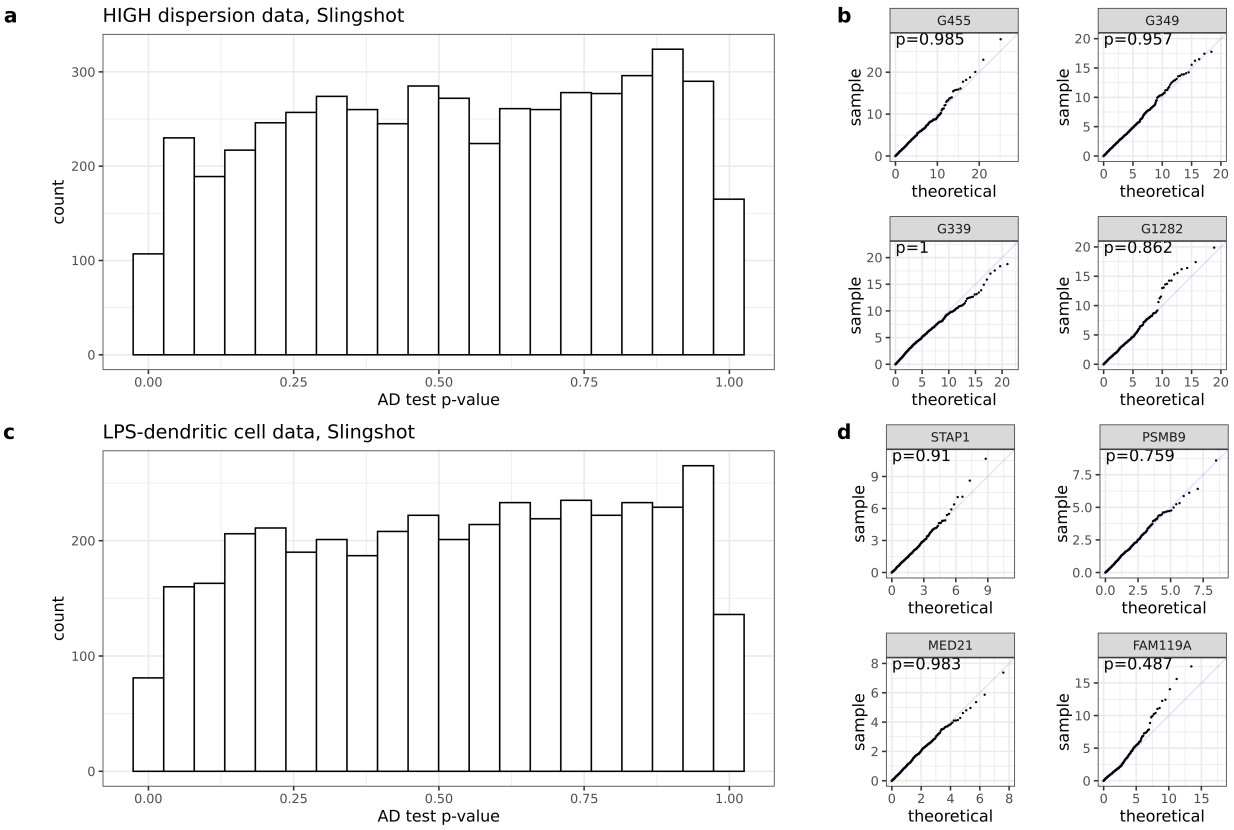


Figure 3.29: Goodness-of-fit of the parametric distribution.

The p-values are from Anderson-Darling (AD) test, which measures the goodness-of-fit of the gamma/two-component gamma mixture distribution to the empirical null distribution generated by subsampling and permutation. **(a)** Histogram of AD test p-values based on the synthetic high dispersion dataset and pseudotime inferred by Slingshot. The distribution is approximately Uniform$[0, 1]$, indicating that the parametric distribution fits the empirical null distribution well. **(b)** Quantile-quantile plots comparing the empirical null distribution and its corresponding parametric fit for four random genes. **(c)** Histogram of AD test p-values based on the LPS-dendritic cell dataset and pseudotime inferred by Slingshot. The distribution is approximately Uniform$[0, 1]$, indicating that the parametric distribution fits the empirical null distribution well. **(d)** Quantile-quantile plots comparing the empirical null distribution and its corresponding parametric fit for four random genes.

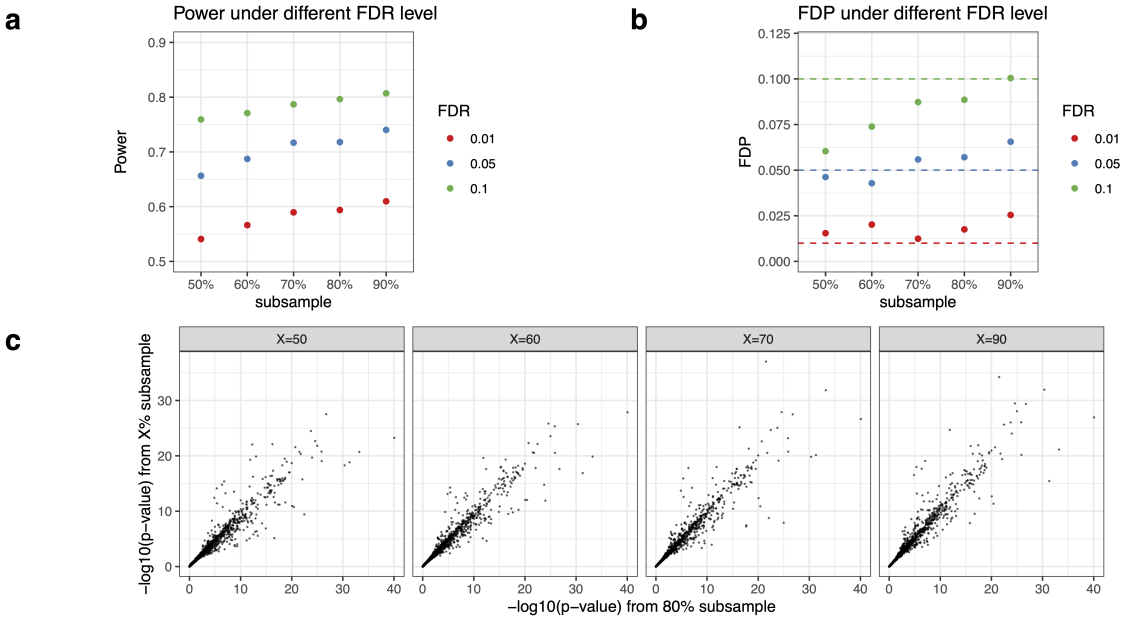


Figure 3.30: Robustness of PseudotimeDE to the subsampling proportion.

Results are based on the synthetic high dispersion dataset and pseudotime inferred by Slingshot. **(a)** Power of PseudotimeDE using different subsampling proportions under FDR levels 0.01, 0.05, and 0.1. **(b)** FDP of PseudotimeDE using different subsampling proportions under FDR levels 0.01, 0.05, and 0.1. **(c)** Scatter plots of the default p-values using 80% as the subsampling proportion vs. the p-values using 50%, 60%, 70% or 90% as the subsampling proportion. The strong linearity (Pearson correlation coefficient $R \geqslant 0.96$) of p-values under different subsampling proportions confirms the robustness of PseudotimeDE to the subsampling proportion.

CHAPTER 4

scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data

4.1 Introduction

Single-cell RNA sequencing (scRNA-seq) technologies have undergone rapid development in recent years. A remarkable achievement is the generation of large-scale datasets, and there are even datasets containing a million cells (See Table 4.1). Such massive scRNA-seq datasets have impeded exploratory data analysis (e.g., visualization) on standard computers.

An intuitive solution to this "big data" challenge is to subsample (downsample) a large-scale dataset, i.e., to select a subset of representative cells. Random subsampling is fast and unbiased, and it has been implemented in popular pipelines such as `Seurat` [119] and `Scanpy` [120]. However, random subsampling may miss rare cell types and is thus not ideal for preserving the transcriptome diversity. To overcome this drawback of random subsampling, Hie et al. proposed the first algorithm Geosketch for "intelligently selecting a subset of single cells", which they called "sketching" [121]. Geosketch aims to evenly sample cells across the transcriptome space by minimizing the Hausdorff distance between the subsample and the original sample (i.e., the large-scale dataset) (Section 4.2). In the follow-up algorithm Hopper, the authors improved the performance of Geosketch in terms of minimizing the Hausdorff distance. Moreover, prior to Geosketch and Hopper and outside of the single-cell field, this "intelligent subsampling" problem has been well studied in the field of computer experiment design, in which the "space-filling designs" implement the idea of even subsampling across the transcriptomic space [122]. The most popular space-filling designs are the minimax distance design and the maximin distance design [123]. Geosketch

and Hopper conceptually belong to the minimax distance design, which, however, is much more computationally intensive than the maximin distance design [123]. Here we propose scSampler, a Python package for fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. By "diversity-preserving sampling," scSampler implements the maximin distance design to make cells in the subsample as separative as possible. Using 8 simulated datasets and 10 real datasets, we show that scSampler outperforms existing subsampling methods in minimizing the Hausdorff distance between the subsample and the original sample. Moreover, scSampler is fast and scalable for million-level data.

4.2 scSampler methodology

scSampler is implemented in Python and can be installed by `pip install scsampler`. The input is a matrix or an `anndata` object from `scanpy` pipeline. Denote the input matrix by $\mathbf{X} \in \mathbb{R}^{n \times p}$, whose columns correspond to p features (by default, top p PCs from a cell-by-gene $\log(\text{count} + 1)$ matrix and scaled to $[0, 1]$) and whose rows correspond to n cells. Therefore, $\mathbf{X}$ can also represent a set $\mathcal{X} = \{x_1, \cdots, x_n\}$, where $x_i \in \mathbb{R}^p$. Our goal is to find a size n_s subset $\mathcal{X}_s \subset \mathcal{X}$, which satisfies:

$$\max_{\mathcal{X}_s} \min_{x_i, x_j \in \mathcal{X}_s} d\left(x_i, x_j\right) , \tag{4.1}$$

where $d(\cdot, \cdot)$ is the Euclidean distance. The optimality in (4.1) can be achieved by minimizing a scalar loss function:

$$\min_{\mathcal{X}_s} \sum_{i=1}^{n_s - 1} \sum_{j=i+1}^{n_s} \frac{1}{d^{\alpha}\left(x_i, x_j\right)} , \tag{4.2}$$

for a sufficiently large α [124]. It is found by [125] and in our numerical results that $\alpha = 4p$ is big enough and keeps the algorithm numerically stable, so we set $\alpha = 4p$ by default. For computational efficiency, scSampler can randomly split the original sample into subsets and perform subsampling on each subset.

4.2.1 scSampler algorithm

Algorithm 1 scSampler: sequential selection

Input: a sample (dataset) $\mathcal{X}$, a required subsample size n_s, and a split fold B
Output: a subsample $\mathcal{X}_s$
Evenly split $\mathcal{X}$ into B random subsets $\mathcal{X}^1, \cdots, \mathcal{X}^b \cdots, \mathcal{X}^B$
for each $\mathcal{X}^b$ **do**
  $m = 1$
  Randomly select a point x_1 from $\mathcal{X}^b$
  Initialize $\mathcal{X}_s^b$ as a subsample with a single point x_1 and remove x_1 from $\mathcal{X}^b$
  for each $x \in \mathcal{X}^b$ **do**
    Compute $\Delta(x|\mathcal{X}_s^b) = 1/\left[d(x_1, x)\right]^\alpha$
  end for
  while $m < \lceil n_s/B \rceil$ **do**
    Find x_{m+1} by (4.3)
    Include x_{m+1} in $\mathcal{X}_s^b$ and remove it from $\mathcal{X}^b$
    for each $x \in \mathcal{X}^b$ **do**
      Update $\Delta(x|\mathcal{X}_s^b) \leftarrow \Delta(x|\mathcal{X}_s^b) + 1/\left[d(x_{m+1}, x)\right]^\alpha$
    end for
    $m \leftarrow m + 1$
  end while
end for
$\mathcal{X}_s = \cup_{b=1}^B \mathcal{X}_s^b$ =0

4.2.2 The sequential criterion

We want to generate a subsample, denoted by $\mathcal{X}_s^b$, with $n_0 = \lceil n_s/B \rceil$ points from each split sample $\mathcal{X}^b$, $b = 1, \ldots, B$. We start with a random point x_1 and select $x_2, \ldots, x_{n_0}$ sequentially. Suppose we have already selected m points. Then the $(m+1)$th point is obtained as

$$\begin{aligned} x_{m+1} &= \arg\min_{x \in \mathcal{X}^b} \left\{ \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{1}{\left[d\left(x_i, x_j\right)\right]^\alpha} + \sum_{i=1}^{m} \frac{1}{\left[d\left(x_i, x\right)\right]^\alpha} \right\} \\ &= \arg\min_{x \in \mathcal{X}^b} \Delta(x|\mathcal{X}_s^b)\,, \end{aligned} \tag{4.3}$$

where $\alpha = 4p$ and

$$\Delta(x|\mathcal{X}_s^b) = \sum_{i=1}^{m} \frac{1}{\left[d\left(x_i, x\right)\right]^\alpha}\,.$$

4.2.3 Other used metrics

4.2.3.1 Gini coefficient

The Gini coefficient measures the inequality of cell types. A smaller Gini coefficient means that the cell types are more balanced in a sample. Here we denote the subsample as X_s, and the cell type proportions in X_s are $p_1, \cdots, p_c, \cdots, p_C$. Note that the C is the total number of cell types in the original data. Therefore, if a cell c type is missing in the subsample, its proportion $p_c = 0$. Then the Gini coefficient G is calculated as:

$$G = \frac{\sum_{c=1}^{C}(2c - C - 1)p_{(c)}}{C\sum_{c=1}^{C} p_{(c)}}, \tag{4.4}$$

where $p_{(c)}$ is the c-th smallest cell type proportion. The code for calculation is from `https://github.com/oliviaguest/gini`.

4.3 Results

4.3.1 scSampler outperforms other subsampling methods

To comprehensively benchmark scSampler—three variants: scSampler-sp1 (no sample splitting; the slowest), scSampler-sp4 (splitting the sample into 4 subsets), and scSampler-sp16 (splitting the sample into 16 subsets; the fastest)—against random sampling and two state-of-the-art subsampling methods, Geosketch and Hopper, we use the scRNA-seq simulator `Splatter` [126] to generate 8 simulated datasets and collect 10 real datasets (See Table 4.1). On each dataset, we subsample 1000, 3000, 5000 and 10000 cells using each subsampling method. Fig. 4.1a shows an example that illustrates the difference between random subsampling and scSampler: compared to random sampling, scSampler selects more cells from small cell clusters. Quantitatively, we compare subsampling methods by two measures: (1) the Hausdorff distance between the subsample and the original sample, (2) computation time, both of which are better if smaller (Fig. 4.1b). Fig. 4.1b summarizes the performance of subsampling methods in the two measures. Notably, scSampler-sp1 consistently yields the

smallest Hausdorff distances across all datasets and all subsample sizes. Moreover, scSampler is fast: on the largest cortex dataset (more than 1 million cells), scSampler-sp1 finishes in 15 minutes, and scSampler-sp16 takes only 1 minute and still outperforms Geosketch and Hopper by achieving a lower Hausdorff distance. Fig. 4.1c shows that scSampler is consistently ranked the top (smaller ranks are better) across the 18 datasets.

To verify if rare cell types are better captured by scSampler than other methods, we calculate the Gini coefficient of cell type proportions in each subsample; a smaller Gini coefficient indicates more balanced cell types (Section 4.2.3.1). In more than 60% of the combinations of 18 datasets and 4 subsmaple sizes, the fastest scSampler-sp16 leads to the smallest Gini coefficient. Considering that the real datasets may not have accurately annotated cell types, we examine the simulated datasets and find that scSampler-sp16 leads to the smallest Gini coefficient in 90% of the combinations of 8 simulated datasets and 4 sample sizes, confirming that scSampler well preserves rare cell types.

4.3.2 The computation time of scSampler with splitting

The computation time of Algorithm 1 without splitting the full dataset (i.e., $B = 1$), denoted by T, is $O(nn_s)$. If the data is split into B subsets, the computation time on each subset is $O(nn_s/B^2)$, so the total computation time, denoted by T_B, is $O(nn_s/B)$, which is $1/B$ of T. The splitting procedure, although does not improve the asymptotic scalability in n if B is a constant, can dramatically accelerate the computation in practice. To see this, Figure 4.2a plots T_B against B on four datasets for $B = 1$, 2, 4, 8, 16, and 32, and shows that roughly $T_B = T/B$. For example, on the dataset "splatter8," when $B = 1$, the computation time is around 700 seconds, which decreases to around 350, 175, and 87.5 for $B = 2, 4$, and 8. We further plot T_B against $1/B$ in Figure 4.2b, which shows that T_B is roughly linear in $1/B$, that is, $T_B = T \cdot (1/B) = T/B$. All computations are carried out on a server running Ubuntu 20.04 system with an Intel Xeon E5-2687W v4 CPU and 256 GB memory.

4.4 Discussion

We proposed scSampler, an unsupervised diversity-preserving subsampling methods for large-scale single-cell transcriptomic data. Compared to the state-of-the-art subsampling methods, scSampler finds subsamples with smaller Hausdorff distances to the original sample, indicating its superiority in preserving transcriptome diversity. Moreover, scSampler is fast, scalable to million-level data, and can be further accelerated by using random splits. As a Python package, scSampler is open source and adaptive to the `Scanpy` pipeline.

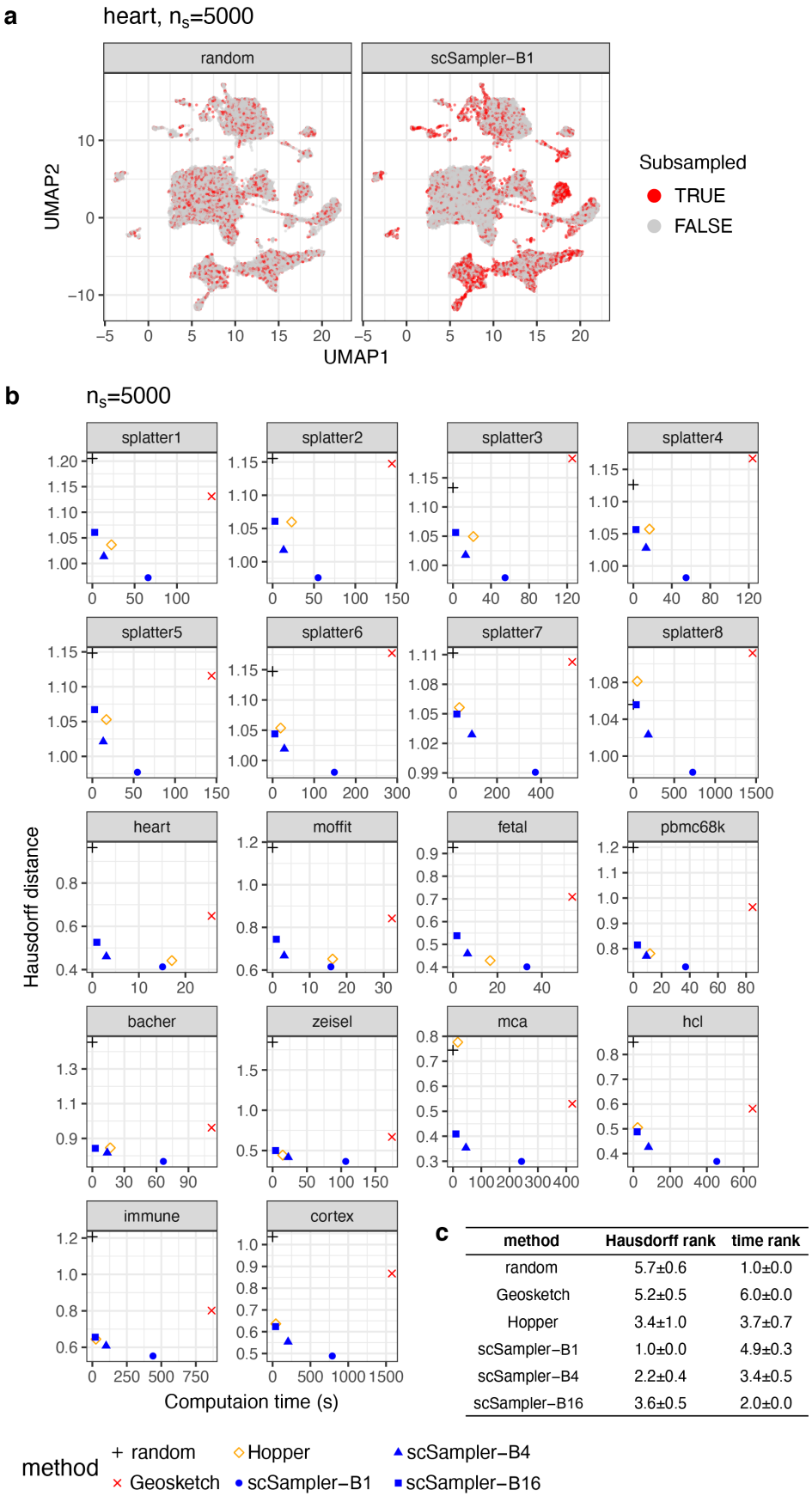
4.5 Code and data availability

The R package and the tutorials of scSampler are available at:
https://github.com/SONGDONGYUAN1994/scsampler.

4.6 Acknowledgments

This chapter is based on my joint work with Dr. Nan Miles Xi, Dr. Lin Wang, and my Ph.D. advisor Dr. Jingyi Jessica Li.

4.7 Figures



c

method	Hausdorff rank	time rank
random	5.7±0.6	1.0±0.0
Geosketch	5.2±0.5	6.0±0.0
Hopper	3.4±1.0	3.7±0.7
scSampler−B1	1.0±0.0	4.9±0.3
scSampler−B4	2.2±0.4	3.4±0.5
scSampler−B16	3.6±0.5	2.0±0.0

Figure 4.1: Benchmarking scSampler against other subsampling methods.

(a) UMAP visualization of selected cells in the original data by random subsampling and scSampler-sp1, respectively. (b) Scatter plots of Hausdorff distance against computation time. (c) Summary of the performance of each methods. The table shows the mean ranks and standard deviations across all datasets and subsample sizes.

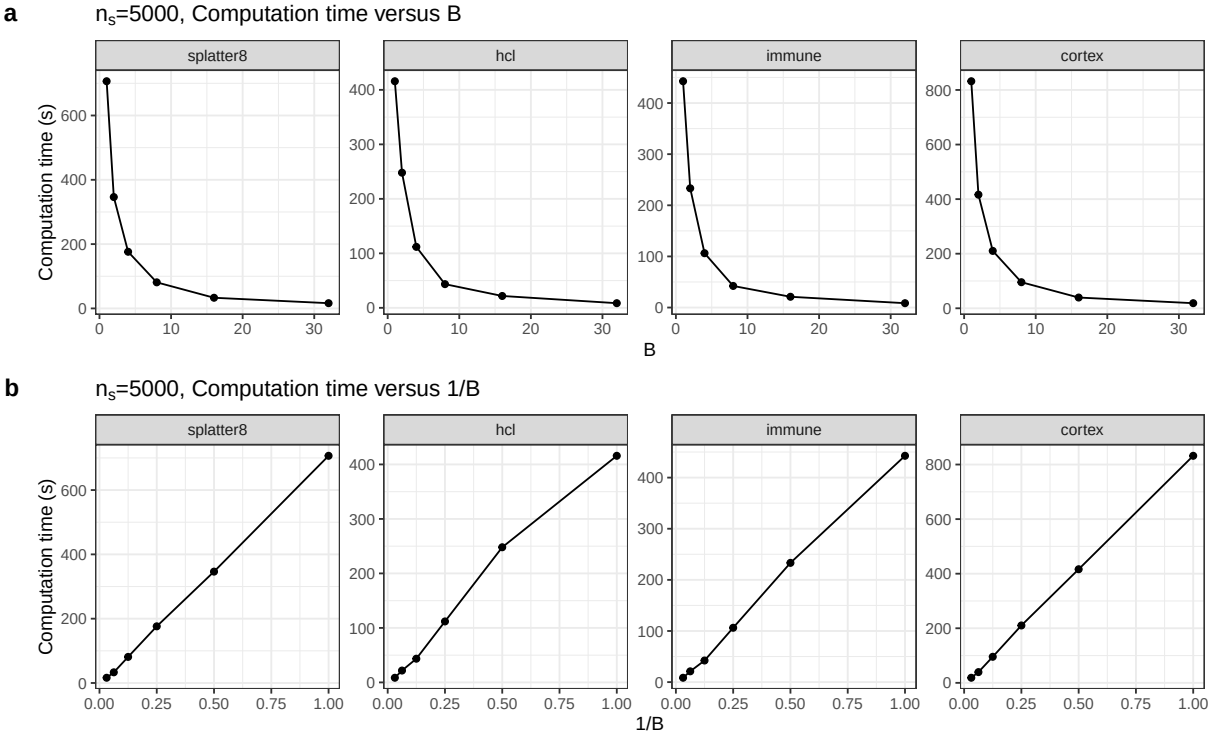


Figure 4.2: Computational time of scSampler

Computational time of four datasets against B and $1/B$, respectively.

4.8 Supplementary materials

4.8.1 Supplementary tables

Table 4.1: Overview of datasets used in scSampler

dataset	# of cells	# of HVGs	# of cell types	λ or reference
splatter1	100000	529	5	0.1
splatter2	100000	524	5	0.1
splatter3	100000	515	5	0.15
splatter4	100000	497	5	0.2
splatter5	100000	500	5	0.25
splatter6	200000	524	5	0.2
splatter7	500000	510	5	0.2
splatter8	1000000	517	5	0.2
heart	29552	2350	32	[127]
moffit	30332	1729	12	[18]
fetal	61006	3330	32	[68]
pbmc68k	68579	1488	10	[8]
bacher	104417	803	6	[128]
zeisel	160796	3389	39	[129]
mca	333778	3587	52	[130]
hcl	599926	1864	63	[131]
immune	606606	1566	34	[132]
cortex	1089022	2124	43	[133]

CHAPTER 5

Summary and future directions

In this article, we discussed three challenges in single-cell and spatial omics and their computational solutions. In this section, I summarise the three computational methods that I developed and discuss some potential future directions.

5.1 scDesign3: generation of realistic in silico data for multimodal single-cell and spatial omics

In Chapter 2, we proposed scDesign3, a realistic and versatile simulator for single-cell and spatial multi-omics. scDesign3 can generate realistic synthetic data from diverse settings, including different cell variations (discrete, continuous, and spatial), feature modalities, and experimental design.

For future directions, one remaining question is the scalability of scDesign3. For the current model, scDesign3 requires fitting regression models for each individual gene. If the marginal regression model is complicated (e.g., in spatial omics), the marginal fitting can be very time-consuming for thousands of genes. One solution is to propose a novel optimization procedure for fitting a large number of regression models jointly with the same design matrix.

5.2 PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data

In Chapter 3, we proposed PseudotimeDE, a novel statistical method to identify DE genes along inferred cell pseudotime. PseudotimeDE focuses on generating well-calibrated p-values while using subsampling and permutation to incorporate the randomness of inferred pseudotime. We use comprehensive studies on simulated and real data to demonstrate that PseudotimeDE yields better FDR control and higher power than other existing methods do.

For future directions, one remaining question is the "double-dipping" issue, which means that the same dataset is used both for inferring latent variables and testing differential expressed genes. Similar to the double-dipping problem in ClusterDE [3], it is possible that there is no real biological trajectory in the dataset while a pseudotime inference algorithm still returns a fake trajectory. Currently, PseudotimeDE still assumes the existence of a biological trajectory and cannot handle the double-dipping issue. To solve this problem, we will design a new frameworj to create a valid in silico control dataset without a trajectory but still resembles the real dataset from different statistical aspects.

5.3 scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data

In Chapter 4, we proposed scSampler, a novel method for fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. The goal of scSampler is to make cells in the subsample as separative as possible compared to the original sample. Using simulated datasets and real datasets, we show that scSampler outperforms existing subsampling methods in minimizing the Hausdorff distance between the subsample and the original sample on million-level data.

For future directions, we can extend the current sampling from single-cell transcriptomics

to spatial transcriptomics. For the newest spatial transcriptomics technology (e.g., the 10X Visium HD [134]), the number of measured locations can be million-level. Therefore, sub-sampling may be applied on Visium HD data. The major challenge will be to take into account both spatial locations and cell-type identities in the sampling process.

Bibliography

[1] Dongyuan Song et al. "scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics". In: *Nature Biotechnology* 42.2 (2024), pp. 247–252.

[2] 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.

[3] Dongyuan Song et al. "ClusterDE: a post-clustering differential expression (DE) method robust to false-positive inflation caused by double dipping". In: *Research Square* (2023).

[4] Dongyuan Song et al. "scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data". In: *Bioinformatics* 38.11 (2022), pp. 3126–3127.

[5] Dongyuan Song et al. "scPNMF: sparse gene encoding of single cells to facilitate gene selection for targeted gene profiling". In: *Bioinformatics* 37.Supplement_1 (2021), pp. i358–i366.

[6] Elvis Han Cui et al. "Single-cell generalized trend model (scGTM): a flexible and interpretable model of gene expression trend along cell pseudotime". In: *Bioinformatics* 38.16 (2022), pp. 3927–3934.

[7] Fuchou Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5 (2009), pp. 377–382.

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

[9] Jason D Buenrostro et al. "Single-cell chromatin accessibility reveals principles of regulatory variation". In: *Nature* 523.7561 (2015), pp. 486–490.

[10] Darren A Cusanovich et al. "Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing". In: *Science* 348.6237 (2015), pp. 910–914.

[11] Ino D Karemaker and Michiel Vermeulen. "Single-cell DNA methylation profiling: technologies and biological applications". In: *Trends in biotechnology* 36.9 (2018), pp. 952–965.

[12] Sean C Bendall et al. "Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum". In: *Science* 332.6030 (2011), pp. 687–696.

[13] Song Chen, Blue B Lake, and Kun Zhang. "High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell". In: *Nature biotechnology* 37.12 (2019), pp. 1452–1457.

[14] Marlon Stoeckius et al. "Simultaneous epitope and transcriptome measurement in single cells". In: *Nature methods* 14.9 (2017), pp. 865–868.

[15] Nikhil Rao, Sheila Clark, and Olivia Habern. "Bridging genomics and tissue pathology: 10x genomics explores new frontiers with the visium spatial gene expression solution". In: *Genetic Engineering & Biotechnology News* 40.2 (2020), pp. 50–51.

[16] Samuel G Rodriques et al. "Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution". In: *Science* 363.6434 (2019), pp. 1463–1467.

[17] Robert R Stickels et al. "Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2". In: *Nature biotechnology* 39.3 (2021), pp. 313–319.

[18] Jeffrey R Moffitt et al. "Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region". In: *Science* 362.6416 (2018).

[19] Luke Zappia, Belinda Phipson, and Alicia Oshlack. "Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database". In: *PLoS computational biology* 14.6 (2018), e1006245.

[20] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. "Challenges in unsupervised clustering of single-cell RNA-seq data". In: *Nature Reviews Genetics* 20.5 (2019), pp. 273–282.

[21] Wouter Saelens et al. “A comparison of single-cell trajectory inference methods”. In: *Nature biotechnology* 37.5 (2019), pp. 547–554.

[22] Charlotte Soneson and Mark D Robinson. “Bias, robustness and scalability in single-cell differential expression analysis”. In: *Nature methods* 15.4 (2018), p. 255.

[23] Junyue Cao et al. “The single-cell transcriptional landscape of mammalian organogenesis”. In: *Nature* 566.7745 (2019), pp. 496–502.

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

[25] Yue Cao, Pengyi Yang, and Jean Yee Hwa Yang. “A benchmark study of simulation methods for single-cell RNA sequencing data”. In: *Nature communications* 12.1 (2021), pp. 1–12.

[26] Helena L Crowell et al. “Built on sand: the shaky foundations of simulating single-cell RNA sequencing data”. In: *bioRxiv* (2021).

[27] 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), pp. 1–37.

[28] Davide Risso et al. “A general and flexible method for signal extraction from single-cell RNA-seq data”. In: *Nature communications* 9.1 (2018), pp. 1–17.

[29] Helena L Crowell et al. “Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data”. In: *Nature communications* 11.1 (2020), pp. 1–12.

[30] Robrecht Cannoodt et al. “Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells”. In: *Nature Communications* 12.1 (2021), pp. 1–9.

[31] Payam Dibaeinia and Saurabh Sinha. “SERGIO: a single-cell expression simulator guided by gene regulatory networks”. In: *Cell systems* 11.3 (2020), pp. 252–271.

[32] Nikolaos Papadopoulos, Parra R Gonzalo, and Johannes Söding. "PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes". In: *Bioinformatics* 35.18 (2019), pp. 3517–3519.

[33] Jinjin Tian, Jiebiao Wang, and Kathryn Roeder. "ESCO: single cell expression simulation incorporating gene co-expression". In: *Bioinformatics* 37.16 (2021), pp. 2374–2381.

[34] Zeinab Navidi, Lin Zhang, and Bo Wang. "simATAC: a single-cell ATAC-seq simulation framework". In: *Genome biology* 22.1 (2021), pp. 1–16.

[35] D Mikis Stasinopoulos and Robert A Rigby. "Generalized additive models for location scale and shape (GAMLSS) in R". In: *Journal of Statistical Software* 23 (2008), pp. 1–46.

[36] Yuqing Zhang, Giovanni Parmigiani, and W Evan Johnson. "ComBat-seq: batch effect adjustment for RNA-seq count data". In: *NAR genomics and bioinformatics* 2.3 (2020), lqaa078.

[37] Simon N Wood. *Generalized additive models: an introduction with R*. chapman and hall/CRC, 2006.

[38] EE Kammann and Matthew P Wand. "Geoadditive models". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 52.1 (2003), pp. 1–18.

[39] Claudia Czado. *Analyzing Dependent Data with Vine Copulas*. New York: Springer, 2019.

[40] Wei Vivian Li and Jingyi Jessica Li. "A statistical simulator scDesign for rational scRNA-seq experimental design". In: *Bioinformatics* 35.14 (2019), pp. i41–i50.

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

[42] Mohamed Marouf et al. "Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks". In: *Nature communications* 11.1 (2020), pp. 1–12.

[43] Ying Ma and Xiang Zhou. "Spatially informed cell-type deconvolution for spatial transcriptomics". In: *Nature Biotechnology* (2022), pp. 1–11.

[44] Dylan M Cable et al. "Robust decomposition of cell type mixtures in spatial transcriptomics". In: *Nature Biotechnology* 40.4 (2022), pp. 517–526.

[45] Marc Elosua-Bayes et al. "SPOTlight: seeded NMF regression to deconvolute spatial transcriptomics spots with single-cell transcriptomes". In: *Nucleic acids research* 49.9 (2021), e50–e50.

[46] Guanao Yan and Jingyi Jessica Li. "scReadSim: a single-cell multi-omics read simulator". In: *bioRxiv* (2022).

[47] Ricard Argelaguet et al. "Computational principles and challenges in single-cell data integration". In: *Nature biotechnology* 39.10 (2021), pp. 1202–1215.

[48] 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), pp. 1–28.

[49] 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).

[50] 10x Genomics. *Datasets - 10x Genomics.* https://www.10xgenomics.com/resources/datasets/. 2022.

[51] Ansuman T Satpathy et al. "Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion". In: *Nature biotechnology* 37.8 (2019), pp. 925–936.

[52] Jiarui Ding et al. "Systematic comparison of single-cell and single-nucleus RNA-sequencing methods". In: *Nature biotechnology* 38.6 (2020), pp. 737–746.

[53] Sophie Petropoulos et al. "Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos". In: *Cell* 165.4 (2016), pp. 1012–1026.

[54] Hyun Min Kang et al. “Multiplexed droplet single-cell RNA-sequencing using natural genetic variation”. In: *Nature biotechnology* 36.1 (2018), pp. 89–94.

[55] Franziska Paul et al. “Transcriptional heterogeneity and lineage commitment in myeloid progenitors”. In: *Cell* 163.7 (2015), pp. 1663–1677.

[56] Burak Tepe et al. “Single-cell RNA-seq of mouse olfactory bulb reveals cellular heterogeneity and activity-dependent molecular census of adult-born neurons”. In: *Cell reports* 25.10 (2018), pp. 2689–2703.

[57] Patrik L Ståhl et al. “Visualization and analysis of gene expression in tissue sections by spatial transcriptomics”. In: *Science* 353.6294 (2016), pp. 78–82.

[58] Bin Li et al. “Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution”. In: *Nature Methods* (2022), pp. 1–9.

[59] Aimée Bastidas-Ponce et al. “Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis”. In: *Development* 146.12 (2019), dev173849.

[60] Lih Feng Cheow et al. “Single-cell multimodal profiling reveals cellular epigenetic heterogeneity”. In: *Nature methods* 13.10 (2016), pp. 833–836.

[61] Darren A Cusanovich et al. “A single-cell atlas of in vivo mammalian chromatin accessibility”. In: *Cell* 174.5 (2018), pp. 1309–1324.

[62] Jiaqiang Zhu, Shiquan Sun, and Xiang Zhou. “SPARK-X: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies”. In: *Genome Biology* 22.1 (2021), p. 184.

[63] Satija Lab. *stxBrain.SeuratData: 10X Genomics Visium Mouse Brain Dataset*. R package version 0.1.1. 2019.

[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] Efthymia Papalexi and Rahul Satija. "Single-cell RNA sequencing to explore immune cell heterogeneity". In: *Nature Reviews Immunology* 18.1 (2018), p. 35.

[66] Sophie Tritschler et al. "Concepts and limitations for learning developmental trajectories from single cell genomics". In: *Development* 146.12 (2019), dev170506.

[67] Adam P Croft et al. "Distinct fibroblast subsets drive inflammation and damage in arthritis". In: *Nature* 570.7760 (2019), pp. 246–251.

[68] Roser Vento-Tormo et al. "Single-cell reconstruction of the early maternal–fetal interface in humans". In: *Nature* 563.7731 (2018), pp. 347–353.

[69] Jong-Eun Park et al. "A cell atlas of human thymic development defines T cell repertoire formation". In: *Science* 367.6480 (2020), eaay3224.

[70] Cole Trapnell et al. "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". In: *Nature biotechnology* 32.4 (2014), p. 381.

[71] Zhicheng Ji and Hongkai Ji. "TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis". In: *Nucleic acids research* 44.13 (2016), e117–e117.

[72] Kelly Street et al. "Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics". In: *BMC genomics* 19.1 (2018), p. 477.

[73] Xiaojie Qiu et al. "Reversed graph embedding resolves complex single-cell trajectories". In: *Nature methods* 14.10 (2017), p. 979.

[74] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*. Vol. 43. CRC press, 1990.

[75] Simon N Wood. "mgcv: GAMs and generalized ridge regression for R". In: *R news* 1.2 (2001), pp. 20–25.

[76] Thomas W Yee. "The VGAM package". In: *R News* 8.2 (2008), pp. 28–39.

[77] Koen Van den Berge et al. "Trajectory-based differential expression analysis for single-cell sequencing data". In: *Nature communications* 11.1 (2020), pp. 1–13.

[78] Nan Hao and Erin K O'shea. "Signal-dependent dynamics of transcription factor translocation controls gene expression". In: *Nature structural & molecular biology* 19.1 (2012), p. 31.

[79] Xu Ren and Pei-Fen Kuan. "Negative binomial additive model for RNA-Seq data analysis". In: *BMC bioinformatics* 21 (2020), pp. 1–15.

[80] David S Fischer, Fabian J Theis, and Nir Yosef. "Impulse model-based differential expression analysis of time course sequencing data". In: *Nucleic acids research* 46.20 (2018), e119–e119.

[81] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome biology* 15.12 (2014), p. 550.

[82] Daniel Spies et al. "Comparative analysis of differential gene expression tools for RNA sequencing time course data". In: *Briefings in bioinformatics* 20.1 (2019), pp. 288–298.

[83] Peter V Kharchenko, Lev Silberstein, and David T Scadden. "Bayesian approach to single-cell differential expression analysis". In: *Nature methods* 11.7 (2014), pp. 740–742.

[84] Greg Finak et al. "MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data". In: *Genome biology* 16.1 (2015), pp. 1–13.

[85] Trung Nghia Vu et al. "Beta-Poisson model for single-cell RNA-seq data analyses". In: *Bioinformatics* 32.14 (2016), pp. 2128–2135.

[86] Keegan D Korthauer et al. "A statistical approach for identifying differential distributions in single-cell RNA-seq experiments". In: *Genome biology* 17.1 (2016), p. 222.

[87] Kieran R Campbell and Christopher Yau. "Order under uncertainty: robust differential expression analysis using probabilistic models for pseudotime inference". In: *PLoS computational biology* 12.11 (2016), e1005212.

[88] Magdalena E Strauß, John E Reid, and Lorenz Wernisch. "GPseudoRank: a permutation sampler for single cell orderings". In: *Bioinformatics* 35.4 (2019), pp. 611–618.

[89] Dimitris N Politis, Joseph P Romano, and Michael Wolf. *Subsampling*. Springer Science & Business Media, 1999.

[90] George C Tseng and Wing H Wong. "Tight clustering: a resampling-based approach for identifying stable and tight patterns in data". In: *Biometrics* 61.1 (2005), pp. 10–16.

[91] Yidan Sun, Heather Zhou, and Jingyi Jessica Li. "Bipartite Tight Spectral Clustering (BiTSC) Algorithm for Identifying Conserved Gene Co-clusters in Two Species". In: *bioRxiv* (2019), p. 865378.

[92] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[93] Simon N Wood. *Generalized additive models: an introduction with R*. CRC press, 2017.

[94] Simon Wotherspoon and Paul Burch. *zigam: EM implementation of zero-inflated GAMs*. R package version 0.1.1. 2016.

[95] Kenneth P Burnham and David R Anderson. "Multimodel inference: understanding AIC and BIC in model selection". In: *Sociological methods & research* 33.2 (2004), pp. 261–304.

[96] Simon N Wood. "On p-values for smooth components of an extended generalized additive model". In: *Biometrika* 100.1 (2013), pp. 221–228.

[97] Belinda Phipson and Gordon K Smyth. "Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn". In: *Statistical applications in genetics and molecular biology* 9.1 (2010), Article 39.

[98] Marie Laure Delignette-Muller and Christophe Dutang. "fitdistrplus: An R Package for Fitting Distributions". In: *Journal of Statistical Software* 64.4 (2015), pp. 1–34. URL: http://www.jstatsoft.org/v64/i04/.

[99] Tatiana Benaglia et al. "mixtools: An R Package for Analyzing Finite Mixture Models". In: *Journal of Statistical Software* 32.6 (2009), pp. 1–29. URL: http://www.jstatsoft.org/v32/i06/.

[100] Gregory Giecold et al. "Robust lineage reconstruction from high-dimensional single-cell data". In: *Nucleic acids research* 44.14 (2016), e122–e122.

[101] Alex K Shalek et al. "Single-cell RNA-seq reveals dynamic paracrine control of cellular variation". In: *Nature* 510.7505 (2014), pp. 363–369.

[102] Aravind Subramanian et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550.

[103] Wei-Lin Qiu et al. "Deciphering pancreatic islet β cell and α cell maturation pathways and characteristic features at the single-cell level". In: *Cell metabolism* 25.5 (2017), pp. 1194–1205.

[104] Åsa Apelqvist et al. "Notch signalling controls pancreatic cell differentiation". In: *Nature* 400.6747 (1999), pp. 877–881.

[105] Rebecca Lawson, Wolfgang Maret, and Christer Hogstrand. "Expression of the ZIP/SLC39A transporters in β-cells: a systematic review and integration of multiple datasets". In: *BMC genomics* 18.1 (2017), p. 719.

[106] Isaac Engel et al. "Innate-like functions of natural killer T cell subsets result from highly divergent gene programs". In: *Nature immunology* 17.6 (2016), pp. 728–739.

[107] Chiaowen Joyce Hsiao et al. "Characterizing and inferring quantitative cell cycle phase in single-cell RNA-seq data analysis". In: *Genome research* 30.4 (2020), pp. 611–621.

[108] Stephanie C Hicks et al. "Missing data and technical variability in single-cell RNA-sequencing experiments". In: *Biostatistics* 19.4 (2018), pp. 562–578.

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

[110] Justin D Silverman et al. "Naught all zeros in sequence count data are the same". In: *Computational and structural biotechnology journal* 18 (2020), p. 2789.

[111] Kwangbom Choi et al. "Bayesian model selection reveals biological origins of zero inflation in single-cell transcriptomics". In: *Genome biology* 21.1 (2020), pp. 1–16.

[112] 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.

[113] Tapio Lönnberg et al. "Single-cell RNA-seq and computational analysis using temporal mixture modelling resolves Th1/Tfh fate bifurcation in malaria". In: *Science immunology* 2.9 (2017), eaal2192.

[114] Xiaojie Qiu et al. "Single-cell mRNA quantification and differential analysis with Census". In: *Nature methods* 14.3 (2017), pp. 309–315.

[115] Richard Berk et al. "Valid post-selection inference". In: *The Annals of Statistics* 41.2 (2013), pp. 802–837.

[116] Jason D Lee et al. "Exact post-selection inference, with application to the lasso". In: *The Annals of Statistics* 44.3 (2016), pp. 907–927.

[117] Adrian Alexa and Jörg Rahnenführer. "Gene set enrichment analysis with topGO". In: *Bioconductor Improv* 27 (2009), pp. 1–26.

[118] Guangchuang Yu et al. "clusterProfiler: an R package for comparing biological themes among gene clusters". In: *OMICS: A Journal of Integrative Biology* 16.5 (2012), pp. 284–287. DOI: 10.1089/omi.2011.0118.

[119] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature biotechnology* 33.5 (2015), pp. 495–502.

[120] 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.

[121] Brian Hie et al. “Geometric sketching compactly summarizes the single-cell transcriptomic landscape”. In: *Cell systems* 8.6 (2019), pp. 483–493.

[122] V Roshan Joseph. “Space-filling designs for computer experiments: A review”. In: *Quality Engineering* 28.1 (2016), pp. 28–35.

[123] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. “Minimax and maximin distance designs”. In: *Journal of statistical planning and inference* 26.2 (1990), pp. 131–148.

[124] Max D Morris and Toby J Mitchell. “Exploratory designs for computational experiments”. In: *Journal of statistical planning and inference* 43.3 (1995), pp. 381–402.

[125] V Roshan Joseph et al. “Sequential exploration of complex surfaces using minimum energy designs”. In: *Technometrics* 57.1 (2015), pp. 64–74.

[126] Luke Zappia, Belinda Phipson, and Alicia Oshlack. “Splatter: simulation of single-cell RNA sequencing data”. In: *Genome biology* 18.1 (2017), pp. 1–15.

[127] Micheal A McLellan et al. “High-resolution transcriptomic profiling of the heart during chronic stress reveals cellular drivers of cardiac fibrosis and hypertrophy”. In: *Circulation* 142.15 (2020), pp. 1448–1463.

[128] Petra Bacher et al. “Low-avidity CD4+ T cell responses to SARS-CoV-2 in unexposed individuals and humans with severe COVID-19”. In: *Immunity* 53.6 (2020), pp. 1258–1271.

[129] Amit Zeisel et al. “Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq”. In: *Science* 347.6226 (2015), pp. 1138–1142.

[130] Xiaoping Han et al. “Mapping the mouse cell atlas by microwell-seq”. In: *Cell* 172.5 (2018), pp. 1091–1107.

[131] Xiaoping Han et al. “Construction of a human cell landscape at single-cell level”. In: *Nature* 581.7808 (2020), pp. 303–309.

[132] Irene Papatheodorou et al. “Expression Atlas update: from tissues to single cells”. In: *Nucleic acids research* 48.D1 (2020), pp. D77–D83.

[133] Zizhen Yao et al. “A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation”. In: *Cell* 184.12 (2021), pp. 3222–3241.

[134] Monica Nagendran et al. *1457 Visium HD enables spatially resolved, single-cell scale resolution mapping of FFPE human breast cancer tissue*. 2023.