UNIVERSITY OF CALIFORNIA

Los Angeles

Statistical Simulation and Analysis of Single-cell RNA-seq Data

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Tianyi Sun

2023

ABSTRACT OF THE DISSERTATION

Statistical Simulation and Analysis of Single-cell RNA-seq Data

by

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

The recent development of single-cell RNA sequencing (scRNA-seq) technologies has revolutionized transcriptomic studies by revealing the genome-wide gene expression levels within individual cells. In contrast to bulk RNA sequencing, scRNA-seq technology captures cell-specific transcriptome landscapes, which can reveal crucial information about cell-to-cell heterogeneity across different tissues, organs, and systems and enable the discovery of novel cell types and new transient cell states. According to search results from PubMed, from 2009-2023, over 5,000 published studies have generated datasets using this technology. Such large volumes of data call for high-quality statistical methods for their analysis. In the three projects of this dissertation, I have explored and developed statistical methods to model the marginal and joint gene expression distributions and determine the latent structure type for scRNA-seq data. In all three projects, synthetic data simulation plays a crucial role.

My first project focuses on the exploration of the Beta-Poisson hierarchical model for the marginal gene expression distribution of scRNA-seq data. This model is a simplified mechanistic model with biological interpretations. Through data simulation, I demonstrate three typical behaviors of this model under different parameter combinations, one of which can be interpreted as one source of the sparsity and zero inflation that is often observed in scRNA-seq datasets. Further, I discuss parameter estimation methods of this model and its other applications in the analysis of scRNA-seq data.

My second project focuses on the development of a statistical simulator, scDesign2, to

generate realistic synthetic scRNA-seq data. Although dozens of simulators have been developed before, they lack the capacity to simultaneously achieve the following three goals: preserving genes, capturing gene correlations, and generating any number of cells with varying sequencing depths. To fill in this gap, scDesign2 is developed as a transparent simulator that achieves all three goals and generates high-fidelity synthetic data for multiple scRNA-seq protocols and other single-cell gene expression count-based technologies. Compared with existing simulators, scDesign2 is advantageous in its transparent use of probabilistic models and is unique in its ability to capture gene correlations via copula. We verify that scDesign2 generates more realistic synthetic data for four scRNA-seq protocols (10x Genomics, CEL-Seq2, Fluidigm C1, and Smart-Seq2) and two single-cell spatial transcriptomics protocols (MERFISH and pciSeq) than existing simulators do. Under two typical computational tasks, cell clustering and rare cell type detection, we demonstrate that scDesign2 provides informative guidance on deciding the optimal sequencing depth and cell number in single-cell RNA-seq experimental design, and that scDesign2 can effectively benchmark computational methods under varying sequencing depths and cell numbers. With these advantages, scDesign2 is a powerful tool for single-cell researchers to design experiments, develop computational methods, and choose appropriate methods for specific data analysis needs.

My third project focuses on deciding latent structure types for scRNA-seq datasets. Clustering and trajectory inference are two important data analysis tasks that can be performed for scRNA-seq datasets and will lead to different interpretations. However, as of now, there is no principled way to tell which one of these two types of analysis results is more suitable to describe a given dataset. In this project, we propose two computational approaches that aim to distinguish cluster-type vs. trajectory-type scRNA-seq datasets. The first approach is based on building a classifier using eigenvalue features of the gene expression covariance matrix, drawing inspiration from random matrix theory (RMT). The second approach is based on comparing the similarity of real data and simulated data generated by assuming the cell latent structure as clusters or a trajectory. While both approaches have limitations, we show that the second approach gives more promising results and has room for further improvements.

The dissertation of Tianyi Sun is approved.

Wei Vivian Li

Qing Zhou

Arash Ali Amini

Mark S. Handcock

Jingyi Li, Committee Chair

University of California, Los Angeles

2023

To my mother Jiping Sun and my alma mater Bejing No. 8 Middle School

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Reflecting on my doctoral journey at UCLA, I could not have made it without the invaluable support from my mentors, collaborators, family, and friends.

First and foremost, I would like to thank my advisor Dr. Jingyi Jessica Li, for her tremendous support of my research and personal development. It has been a privilege for me to be able to study and work alongside her for over six years and I have been deeply moved by her quality of perseverance, rigor, courage, and diligence in the pursuit of scientific knowledge. She has always encouraged us to be innovative and has been supportive of me when I wanted to do difficult projects. Her broad interest outside statistics and the STEM field also motivates me to think about how I can become a better person and how I can contribute what I do to a better society. In my future career and life, I hope I can challenge myself more like she did.

I would like to thank Dr. Mark Stephen Handcock, Dr. Arash Ali Amini, Dr. Qing Zhou, and Dr. Wei Vivian Li for serving on my doctoral committee and for providing valuable feedback on my research.

I would like to thank all former and current members of Jessica's group for their helpful discussions and suggestions about my research. Among them, I want to especially thank Dongyuan Song for his valuable suggestions for my third project in this dissertation. He has been extremely helpful in providing the "biologist's perspective" to my projects and I wish him the best of luck in his academic career. I also thank Dr. Wei Vivian Li for helping revise the scDesign2 manuscript and Dr. Ruochen Jiang for collaborating on the "zeros in scRNA-seq data" project.

Last but not least, I would like to thank my family and friends for their unconditional love and support. I would like to thank my mother Jiping Sun for always loving me and believing in me. I could not become who I am without her. I would also like to thank my dear friends, Mengjia Chen, Guan'ao Yan, Yiling Chen, Xinzhou Ge, Kexin Li, Yuhao Yin, Gabriel Ruiz, Boliang Wu, Wenbin Guo, Siyuan Yu, Wenchang Chen, Meixi Lin, Zilin Miao, Ping Zhu, Zhuangzhuang Ding, Yilin Lyu, and Yuning Cao.

VITA

2013–2017	B.Eng. in Automation, Department of Automation, Tsinghua University.
2017–2023	Ph.D. Student, Department of Statistics, University of California, Los Angeles.

PUBLICATIONS

Sun, T.; Song, D.; Li, W.V.; Li, J.J. (2021). scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. Genome Biology 22, 163.

Sun, T.; Song, D.; Li, W.V.; Li, J.J. Simulating single-cell gene expression count data with preserved gene correlations by scDesign2. (2022). Journal of Computational Biology 29(1): 23-26.

Jiang, R.; **Sun, T.**; Song, D.; Li, J.J. (2022). Statistics or biology: the zero-inflation controversy about scRNA-seq data. Genome Biology 23, 31.

Song, D.; Wang, Q.; Yan, G.; Liu, T.; **Sun, T.**; Li, J.J. (2023). scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. Nature Biotechnology.

CHAPTER 1

Introduction

Single-cell RNA sequencing (scRNA-seq) stands at the forefront of genomic research, offering a revolutionary approach to scrutinize gene expression at a cellular level [1, 2]. Unlike conventional bulk RNA sequencing, scRNA-seq allows scientists to delve into the transcriptomic profiles of individual cells within a heterogeneous population. This technology enables the comprehensive examination of gene expression patterns across diverse cell types, revealing invaluable insights into cellular heterogeneity, developmental processes, disease mechanisms, and complex biological systems [3–8]. Due to its importance, numerous biological datasets have been generated [9–13], which calls for high-quality statistical analysis.

The raw data of a scRNA-seq experiment are short DNA sequences called "reads," which are sequenced fragments of amplified RNA molecules with cellular barcodes attached [14, 15]. After these reads are mapped to the reference genome and sorted to individual cells, a gene-by-cell matrix of expression count values can be obtained, which can be used for downstream statistical analysis. As shown in Table 1.1, a typical analysis procedure may involve the following tasks [16]: (1) cell/gene filtering and data normalization [17, 18], (2) visualization and dimensionality reduction [19–24], (3) cell-level analysis, e.g., clustering [25–27] or trajectory inference [28, 29], (4) gene-level analysis, e.g., differentially-expressed (DE) gene detection [30–33].

For each of the statistical tasks described above, many different computational methods have been developed [34, 35]. Therefore, method benchmarking naturally becomes necessary to illustrate different methods' strengths, weaknesses, and best application scenarios [36–40]. For this purpose, simulated data can be extremely helpful due to its ease of generation (little to no financial cost) and presence of ground truth information. Chapter 3 of this dissertation

describes scDesign2, a high-quality statistical simulator that generates realistic synthetic datasets [41]. From a methodological standpoint, the main innovation and advantage of scDesign2 is in its modeling of not only the gene marginal distributions but also the gene correlation structure. Due to this property, compared with existing simulators, scDesign2 can generate more realistic synthetic scRNA-seq datasets. We also demonstrate its use in (1) the benchmarking of two typical computational tasks, cell clustering and rare cell type detection, and (2) the design of scRNA-seq experiments in choosing the optimal cell number and total sequencing depth.

<table>
<tr><th>Task category</th><th>Task type</th><th>Example task(s)</th><th colspan="2">Projects in this dissertation</th></tr>
<tr><td rowspan="8">Data preprocessing</td><td rowspan="2">Quality control</td><td>Cell filtering</td><td rowspan="4"></td><td rowspan="8"></td></tr>
<tr><td>Doublet detection</td></tr>
<tr><td colspan="2">Data normalization</td></tr>
<tr><td>Data correction</td><td>Batch effect correction</td></tr>
<tr><td>Feature selection</td><td>Gene filtering</td><td rowspan="4">Chapter 3
scDesign2</td></tr>
<tr><td colspan="2">Visualization</td></tr>
<tr><td colspan="2">Dimesionality reduction</td></tr>
<tr><td rowspan="6">Cell-level analysis</td><td rowspan="3">Cluster analysis</td><td>Clustering</td><td rowspan="6">Chapter 4
scStructure</td></tr>
<tr><td>Compositional Analysis</td></tr>
<tr><td>Cluster Annotation</td></tr>
<tr><td rowspan="3">Trajectory analysis</td><td>Trajectory inference</td><td rowspan="3"></td></tr>
<tr><td>RNA velocity analysis</td></tr>
<tr><td>Cell potential analysis</td></tr>
<tr><td rowspan="3">Gene-level analysis</td><td colspan="2">Differentially-expressed (DE)
gene identification</td><td>Chapter 3
scDesign2</td><td>Chapter 2
Beta-Poisson
model</td></tr>
<tr><td colspan="2">Gene set enrichment analysis</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td colspan="2">Gene regulatory network analysis</td></tr>
</table>

Table 1.1: Typical computational tasks in scRNA-seq data analysis (adapted from Figures 1 and 5 of [16]) and their relationship to projects in this dissertation.

For cell-level analysis, two different types of tasks can be performed, which are cell clustering and trajectory inference. With clustering methods, cells can be partitioned into different cell types, representing discrete cellular heterogeneity [25–27]. In contrast, trajectory inference methods will order cells into single-branch or multiple-branch trajectories, representing continuous heterogeneity [28, 29]. These are two different interpretations of a given dataset. However, as of now, there is no principled way of choosing which type of method is more suitable to describe a given dataset. Chapter 4 of this dissertation describes two computa-

tional approaches designed to solve this problem. We demonstrate that Approach 2 based on data simulation gives more promising results compared to Approach 1.

Finally, for gene-level analysis tasks such as DE gene identification, statistical modeling of the gene expression distribution of a single gene is often needed. However, most existing statistical models used for gene-level analysis do not give insights into the underlying biological processes [30–33]. Chapter 2 of this dissertation examines the Beta-Poisson distribution, which is based on a two-state Markovian model of gene expression [42]. Data simulation results of this model based on three different parameter combinations show that it can be interpreted as one of the sources that contribute to the sparsity and zero inflation that is observed in real scRNA-seq datasets.

To summarize, during my doctoral studies, I have led the above three projects of statistical method development and applications. One common component in all three projects is data simulation. In Chapters 2-4, I will describe the details of these three projects. Unlike the organization of this chapter, the order of Chapters 2-4 is chosen as the chronological order of project completion, which is the same as my personal journey of research curiosity.

CHAPTER 2

Application of the Beta-Poisson model in single-cell gene expression data analysis

2.1 Introduction

The technology of single-cell RNA sequencing (scRNA-seq) has developed rapidly in recent years. From its first appearance in 2009 [43], the number of cells that can be profiled has increased to hundreds and thousands in a single biological sample [44, 45]. However, the analysis of scRNA-seq data remains difficult due to high technical and biological noise. Technical noise is prevalent because of limited RNA capture efficiency, PCR bias, sequencing noise (Poisson noise), limited sequencing depth, etc. Biological noise refers to the randomness in the underlying biological process that generates the mRNA. Both of these two sources of noise or variability are hard to model. For modeling technical noise, the difficulty is due to the complexity and certain limitations of the experiment. For modeling biological noise, the difficulty is due to the limited knowledge we have about the true biological process.

Empirically, in contrast to bulk RNA-seq data, a typical feature of scRNA-seq data is the zero mode in the distribution of the expression of a large proportion of genes. The zeros in the dataset could be due to technical noise, i.e., failure of detection, or due to true biological variability, i.e., truly no expression. A common approach of modeling would be to use a two-component mixture model, where one of the components accounts for the false zeros due to failure of detection, and the other component accounts for the part with true biological signals [32, 46–48]. However, for the second component of the model, choices like Poisson, negative binomial, or Gaussian after log-transformation lack biological justification and may not explain the true zeros in gene expression. In particular, the use of Poisson

and negative binomial distribution in the analysis of bulk RNA-seq data does not justify their correctness in the analysis of scRNA-seq data. The reason is that each measurement in a bulk RNA-seq experiment represents the average expression level of one gene among a group of cells, in contrast to the case of a scRNA-seq experiment, where each measurement represents the expression level of one gene in one individual cell.

Here, we discuss the Beta-Poisson hierarchical model, which is derived from a simplified mathematical model of gene expression. Compared to the Poisson model and the negative binomial model, it reflects some properties of the underlying gene expression process and can be interpreted biologically. We discuss some of its properties and some of its existing and further potential applications in the analysis of scRNA-seq data.

2.2 The Beta-Poisson hierarchical model

2.2.1 A two-state gene expression model

Many experiments have shown that gene expression among a homogeneous population of genetically identical cells is highly heterogeneous [49, 50]. Imaging technologies reveal that the transcription of a gene is a discontinuous process [51], where the gene switches between a state where it actively transcribes mRNA and another state where no transcription happens. The interval that the gene stays in the two states is irregular and can be described by a probabilistic model. In 1995, Peccoud and Ycart modeled the above process as a birth-and-death process in a Markovian environment and derived some mathematical results [42].

The set of reactions in gene expression is summarized in Figure 2.1. The time evolution of the gene expression process can be modeled as a continuous-time Markov chain. The state space [42] for this process is

$$S = \{(i, n) : i \in \{0, 1\}, n \in \mathbb{N}\}.$$

As can be seen, the state variable for this Markov process consists of two coordinates. The first coordinate i indicates the state of the gene. $i = 0$ when the gene is inactive and 1 when

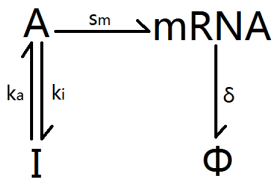

Figure 2.1: A two-state gene expression model.

active. The second coordinate n indicates the number of mRNA molecules.

We further assume that the time that the gene remains in the active state is exponentially distributed with parameter k_i ($1/k_i$ time units on average). Then it switches to the inactive state, and remains in the inactive state for a period of time exponentially distributed with parameter k_a. Only in the active state, the gene is transcribed to mRNA with rate s_m, and mRNA can be degraded in both states with rate δ for each molecule. In other words, the lifetime of each mRNA molecule is independent and identically distributed as an exponential random variable with rate parameter δ. The unit of these rate parameters is the number of chemical reactions per time unit (e.g. second or minute).

With the above assumptions, the Kolmogorov system of differential equations can be obtained, as summarized in equations (5) of [42]. In principle, by solving the set of equations with some initial condition, we can get $p_{i,n}(t)$ for any $t > 0$, which can further tell us the probability distribution of the number of mRNA molecules at any time t. In [42], Peccoud and Ycart introduced a set of moment-generating functions to achieve this goal:

$$G_0(z,t) = \sum_{n=0}^{\infty} z^n p_{0,n}(t); \qquad G_1(z,t) = \sum_{n=0}^{\infty} z^n p_{1,n}(t).$$

Further, define

$$G(z,t) = G_0(z,t) + G_1(z,t),$$

which is the moment-generating function of the number of mRNA molecules at time t. They showed that the cases of $\delta = 0$ and $\delta > 0$ need to be considered separately. The complete solution is only possible for the first case and in the latter and more biologically realistic case, only the result for the stationary distribution can be obtained.

In the last section of the paper, they proposed a moment-matching approach to estimate the parameters of the model when only data from the stationary distribution is available. However, in this case, as the model is time-homogeneous, multiplying the parameters by the same constant will give the same stationary distribution. Therefore, only three of the four parameters can be estimated. The time scale can be adjusted so that one of the parameters, for example, δ is set to 1. In this case, we can express the other three parameters as a function of the exponential moments e_1, e_2, and e_3, where

$$e_n = \mathbb{E}[X(X-1)\cdots(X-n+1)],$$

and

$$(k_a, k_i, s_m) = \phi(e_1, e_2, e_3),$$

where ϕ denotes the functional relationship for the two sets of parameters (the exact form of ϕ is shown in the last two pages of [42]). To estimate the parameters, we can estimate the exponential moments by the sample exponential moments and plug in the values to ϕ to get the estimates for k_a, k_i, and s_m. By the law of large numbers, these estimators are consistent.

This result is very useful in the analysis of single-cell gene expression data, especially for experiments where only the measurement at one time point is possible, which is the case for scRNA-seq and also for experiments based on single molecule fluorescence in situ hybridization (smFISH). The data can be assumed to be generated from the steady state distribution as the Markov process converges to the steady state distribution at an exponential rate related to the parameter values [42]. The development time of a biological tissue or the culture

time of a cell line has a much longer time scale.

2.2.2 The analytic form of the stationary distribution

For the asymptotic behavior of the model when $\delta > 0$, Peccoud and Ycart only derived the closed-form solution of the moment-generating functions. In 2006, Raj et al derived the analytic form of the stationary distribution [52],

$$\begin{aligned}\rho(n) =& \frac{\Gamma(\frac{k_a}{\delta})}{\Gamma(n+1)\Gamma(\frac{k_a}{\delta}+\frac{k_i}{\delta}+n)} \frac{\Gamma(\frac{k_a}{\delta}+\frac{k_i}{\delta})}{\Gamma(\frac{k_a}{\delta})} \left(\frac{s_m}{\delta}\right)^n \\ & {}_1F_1\left(\frac{k_a}{\delta}+n, \frac{k_a}{\delta}+\frac{k_i}{\delta}+n, -\frac{s_m}{\delta}\right),\end{aligned} \tag{2.1}$$

where n is a non negative integer and ${}_1F_1(a,b,c)$ is a confluent hypergeometric function of the first kind. They computed the MLE of the parameters using a numerical method without further showing the details. As can be seen, the closed-form solution is quite complicated and even the evaluation of its value given the parameters is not easy, because of the difficulty of the computation of the confluent hypergeometric function.

2.2.3 The Beta-Poisson hierarchical model

In 2013, Kim and Marioni pointed out that the steady-state distribution can be generated by a Beta-Poisson mixture model in the following way [53],

$$\begin{aligned} p|k_a, k_i &\sim \text{Beta}(k_a, k_i) \\ N|p, s_m &\sim \text{Poisson}(ps_m) \end{aligned}$$

where p is a hidden variable that follows a Beta distribution. The marginal distribution $p(N|s_m, k_a, k_i)$ takes the same form as equation (2.1). They further introduced priors to the model parameters and used Gibbs sampling to estimate the model parameters. Note that in this case, as mentioned before, the value of the parameter δ is set to be equal to 1, which makes the model identifiable.

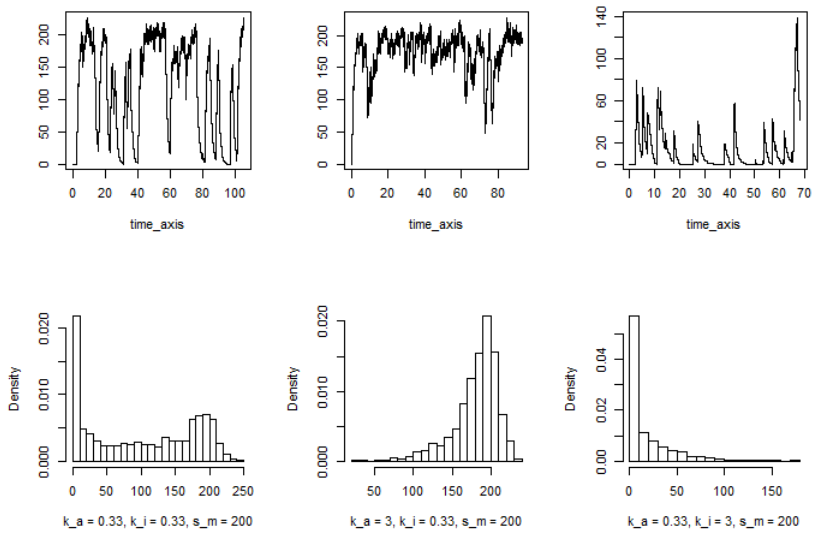

Figure 2.2: Three typical behaviors of the model, assuming $\delta = 1$.

2.3 Simulation and fitting to real data

To illustrate three typical behaviors of the model, a sample path and the steady state distribution are shown in Figure 2.2 for each of the three typical parameter ranges. Here the main focus is on the effect of the relative magnitude of k_a and k_i. In the first column, when k_a and k_i are both small, the gene has slow switching rates between the two states, and the mRNA number in a cell will oscillate between zero and around s_m/δ. For a group of homogeneous cells, the steady-state distribution of the number of mRNA of the gene becomes bimodal, due to the slow switching. In the second column, k_a is much larger than k_i, meaning that the gene stays in the active state most of the time and the oscillation of the number of mRNA is much less compared to the first scenario. The steady-state distribution peaks at around s_m/δ, which is the average number of mRNA if the gene is always active and is left skewed due to the smaller magnitude of k_i. In the last column, k_a is much smaller than k_i, meaning that the gene only activates occasionally and the expression process appears to be a sequence of infrequent pulses. The steady-state distribution peaks around zero and is right skewed. As can be seen here, the model is capable of describing different types of transcriptional behaviors as compared to the classical Poisson or negative binomial model.

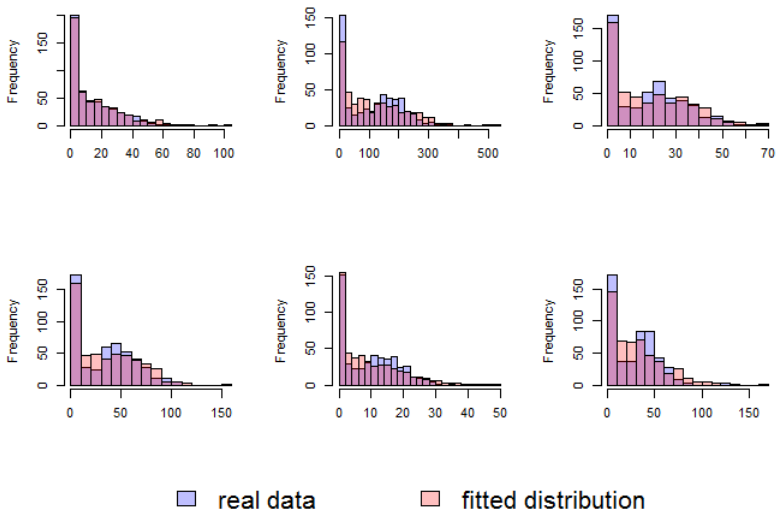

Figure 2.3: Comparison of the distribution of the real data and the fitted distribution.

To investigate how well the model can potentially fit real data, we look at a scRNA-seq dataset from the 10x genomics website. The dataset contains 1045 cells, 498 of which are from a human cell line (HEK293T) and 547 from a mouse cell line (NIH3T3). We filter for genes with mean count values greater than 10 and less than 500 and inspect how well the model can fit the data. We use the moment matching method described in section 2.2.1 to estimate the parameters. Some typical cases are shown in Figure 2.3. For some genes, the model can fit the data very well, as shown in the case of the first panel. However, for most genes, the model does not capture the bimodal shape of the real data very well. One of the reasons is that technical noise is not accounted for here. A simple modification would be to introduce an extra parameter for the proportion of false zeros. Also, normalized count should be a better choice for model fitting instead of the raw count that we are using here, because we need to adjust for factors like cell size and sequencing depth. With these modifications, we can expect that the Beta-Poisson model can be used as a base model for the marginal gene expression distribution of scRNA-seq data.

2.4 Existing applications and further improvements

The Beta-Poisson model was first introduced in the analysis of scRNA-seq data in 2013. Kim and Marioni applied this model to analyze a scRNA-seq dataset that contains the gene expression level in 12 mouse embryonic stem cells [53]. They estimated the parameters using the aforementioned Gibbs sampling method and found that the inferred parameters are consistent with RNA polymerase II binding and chromatin modifications. Their parameter inference procedure is very well specified. However, in their model, they didn't account for the technical noise that is prevalent in scRNA-seq datasets. Also, as the development of the scRNA-seq technology was just beginning at that time, their sample only contains 12 cells, which is a lot less compared to the throughput of the current technology [44, 45]. To develop a good parameter estimation method that is both accurate and fast when dealing with a large number of samples is a problem that needs to be solved.

In 2016, Vu et al used this model to analyze two datasets that were sequenced by the Fluidigm technology [54]. In their analysis, they included a parameter for the proportion of false zeros and also made some adjustments for normalized data. They also proposed a differential gene expression (DE) analysis method based on this model using the generalized linear model (GLM) type of formulation. Specifically, denote Y as the expression level of one gene, and x as the associated covariates. Define $\eta = x^{\mathrm{T}}\beta$, where β is the corresponding parameter vector for the covariates. Then the relationship between Y and x can be expressed in the following way,

$$\begin{aligned} g(\mathbb{E}Y) &= \eta \\ Y|\mathbb{E}Y &\sim \text{Beta - Poisson}(k_a, k_i, s_m) \\ \mathbb{E}Y &= f(k_a, k_i, s_m) \end{aligned}$$

where $f(.)$ is a function that specifies the relationship between $\mathbb{E}Y$ and the model parameters. By calculating the variance function and specifying the link function $g(.)$, we can estimate the parameters in this model and test if β or some components of it equals zero. Notice

that in this formulation, we are testing for whether there is an effect in the difference of the mean. However, as illustrated in Figure 2.2, the mean may not be a representative sample statistic if the expression of a gene is bimodal. Therefore, we might also consider testing for the effect on the difference in the parameter s_m, which is the synthesis rate of mRNA when the gene is in the active state. We can also test for the effect on the difference in k_a and k_i, which is the information that is uniquely provided by the Beta-Poisson model compared to models like Poisson or negative binomial. Based on the difference in the parameter values, we might be able to classify genes based on their transcriptional regulatory behavior and investigate if it is related to the functions of genes.

In the same year, Delmans and Hemberg used this model as the base model for one of the three methods to test for DE [55]. Their method is called D^{3}E, which is short for discrete distributional differential expression. They focused on evaluating whether there is an effect on the difference in the distribution rather than the mean. They used three different types of tests, the Cramer von Mises test, the Kolmogorov-Smirnov (KS) test, and the likelihood ratio test. They used the Beta-Poisson model as the base model for the likelihood ratio test for the overall effect. Their method can test for the difference in distribution, which reveals more information than testing for the difference in mean. However, they didn't develop a formal statistical test for the difference in the three individual parameters.

Aside from improving the computational efficiency of the estimation procedure and testing for DE, this model can also be used to solve many other problems. One important example is that as this model can describe bimodal expression, the proportion of false zeros produced from the experimental procedure can be estimated more accurately compared to using the Poisson or negative binomial model. This can give us a more accurate evaluation of the technical noise. Lastly, this model can also be used as the base model for dimensionality reduction and clustering. The zero-inflation feature of scRNA-seq datasets is one of the reasons that classical methods like PCA and k-means do not perform well on them. Current methods that take this feature into account cannot distinguish between true zeros and false zeros [23, 27].

2.5 Discussion

The Beta-Poisson hierarchical model is a good base model for the marginal distribution of single-cell gene expression, as it is derived from a mechanistic model of the gene expression process. Compared to classical models like Poisson and negative binomial, it has the capacity to model the bimodal expression of a gene and the model parameters can provide biological insights about the transcriptional behavior of a gene. It has already been applied in the analysis of scRNA-seq data, and it can certainly be used to answer other interesting questions.

Gene expression is a complicated process and this model certainly has its limitations. Although some experiments have shown that the three rate parameters of the model are independent of the development time of the cell and other extrinsic factors [51, 52], whether this is true for any cell populations is hard to determine. As Kærn et al pointed out, this model "is simple in comparison with the true complexity of gene expression. However, it has provided a good theoretical framework for understanding the effects of stochasticity on prokaryotic and eukaryotic gene expression" [56].

2.6 Acknowledgements

I would like to thank my advisor, Dr. Jingyi Jessica Li, for the overall guidance of this project and discussion on the potential further applications of this model.

CHAPTER 3

scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured

3.1 Introduction

The recent development of the single-cell RNA-seq (scRNA-seq) technologies has revolutionized transcriptomic studies by revealing the genome-wide gene expression levels within individual cells [57, 58]. In contrast to bulk RNA sequencing, scRNA-seq technology captures cell-specific transcriptome landscapes, which can reveal crucial information about cell-to-cell heterogeneity across different tissues, organs, and systems and enable the discovery of novel cell types and new transient cell states [9, 59–63]. Already, scRNA-seq technologies have led to breakthroughs in understanding biological processes such as stem cell differentiation and embryogenesis [64, 65], neurological disorders [66, 67], and tumorigenesis [68, 69].

Since the first scRNA-seq study was published in 2009 [70], many experimental protocols have been developed [71–73]. Broadly speaking, the existing protocols fall into two categories: tag-based and full-length [74]. Tag-based protocols (e.g., 10x Genomics [15], CEL-Seq2 [75], Drop-seq [76], and Seq-Well [77]) only capture and sequence one end of RNA transcripts, while full-length protocols (e.g., Smart-Seq2 [14], Fluidigm C1 [78], and MATQ-seq [79]) sequence fragments from full-length RNA transcripts [73, 80]. Typically, compared to full-length protocols (given the sequencing depth), tag-based protocols sequence more cells but with fewer transcripts captured per cell [81]. In addition to this cell-number vs. per-cell-depth trade-off, tag-based protocols use unique molecular identifiers (UMIs) to remove

polymerase chain reaction (PCR) amplification biases [82], while full-length protocols do not have this advantage and can only output reads without UMIs. Therefore, these protocols have different advantages in throughput (number of cells and number of genes captured) and accuracy (number of non-biological zeros and PCR biases) [73, 83, 84]. Moreover, when designing experiments, researchers often face the practical issue of having a limited budget. In this case, they need guidance to choose either sequencing more cells with fewer reads (or UMIs) in each cell, or sequencing fewer cells with more reads (or UMIs) in each cell [85–87].

In addition to selecting experimental protocols before conducting scRNA-seq experiments, a common challenge after collecting scRNA-seq data is to choose among the many available data analysis methods in an unbiased manner. For example, many algorithms have been developed for missing gene expression imputation [88, 89], dimensionality reduction [24, 90, 91], cell clustering [26, 27, 92, 93], rare cell type detection [94–96], differentially expressed gene identification [97–99], and trajectory inference [28, 29, 100–102]. Even though several benchmark and comparative studies have been carried out for common analysis tasks [36, 38, 103–105], most of them have only evaluated a subset of available computational methods using data from limited experimental protocols. Hence, they cannot meet the diverse needs of ongoing and future analyses of scRNA-seq data. In short, single-cell researchers lack a systematic and flexible approach to select appropriate computational methods for their specific data analysis needs.

One solution to the above two issues is to use *in silico* synthetic datasets, which carry ground truths (cell types, cell trajectories, differentially expressed genes, etc.) and do not induce extra experimental costs. Below we summarize six properties that an ideal simulator should achieve.

1. The simulator can be trained by real data so that it is adaptive to various experimental protocols and biological conditions.

2. The simulator can preserve genes so that its synthetic cells contain expression levels of real genes. The simulator should retain every gene's distribution of expression levels in its synthetic data without deleting genes in real data. This property is essential for

benchmarking differential gene expression analysis.

3. The simulator can capture gene correlations so that its synthetic data maintain a similar gene correlation structure to that in real data. This property relies on the last property and is essential for benchmarking multi-gene analyses such as cell dimensionality reduction (e.g., principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE) [20, 106], and uniform manifold approximation and projection (UMAP) [21, 22]), cell clustering, rare cell type detection, and cell trajectory inference.

4. The simulator can generate synthetic data with both varying cell number and sequencing depth, under the same biological condition of training data. This property is essential for guiding experimental design and benchmarking robustness of computational methods.

5. The simulator is transparent so that its model parameters can be easily understood and adjusted. For example, key statistical properties, such as every gene' expression mean, variance, zero proportions, and every gene pair's expression correlation, can be easily accessed from the model. This property is essential for model diagnostics and customized simulation. Specifically, with a transparent model, whenever the synthetic data do not resemble the real data, computational researchers can easily access how well the model fits to each gene's marginal distribution and what genes' correlations are well captured or missed. Moreover, a transparent model offers users an opportunity to generate data from their specified parameter values, e.g., gene expression means.

6. The simulator is computationally and sample efficient so that its training does not require expensive hardware, take excessive computational time, or rely on an enormous number of real cells to achieve good training. This property is essential for the simulator to be user-friendly and adaptive to full-length protocols that generate hundreds to thousands of cells, e.g., Fluidigm C1 and Smart-Seq2.

Although many simulators have been developed for scRNA-seq data and various method-

ological advances have been made [24, 87, 107–115], to the best of our knowledge, none of them achieves all the six properties. We summarize 14 representative simulators in Table 3.1. Except scGAN [111], these simulators all use probabilistic models or differential equations that are transparent and easy to fit, thus satisfying properties 5 and 6. However, scDesign [87], three simulators in the splatter package (splat simple, splat, and kersplat) [108], and SymSim [109] do not preserve genes, failing properties 2 and 3; ZINB-WaVE[1] [24, 108] and SPARSim [110] cannot vary cell number or sequencing depth, failing property 4; SERGIO [115] requires a user-specified gene regulatory network as input and does not estimate gene correlations from real data[2], thus not achieving property 3. Although scGAN preserves genes and uses a deep neural network to capture gene correlations, it cannot vary sequencing depth (not satisfying property 4), and its black-box nature, requirement for GPU, and long computational time make it fail properties 5 and 6. Hence, a simulator that achieves all the six properties is in demand.

Here we propose scDesign2 as the first simulator that achieves all the six properties and generates realistic synthetic data for multiple single-cell gene expression count-based technologies. Inheriting its name from our previous simulator scDesign, scDesign2 has achieved a significant methodological advance and become the first transparent simulator that reliably captures gene correlations. This advance is enabled by probabilistic modeling of not only marginal distributions of individual genes but also the joint distribution of thousands of genes. Thanks to its achievement of the six properties, scDesign2 will serve as a powerful tool for guiding experimental design and benchmarking computational methods in the single-cell transcriptomics field.

[1]ZINB-WaVE was not proposed as a simulator in its original publication [24] but was later implemented as a simulator in the splatter package [108].

[2]A quote from the SERGIO paper [115]: "It is worth noting here that several existing single-cell expression simulators employ a probabilistic model whose parameters are directly estimated from a real dataset and then sample synthetic data from the model. This approach is not feasible in SERGIO since the true GRN underlying the real dataset is unknown and notoriously hard to reconstruct, and the explicit use of a GRN is a crucial distinguishing feature of SERGIO. As such, SERGIO uses a randomly generated GRN to first synthesize clean expression data and uses the real dataset only in the second phase, to determine the extent of technical noise to add to the clean data."

3.2 Results

3.2.1 An overview of scDesign2

The statistical framework of scDesign2 consists of two steps: (1) model-fitting and (2) synthetic data generation (Fig. 3.1). In the model-fitting step, scDesign2 fits a multivariate generative model to a real scRNA-seq dataset. If the dataset contains more than one cell type (defined by marker genes or cell clustering; see Methods), then scDesign2 divides the dataset into subsets, one per cell type, and fits a cell-type-specific model to each subset. In the data-generation step, scDesign2 generates synthetic scRNA-seq data from the fitted model for each cell type.

The model-fitting step is composed of the following two sub-steps. First, scDesign2 fits a univariate count distribution to each gene's counts in cells of the same type. Four count distributions are considered: Poisson, zero-inflated Poisson (ZIP), negative binomial (NB), and zero-inflated negative binomial (ZINB), with the former three as special cases of the ZINB. All the four distributions have been widely used to model a gene's read or UMI counts in a homogeneous group of cells [24, 116–118]. From these four distributions, scDesign2 chooses one distribution for every gene in every cell type in a data-driven way. Second, scDesign2 captures the correlations of thousands of genes (all the moderately to highly expressed genes) by fitting a Gaussian copula in each cell type. We choose the Gaussian copula for its easiness to fit and good transparency, and we find it capturing gene correlations well (Fig. 3.2 and Supplementary Figs. 3.13–3.17).

As the first simulator that explicitly captures gene correlations, scDesign2 leverages a unique advantage of the copula framework: the separate modeling of each gene's marginal distribution and the correlation structure of thousands of genes together. This separation and its resulting flexibility are critical for scDesign2 to model single-cell gene expression count data generated by various experimental protocols. Thanks to this flexibility, scDesign2 can choose a count distribution from Poisson, ZIP, NB, and ZINB to fit each gene's expression counts and reveal biological insights of that gene's expression pattern.

3.2.2 Synthetic data generated by scDesign2 most resemble real scRNA-seq data in benchmarking against existing simulators

We benchmark scDesign2 against eight existing simulators—ZINB-WaVE, SPARSim, scGAN, scDesign, three variants of splat in the splatter package (splat simple, splat, and kersplat), and SymSim. We also compare scDesign2 with its own variant that only uses gene marginal distributions and no copula (w/o copula). Among these ten simulators, only scDesign2, its w/o copula variant, ZINB-WaVE, SPARSim, and scGAN preserve genes. We apply these ten simulators to four scRNA-seq datasets (in which cells are labelled with curated cell types) generated by different experimental protocols (10x Genomics [119], CEL-Seq2 [120], Fluidigm C1 [121], and Smart-Seq2 [122]). For each dataset, we randomly split its cells into two halves, with one half ("training data") to be used for training every simulator on each cell type individually and the other half ("test data") to serve as the benchmark standard to be compared with the synthetic data generated by each simulator.

We use three sets of benchmark analyses to compare synthetic data with the corresponding test data. Here is an overview. First, we select three cell types from each dataset (measured by each experimental protocol), obtaining a total of 12 cell-type–protocol combinations. For each combination, we evaluate eight key statistics: four gene-wise (expression mean, variance, coefficient of variation (cv), and zero proportion); two cell-wise (zero proportion and library size); two gene-pair-wise (Pearson correlation and Kendall's tau). (Note that we include Kendall's tau instead of Spearman rank correlation as a rank-based correlation statistic because Kendall's tau can account for ties.) For each statistic, we compare its empirical distribution—across genes (for gene-wise statistics), across cells (for cell-wise statistics), or across gene-pairs (for gene-pair-wise statistics)—in the test data with that in the synthetic data generated by each simulator. For the four gene-wise and two gene-pair-wise statistics, we also directly compare their values in the test data with those in the synthetic data generated by scDesign2, ZINB-WaVE, SPARSim, and scGAN—the four simulators that preserve genes. We cannot do this for the other simulators, because the values of these gene-related statistics are not comparable if the genes are not preserved. The re-

sults are summarized in Fig. 3.3 and Supplementary Figs. 3.18–3.40. Second, for each of the 12 cell-type–protocol combinations, we compare the gene correlation matrix estimated from the test data with that from the synthetic data generated by each simulator that preserves genes. We exclude the simulators that do not preserve genes because the gene expression matrices estimated from their synthetic data do not align with those from real data (i.e., the genes of the synthetic data matrix cannot be matched one-to-one to the genes of the training data matrix). The results are summarized in Fig. 3.2 and Supplementary Figs. 3.13–3.17. Third, for each of the four protocols, we use 2D visualization—t-SNE and PCA—to compare cells of multiple types in the test data and the synthetic data generated by each simulator that preserves genes. Again, we exclude the simulators that do not preserve genes because their synthetic cells cannot be combined with real cells for joint visualization (dimesionality reduction requires all cells to have the same original dimensions, i.e., genes). The results are summarized in Fig. 3.4 and Supplementary Figs. 3.41–3.43.

Overall, we find that the synthetic data generated by scDesign2 most resemble the test data for all four protocols. In our first set of analyses, we categorize the eight existing simulators into two types: simulators that preserve genes (ZINB-WaVE, SPARSim, and scGAN) and others. First, by comparing the distributions of eight key statistics between test data and synthetic data, we find that the simulators capable of preserving genes have overall better performance than other simulators, across cell types and protocols (Fig. 3.3a and Supplementary Figs. 3.18a–3.28a).

Second, we further benchmark the gene-preserving simulators by directly comparing their synthetic data and test data in terms of the gene-wise and gene-pair-wise statistics' values. Note that we cannot compare these statistics' values for simulators that do not preserve genes because the "genes" in those simulators' synthetic data cannot be matched to any genes in real data. In detail, we calculate the mean-squared errors (MSEs) of the four gene-wise statistics and the two gene-pair-wise statistics between test data and synthetic data generated by scDesign2, ZINB-WaVE, SPARSim, and scGAN. Fig. 3.3b shows that scGAN, a deep-learning-based method, consistently has the worst MSEs for all the six statistics. Due

to its long computational time[3], difficult implementation, and unsatisfactory performance, we exclude it from the following comparisons.

Out of 48 comparisons of gene-wise statistics (4 statistics times 12 cell-type–protocol combinations), scDesign2 achieves the best MSEs in 37 comparisons and demonstrates a clear advantage over ZINB-WaVE and SPARSim (Fig. 3.3b and Supplementary Figs. 3.18b–3.28b). Out of 24 comparisons of gene-pair-wise statistics (2 correlation statistics times 12 cell-type–protocol combinations) based on the 500 most highly expressed genes (in terms of their mean expression levels across cells) in each cell-type–protocol combination, scDesign2 achieves the best MSEs in 15 comparisons (Fig. 3.3b and Supplementary Figs. 3.18b–3.28b). We highlight the highly expressed genes because their Pearson correlations and Kendall's tau values are more biologically meaningful; in all cell-type–protocol combinations, the top 500 highly expressed genes, ranked by either mean expression levels or non-zero proportions across cells, explain at least 50% of reads or UMIs (Supplementary Figs. 3.29c–3.40c), confirming that these genes play dominant roles in transcriptional programs in cells. In addition, we include the comparison results based on more genes in Supplementary Figs. 3.29d&e–3.40d&e, which show that, as more lowly expressed genes are included, the MSEs of all these simulators decrease and become less distinguishable (because lowly expressed gene pairs have correlations close to zero in test data and all synthetic data), making the comparison less meaningful.

Third, we examine correlations of individual gene pairs and observe that scDesign2 can preserve strong negative gene correlations missed by ZINB-WaVE and SPARSim, which wrongly capture these correlations as weak or even positive (Fig. 3.3c-d and Supplementary Figs. 3.18c-d–3.28c-d). This observation is further confirmed by our second set of analyses below. Furthermore, we compare the relationships of three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) between test data and synthetic data generated by each simulator, and we find that scDesign2 better captures the relationships than existing simulators do across cell types and experimental protocols

[3]The training of scGAN takes 1-2 days (with NVIDIA GeFore GTX 2080 Ti GPU) on 255 cells and 15926 genes, in contrast to the other simulators that take at most minutes to train with CPU.

(Fig. 3.3e and Supplementary Figs. 3.18e–3.28e).

In our second set of analyses, we compare gene correlation matrices in terms of both Pearson correlation and Kendall's tau between test data and synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Heatmap visualization shows that scDesign2 captures gene correlations most accurately and consistently across cell types and experimental protocols (Fig. 3.2 and Supplementary Figs. 3.13–3.17). Notably, for highly expressed genes in Smart-Seq2 data, ZINB-WaVE and SPARSim miss almost all the gene correlations, while scDesign2 well preserves positive and negative gene correlations in its synthetic data (Fig. 3.2b & d and Fig. 3.17b & d).

In our third set of analyses, we use 2D visualization to compare cells in test data and those in synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Both t-SNE and PCA 2D plots show that cells in synthetic data generated by scDesign2 most resemble cells in test data (Fig. 3.4 and Supplementary Figs. 3.41–3.43). In particular, by overlaying real and synthetic cells in the same 2D plot, we find synthetic cells generated by scDesign2 least distinguishable from real cells. On the contrary, synthetic cells generated by ZINB-WaVE and SPARSim exhibit spurious patterns unseen in real cells.

To quantify the similarity between synthetic cells and real test cells, we use the median integration local inverse Simpson's index (miLISI) [123], whose value is between 1 and 2, with a larger value indicating a greater similarity. Specifically, we compute an integration local inverse Simpson's index (iLISI) to represent the effective number of cell labels (with 1 meaning synthetic or real cells only, and 2 meaning equal numbers of synthetic and real cells) in the local neighborhood of each (synthetic or real) cell; the closer iLISI is to 2, the more equal presence synthetic and real cells have in the local neighborhood. Taking the median of the iLISIs of all cells, we obtain the miLISI, which quantifies the overall mixing of synthetic cells with real cells. Using the R package LISI [123], we calculate the miLISI value for each of the overlaying 2D plots containing real and synthetic cells (Fig. 3.4 and Supplementary Figs. 3.41–3.43), and we find that scDesign2 consistently leads to the highest miLISI value, with greater advantages in 2D tSNE plots than 2D PCA plots. Since 2D t-SNE projection preserves cell clusters better than 2D PCA does and is more widely used

for visualizing single-cell gene expression data, our results suggest that the synthetic data by scDesign2 best capture the cluster structure in real cells. Together, the miLISI values confirm the superb performance and the realistic nature of scDesign2.

These three sets of analyses also verify the advantage of using copula in scDesign2. Compared with scDesign2, its w/o copula variant, as expected, cannot capture gene correlations at all (Fig. 3.3a, Fig. 3.2, Supplementary Figs. 3.18a–3.28a, and Supplementary Figs. 3.13–3.17). As a result, the synthetic data generated by the w/o copula variant do not resemble the corresponding real data in 2D visualization (Fig. 3.4 and Supplementary Figs. 3.41–3.43).

In addition to its realistic nature, scDesign2 also has two more unique advantages over ZINB-WaVE and SPARSim. Unlike the other two simulators, scDesign2 only considers genes as features and models their joint distribution, and it regards cells as observations instead of features. This formulation is aligned with the statistical thinking that genes are fixed quantities but cells are randomly sampled from a population of cells. Thanks to this principled formulation, scDesign2 can generate synthetic cells of any number, in contrast to ZINB-WaVE and SPARSim that can only generate the same number of synthetic cells as real cells. It is also worth noting that, although scDesign2 does not explicitly model the distribution of cell library sizes, it recovers that distribution rather faithfully (see the cell library size distributions in Fig. 3.3a and Supplementary Figs. 3.18a–3.28a). This is achieved by modeling joint gene distributions and accounting for gene correlations through the use of copula. Compared to scGAN, the training of scDesign2 is fast and does not rely on a large number of input real cells for good training quality.

3.2.3 Refinement of scDesign2 training: calibration of cell types by ROGUE scores

For a dataset containing multiple cell types, scDesign2 needs to fit a model to each cell type before generating synthetic data. To ensure the quality of its synthetic data, scDesign2 must have one of its count models (ZINB model and its three simplified variants) fit well to each gene's real expression levels in each cell type; otherwise, the synthetic data may

not well mimic real data due to the poorness of model fitting. We observe this issue in the 10x Genomics dataset (Fig. 3.4a), where some cell types such as transit-amplifying early (TA.Early) cells and goblet cells are composed of discrete sub-clusters in 2D tSNE illustration. As a result, some genes' expression levels within one of such cell types cannot be fit well by scDesign2's count models, leading to a discrepancy between synthetic data and real data (synthetic TA.Early and goblet cells do not appear to have cell sub-clusters in 2D tSNE illustration).

To address this issue, we calibrate each cell type using the ROGUE score [124], which measures the homogeneity of that cell type, before training scDesign2. Concretely, we first partition the cell type into sub-clusters using the Louvain clustering algorithm [25] in the Seurat R package [26]. Employing varying resolution parameters in the Louvain algorithm, we partition the cell type into a varying number of sub-clusters. Second, we calculate the ROGUE score of every sub-cluster, and then we compute the average ROUGE score across sub-clusters for each number of sub-clusters, ranging from 1 to 6. Third, we examine how the average ROGUE score increases as the number of sub-clusters increases (Fig. 3.5a), together with 2D t-SNE visualization (Fig. 3.5b), to determine an appropriate number of sub-clusters, which is usually the "elbow point" where the average ROGUE score saturates.

Applying this strategy to refining the six cell types in the 10x Genomics dataset, we observe that, after being trained with the refined cell types, scDesign2 generates more realistic synthetic data (Fig. 3.5c; the miLISI value increases from 1.596 to 1.779).

3.2.4 Application 1: scDesign2 generates realistic synthetic data for other single-cell expression count-based technologies

Beyond scRNA-seq data, we demonstrate that scDesign2 can also generate realistic synthetic data for other single-cell count-based technologies that do not necessarily use next-generation sequencing, as long as individual genes' count distributions can be well approximated by Poisson, ZIP, NB or ZINB. For instance, single-cell spatial transcriptomics technologies, usually based on fluorescence in situ hybridization (FISH), are known to yield Poisson or

NB distributed counts [125, 126]. The versatility of scDesign2 is endowed by its data-driven way of selecting marginal distributions for individual genes, regardless of each distribution being Poisson or NB, zero-inflated or not.

We demonstrate the accuracy of scDesign2 based on two single-cell spatial transcriptome datasets: one dataset of cells in the mouse hypothalamic preoptic region measured by multiplexed error robust fluorescence in situ hybridization (MERFISH) [127] and another dataset of cells in the mouse hippocampal area CA1 measured by probabilistic cell typing by in situ sequencing (pciSeq) [128], a newly developed spatial transcriptome profiling technology. Both datasets contain labeled cell types. Due to the lack of simulators specifically designed for single-cell spatial transcriptome data, we still benchmark scDesign2 against its w/o copula variant, as well as ZINB-WaVE and SPARSim, the two simulators that preserve genes. Note that for all the simulators considered, they only generate gene counts, not spatial coordinates, for synthetic cells. Similar to our previous analysis, for each cell type in each dataset, we randomly split the cells into two halves, with one half ("training data") to be used for training every simulator and the other half ("test data") to serve as the benchmark standard to be compared with the synthetic data generated by each simulator. Fig. 3.6 and Supplementary Fig. 3.44 demonstrate the 2D visualization of each real dataset, the corresponding synthetic data generated by each simulator, as well as the combination of test data and each synthetic dataset. For both technologies and in both t-SNE and PCA visualization, scDesign2 outperforms SPARSim and ZINB-WaVE by generating synthetic data that most resemble the real data. In particular, scDesign2 consistently achieves the highest miLISI values in the 2D visualization plots of combined data, indicating that the synthetic cells generated by scDesign2 are least distinguishable from real cells. These results confirm the versatility and robustness of scDesign2.

3.2.5 Application 2: scDesign2 guides experimental design and computational method benchmarking in cell clustering

Cell clustering is a ubiquitous computational task in single-cell research. Here we demonstrate how scDesign2 can guide experimental design (i.e., deciding the optimal cell number and sequencing depth) and benchmark computational methods for the cell clustering task.

After training scDesign2 on each of the four scRNA-seq datasets generated by different experimental protocols (10x Genomics [119], CEL-Seq2 [120], and Fluidigm C1 [121], Smart-Seq2 [122]), we apply the trained scDesign2 to generate synthetic data under three experimental design scenarios: (1) varying sequencing depths, where the total number of reads (or UMIs) varies but the cell number is fixed; (2) varying cell numbers, where the number of sequenced cells varies but the sequencing depth is fixed; (3) fixing the per-cell average sequencing depth, where the both the number of sequenced cells and the total sequencing depth vary, but the average number of reads (or UMIs) in each cell is fixed. For each protocol, scDesign2 generates a synthetic dataset per sequencing depth and cell number.

To guide the choices of sequencing depth and cell number based on clustering accuracy, we apply two popular scRNA-seq cell clustering methods—Seurat (the kNN-Jaccard-Louvain algorithm) [25, 26] and SC3 [27]—to the synthetic datasets and use the adjusted mutual information (AMI) [129] and the adjusted Rand index (ARI) [130] as two clustering accuracy measures. Note that SC3 can be specified to output the same number of cell clusters as the annotated cell types, while Seurat cannot due to the nature of the Louvain algorithm it uses [25]. The results are summarized in Figs. 3.7–3.9 and Supplementary Figs. 3.45–3.53.

For the first, varying-sequencing-depth scenario, we expect the clustering accuracy to improve as the sequencing depth increases, because a larger sequencing depth would better reveal every cell's transcriptome profile and thus lead to better clustering. Moreover, we also expect there to be a saturation effect: the clustering accuracy no longer improves much after the sequencing depth increases to a point, which we regard as the optimal sequencing depth that balances clustering accuracy and budget. The results confirm our expectation. For the two UMI-based protocols 10x Genomics and CEL-Seq2, we observe the improvement

and the saturation effect in clustering accuracy, based on both Seurat and SC3, as the sequencing depth increases. In detail, the saturation for 10x Genomics data occurs at 113.05 million UMIs for 3793 cells, while the real dataset has only 28.57 million UMIs (Fig. 3.7); the saturation for CEL-Seq2 data occurs at 42.72 million UMIs for 2279 cells, while the real dataset contains 172.14 million UMIs (Fig. 3.45). For the two non-UMI-based protocols Fluidigm C1 and Smart-Seq2, we observe the saturation effect even at the lowest sequencing depth we consider, likely due to the fact that these two protocols provide a deeper profiling of every cell than the UMI-based protocols do. In detail, the saturation for Fluidigm C1 data occurs at 26.74 (based on Seurat) or 110.52 (based on SC3) million reads for 317 cells, while the real dataset contains 869.24 million reads (Fig. 3.48); the saturation for Smart-Seq2 data occurs at 33.68 million reads for 1078 cells, based on both Seurat and SC3, while the real dataset contains 1074.97 million reads (Fig. 3.51). The t-SNE visualization supports the observed trends of clustering accuracy. In each t-SNE plot that corresponds to one sequencing depth and one set of cell clusters/types (by Seurat, SC3, or annotated cell types), synthetic cells are labelled by their cell clusters/types; contrasting a tSNE plot of cell clusters with that showing cell types illustrates clustering accuracy (Fig. 3.7a and Supplementary Figs. 3.45a, 3.48a, and 3.51a). In conclusion, for clustering purpose, we would recommend increasing the 10x Genomics sequencing depth to 113.05 million UMIs, if budget allows, and using SC3 for clustering; for CEL-Seq2, Fluidigm C1, and Smart-Seq2, we would recommend decreasing the sequencing depths to 42.72 million UMIs, 110.52 million reads, and 33.68 million reads, respectively, to save budget and using either Seurat or SC3 for clustering.

For the second, varying-cell-number scenario, we expect the clustering accuracy to first increase and then decrease as the cell number increases. The reason is that good clustering requires both a reasonable number of cells of each type and a clear-enough gene expression profile (where enough genes are captured) of every cell, thus posing a tradeoff—given the sequencing depth, the larger the cell number, the less clear each cell's profile would be. Hence, as the cell number increases from low, while every cell's profile is still clear, clustering accuracy increases; however, as the cell number reaches a point where every cell type has more than enough cells, further increasing the cell number would obscure every cell's profile and

deteriorate clustering accuracy. For the two UMI-based protocols 10x Genomics and CEL-Seq2, our expectation is confirmed: we observe an overall trend of clustering accuracy that first increases and then decreases (Fig. 3.8b and Supplementary Fig. 3.46b). In detail, for 10x Genomics data, both Seurat and SC3 have their accuracy maximized at 948 cells. This optimality is supported by the t-SNE visualization, which shows that the Seurat and SC3 cell clusters best agree with the annotated cell types at this optimal cell number (Fig. 3.8a). Hence, the real data cell number 3,793 is not optimal for distinguishing the annotated cell types by Seurat or SC3. For CEL-Seq2 data, Seurat and SC3 have optimal accuracy at 2,279 and 570 cells, respectively, also supported by the t-SNE visualization (Supplementary Fig. 3.46a). This suggests that the real data cell number 2,279 can lead to optimal cell clustering by Seurat. In contrast, for the two non-UMI-based protocols Fluidigm C1 and Smart-Seq2, we only observe a first-increasing-and-then-saturated trend of clustering accuracy as the cell number increases, without seeing the trend decreasing (except for SC3 on Smart-Seq2 data) (Supplementary Figs. 3.49b and 3.52b). A likely reason is that these two protocols can provide a clear profile of every cell up to a large cell number around 10,000 given their deep sequencing depths in real data (869.24 million reads in the Fluidigm C1 data and 1074.95 million reads in the Smart-Seq2 data). For both Seurat and SC3, the cell numbers at which their performance saturates are close to the cell numbers in real data: 317 cells in the Fluidigm C1 data and 1078 cells in the Smart-Seq2 data. In conclusion, we use scDesign2 to find that the cell numbers are close to being optimal in the CEL-Seq2, Fluidigm C1, and Smart-Seq2 datasets. For 10x Genomics, we would recommend decreasing the cell number to 948 cells (while keeping the sequencing depth at 28.58 million UMIs) to optimize the clustering accuracy by either Seurat or SC3.

For the third, fixing-average-sequencing-depth scenario, we expect the clustering accuracy to improve as the cell number (and also the total sequencing depth) increases, because more cells will make the identification of cell types easier. Moreover, we expect there to be a saturation effect: the clustering accuracy no longer improves much after the cell number increases to a point, which we regard as the optimal cell number that balances clustering accuracy and budget. The results confirm our expectation. In all four protocols, we observe

the expected trend of clustering accuracy for both Seurat and SC3, as well as the saturation effect, which is more obvious for Seurat. In detail, the saturation for 10x Genomics data occurs at 948 cells (based on Seurat), or 3793 cells (based on SC3), while the real dataset has 3,793 cells (Fig. 3.9); the saturation for CEL-Seq2 data occurs at 1,140 cells, while the real dataset has 2,279 cells (Fig. 3.47); the saturation for Fluidigm C1 data occurs at 317 cells (based on Seurat), which is the same cell number as the real dataset, while the optimal clustering accuracy occurs at 1,268 cells based on SC3 (Fig. 3.50); the saturation for Smart-Seq2 data occurs at 4,312 cells, while the real dataset has 1,078 cells (Fig. 3.53). In conclusion, when the average read (or UMI) count per cell is kept as fixed, for clustering purpose, we recommend keeping the cell number as in the original design for 10x Genomics and using SC3 for clustering; for CEL-Seq2, we recommend decreasing the cell number to 1,140 to save budget and using Seurat for clustering; for Fluidigm C1, if budget allows, we recommend increasing the cell number to 1,268 and using SC3 for clustering; for Smart-Seq2, if budget allows, we recommend increasing the cell number to 4,312 and using either Seurat or SC3 for clustering.

Beyond experimental design, scDesign2 also provides a comprehensive comparison of Seurat and SC3 across sequencing depths and cell numbers. Overall, both methods demonstrate superb accuracy in a wide range of sequencing depths and cell numbers for every protocol. At close-to-optimal sequencing depths and cell numbers for each method, SC3 has better accuracy than Seurat. However, Seurat and SC3 has different robustness: Seurat is a more robust method for 10x genomics data when the sequencing depth is too low or the cell number is too large (Figs. 3.7b–3.9b), while SC3 is more robust when the cell number is small for CEL-Seq2 (Supplementary Figs. 3.46b–3.47b), Fluidigm C1 (Supplementary Figs. 3.49b–3.50b), and Smart-Seq2 (Supplementary Figs. 3.52b–3.53b). This finding is consistent with the fact that SC3 is an ensemble method that is more robust against a small number of cells but cannot be easily scaled up when the cell number is too large.

3.2.6 Application 3: scDesign2 guides experimental design and computational method benchmarking in rare cell type detection

Rare cell type detection is another important application of scRNA-seq, whose high-throughput profiling of cells opens an unprecedented opportunity to identify unknown cell types that are often rare but critical. Here we demonstrate how scDesign2 can guide experimental design (i.e., deciding the optimal cell number and sequencing depth) and benchmark computational methods for the rare cell type detection task.

From the 10x Genomics dataset of mouse intestine epithelial tissue [119], we select six cell types—stem cells (Stem), goblet cells (Goblet), tuft cells (Tuft), early transit amplifying cells (TA.Early), enterocyte progenitors (EP), and early enterocyte progenitors (EP.Early), among which Tuft is the rare cell type [131] and has a proportion less than 5% among the six cell types. After training scDesign2 on this dataset, we use scDesign2 to generate synthetic data under three experimental design scenarios: (1) varying sequencing depths, where the total number of UMIs varies but the cell number is fixed; (2) varying cell numbers, where the number of sequenced cells varies but the sequencing depth is fixed; (3) fixing the per-cell average sequencing depth, where the both the number of sequenced cells and the total sequencing depth vary, but the average number of reads (or UMIs) in each cell is fixed. For every sequencing depth and cell number, scDesign2 generates a synthetic dataset.

To guide the choices of sequencing depth and cell number based on rare-cell-type detection accuracy, we apply two popular methods—FiRE [96] and GiniClust2 [95]—to the synthetic datasets and evaluate four accuracy measures: precision (the percentage of truly rare cells among the detected rare cells), recall (the percentage of detected rare cells among the truly rare cells), F1-score (the harmonic mean of the precision and recall), and AUPRC (the area under the precision-recall curve). Since GiniClust2 does not allow adjustment of its detection threshold, we cannot calculate its AUPRC. However, as most users of FiRE would stick with its default threshold, the AUPRC measure is not as informative as the other three measures from a user's perspective.

For the first, varying-sequencing-depth scenario, we expect that the detection accuracy

would improve as the sequencing depth increases and there would be a saturation effect, similar to our expectation for cell clustering. The detection accuracy of FiRE and GiniClust2 roughly confirm our expectation. Across twelve sequencing depths ranging from 1.76 to 3612.4 million UMIs (with the cell number fixed as 3793, the number of cells in real data), we observe an overall trend of increasing detection accuracy with few exceptions (Fig. 3.10). For FiRE, its accuracy exhibits saturation after the sequencing depth reaches 457.23 million UMIs (Fig. 3.10a & c), while for GiniClust2 the saturation occurs earlier at a sequencing depth of 113.05 million UMIs (Fig. 3.10b & d). The t-SNE visualization supports the observed trends of precision and recall. In each t-SNE plot that corresponds to one sequencing depth and one detection method (FiRE or GiniClust2), synthetic cells are labelled as one of four types: true positive (TP; the rare cells correctly detected), false positive (FP; the unrare cells falsely detected), false negative (FN; the rare cells falsely undetected), and true negative (TN; the unrare cells correctly undetected). The numbers of TP, FP, FN, and TN cells determine the precision and recall: a large precision requires many TP cells and few FP cells; a large recall requires many TP cells and few FN cells. Notably, the abnormal accuracy of GiniClust2 at 457.23 million UMIs (Fig. 3.10d) is explained by the t-SNE visualization (Fig. 3.10b), which shows that GiniClust2 misidentifies the second largest cell cluster as the rare cell type, leads to many FP and FN cells, and results in close to zero precision and recall. Combining the FiRE and GiniClust2 results, we conclude that the real data sequencing depth at 28.57 million UMIs for 3793 cells is not optimal for detecting the rare cell type Tuft (Fig. 3.10c–d). If budget allows, we would recommend increasing the sequencing depth to 113.06 million UMIs and use GiniClust2 to detect tuft cells.

For the second, varying-cell-number scenario, we expect the detection accuracy to first increase and then decrease as the cell number increases, similar to our expectation for cell clustering. Again, the detection accuracy of FiRE and GiniClust2 confirm our expectation. Across thirteen cell numbers ranging from 29 to 121,376 (with the sequencing depth fixed as 28.57 million UMIs, the same as in real data), we observe an overall trend of detection accuracy that first increases and then decreases (Fig. 3.11). For both FiRE and GiniClust2, their F1-scores are optimal at 1,896 cells (Fig. 3.11c–d). This optimality is supported by the

t-SNE visualization, which shows a plot of synthetic cells with TP, FP, FN, and TN labels for every cell number and each detection method (Fig. 3.11a–b). Hence, the real data cell number 3793 is not optimal for detecting tuft cells given the total sequencing depth of 28.57 million UMIs. If the detection of tuft cells is a primary goal and the sequencing depth cannot be increased due to budget constraints, we would recommend decreasing the cell number of 1896 cells and use GiniClust2 to detect tuft cells.

For the third, fixing-average-sequencing-depth scenario, we expect the detection accuracy to first increase and then saturate as the cell number increases, similar to our expectation for cell clustering. The detection accuracy of FiRE roughly confirms our expectation, while the detection accuracy of GiniClust2 deviates from this trend (Fig. 3.12). For FiRE, across thirteen cell numbers ranging from 29 to 121,376 (with the average sequencing depth fixed as 7.53k UMIs per cell, the same as in real data), the F1-score reaches an early local maximum at 474 cells, and then stays relatively stable. A similar trend can be seen for the other three accuracy measures: precision, recall, and AUPRC. For GiniClust2, across nine cell numbers ranging from 29 to 7,586, the F1-score reaches a global maximum at 237 cells, and then it decreases as the cell number further increases. This is mainly due to the increasing proportion of FPs in the discovered rare cells, as indicated by the plunging precision curve. The recall, on the other hand, stays relatively stable after the optimal cell number. The t-SNE visualization supports the observed trends of these accuracy measures. For example, we can see that for GiniClust2, when the cell number reaches 1000, more cells are labelled as FP, as shown in subpanels (4)-(6) of Fig. 3.12b. In summary, if the goal is to detect tuft cells and the average sequencing depth is fixed as 7.53k UMIs per cell, we recommend using GiniClust2 and decreasing the number of cells to 237.

In addition to assisting experimental design, scDesign2 also provides an objective comparison of FiRE and GiniClust2 across sequencing depths and cell numbers. Figs. 3.10–3.12 show that GiniClust2 has much better accuracy than FiRE at close-to-optimal sequencing depths and cell numbers. However, FiRE is a more robust method that it can successfully run at all sequencing depths and cell numbers, while GiniClust2 fails when the cell number is too small or too large (GiniClust3 may have addressed this large-cell-number issue [132]).

This finding is consistent with the methodological difference between the two methods: FiRE detects rare cells via an outlier detection approach, while GiniClust2 first performs cell clustering and then identifies the cells in small clusters as rare cells. The requirement of cell clustering explains why GiniClust2 fails when the cells exhibit no clear clusters and why it works well when rare cells form small clear clusters. In contrast, outlier detection has no requirement on cluster patterns, and this explains why FiRE is robust.

3.3 Discussion

In this article, we propose scDesign2, a transparent simulator for single-cell gene expression count data. Our development of scDesign2 is motivated by the pressing challenge to generate realistic synthetic data for various scRNA-seq protocols and other single-cell gene expression count-based technologies. Unlike existing simulators including our previous simulator scDesign, scDesign2 achieves six properties: protocol adaptiveness, gene preservation, gene correlation capture, flexible cell number and sequencing depth choices, transparency, and computational and sample efficiency. This achievement of scDesign2 is enabled by its unique use of the copula statistical framework, which combines marginal distributions of individual genes and the global correlation structure among genes. As a result, scDesign2 has the following methodological advantages that contribute to its high degree of transparency. First, it selects a marginal distribution from four options (Poisson, ZIP, NB, and ZINB) for each gene in a data-driven manner to best capture and summarize the expression characteristics of that gene. Second, it uses a Gaussian copula to estimate gene correlations, which will be used to generate synthetic single-cell gene expression counts that preserve the correlation structures. Third, it can generate gene expression counts according to user-specified sequencing depth and cell number.

We have performed a comprehensive set of benchmarking and real data studies to evaluate scDesign2 in terms of its accuracy in generating synthetic data and its efficacy in guiding experimental design and benchmarking computational methods. Based on four scRNA-seq protocols and 12 cell types, our benchmarking results demonstrate that scDesign2 better

captures gene expression characteristics in real data than eight existing scRNA-seq simulators do. In particular, among the four simulators that aim to preserve gene correlations, scDesign2 achieves the best accuracy. Moreover, we demonstrate the capacity of scDesign2 in generating synthetic data of other single-cell count-based technologies including MERFISH or pciSeq, two single-cell spatial transcriptomics technologies. After validating the realistic nature of synthetic data generated by scDesign2, we use real data applications to demonstrate how scDesign2 can guide the selection of cell number and sequencing depth in experimental design, as well as how scDesign2 can benchmark computational methods for cell clustering and rare cell type identification.

Since scRNA-seq data typically contain tens of thousands of genes, the estimation of the copula gene correlation matrix is a high dimensional problem. This problem can be partially avoided by only estimating the copula correlation matrix of thousands of moderately to highly expressed genes. We use a simulation study to demonstrate why this approach is reasonable (Supplementary Figs. 3.54 and 3.55), and a more detailed discussion is in the Methods section. To summarize, the simulation results suggest that, to reach an average estimation accuracy of ± 0.3 of true correlation values among the top 1000 highly expressed genes, at least 20 cells are needed. To reach an accuracy level of ± 0.2 for the top 1500 highly expressed genes, at least 50 cells are needed. With 100 cells, an accuracy level of ± 0.1 can be reached for the top 200 highly expressed genes, and a slightly worse accuracy level can be reached for the top 2000 genes.

In the implementation of the `scDesign2 R` package, we control the number of genes for which copula correlations need to be estimated by filtering out the genes whose zero proportions exceed a user-specified cutoff. For all the results in this paper, the cutoff is set as 0.8. In Supplementary Table S3.2, we summarize the number of cells (n), i.e., the sample size, and the number of genes included for copula correlation estimation (p) in each of the 12 datasets used for benchmarking simulators. Based on Supplementary Figs. 3.54 and 3.55, we see that p appears to be too large for the CEL-Seq2, Fluidigm C1, and Smart-Seq2 datasets. This suggests that the results in this paper may be further improved by setting a more stringent cutoff for gene selection.

For future methodological improvement, there are other ways to address this high-dimensional estimation problem. For example, we can consider implementing sparse estimation (e.g., [133]) for the copula correlation matrix. Moreover, we can build a hierarchical model to borrow information across cell types/clusters. This will be useful for improving the model fitting for small cell types/clusters that may share similar gene correlation structures.

The current implementation of scDesign2 is restricted to single-cell datasets composed of discrete cell types, because the generative model of scDesign2 assumes that cells of the same type follow the same distribution of gene expression. However, many single-cell datasets exhibit continuous cell trajectories instead of discrete cell types. A nice property of the probabilistic model used in scDesign2 is that it is generalizable to account for continuous cell trajectories. First, we can use the Generalized Additive Model (GAM) [134–136] to model each gene's marginal distribution of expression as a function of cell pseudotime, which can be computationally inferred from real data [28, 29, 100]. Second, the copula framework can be used to incorporate gene correlation structures along the cell pseudotime. Combining these two steps into a generative model, this extension of scDesign2 has the potential to overcome the current challenge in preserving gene correlations encountered by existing simulators for single-cell trajectory data, such as Splatter Path [108], dyngen [137], and PROSSTT [107]. Another note is that scDesign2 does not generate synthetic cells based on outlier cells that do not cluster well with any cells in well-formed clusters. This is not necessarily a disadvantage, neither is it a unique feature to scDesign2. In fact, all model-based simulators that learn a generative model from real data must ignore certain outlier cells that do not fit well to their model. Some outlier cells could either represent an extremely rare cell type or are just "doublets" [138–141], artifacts resulted from single-cell sequencing experiments. Hence, our stance is that ignorance of outlier cells is a sacrifice that every simulator has to make; the open question is the degree to which outlier cells should be ignored, and proper answers to this question must resort to statistical model selection principles.

Regarding the use of scDesign2 to guide the design of scRNA-seq experiments, although scDesign2 can model and simulate data from different scRNA-seq protocols and other single-cell expression count-based technologies, the current scDesign2 implementation is not yet

applicable to cross-protocol data generation (i.e., training scDesign2 on real data of one protocol and generating synthetic data for another protocol) because of complicated differences in data characteristics among protocols. To demonstrate this issue, we use a multi-protocol dataset of peripheral blood mononuclear cells (PBMCs) generated for benchmarking purposes [74]. We select data of five cell types measured by three protocols, 10x Genomics, Drop-Seq, and Smart-Seq2, and we train scDesign2 on the 10x Genomics data. Then we adjust the fitted scDesign2 model for the Drop-Seq and Smart-Seq2 protocols by rescaling the mean parameters in the fitted model to account for the total sequencing depth and cell number, which are protocol-specific (see Methods for details). After the adjustment, we use the model for each protocol to generate synthetic data. Supplementary Fig. 3.56 illustrates the comparison of real data and synthetic data for each protocol. From the comparison, we observe that the synthetic cells do not mix well with the real cells for the two cross-protocol scenarios; only for 10x Genomics, the same-protocol scenario, do the synthetic cells mix well with the real cells.

To further illustrate the different data characteristics of different protocols, we compare individual genes' mean expression levels in the aforementioned three protocols. We refer to Drop-Seq and Smart-Seq2 as the target protocols, and 10x Genomics as the reference protocol. First, we randomly partition the two target-protocol datasets and the reference-protocol dataset into two halves each; we repeat the partitions for 100 times and collect 100 sets of partial datasets, with each set containing two target-protocol partial datasets (one Drop-Seq and one Smart-Seq2) and two reference-protocol partial datasets (split from the 10x-Genomics dataset)—one of the latter is randomly picked and referred to as the "reference data." Second, For every gene in each cell type, we take each set of partial datasets and compute two cross-protocol ratios, defined as the gene's mean expression levels in the target-protocol partial datasets divided by its mean expression level in the reference data, and a within-protocol ratio, defined as the ratio of the gene's mean expression level in the other reference-protocol partial dataset divided by that in the reference data; together, with the 100 sets of partial dataset, every gene in each cell type has 100 ratios for each of the two cross-protocol comparisons and 100 ratios for the within-protocol comparison. We apply this

procedure to the top 50 and 2000 highly expressed genes in five cell types. Supplementary Figs. 3.57 and 3.58 show that, with the within-protocol ratios as a baseline control for each cell type and each target protocol, the cross-protocol ratios exhibit a strongly gene-specific pattern; moreover, there is no monotone relationship between the cross-protocol ratios and the mean expression levels of genes. This result confirms that there does not exist a single scaling factor to convert all genes' expression levels from one protocol to another. However, an interesting phenomenon is that, for each target protocol, the cross-protocol ratios have similar patterns across cell types. This phenomenon sheds light on a future research direction of cross-protocol simulation for the cell types that exist in only one protocol, if the two protocols have shared cell types. In this scenario, we may train a model for each cell type in each protocol, learn a gene-specific but cell-type-invariant scaling factor from the shared cell types, and simulate data for the cell types missing in one protocol.

We note that the above analysis is only conducted for the genes' mean expression levels. The difficulty of cross-protocol simulation is in fact even larger because realistic simulation requires the rescaling of the other distributional parameter(s) in a two-parameter distribution such as NB and ZIP or a three-parameter distribution such as ZINB. Existing work has provided extensive empirical evidence on the vast differences between protocols in terms of data characteristics [26, 123].

In applications 2 and 3, we have demonstrated how to use scDesign2 to guide experimental design and benchmark computational methods for the tasks of cell clustering and rare cell type detection. Note that in these analyses, the optimized sequencing depths and cell numbers are only applicable to the same experimental protocols and biological samples. Yet, this limitation does not disqualify scDesign2 as a useful tool to guide experimental design. For example, researchers usually perform a coarse-grained, low-budget experiment to obtain a preliminary dataset, and then they may use scDesign2 to guide the optimal design of the later, more refined experiment. As another example, if scRNA-seq data need to be collected from many individuals, researchers usually first perform a pilot study on a small number of individuals. Then they may train scDesign2 using the pilot data to guide the design of the subsequent, large-scale experiments. Moreover, in addition to guiding the experimental

design, scDesign2 is useful as a general benchmarking tool for various experimental protocols and computational methods. For example, the analyses we performed in applications 2 and 3 are easily generalizable to other computational methods for a more comprehensive benchmarking.

Although we only use cell clustering and rare cell type detection to demonstrate scDesign2's use in guiding experimental design and benchmarking computational methods, we want to emphasize that scDesign2 has broad applications beyond these two tasks. Inheriting the flexible and transparent modeling nature of our previous simulator scDesign, scDesign2 can also benchmark other computational analyses we have demonstrated in our scDesign paper [87], including differential gene expression analysis and cell dimensionality reduction. Moreover, beyond its role as a simulator, scDesign2 may benefit single-cell gene expression data analysis by providing its estimated parameters about gene expression and gene correlations. Here we discuss three potential directions. First, scDesign2 can assist differential gene expression analysis. Its estimated marginal distributions of individual genes in different cell types can be used to investigate more general patterns of differential expression (such as different variances and different zero proportions), in addition to comparing gene expression means between two groups of cells [142]. Second, its estimated gene correlation structures can be used to construct cell-type-specific gene networks [143] and incorporated into gene set enrichment analysis to enhance statistical power [144, 145]. Third, scDesign2 has the potential to improve the alignment of cells from multiple single-cell datasets [146]. Its estimated gene expression parameters can guide the calculation of cell type or cluster similarities between batches, and its estimated gene correlation structures can be used to align cell types or clusters across batches based on the similarity in gene correlation structures. [147].

3.4 Methods

3.4.1 The statistical framework of scDesign2

3.4.1.1 Fitting a generative model of single-cell gene expression count data with gene correlations

Given an scRNA-seq count matrix $\mathbf{X} \in \mathbb{N}^{p \times n}$ with p genes and n cells, we assume that the n cells belong to K cell types and that the cell memberships have been assigned by clustering, labeled by marker genes, or known in advance. (For input data without pre-defined cell types, our recommendation for cell clustering is in two subsections.) Our goal is to fit a parametric count model to characterize the joint distribution of genes' counts in each cell type. For cell type k, We denote its number of cells by $n^{(k)}$, its count sub-matrix by $\mathbf{X}^{(k)}$, and its set of model parameters by $\Theta^{(k)}$, $k = 1, \ldots, K$. For simplicity of notation, we drop the superscript (k) in the following discussion about the generative model for one single cell type.

We denote $X_{\cdot j} = (X_{1j}, \ldots, X_{pj})^\mathsf{T} \in \mathbb{R}^p$ as a random p-dimensional gene count vector in cell j, $j = 1, \ldots, n$. We denote its realization, i.e., the observed gene count vector as the j-th column in $\mathbf{X}$, by $x_{\cdot j} = (x_{1j}, \ldots, x_{pj})^\mathsf{T}$. Jointly for the p genes, we assume that $X_{\cdot j}$ independently follows a p-dimensional distribution F, which we will specify by a copula in the next paragraph. Marginally for each gene i, we assume that X_{ij} independently follows a univariate count distribution F_i. For example, if F_i is the ZINB distribution, we write $X_{ij} \overset{\text{ind}}{\sim} \text{ZINB}(p_i, \psi_i, \mu_i)$, which can be interpreted as a hierarchical model: (1) $Z_{ij} \overset{\text{ind}}{\sim} \text{Ber}(p_i)$ is a hidden latent variable indicating whether gene i drops out in cell j; (2) $X_{ij}|Z_{ij} \overset{\text{ind}}{\sim} 1_0 Z_{ij} + \text{NB}(\psi_i, \mu_i)(1 - Z_{ij})$, where 1_0 indicates a point mass at 0. That is,

$$\mathbb{E}(X_{ij}|Z_{ij} = 0) = \mu_i\,, \quad \text{Var}(X_{ij}|Z_{ij} = 0) = \mu_i + \frac{\mu_i^2}{\psi_i}\,.$$

Note that the Z_{ij}'s are unobserved and introduced only to describe the zero-inflation component. The Poisson, the zero-inflated Poisson (ZIP), and the negative binomial (NB) distributions are three special cases of the ZINB distribution, where $p_i = 0$ for Poisson and

NB, and $\psi_i = \infty$ for Poisson and ZIP. From these four distributions, scDesign2 automatically chooses the one that best fits to gene i's observed counts. Specifically, for the i-th row of $\mathbf{X}$, $x_{i\cdot} = (x_{i1}, \ldots, x_{in})^\mathsf{T}$, if its sample mean $\bar{x}_{i\cdot} = n^{-1} \sum_{j=1}^{n} x_{ij} \geqslant$ its sample variance $\hat{\sigma}_i^2 = (n-1)^{-1} \sum_{j=1}^{n} (x_{ij} - x_{i\cdot})^2$, i.e., there is no over-dispersion, scDesign2 fits the Poisson and the ZIP distributions separately to $x_{i\cdot}$ by maximum likelihood estimation (MLE), and then performs a likelihood ratio test with χ_1^2 as the null distribution to determine if zero-inflation is significant, i.e., the ZIP distribution should be chosen over the Poisson distribution. Otherwise if there is over-dispersion, i.e., $\bar{x}_{i\cdot} < \hat{\sigma}_i^2$, scDesign2 fits the NB and the ZINB distributions separately to $x_{i\cdot}$ by MLE and then performs a likelihood ratio test with χ_1^2 as the null distribution to determine if zero-inflation is significant, i.e., the ZINB distribution should be chosen over the NB distribution. The default significance level (i.e., p-value cutoff) for both tests is 0.05.

After estimating the marginal distributions of the p genes, i.e., $F_1, \ldots, F_p$, scDesign2 uses a copula to model the joint p-dimensional distribution F. A copula is defined as a joint cumulative distribution function (CDF), $C(\cdot) : [0,1]^p \to [0,1]$, which includes p uniform marginal distributions on $[0,1]$. That is, C is the CDF of a random vector $U = (U_1, \ldots, U_p)^\mathsf{T} \in [0,1]^p$, with $U_i \sim \text{Uniform}[0,1]$, $i = 1, \ldots, p$. For cell j's gene count vector $X_{\cdot j} \in \mathbb{R}^p$, although its i-th component X_{ij} may not follow the Uniform$[0,1]$ distribution, we can transform X_{ij} by applying the marginal CDF F_i so that $F_i(X_{ij}) \sim \text{Uniform}[0,1]$. This allows us to write the joint CDF F as

$$F(x_{1j}, \ldots, x_{pj}) = C(F_1(x_{1j}), \ldots, F_p(x_{pj}))\,,$$

which is decomposable into the copula C and the marginal distributions $F_1, \ldots, F_p$. Sklar's theorem states that such a decomposition exists uniquely for any continuous distribution F [148]. If F is discrete in any dimension, the copula C still exists but may not be unique, i.e., not identifiable [149, 150]. To resolve this unidentifiability issue, scDesign2 uses the technique of distributional transform [151]: first draw $V_{ij} \sim \text{Uniform}[0,1]$ independently for

$i = 1, \ldots, p$ and $j = 1, \ldots, n$; second define U_{ij} as

$$U_{ij} = (1 - V_{ij})F_i(X_{ij} - 1) + V_{ij}F_i(X_{ij})\,. \tag{3.1}$$

The effect of this transform is illustrated in Supplementary Fig. 3.59. Essentially, for a discrete random variable X_{ij} with CDF F_i, this transform distributes the non-zero probability mass X_{ij} has at every value x uniformly to the interval $[x, x + 1)$, thus transforming the discrete CDF F_i to a continuous CDF $\tilde{F}_i$ as

$$\tilde{F}_i(y) = F_i(\lfloor y \rfloor - 1) + (y - \lfloor y \rfloor)\left(F_i(\lfloor y \rfloor) - F_i(\lfloor y \rfloor - 1)\right)\,,$$

where $\lfloor y \rfloor$ denotes the largest integer no greater than y.
With V_{ij} and X_{ij}, if we define

$$\tilde{X}_{ij} = X_{ij} + V_{ij}\,, \tag{3.2}$$

then the probability density function of $\tilde{X}_{ij}$ is

$$\tilde{f}(y) = \Pr(X_{ij} = \lfloor y \rfloor, V_{ij} = y - \lfloor y \rfloor) = \Pr(X_{ij} = \lfloor y \rfloor) = F_i(\lfloor y \rfloor) - F_i(\lfloor y \rfloor - 1)\,,$$

and the CDF of $\tilde{X}_{ij}$ is

$$\int_{-\infty}^{y} \tilde{f}(t)dt = F_i(\lfloor y \rfloor - 1) + (y - \lfloor y \rfloor)\left(F_i(\lfloor y \rfloor) - F_i(\lfloor y \rfloor - 1)\right)\,.$$

Hence, $\tilde{X}_{ij} \sim \tilde{F}_i$; that is, the continuous random variable $\tilde{X}_{ij}$ constructed from X_{ij} and V_{ij} follows $\tilde{F}_i$. Defining $U_{ij} = \tilde{F}_i(\tilde{X}_{ij})$, we have $U_{ij} \sim \text{Uniform}[0, 1]$ and

$$\begin{aligned} U_{ij} &= \tilde{F}_i(X_{ij} + V_{ij}) = F_i(X_{ij} - 1) + V_{ij}\left(F_i(X_{ij}) - F_i(X_{ij} - 1)\right) \\ &= (1 - V_{ij})F_i(X_{ij} - 1) + V_{ij}F_i(X_{ij})\,, \end{aligned}$$

which is (3.1). This proves that U_{ij} constructed by (3.1) follows Uniform$[0, 1]$ and is thus desirable.

After this transform, the CDF F of $X_{\cdot j}$ is defined as the copula C of $U_{\cdot j} = (U_{1j}, \ldots, U_{pj})^\mathsf{T}$:

$$F(x_{1j}, \ldots, x_{pj}) = C(u_{1j}, \ldots, u_{pj}) \,,$$

where $(u_{1j}, \ldots, u_{pj})^\mathsf{T}$ is a realization of $(U_{1j}, \ldots, U_{pj})^\mathsf{T}$. In scDesign2, we choose C as the Gaussian copula. Denoting by Φ the CDF of a standard Gaussian distribution, we define F as

$$F(x_{1j}, \ldots, x_{pj}) = \Phi_p(\Phi^{-1}(u_{1j}), \ldots, \Phi^{-1}(u_{pj}); \mathbf{R})$$

where $\Phi_p(\cdot; \mathbf{R})$ is the CDF of a p-dimensional Gaussian distribution with a zero mean vector and a covariance matrix that is equal to the correlation matrix $\mathbf{R}$. If we denote R_{hl} as the Gaussian copula correlation between genes h and l, i.e., the (h, l)-th entry of $\mathbf{R}$, and τ_{hl} as the Kendall's tau between the same two genes on the original scale, i.e., $\tau_{hl} = \tau(X_{hj}, X_{lh})$, then we have the following relationship [152, 153],

$$R_{hl} = \sin\left(\frac{\pi}{2}\tau_{hl}\right).$$

This relationship links the copula correlation with the Kendall's tau of the two original variables, thus providing an interpretation of the copula correlation. It also suggests that $\mathbf{R}$ can be estimated by plugging the sample tau matrix into the above formula; however, this estimate of R may not always be positive semidefinite [154, 155]. Therefore, we use another procedure to estimate $\mathbf{R}$.

Denote by $(\hat{p}_i, \hat{\psi}_i, \hat{\mu}_i)$ the estimated parameters of F_i, which specify a fitted marginal distribution $\widehat{F}_i$. We sample v_{ij}^* from Uniform$[0, 1]$ independently for $i = 1, \ldots, p$ and $j = 1, \ldots, n$, and we calculate u_{ij}^* as

$$u_{ij}^* = v_{ij}^* \widehat{F}_i(x_{ij} - 1) + (1 - v_{ij}^*)\widehat{F}_i(x_{ij}) \,.$$

Then we estimate $\mathbf{R}$ by the sample covariance matrix $\widehat{\mathbf{R}}$ of $\left(\Phi^{-1}(u_{1j}^*), \ldots, \Phi^{-1}(u_{pj}^*)\right)^\mathsf{T}$, $j =$

$1, \ldots, n$.

As a side note, since this estimation procedure requires the random sampling of v_{ij}^*'s, it introduces additional randomness into the estimation of $\mathbf{R}$; that is, $\widehat{\mathbf{R}}$ is not a deterministic function of data. However, this additional randomness has a negligible effect on the synthetic data. As demonstrated in Supplementary Fig. 3.60, the gene correlation matrices estimated from synthetic data generated by scDesign2, with $\widehat{\mathbf{R}}$ estimated under two different random samples of v_{ij}^*'s, are very similar to each other.

To summarize, scDesign2 first estimate the marginal distributions $F_1, \ldots, F_p$ as $\widehat{F}_1, \ldots, \widehat{F}_p$, each of which may be a fitted Poisson, ZIP, NB, or ZINB distribution. Then scDesign2 calculates u_{ij}^*'s as described above and estimates a $p \times p$ covariance matrix as $\widehat{\mathbf{R}}$. Finally, scDesign2 estimates the p-dimensional joint distribution F as

$$\widehat{F}(x_{1j}, \ldots, x_{pj}) = \Phi_p(\Phi^{-1}(u_{1j}^*), \ldots, \Phi^{-1}(u_{pj}^*); \widehat{\mathbf{R}}) \,,$$

whose estimated model parameters are $\widehat{\Theta} = \{\hat{p}_1, \hat{\psi}_1, \hat{\mu}_1, \ldots, \hat{p}_p, \hat{\psi}_p, \hat{\mu}_p, \widehat{\mathbf{R}}\}$.

As a practical note, since the data matrix $\mathbf{X}$ typically contains tens of thousands of genes, if the sample size, i.e., the number of cells is not large enough, the estimation of the copula correlation matrix can be problematic [133]. Moreover, many genes are too lowly expressed to be detected in scRNA-seq data, making their correlations uninteresting to estimate. For these two reasons, we argue that the copula correlations should only be estimated for a subset of moderately to highly expressed genes.

In Supplementary Figs. 3.54 and 3.55, we analyze how n (the sample size, i.e., the number of cells) and p (the number of top expressed genes included) affect the estimation of the copula correlation matrix. We use two example datasets: the stem cell data generated by the 10x Genomics protocol and the dendrocyte (subtype 1) data generated by the Smart-Seq2 protocol. For each dataset, we extract the fitted Gaussian copula model for the top 2000 genes with the highest mean expression levels, and we use this model as the ground truth model to generate 1000 samples with a varying n. Then we estimate the copula correlation matrix of a varying p from each sample. For computational efficiency, we use the plug-in

estimation method based on sample tau values: $\hat{R}_{hl} = \sin\left(\frac{\pi}{2}\hat{\tau}_{hl}\right)$. Finally, we calculate the mean squared error (MSE) between the estimated copula correlations and the true copula correlations. That is, for each n and p, we have 1000 MSE values.

In Supplementary Figs. 3.54 and 3.55, from panel (a), we can see that MSEs decrease as n increases. From panel (b), we can see that MSEs increase as p increases, i.e., more lowly expressed genes are included. To ease the interpretation of the results, we mark three horizontal lines at MSE = 0.09, 0.04, and 0.01 to represent three levels of estimation quality. On the scale of correlation values, these three levels indicate that on average the estimated values are within ± 0.3, ± 0.2, and ± 0.1 of the true values. The results suggest that to reach the ± 0.3 level of estimation quality, a reasonable choice of n is at least 20, and the top 1000 highly expressed genes can be included. To reach the ± 0.2 level, a reasonable choice of n is at least 50, and the top 1500 highly expressed genes can be included. For $n = 100$, the ± 0.1 level can be reached for the top 100-200 highly expressed genes, and even the error level for the top 2000 is close to this level. The results confirm that sample size is not a concern for single-cell data because most cell types contain at least a hundred cells that can be measured by current protocols.

In the implementation of the `scDesign2 R` package, before fitting the above generative model for each cell type, scDesign2 partitions the genes into three groups: the first group containing genes with zero proportions less than a cutoff (default 0.8, but can be changed according to the discussion above), the second group containing genes with zero proportions between the cutoff and $(n-2)/n$, where n is the number of cells, and the last group including the remaining genes, i.e., genes expressed in fewer than three cells. For the first group, scDesign2 fits the above generative model jointly for its genes. For the second group, scDesign2 fits a marginal distribution for each individual gene. For the last group, scDesign2 only generates zero counts for all its genes.

3.4.1.2 Generation of synthetic single-cell gene expression count data

To generate synthetic scRNA-seq data for K cell types, scDesign2 first estimates the proportions of K cell types from the real scRNA-seq count matrix $\mathbf{X}$, for which we denote the number of reads mapped to the $n^{(k)}$ cells of type k as $N^{(k)}$, and the total number of reads mapped to all the n cells as $N = \sum_{k=1}^{K} N^{(k)}$. Denoting the cell type proportions as $\pi = (\pi^{(1)}, \ldots, \pi^{(K)})^\mathsf{T}$ such that $\sum_{k=1}^{K} \pi^{(k)} = 1$, scDesign2 estimates them by $\hat{\pi} = (\hat{\pi}^{(1)}, \ldots, \hat{\pi}^{(K)})^\mathsf{T}$, where

$$\hat{\pi}^{(k)} = \frac{n^{(k)}}{n}, \quad k = 1, \ldots, K.$$

We denote the synthetic scRNA-seq data to be generated as $\mathbf{X}'$, which contains n' cells and N' expected number of reads, with n' and N' as user-specified input parameters of scDesign2. Denoting the number of synthetic cells of type k as $n^{(k)\prime}$, scDesign2 draws the numbers of synthetic cells of all K cell types from a multinomial distribution, i.e., $(n^{(1)\prime}, \ldots, n^{(K)\prime})^\mathsf{T} \sim \text{Multinomial}(n', \hat{\pi})$. Then given $n^{(k)\prime}$, the expected number of reads assigned to cell type k in $\mathbf{X}'$ should be

$$N^{(k)0} = \frac{N^{(k)}}{n^{(k)}} n^{(k)\prime}, \quad k = 1, \ldots, K\,.$$

However, given the constraint that the expected total number of reads in $\mathbf{X}'$ is N', we need to rescale $N^{(k)0}$ to

$$N^{(k)\prime} = \frac{N^{(k)0}}{\sum_{s=1}^{K} N^{(s)0}} N', \quad k = 1, \ldots, K\,.$$

As a result, the scaling factor is

$$r = \frac{N^{(k)\prime}}{N^{(k)0}} = \frac{N'}{\sum_{s=1}^{K} N^{(s)0}}\,,$$

which does not depend on the cell type, and scDesign2 uses this scaling factor to rescale the mean parameter of every gene.

Given the fitted generative model $\widehat{F}^{(k)}$ for cell type k with parameters

$$\widehat{\Theta}^{(k)} = \{\hat{p}_1^{(k)}, \hat{\psi}_1^{(k)}, \hat{\mu}_1^{(k)}, \ldots, \hat{p}_p^{(k)}, \hat{\psi}_p^{(k)}, \hat{\mu}_p^{(k)}, \widehat{\mathbf{R}}^{(k)}\}, \quad k = 1 \ldots, K,$$

and the scaling factor r, scDesign2 generates $n^{(k)\prime}$ synthetic cells from a rescaled model $\widehat{F}^{(k)\prime}$, which is defined by parameters

$$\widehat{\Theta}^{(k)\prime} = \{\hat{p}_1^{(k)}, \hat{\psi}_1^{(k)}, r\hat{\mu}_1^{(k)}, \ldots, \hat{p}_p^{(k)}, \hat{\psi}_p^{(k)}, r\hat{\mu}_p^{(k)}, \widehat{\mathbf{R}}^{(k)}\}, \quad k = 1 \ldots, K,$$

Concretely, how the data generation works is that scDesign2 first draws $n^{(k)\prime}$ vectors, denoted as $z_{\cdot j}^{(k)\prime} \in \mathbb{R}^p$; $j = 1, \ldots, n^{(k)\prime}$, independently from $\Phi_p(\cdot; \widehat{\mathbf{R}}^{(k)})$. Then scDesign2 converts $z_{ij}^{(k)\prime}$ to $x_{ij}^{(k)\prime}$ by setting $x_{ij}^{(k)\prime}$ to be the $\Phi(z_{ij}^{(k)\prime})$-th quantile of $\widehat{F}_i^{(k)\prime}$, i.e., $\text{ZINB}(\hat{p}_i^{(k)}, \hat{\psi}_i^{(k)}, r\hat{\mu}_i^{(k)})$ (including the Poisson, ZIP, and NB distributions as special cases), $i = 1, \ldots, p$. Finally, scDesign2 outputs the synthetic count matrix $\mathbf{X}' = [\mathbf{X}^{(1)\prime} \cdots \mathbf{X}^{(K)\prime}]$, where $\mathbf{X}^{(k)\prime} = (x_{ij}^{(k)\prime})$ is a $p \times n^{(k)\prime}$ matrix for cell type k.

Note that the synthetic count matrix $\mathbf{X}'$ does not contain exactly N' reads; rather, N' is the expected total number of reads. We think this setting mimics a real sequencing experiment, where the total number of sequenced reads would not be exactly the same as the preset sequencing depth N' due to experimental randomness.

3.4.1.3 Recommendation for cell clustering when input data do not have labelled cell types

If users would like to train scDesign2 on a gene-by-cell count matrix without cell type labels, a necessary preceding step is cell clustering. We recommend users to choose a state-of-the-art cell clustering method such as Seurat and SC3. For the resulting clusters, we recommend users to visualize them by t-SNE or UMAP and use a goodness-of-fit measure (e.g., Pearson's chi-square statistic and ROGUE score [124]) to check whether each gene approximately follows a NB or ZINB distribution in a cell cluster. This check will guide users to decide on an appropriate number of cell clusters in a data-driven way.

3.4.2 The scDesign2 variant without copula

The only difference between this variant "w/o copula" and scDesign2 is that this variant assumes the p genes to have independent marginal distributions $F_1, \ldots, F_p$. The fitting of the p marginal distributions and the generation of synthetic data is the same as those in scDesign2.

3.4.3 Existing simulators

- **scDesign**: The R package scDesign version 1.0.0 is used for the analysis.
- **scGAN**: This method is executed from this github repository https://github.com/imsb-uke/scGAN, downloaded around March 29, 2020.
- **splat, splat simple, kersplat**: These methods are executed from the R packge splatter version 1.10.1.
- **SPARSim**: The R package SPARSim version 0.9.5 is used for the analysis.
- **SymSim**: The R package SymSim version 0.0.0.9000 is used for the analysis.
- **ZINB-WaVE**: The ZINB-WaVE method is used from the wrapper functions in the R package splatter version 1.10.1.
- **scDesign**: The R package scDesign version 1.0.0 is used for the analysis.

3.4.4 Dimensionality reduction methods

- **t-SNE**: The R package Rtsne version 0.15 is used for generating t-SNE plots. The function Rtsne is used, with all parameters set to default, except check_duplicate = FALSE and perplexity is changed from the default value of 30 to one third of the sample size when the sample size (total number of cells) is less than 90.
- **PCA**: The R function prcomp() is used for generating PCA plots, with parameters set as default.

3.4.5 Cell clustering methods

- **Seurat**: The Seurat clustering method is executed by the following instruction in this tutorial https://satijalab.org/seurat/v3.2/pbmc3k_tutorial.html. R package Seurat version 3.1.5 is used for the analysis.
- **SC3**: The SC3 clustering method is executed by the following instruction in this tutorial https://www.bioconductor.org/packages/release/bioc/vignettes/SC3/inst/doc/SC3.html. R package SC3 version 1.14.0 is used for the analysis.

3.4.6 Rare cell type detection methods

- **FiRE**: The FiRE method is executed by the following instruction in this tutorial https://github.com/princethewinner/FiRE. R package FiRE version 1.0 is used for the analysis.
- **GiniClust2**: This method is executed from this github repository https://github.com/dtsoucas/GiniClust2 downloaded around March 4, 2020. It is executed based on the reference manual in this repository, except no cells are filtered.

3.4.7 Datasets

- **10x Genomics**: The 10x Genomics dataset measures the mouse intestinal epithelial tissue [119]. The raw count dataset is downloaded from Gene Expression Omnibus (GEO) with accession number GSE92332. Data for cell types Stem, Goblet, Tuft, Transit Amplifying Early (TA Early), Enterocyte Progenitor and Enterocyte Progenitor Early were selected for analysis. Spike-in RNA counts were filtered. The resulting count matrix contains 15962 genes and 3793 cells.
- **CEL-Seq2**: The CEL-Seq2 dataset measures the human pancreas [120]. The raw count dataset is downloaded from GEO with accession number GSE85241. Data for cell types alpha, beta, acinar, delta, duct, endothelial, mesenchymal and pancreatic polypeptide cell (pp) were selected for analysis. Spike-in RNA counts were filtered.

The resulting count matrix contains 19049 genes and 2279 cells.

- **Fluidigm C1**: The Fluidigm C1 dataset measures human brain cells [121]. The raw count dataset is downloaded from GEO with accession number GSE67835. Data for cell types astrocytes, endothelial, fetal quiescent, hybrid neurons, oligodendrocytes and oligodendrocyte precursor cell (OPC) were selected for analysis. The resulting count matrix contains 22088 genes and 317 cells.

- **Smart-Seq2**: The Smart-Seq2 dataset measures human blood dendritic cells [122]. The raw count dataset is downloaded from GEO with accession number GSE94820. Data for dendrocyte subtypes 1–6 and monocyte subtypes 1–4 were selected for analysis. Spike-in RNA counts were filtered. The resulting count matrix contains 26586 genes and 1078 cells.

- **MERFISH**: The MERFISH dataset measures the mouse hypothalamic preoptic region [127]. The raw count dataset is downloaded from Dryad (https://datadryad.org/stash/dataset/doi:10.5061/dryad.8t8s248). It contains 155 genes and 6412 cells. Cell subtypes are combined into cell types, e.g., "Endothelial 1" and "Endothelial 2" are combined as "Endothelial", resulting in nine cell types in total.

- **pciSeq**: The pciSeq dataset measures the mouse hippocampal area CA1 [128]. The raw data "cells_left_CA1_3-1" are downloaded from https://su.figshare.com/articles/pciSeq_files_in_csv_format/10318610/1. Gene expression values are rounded as integers, and cell subtypes are combined into cell types, e.g., "Astro.1" to "Astro.5" are combined as "Astro". The cell type "Zero" is removed as it contains cells with almost no genes expressed, so seven cell types are retained. The processed data contain 84 genes and 2253 cells.

3.5 Software and code

The `scDesign2` `R` package is available at https://github.com/JSB-UCLA/scDesign2. The source code and data for reproducing the results are available at https://doi.org/10.

5281/zenodo.4011311.

3.6 Acknowledgements

This chapter is based on my joint work with Dongyuan Song, Dr. Wei Vivian Li, and my advisor Dr. Jingyi Jessica Li [41]. We would also like to thank Dr. Roy Wollman and his Ph.D. student Zach Hemminger for bringing our attention to MERFISH and pciSeq data. We also appreciate the comments and feedback from the members of the Junction of Statistics and Biology at UCLA (http://jsb.ucla.edu).

3.7 Tables

Simulator \ **Property**	1 protocol adaptive	2 gene preserved	3 gene cor. captured	4 cell num. seq. dep. flexible	5 trans-parent	6 comp. & sample efficient
dyngen [137]	✓	✗	✗	✓	✓	✓
Lun2 [156]	✓	✓	✗	✓	✓	✓
powsimR [114]	✓	✓	✗	✓	✓	✓
PROSST [107]	✓	✓	✗	✓	✓	✓
scDD [113]	✓	✗	✗	✓	✓	✓
scDesign [87]	✓	✓	✗	✓	✓	✓
scGAN [111]	✓	✓	✓	✓	✗	✗
splat simple [108]	✓	✗	✗	✗	✓	✓
splat [108]	✓	✗	✗	✗	✓	✓
kersplat [108]	✓	✗	✓	✗	✓	✓
SPARSim [110]	✓	✓	✓	✗	✓	✓
SymSim [109]	✓	✗	✗	✗	✓	✓
ZINB-WaVE [24]	✓	✓	✓	✗	✓	✓
SERGIO [115]	✓	✓	✗*	✓	✓	✓
scDesign2	✓	✓	✓	✓	✓	✓

Property 1: protocol adaptiveness;
Property 2: gene preservation;
Property 3: gene correlation capture;
Property 4: flexible cell number and sequencing depth choices;
Property 5: transparency;
Property 6: computational and sample efficiency.
For each simulator and each property, a checkmark, checkcross, or cross means that the simulator satisfies, partially satisfies, or does not satisfy the property.
*: SERGIO requires a user-specified gene regulatory network, and it does not capture/estimate gene correlations from a real dataset.

Table 3.1: Summary of 14 simulators (including our proposed scDesign2) in six properties.

3.8 Figures

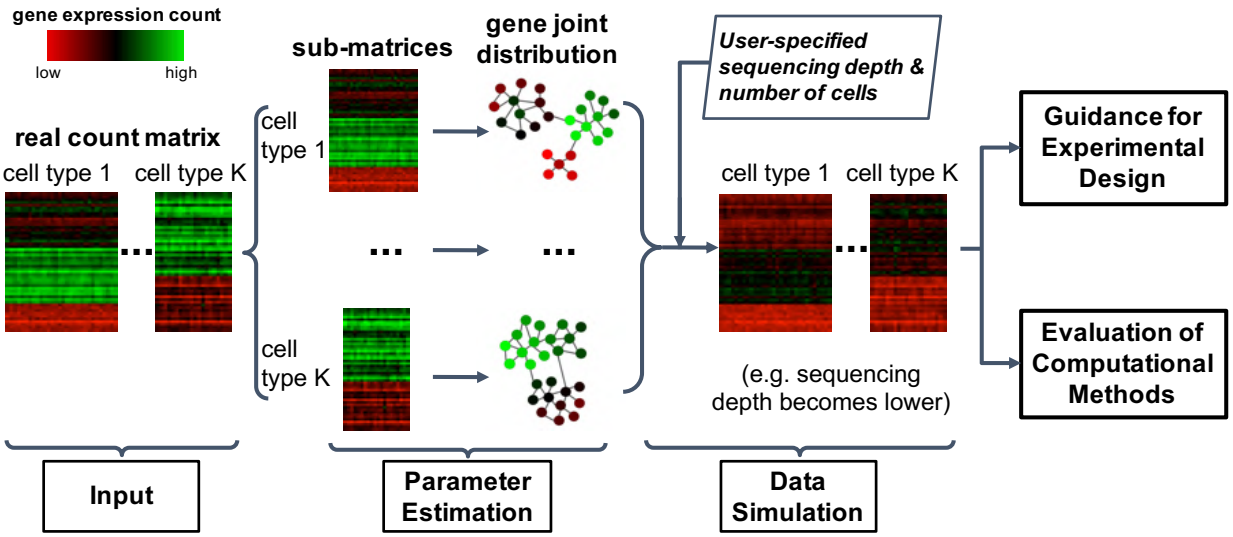

Figure 3.1: An overview of scDesign2.

The input for scDesign2 is a gene-by-cell count matrix with cells labelled as cell types or clusters. For cells in each type or cluster, scDesign2 uses the copula framework to fit a joint distribution of gene expression counts. Then given user-specified sequencing depth and number of cells, scDesign2 generates synthetic data for each cell type or cluster. The synthetic data can be used to guide experimental design and evaluate computational methods.

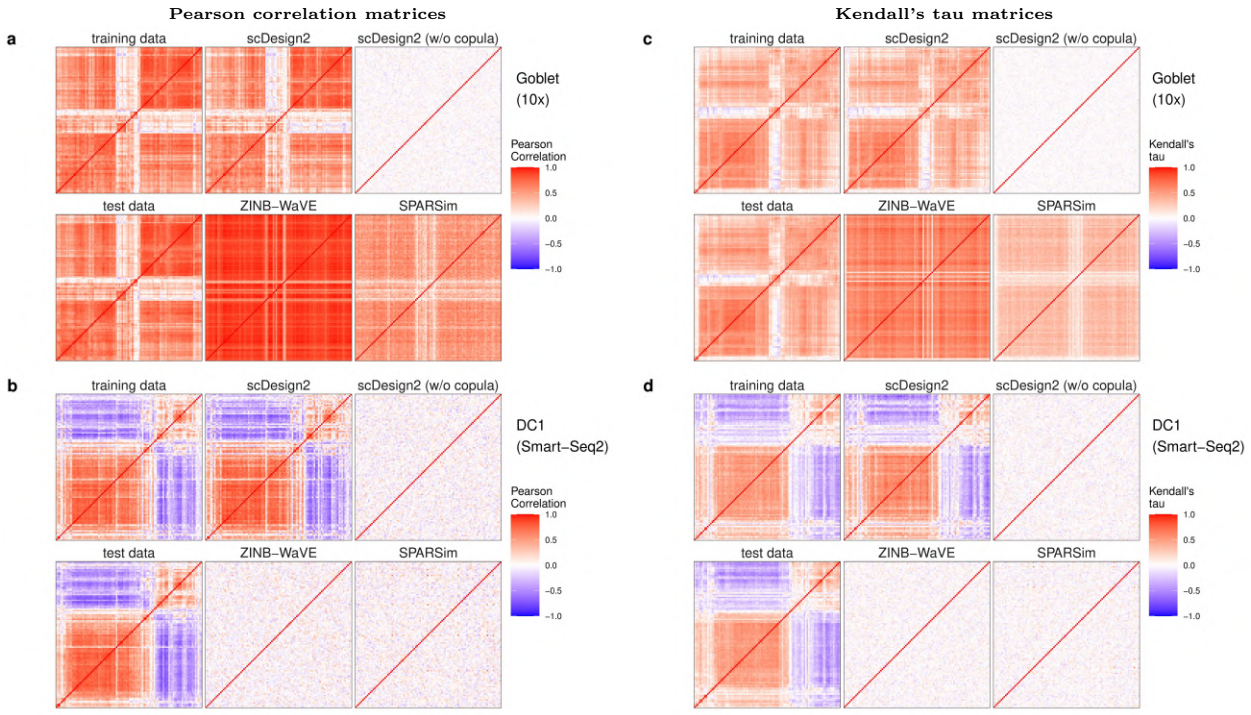

Figure 3.2: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain goblet cells measured by 10x Genomics [119]; In (b) and (d), training and test data contain cells of dendrocytes subtype 1 (DC1) measured by Smart-Seq2 [122]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

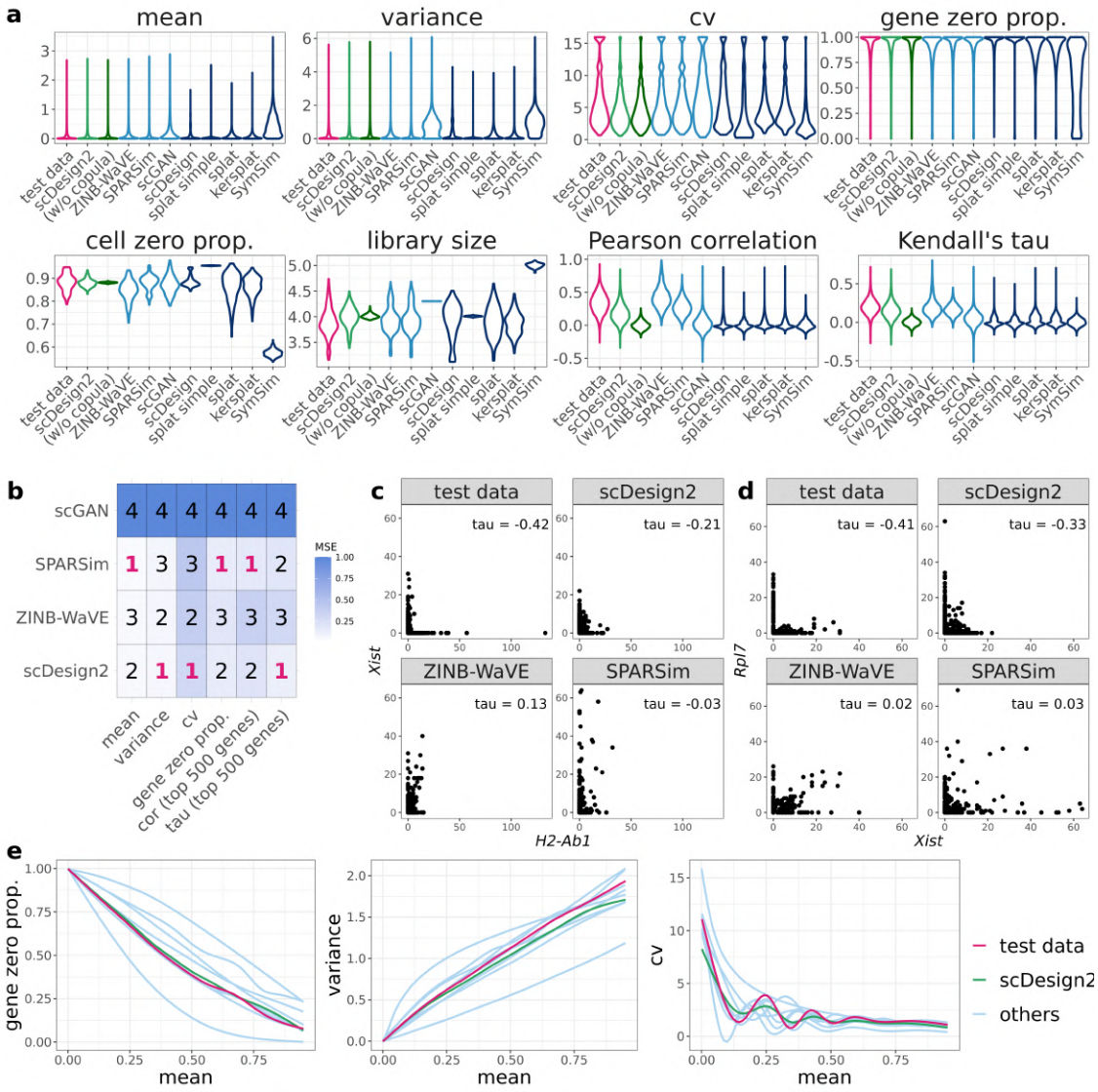

Figure 3.3: Benchmarking scDesign2 against its variant without copula and eight existing scRNA-seq simulators for generating goblet cells measured by 10x Genomics.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scGAN, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, SPARSim, and scGAN, the only four methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for three out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*Xist* vs. *H2-Ab1* and *Rpl7* vs. *Xist*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. The Kendall's tau values in the synthetic data generated by scDesign2 resemble most the values in the test data. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the eight existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

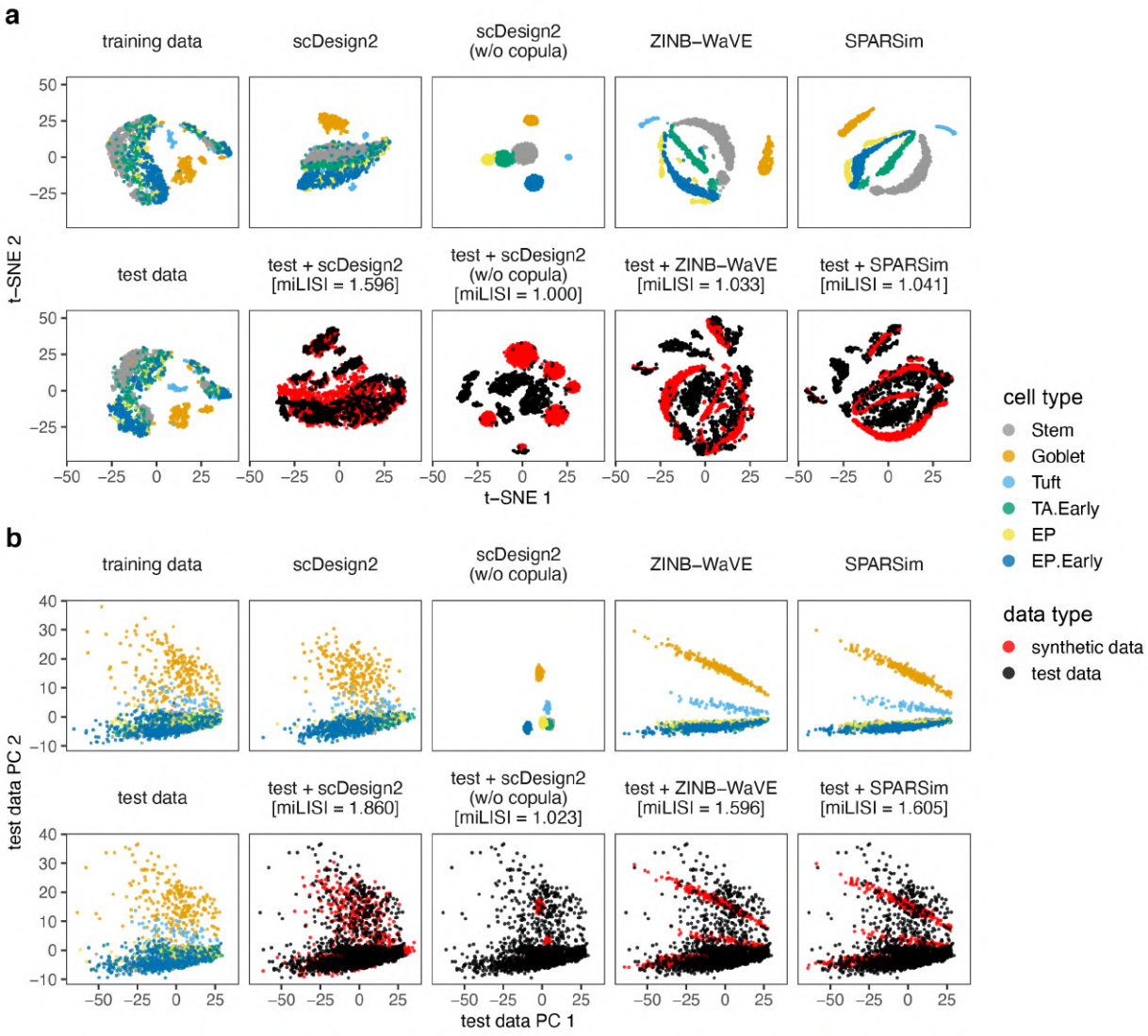

Figure 3.4: Comparison of 10x Genomics data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction. miLISI is short for median integration local inverse Simpson's Index, a higher value of which indicates that the simulated data mix better with the test data in the 2D visualization plot. By visually inspecting the patterns in these plots as well as comparing the miLISI values, we find that the synthetic data generated by scDesign2 most resemble the test data.

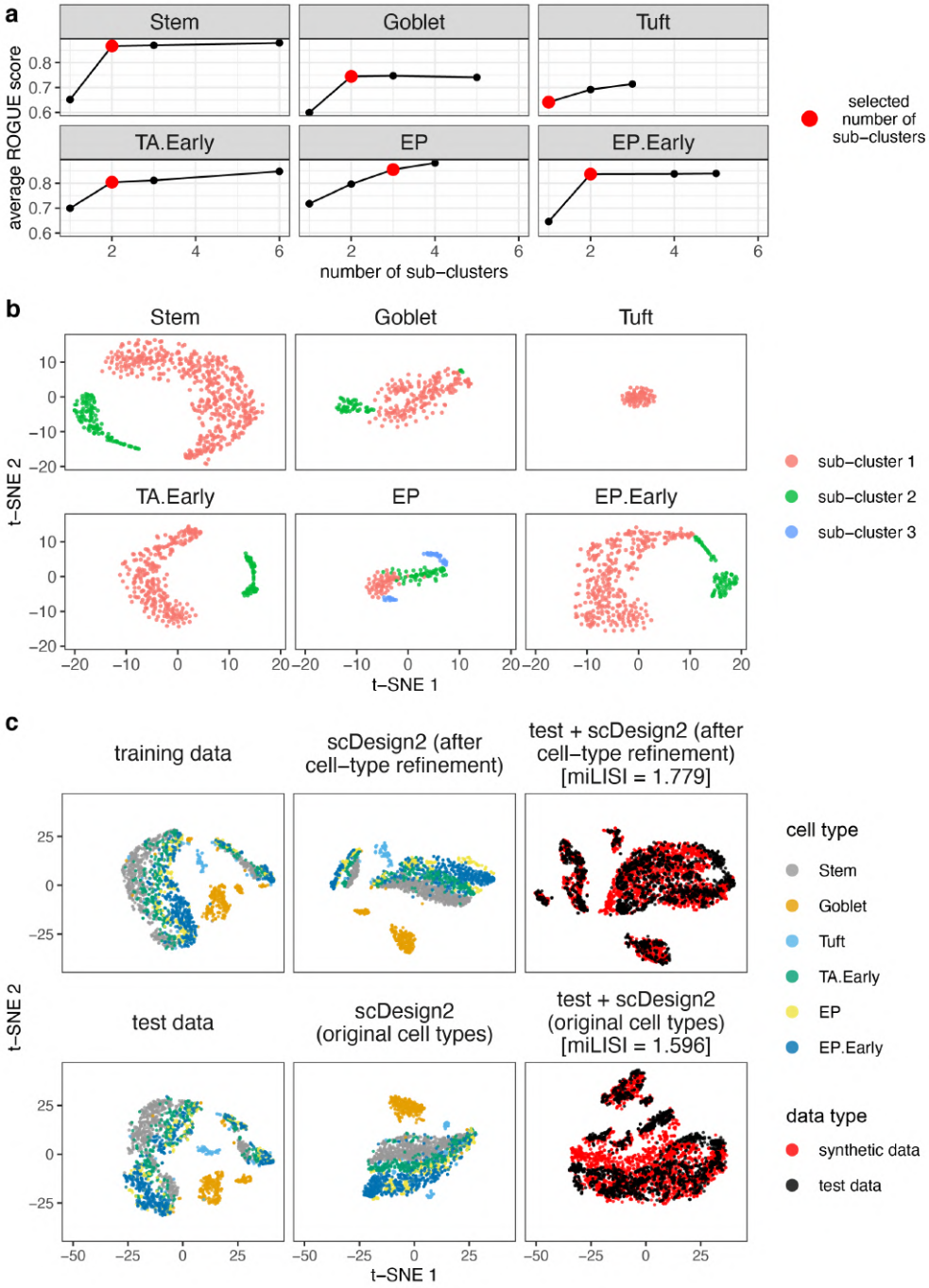

Figure 3.5: Application of ROGUE scores combined with dimensionality reduction plots to refine cell types before training scDesign2.

This refinement approach is demonstrated on the 10x Genomics dataset. (a) In each cell type, the relationship between the average ROGUE score across sub-clusters and the number of sub-clusters. Before a ROGUE score is calculated for each sub-cluster, the Louvain clustering algorithm is applied to each cell type with a varying resolution parameter so that a varying number of sub-clusters is obtained. Based on how the average ROGUE score saturates, a number of sub-clusters is selected and marked in red for each cell type. (b) The t-SNE plots of each cell type with the sub-clusters, whose number is marked in (a), labelled with distinct colors. (c) The t-SNE plots of training data (top left), test data (bottom left), synthetic data of scDesign2 trained with the refined sub-clusters (middle left) or the original cell types (middle bottom), and combination of test data with each set of synthetic data. Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction. We find that, after the cell-type refinement, the simulated data of scDesign2 resemble the real data better, as indicated by the higher miLISI value.

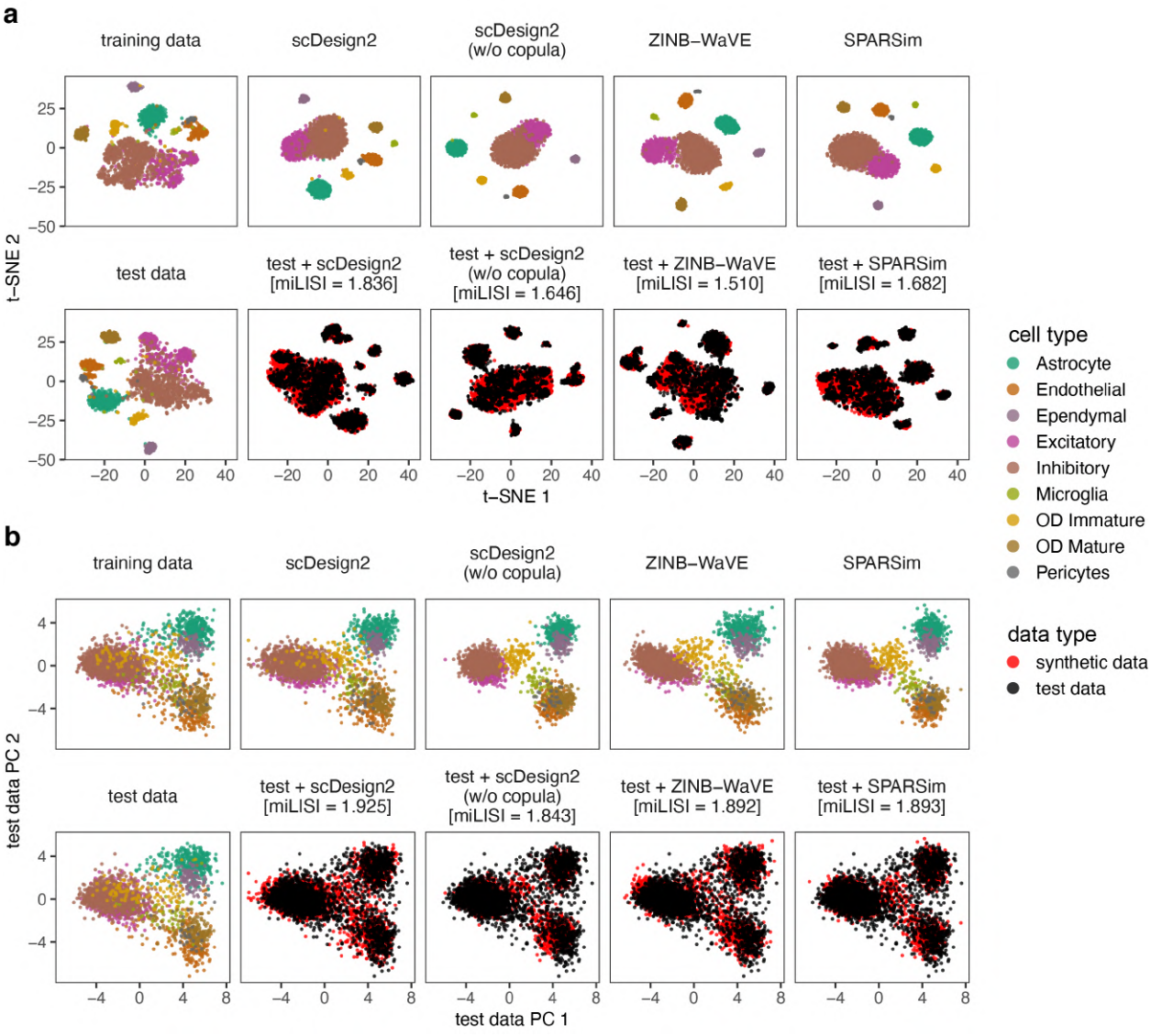

Figure 3.6: Comparison of MERFISH data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as log(1 + count) before dimensionality reduction. miLISI is short for median integration local inverse Simpson's Index, a higher value of which indicates that the simulated data mix better with the test data in the 2D visualization plot. By visually inspecting the patterns in these plots as well as comparing the miLISI values, we find that the synthetic data generated by scDesign2 most resemble the test data.

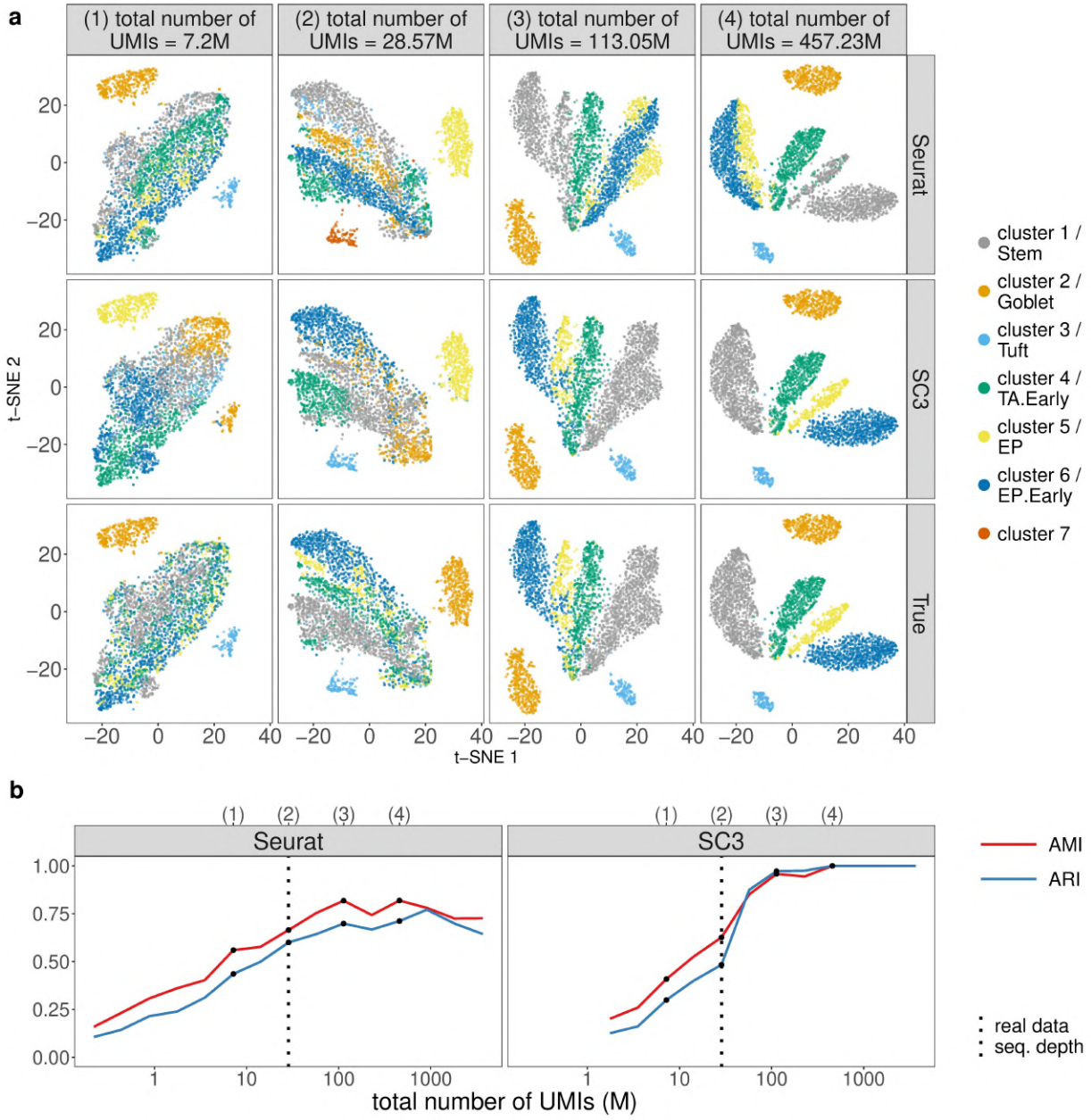

Figure 3.7: scDesign2 guides the choice of sequencing depth in cell clustering.

scDesign2 generates synthetic 10x Genomics data with fifteen sequencing depths. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four sequencing depths in (a) are marked as dots and in the top, and the sequencing depth of the real dataset [119] is marked as vertical dashed lines.

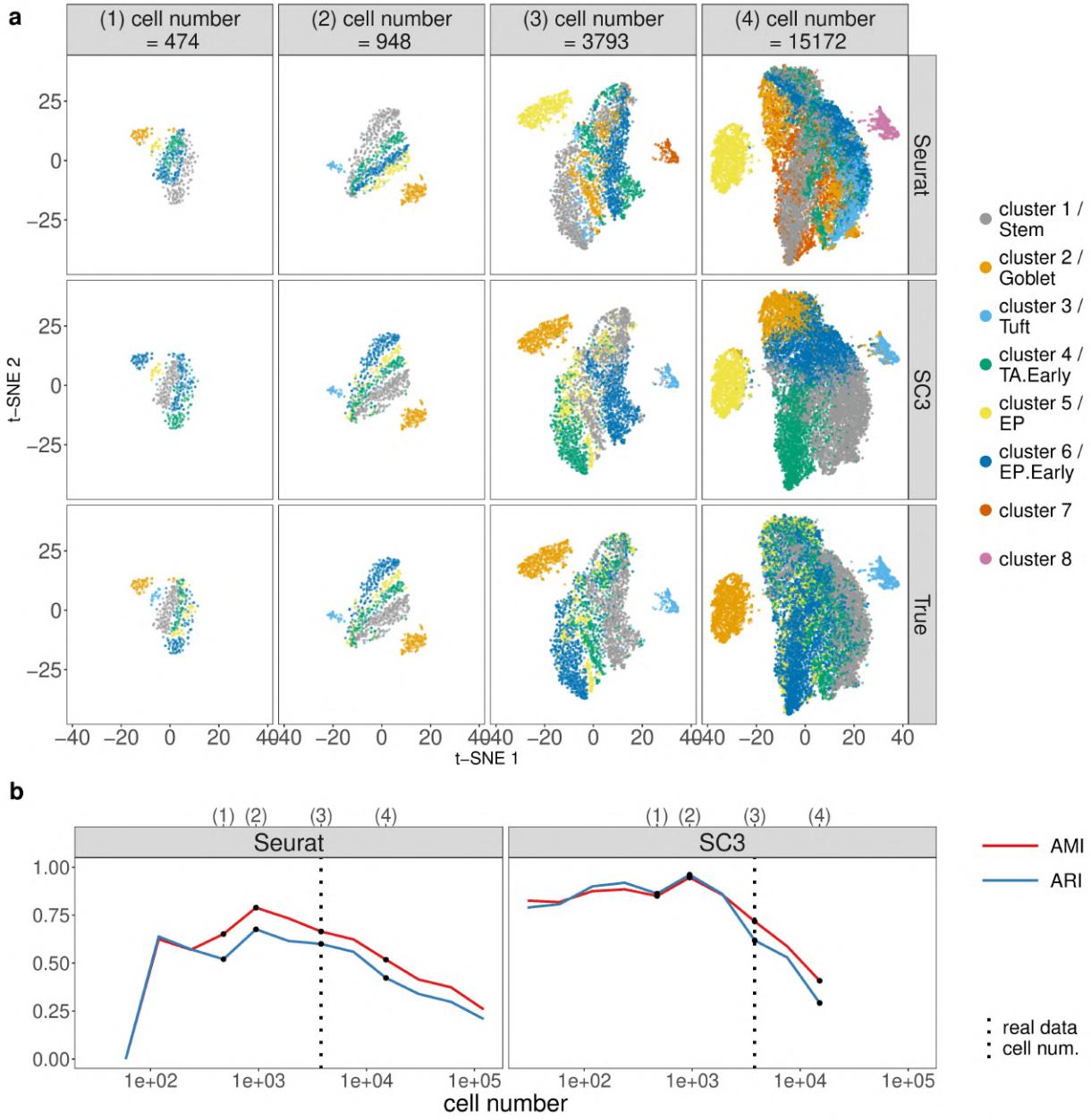

Figure 3.8: scDesign2 guides the choice of cell number in cell clustering, in the case where the total sequencing depth is kept as fixed.

scDesign2 generates synthetic 10x Genomics data with twelve cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [119] is marked as vertical dashed lines.

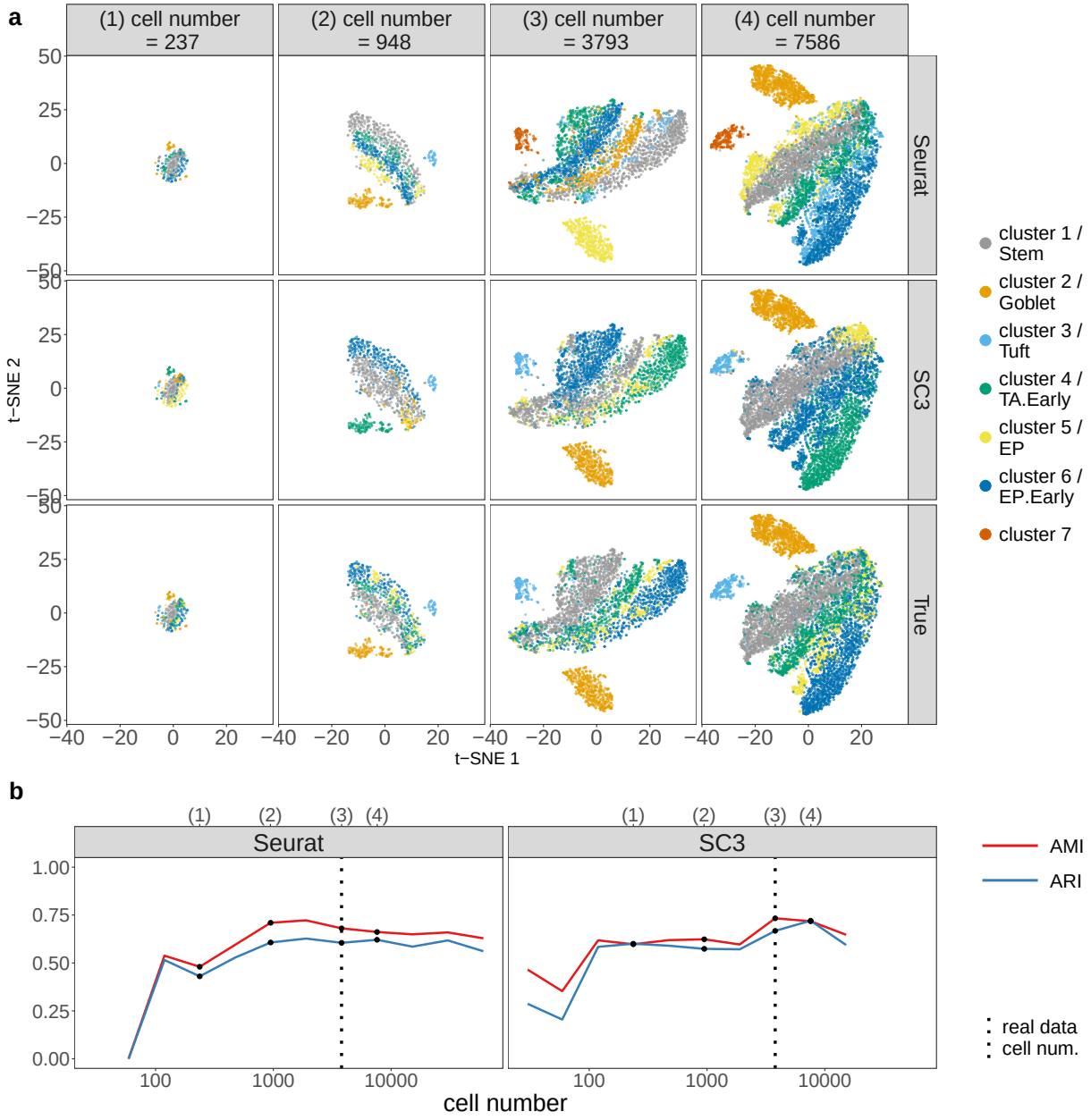

Figure 3.9: scDesign2 guides the choice of cell number in cell clustering, in the case where the average sequencing depth is kept as fixed.

scDesign2 generates synthetic 10x Genomics data with eleven cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [119] is marked as vertical dashed lines.

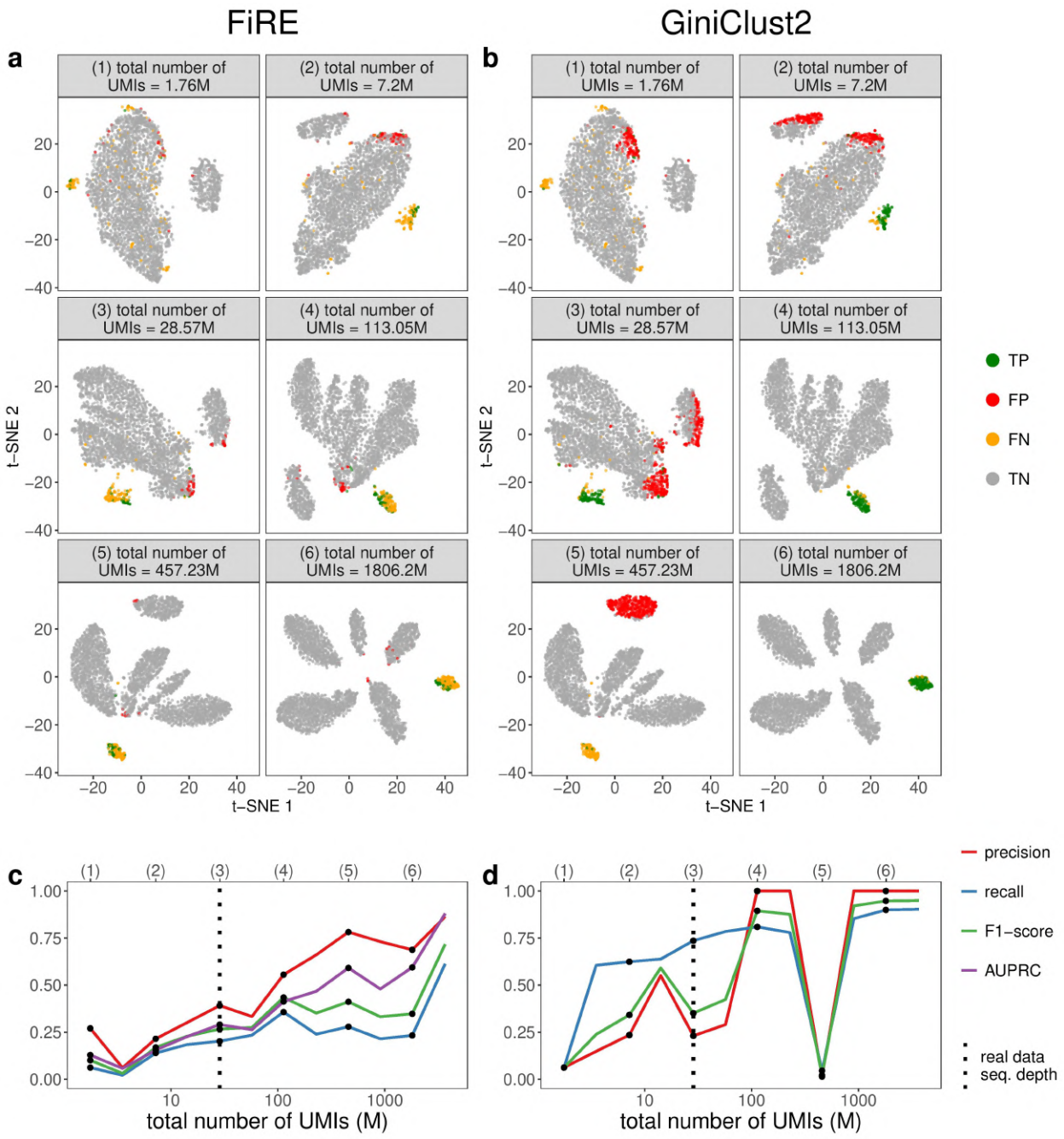

Figure 3.10: scDesign2 guides the choice of sequencing depth in rare cell type detection.

scDesign2 generates synthetic 10x Genomics data with twelve sequencing depths. Two rare-cell-type detection methods—FiRE and GiniClust2—are applied to each synthetic dataset to detect rare cell types. (a) t-SNE visualization of six synthetic datasets and identification results—true positive (TP), false positive (FP), false negative (FN), and true negative (TN) cells—of FiRE in each dataset. (b) t-SNE visualization of the same six synthetic datasets and identification results of GiniClust2 in each dataset. (c) Four identification accuracy measures by FiRE (precision, recall, F1-score, and AUPRC) vs. sequencing depth. (d) Three identification accuracy measures by GiniClust2 (precision, recall, and F1-score) vs. sequencing depth. In (c) and (d), the results of the six sequencing depths in (a) and (b) are marked as dots and in the top, and the sequencing depth of the real dataset [119] is marked as vertical dashed lines.

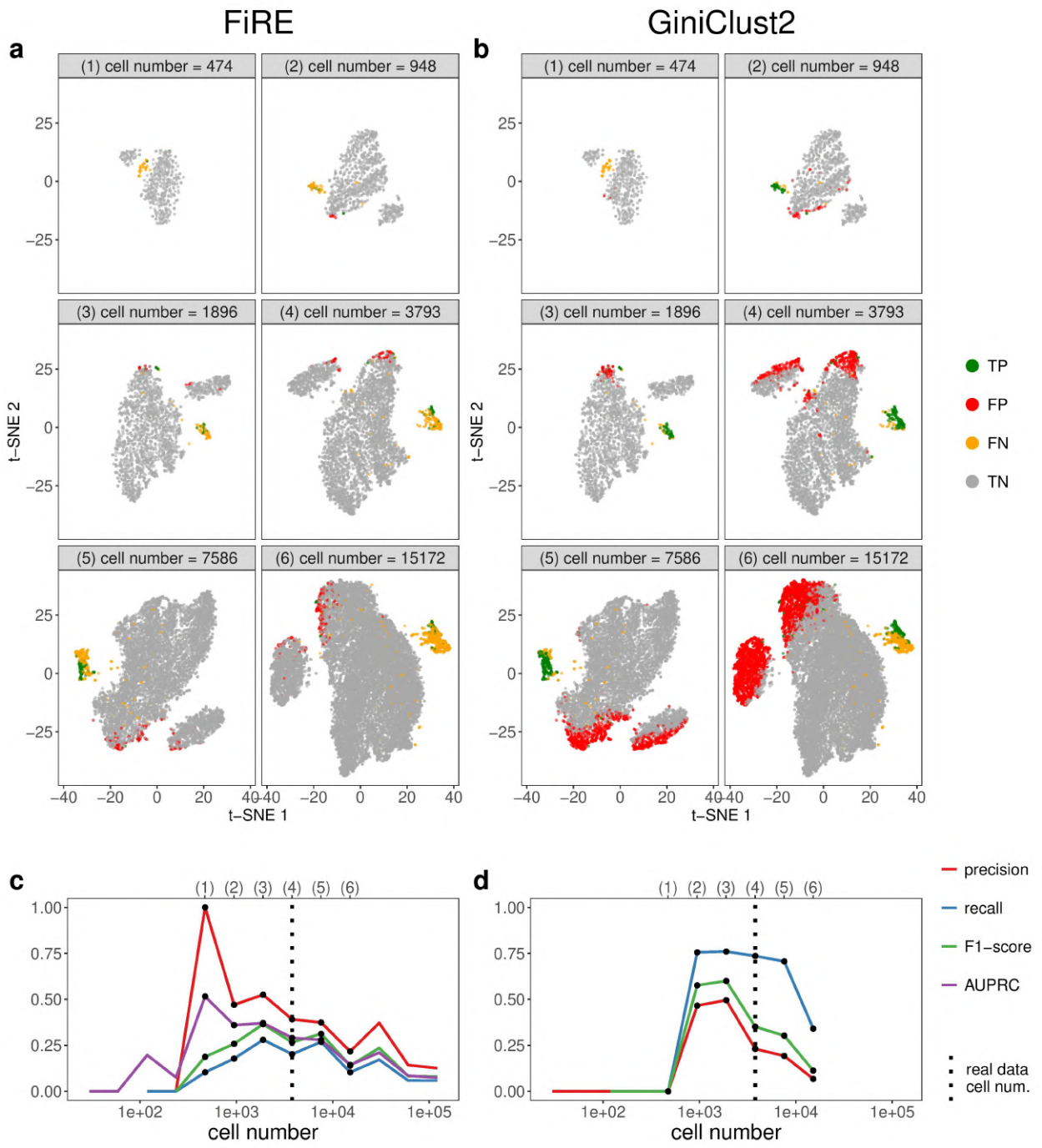

Figure 3.11: scDesign2 guides the choice of cell number in rare cell type detection, in the case where the total sequencing depth is kept as fixed.

scDesign2 generates synthetic 10x Genomics data with thirteen cell numbers. Two rare-cell-type detection methods—FiRE and GiniClust2—are applied to each synthetic dataset to detect rare cell types. (a) t-SNE visualization of six synthetic datasets and identification results—true positive (TP), false positive (FP), false negative (FN), and true negative (TN) cells—of FiRE in each dataset. (b) t-SNE visualization of the same six synthetic datasets and identification results of GiniClust2 in each dataset. (c) Four identification accuracy measures by FiRE (precision, recall, F1-score, and AUPRC) vs. cell number. (d) Three identification accuracy measures by GiniClust2 (precision, recall, and F1-score) vs. cell number. In (c) and (d), the results of the six cell numbers in (a) and (b) are marked as dots and in the top, and the cell number of the real dataset [119] is marked as vertical dashed lines. Whenever there is no line for a cell number, FiRE or GiniClust2 does not detect any rare cells or fails.

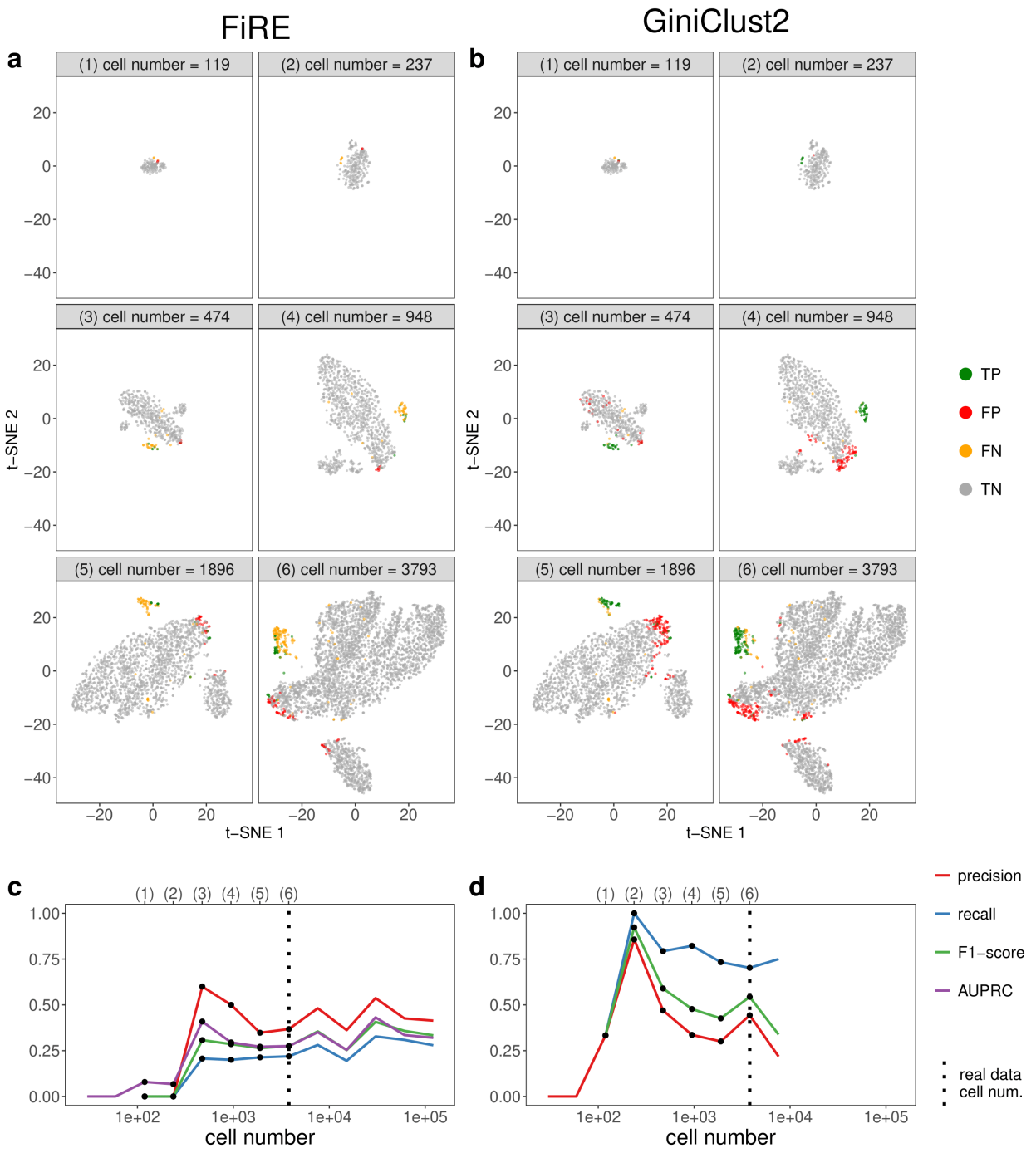

Figure 3.12: scDesign2 guides the choice of cell number in rare cell type detection, in the case where the average sequencing depth is kept as fixed.

scDesign2 generates synthetic 10x Genomics data with thirteen cell numbers. Two rare-cell-type detection methods—FiRE and GiniClust2—are applied to each synthetic dataset to detect rare cell types. (a) t-SNE visualization of six synthetic datasets and identification results—true positive (TP), false positive (FP), false negative (FN), and true negative (TN) cells—of FiRE in each dataset. (b) t-SNE visualization of the same six synthetic datasets and identification results of GiniClust2 in each dataset. (c) Four identification accuracy measures by FiRE (precision, recall, F1-score, and AUPRC) vs. cell number. (d) Three identification accuracy measures by GiniClust2 (precision, recall, and F1-score) vs. cell number. In (c) and (d), the results of the six cell numbers in (a) and (b) are marked as dots and in the top, and the cell number of the real dataset [119] is marked as vertical dashed lines. Whenever there is no line for a cell number, FiRE or GiniClust2 does not detect any rare cells or fails.

S3.9 Supplementary Tables

protocol	cell type	number of cells used for model fitting (n)	number of genes included for copula correlation estimation (p)
10x Genomics	goblet	255	3022
10x Genomics	stem	634	2465
10x Genomics	tuft	83	2981
CEL-Seq2	alpha	426	8285
CEL-Seq2	beta	233	8498
CEL-Seq2	acinar	134	9612
Fluidigm C1	astrocytes	25	7574
Fluidigm C1	neurons	61	9440
Fluidigm C1	oligodendrodytes	19	7232
Smart-Seq2	dendrocyte 1	83	8880
Smart-Seq2	dendrocyte 2	47	8483
Smart-Seq2	monocyte 2	61	8459

Table S3.2: Summary of the sample size (n) and the number of genes included for copula correlation estimation (p), for each of the 12 datasets used for the benchmarking of simulators.

S3.10 Supplementary Figures

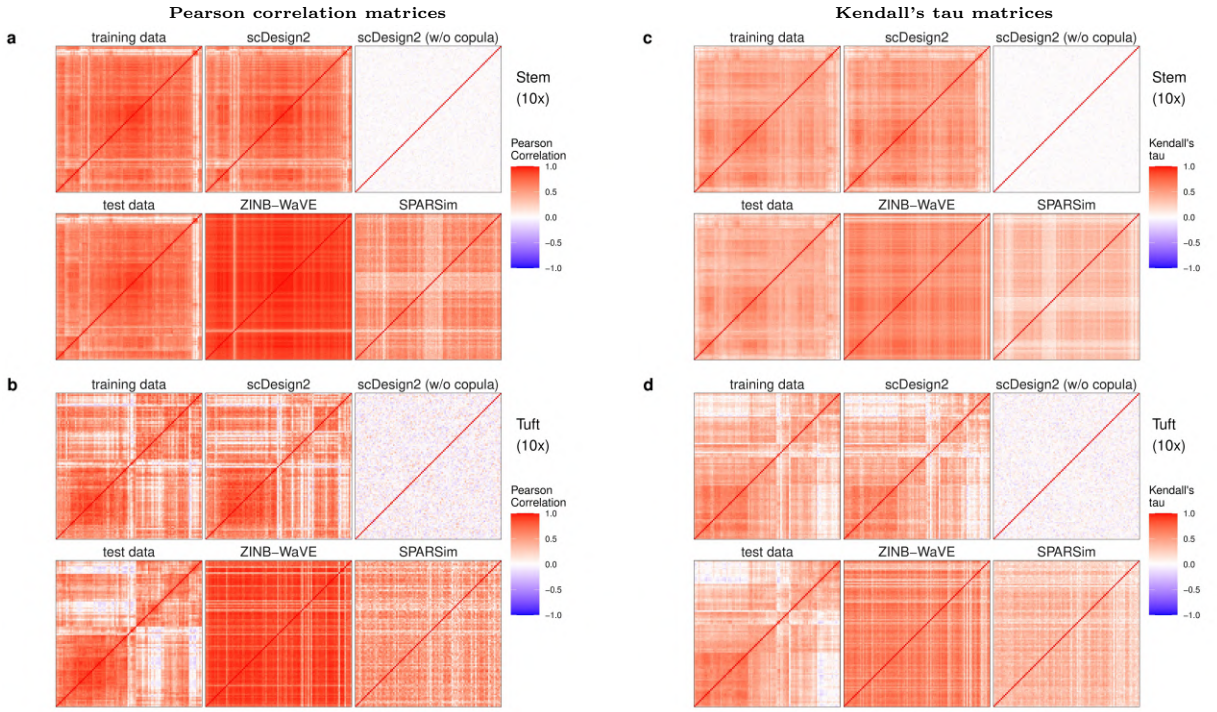

Figure 3.13: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain stem cells measured by 10x Genomics [119]; In (b) and (d), training and test data contain tuft cells measured by 10x Genomics [119]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

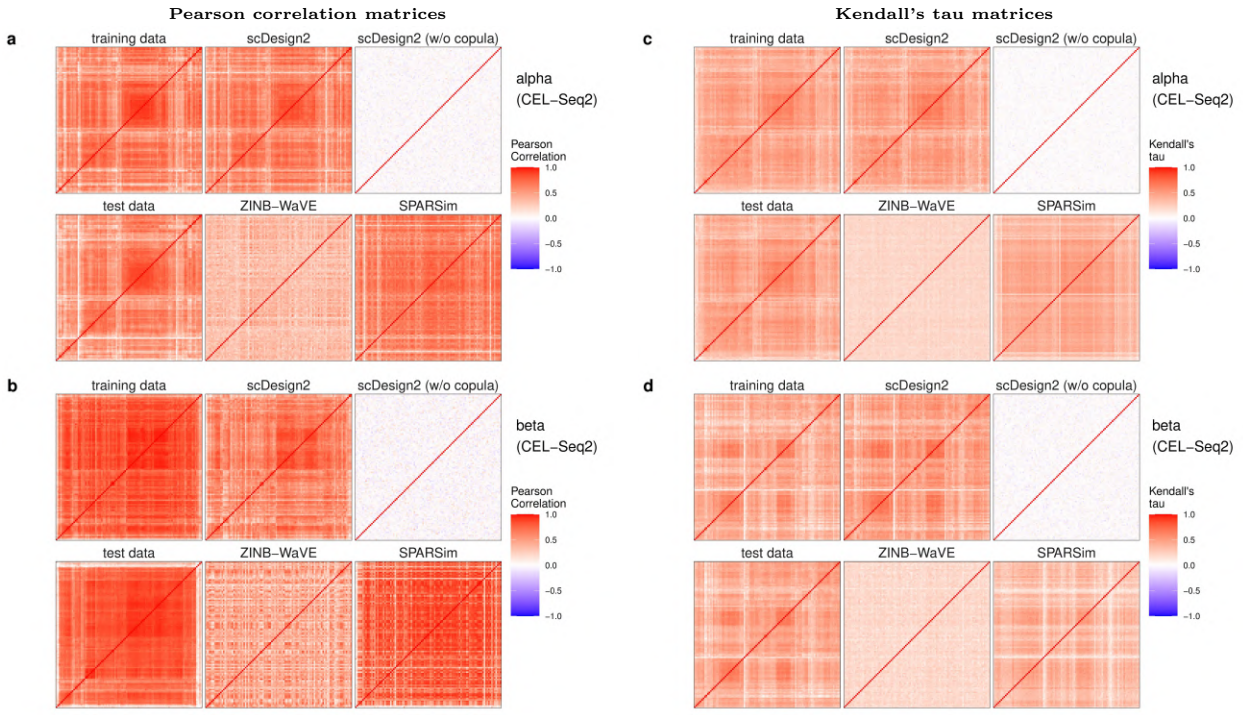

Figure 3.14: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain alpha cells measured by CEL-Seq2 [120]; In (b) and (d), training and test data contain beta cells measured by CEL-Seq2 [120]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

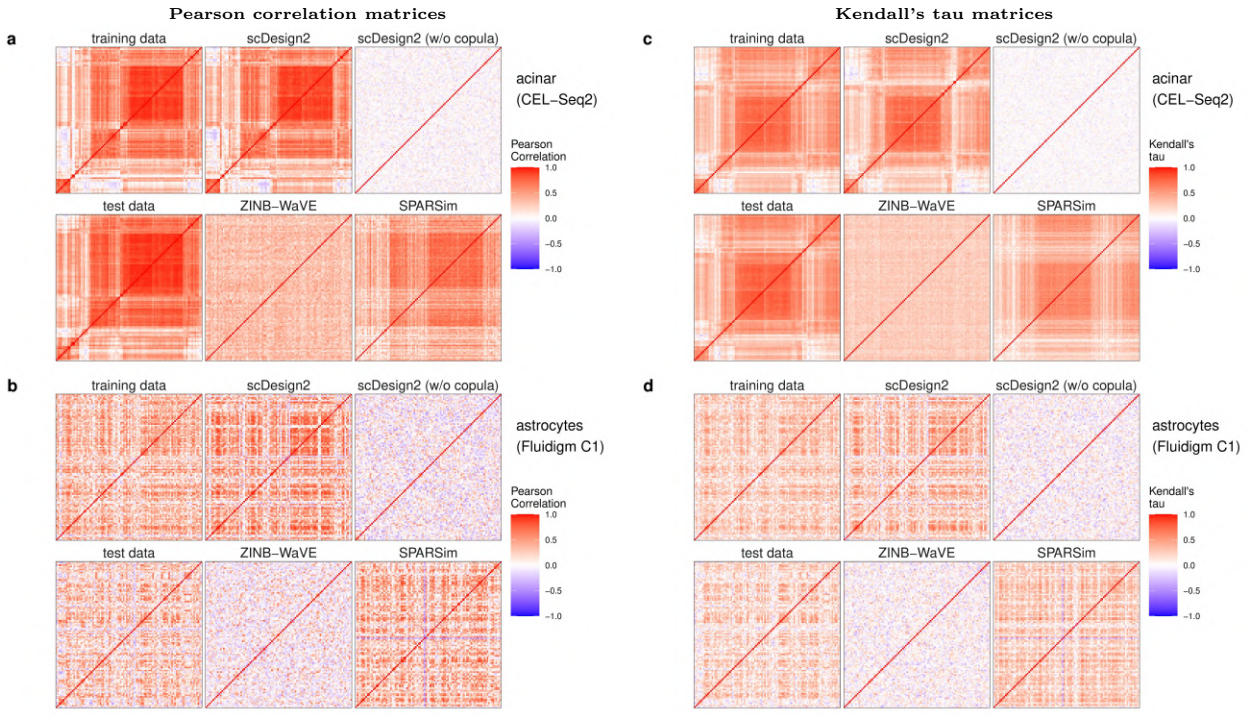

Figure 3.15: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain acinar cells measured by CEL-Seq2 [120]; In (b) and (d), training and test data contain astrocytes measured by Fluidigm C1 [121]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

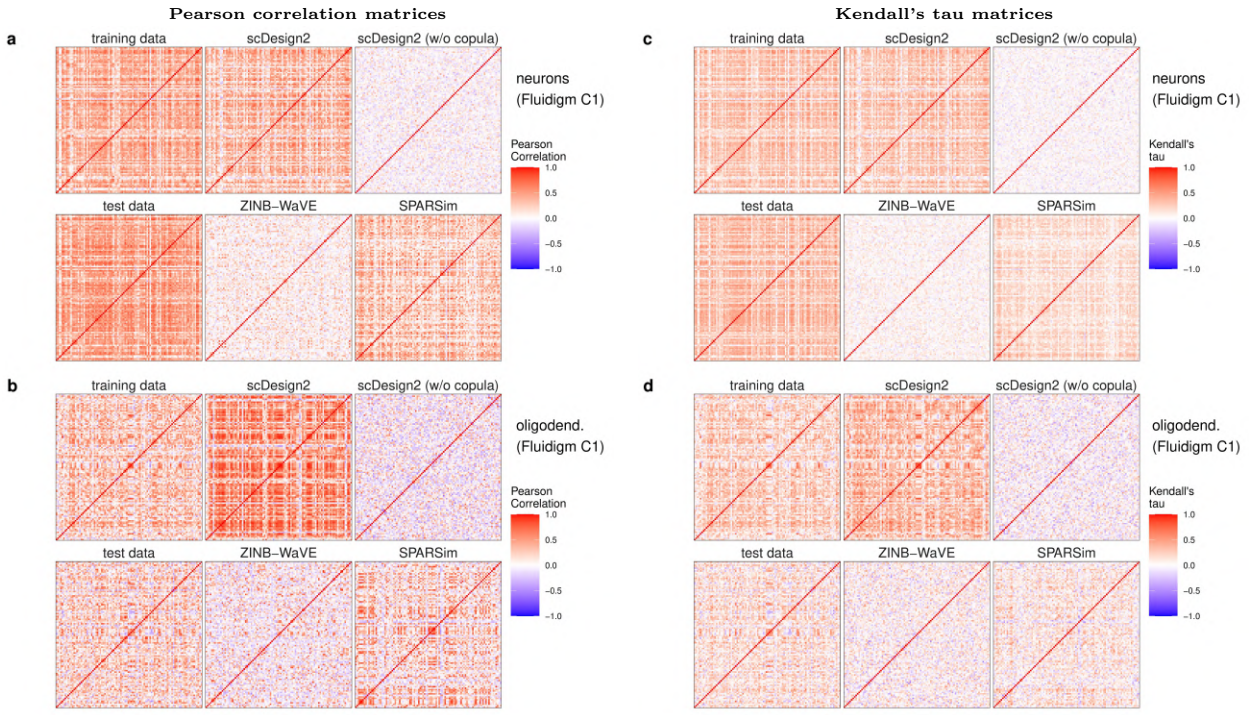

Figure 3.16: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain neurons measured by Fluidigm C1 [121]; In (b) and (d), training and test data contain oligodendrocytes measured by Fluidigm C1 [121]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

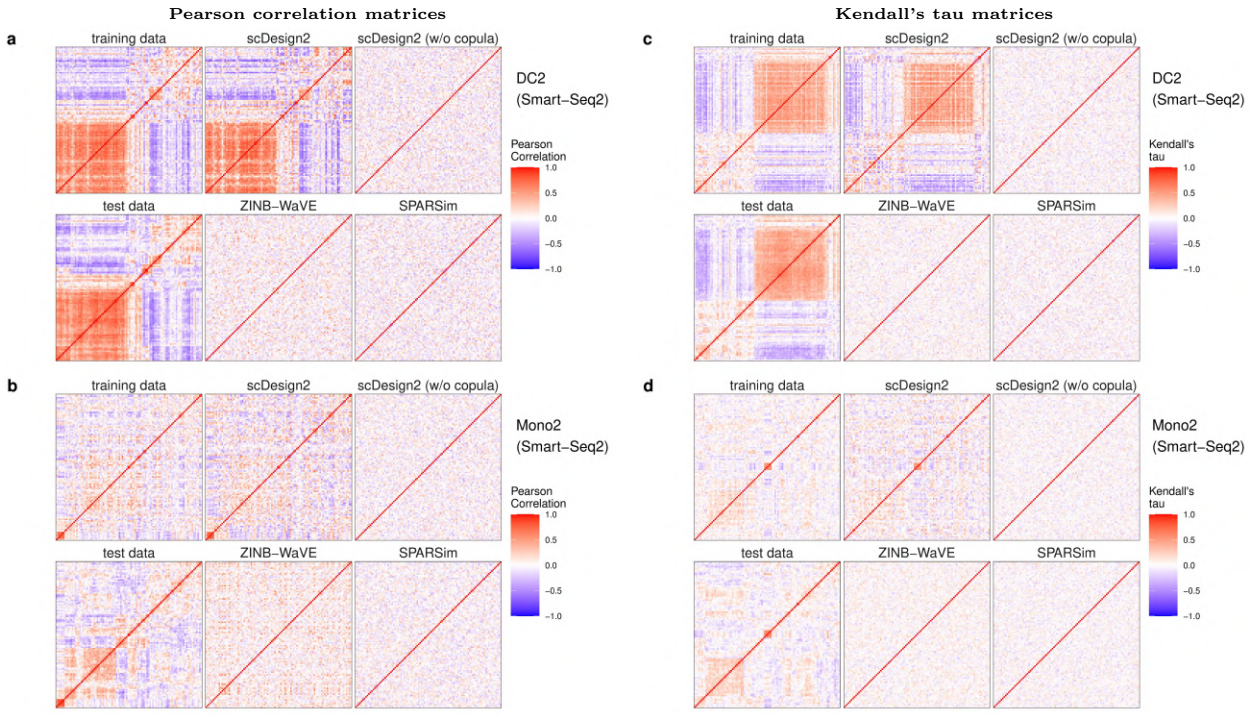

Figure 3.17: Heatmaps of gene correlation matrices estimated from real data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), training and test data contain cells of dendrocytes subtype 2 (DC2) measured by Smart-Seq2 [122]; In (b) and (d), training and test data contain cells of monocytes subtype 2 (Mono2) measured by Smart-Seq2 [122]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the correlation matrices estimated from the synthetic data generated by scDesign2 most resemble those of training and test data.

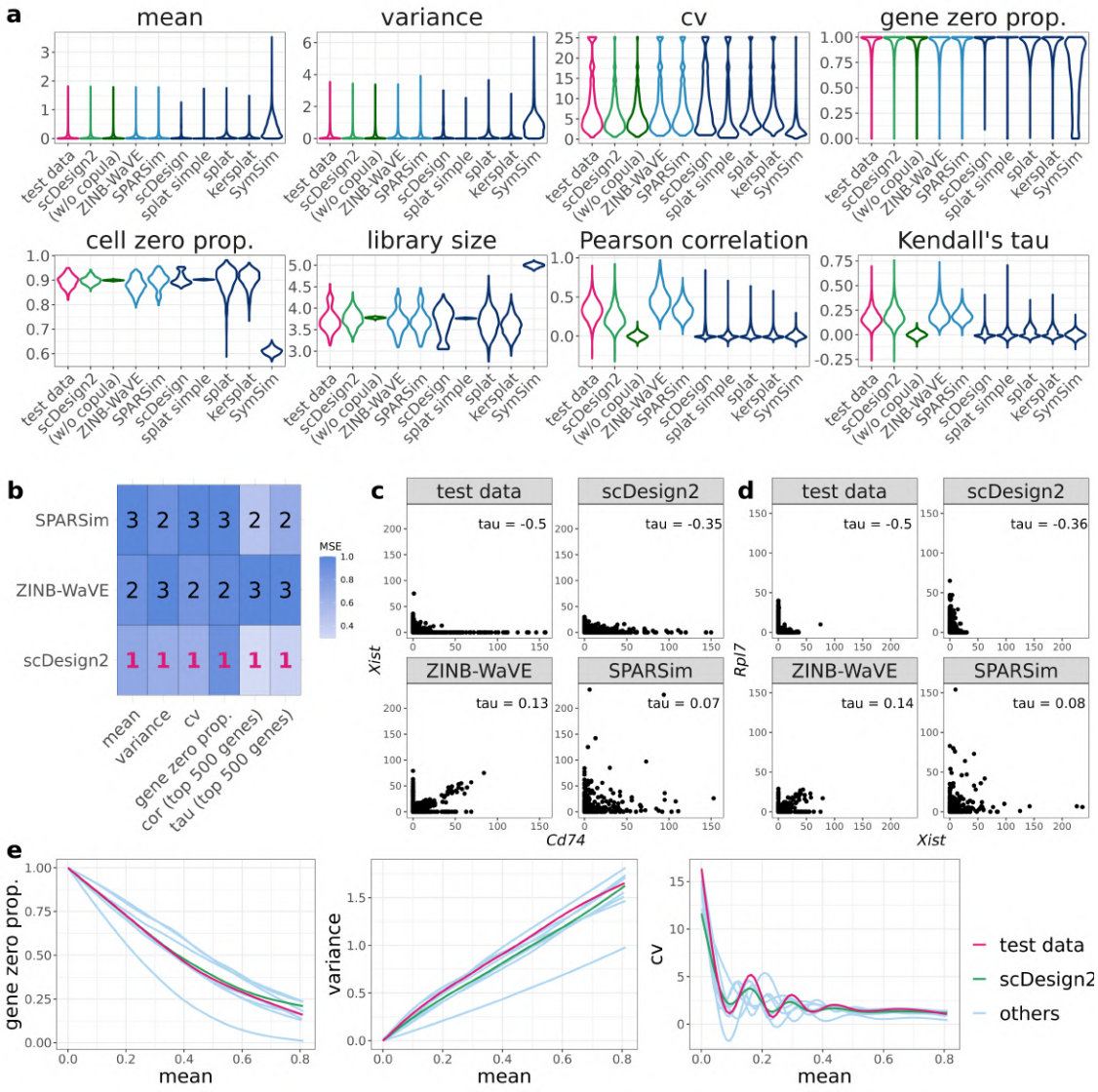

Figure 3.18: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating stem cells measured by 10x Genomics.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign2 is ranked the top for six out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*Xist* vs. *CD74* and *Rpl7* vs. *Xist*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Only scDesign2 captures the negative gene correlations in the real data. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

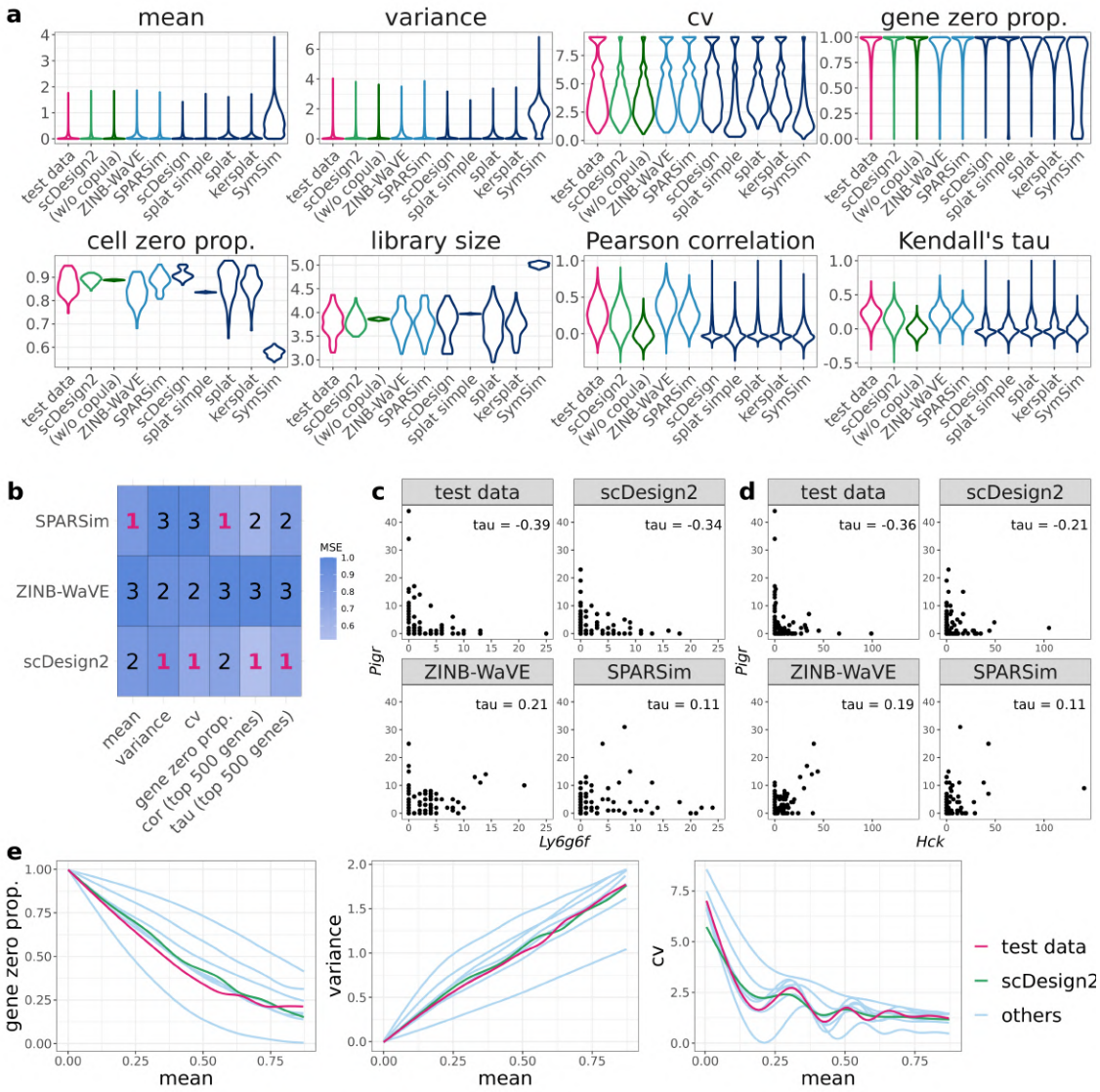

Figure 3.19: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating tuft cells measured by 10x Genomics.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for four out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*Ly6g6f* vs. *Pigr* and *Hck* vs. *Pigr*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Only scDesign2 captures the negative gene correlations in the real data. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

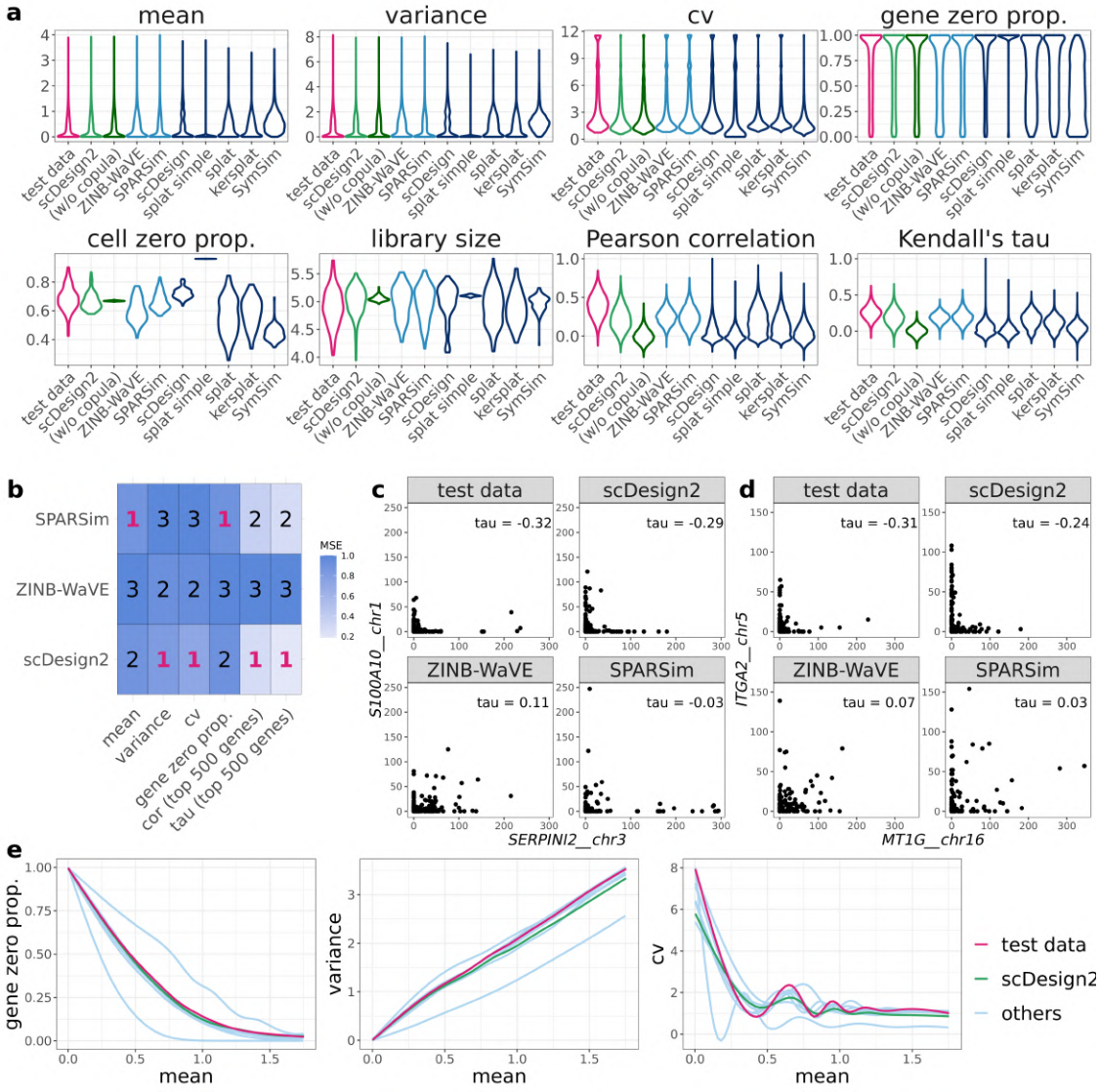

Figure 3.20: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating acinar cells measured by CEL-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for four out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*SERPINI2* vs. *S100A10* and *MT1G* vs. *ITGA2*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

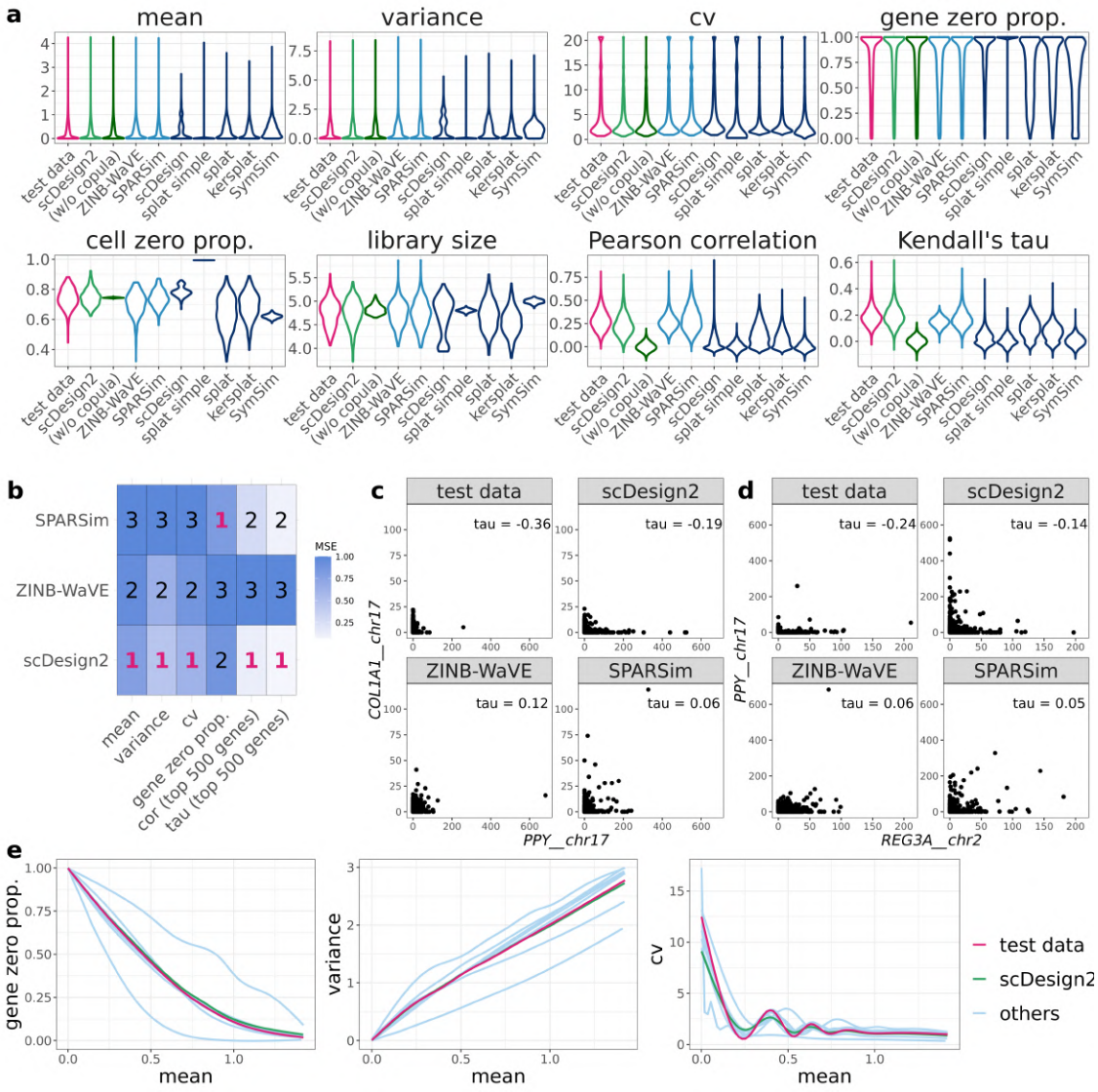

Figure 3.21: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating alpha cells measured by CEL-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for five out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*PPY* vs. *COL1A1* and *REG3A* vs. *PPY*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Only scDesign2 captures the negative gene correlations in the real data. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

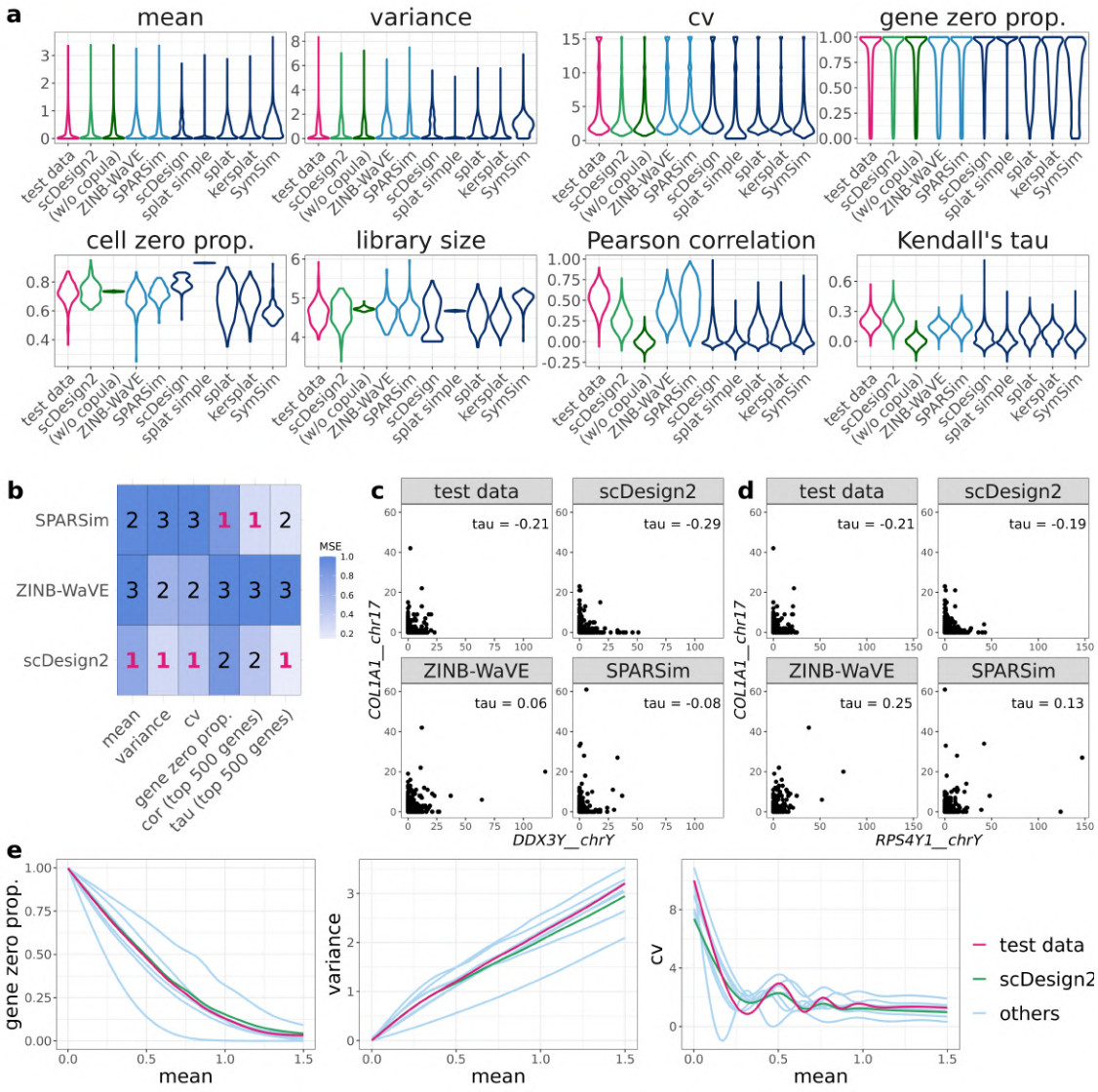

Figure 3.22: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating beta cells measured by CEL-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for four out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*DDX3Y* vs. *COL1A1* and *RPS4Y1* vs. *COL1A1*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

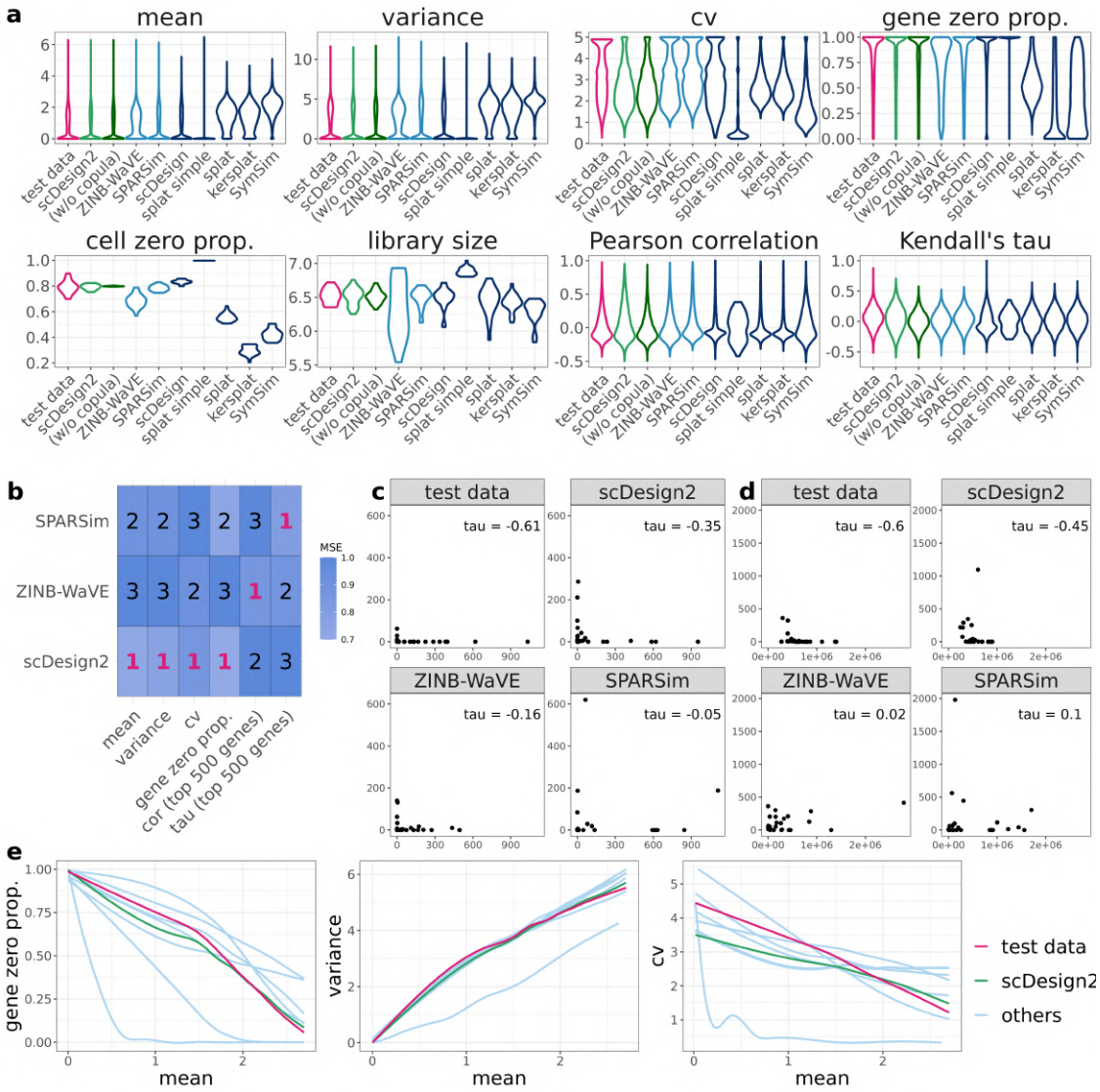

Figure 3.23: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating astrocytes measured by Fluidigm C1 (SMARTer).

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for four out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

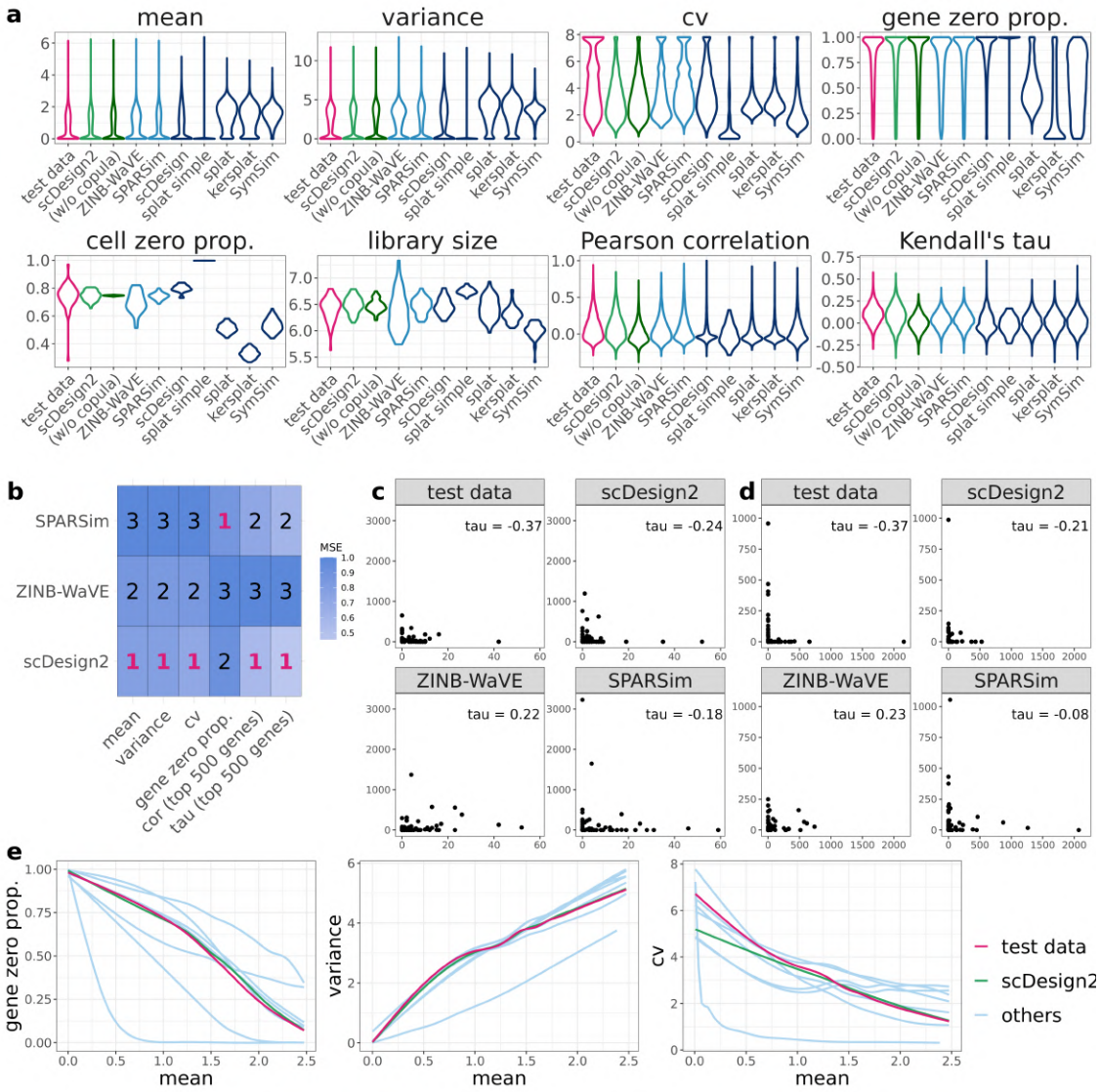

Figure 3.24: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating neurons measured by Fluidigm C1 (SMARTer).

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for five out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

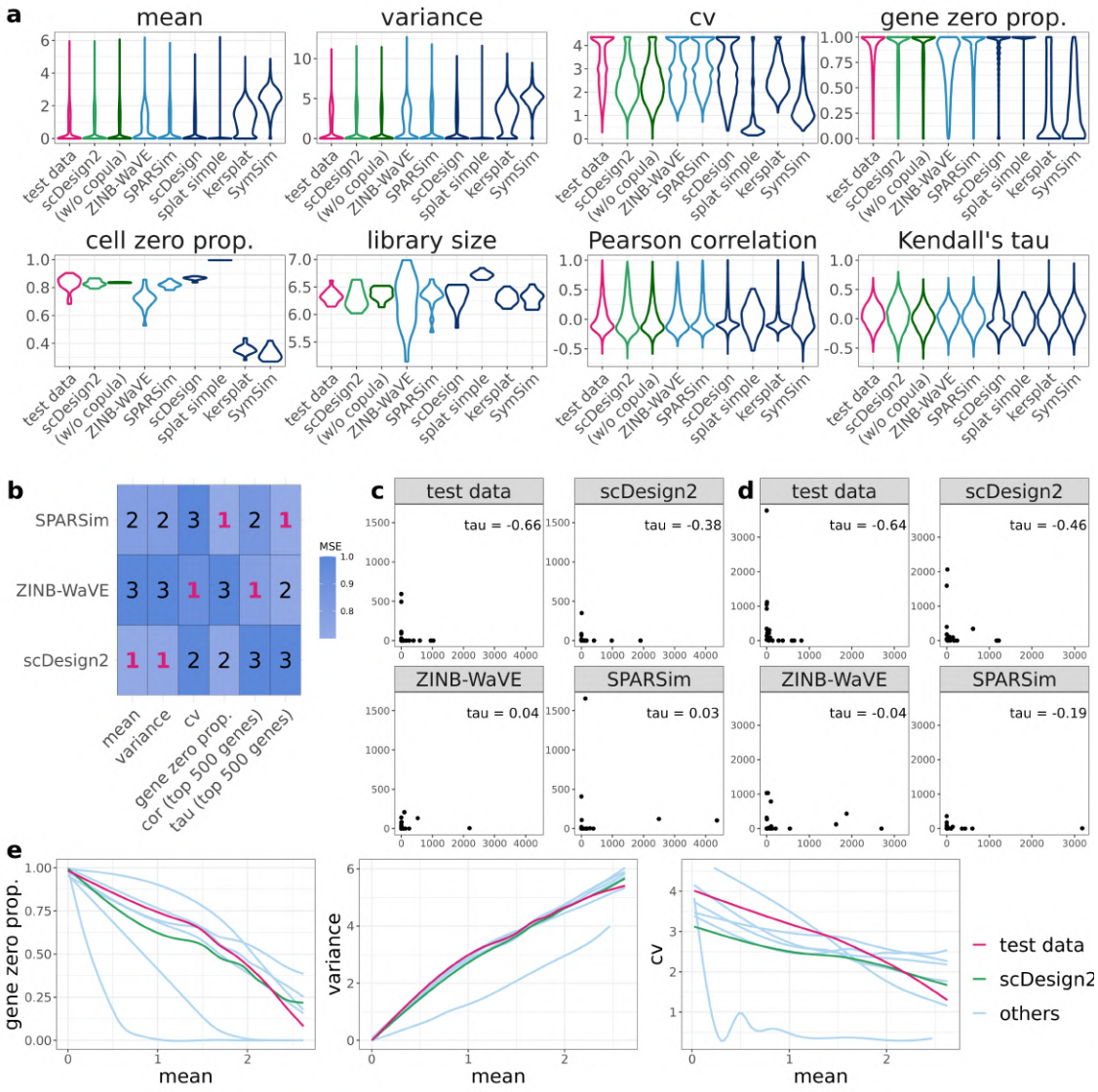

Figure 3.25: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating oligodendrocytes measured by Fluidigm C1 (SMARTer).

The input data is an oligodendrocytes dataset. (a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for two out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

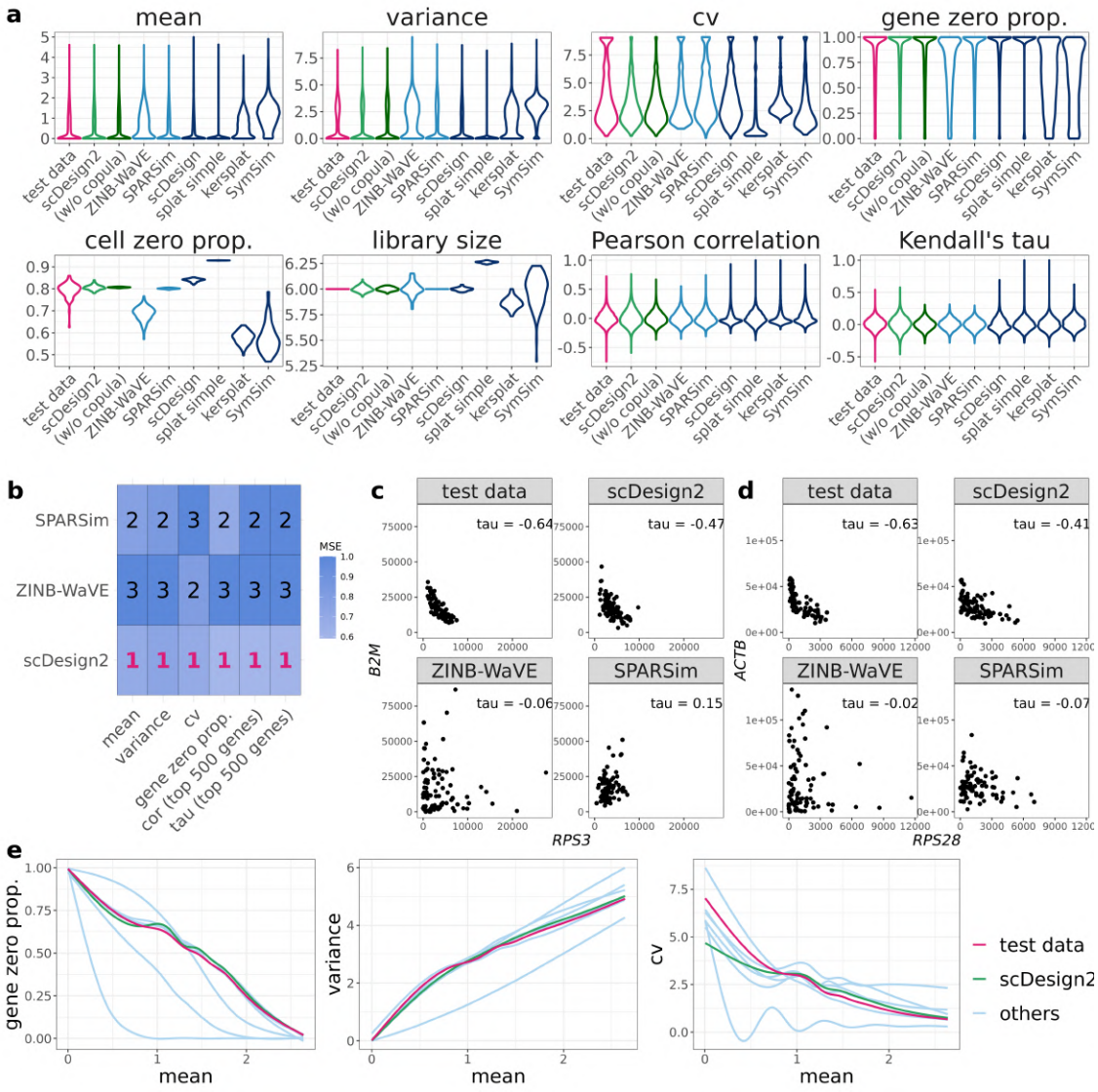

Figure 3.26: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating dendrocytes (subtype 1) measured by Smart-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for six out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*RPS3* vs. *B2M* and *RPS28* vs. *ACTB*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

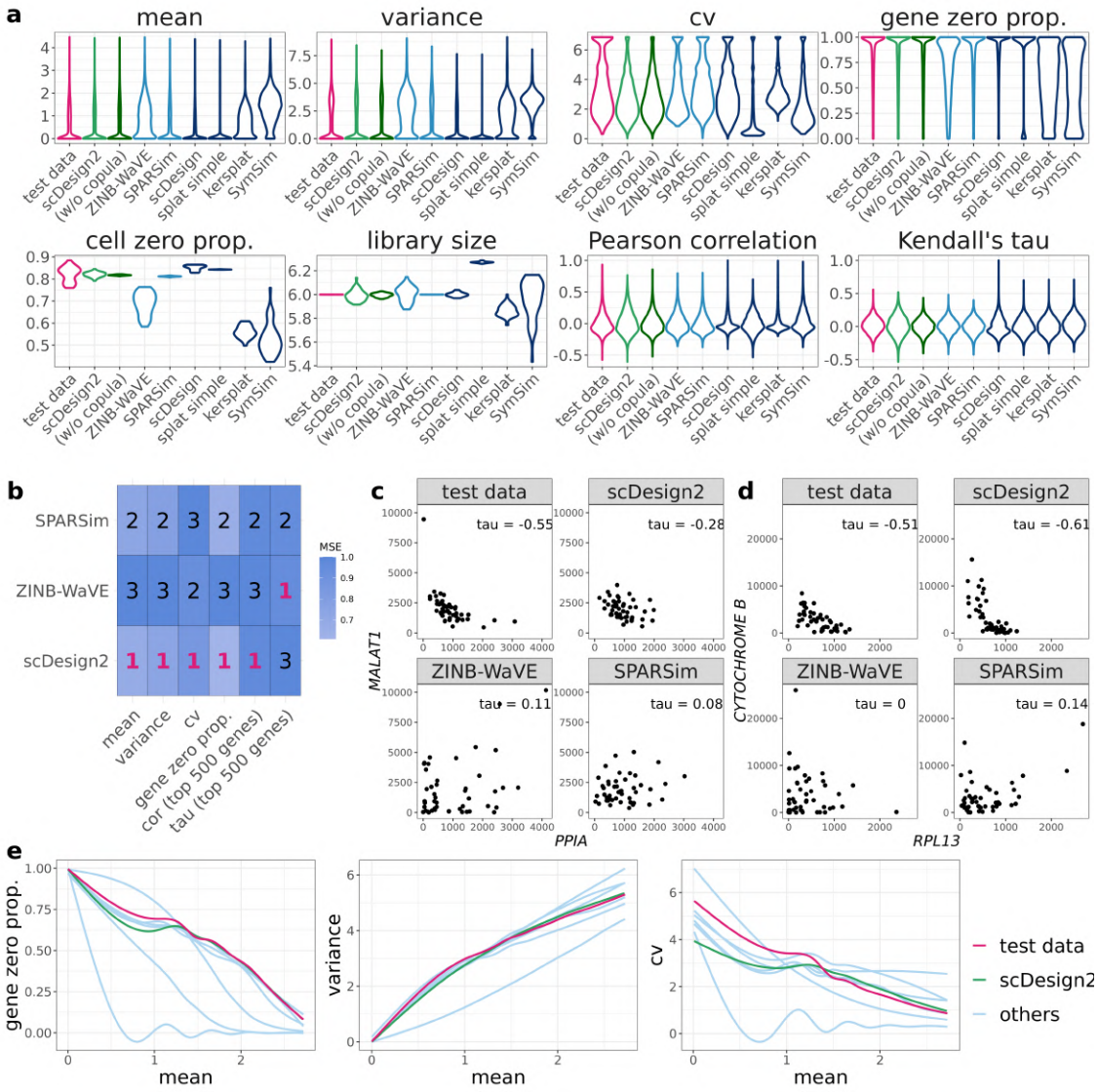

Figure 3.27: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating dendrocytes (subtype 2) measured by Smart-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for five out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*PPIA* vs. *MALAT1* and *RPL13* vs. *CYTOCHROME B*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. Only scDesign2 captures the negative gene correlations in the real data. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

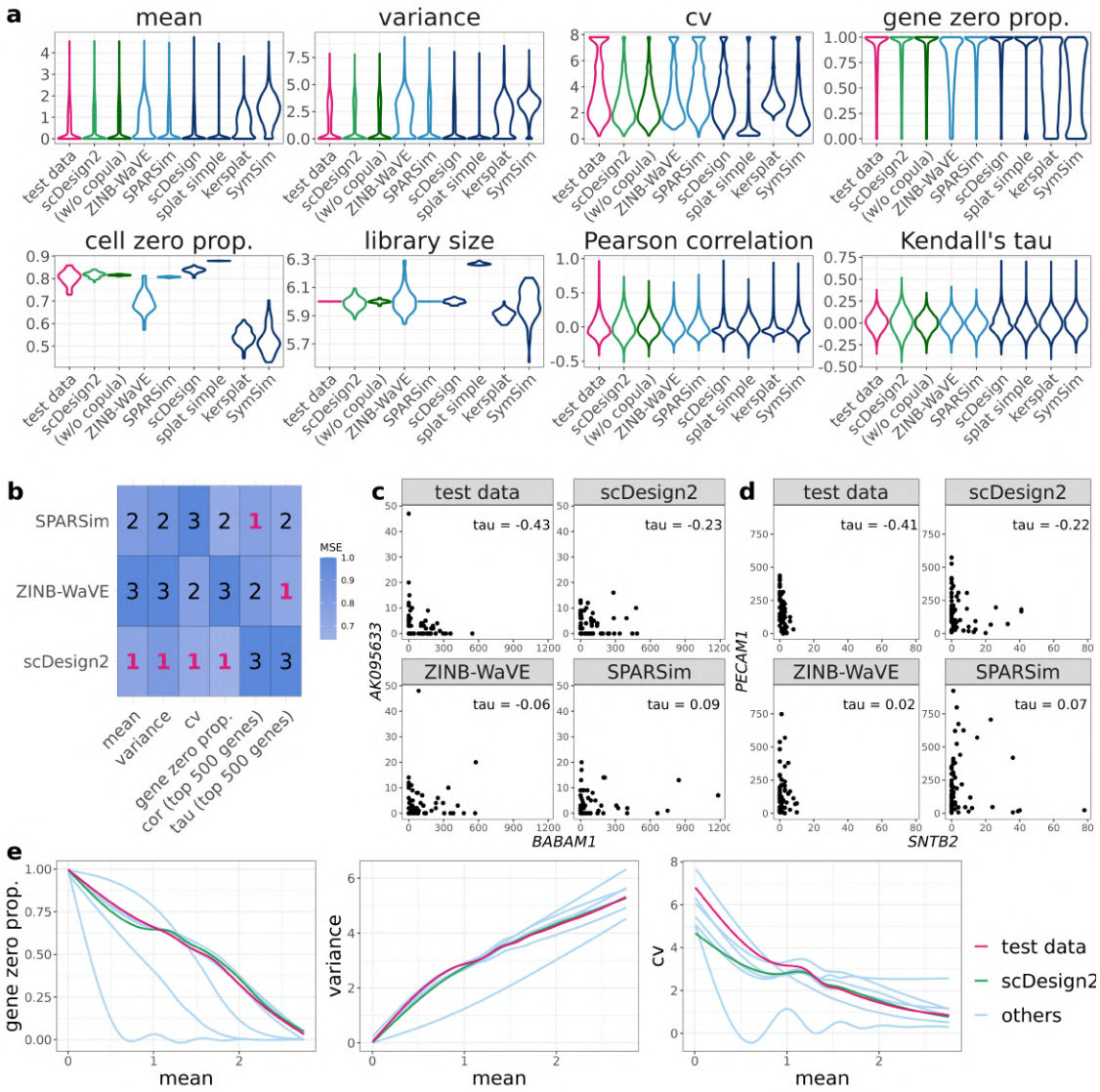

Figure 3.28: Benchmarking scDesign2 against its variant without copula and seven existing scRNA-seq simulators for generating monocytes (subtype 2) measured by Smart-Seq2.

(a) Distributions of eight summary statistics (gene-wise expression mean, variance, coefficient of variation (cv), and zero proportion; cell-wise zero proportion and library size; gene-pair-wise Pearson correlation and Kendall's tau) are plotted based on the real data (test data unused for training simulators) and the synthetic data generated by scDesign2, scDesign2 without copula (w/o copula), ZINB-WaVE, SPARSim, scDesign, three variants of the splatter package (splat simple, splat, and kersplat), and SymSim. (b) Ranking (with 1 being the best-performing method) of scDesign2, ZINB-WaVE, and SPARSim, the only three methods that preserve genes, in terms of the mean-squared error (MSE) of each of six summary statistics (four gene-wise and two gene-pair-wise) between the statistic values in the real data and the synthetic data generated by each simulator. Note that the color scale shows the normalized MSE: for each statistic (column in the table), the normalized MSEs are the MSEs divided by the largest MSE of that statistic. scDesign is ranked the top for four out of the six statistics. For the two gene-pair-wise statistics, we focus on the top 500 highly expressed genes, because as analyzed in the text, they are more meaningful, both biologically and statistically, than the correlations of the lowly expressed genes. (c)-(d) Scatterplots of two example gene pairs—*BABAM1* vs. *AK095633* and *SNTB2* vs. *PECAM1*—based on the real data and the synthetic data generated by scDesign2, ZINB-WaVE, and SPARSim. (e) Smoothed relationships between three pairs of gene-wise statistics (zero proportion vs. mean, variance vs. mean, and cv vs. mean) across all genes (curves plotted by the R function `geom_smooth()`) in the real data and the synthetic data generated by scDesign2 and the seven existing simulators (others). Note that ZINB-WaVE and SymSim filter out certain genes when simulating new data; Pearson correlation and Kendall's tau are only calculated between the genes whose zero proportions are less than 50%; gene-wise mean and variance and cell-wise library size are transformed to the $\log_{10}(1+x)$ scale (where x represents a statistic's value).

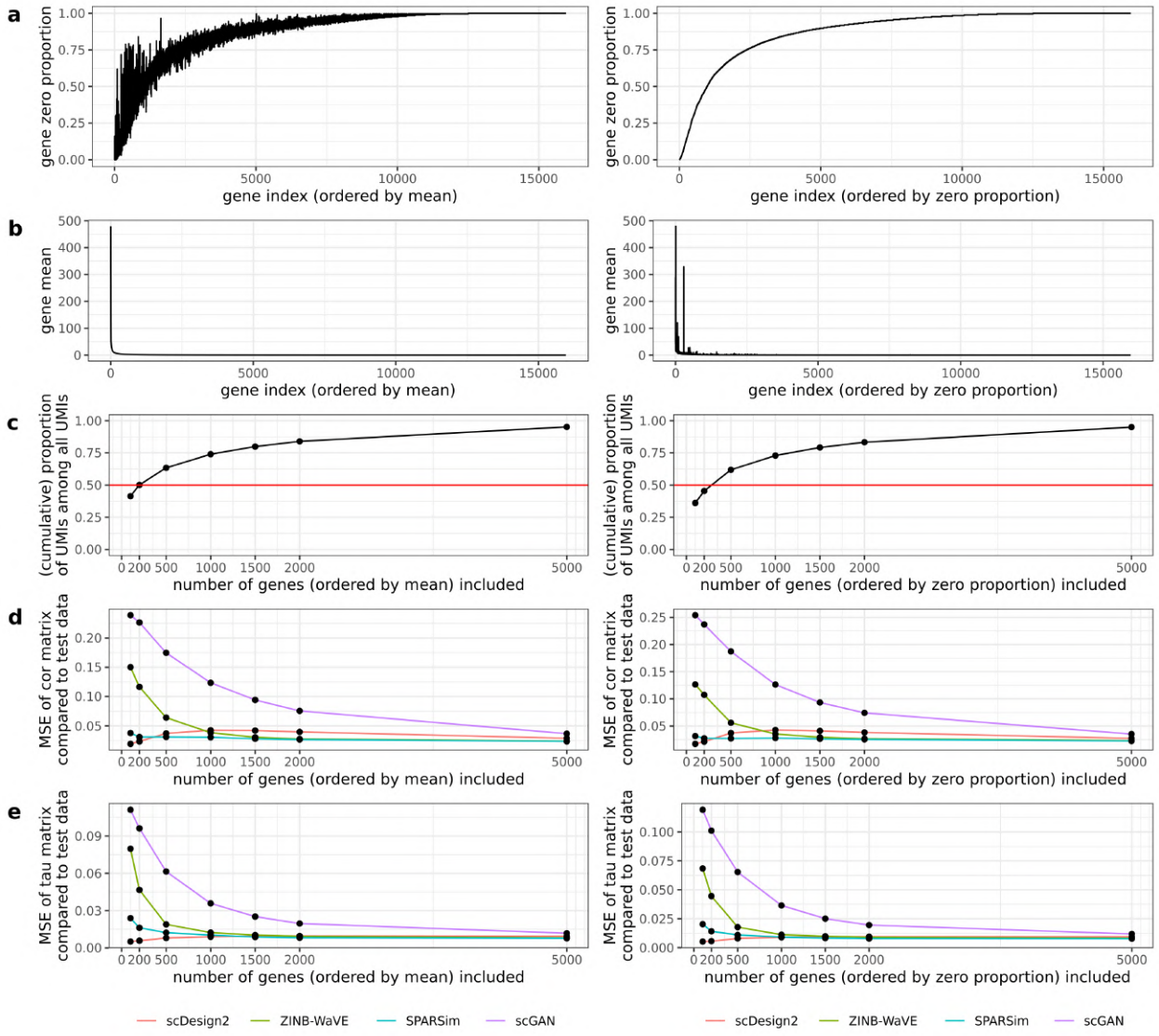

Figure 3.29: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by four simulators trained on the 10x Genomics goblet cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

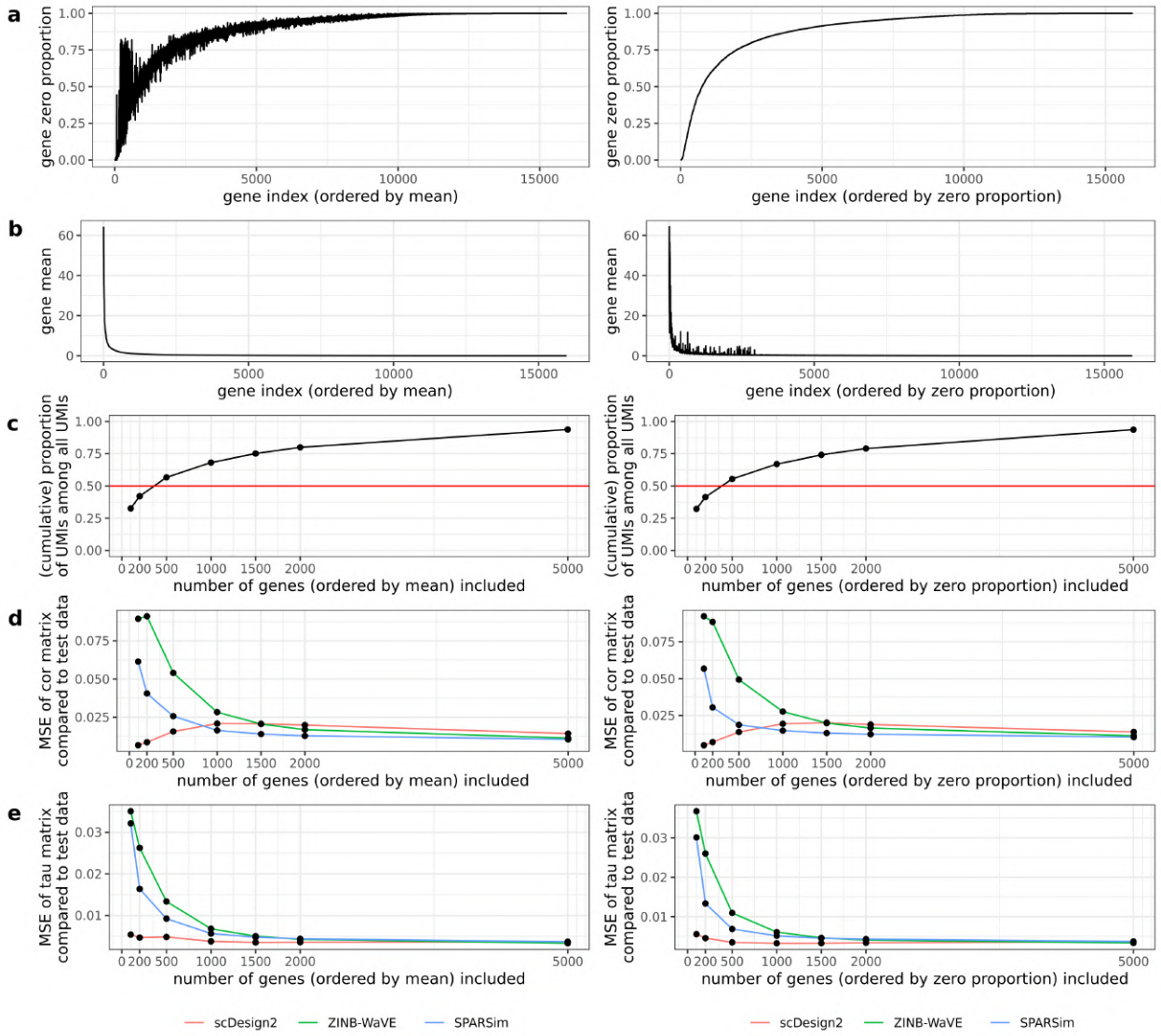

Figure 3.30: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the 10x Genomics stem cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

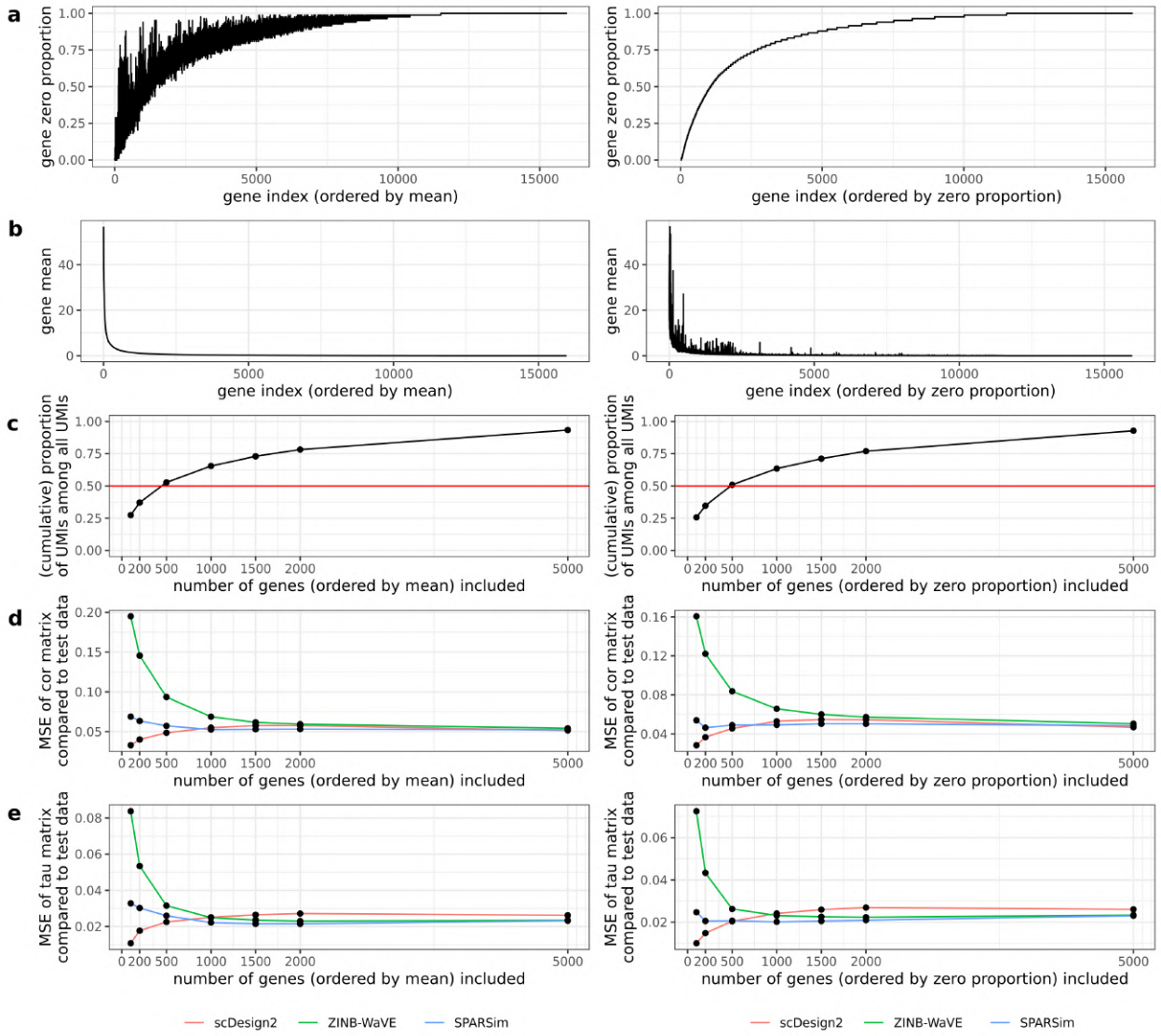

Figure 3.31: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the 10x Genomics tuft cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

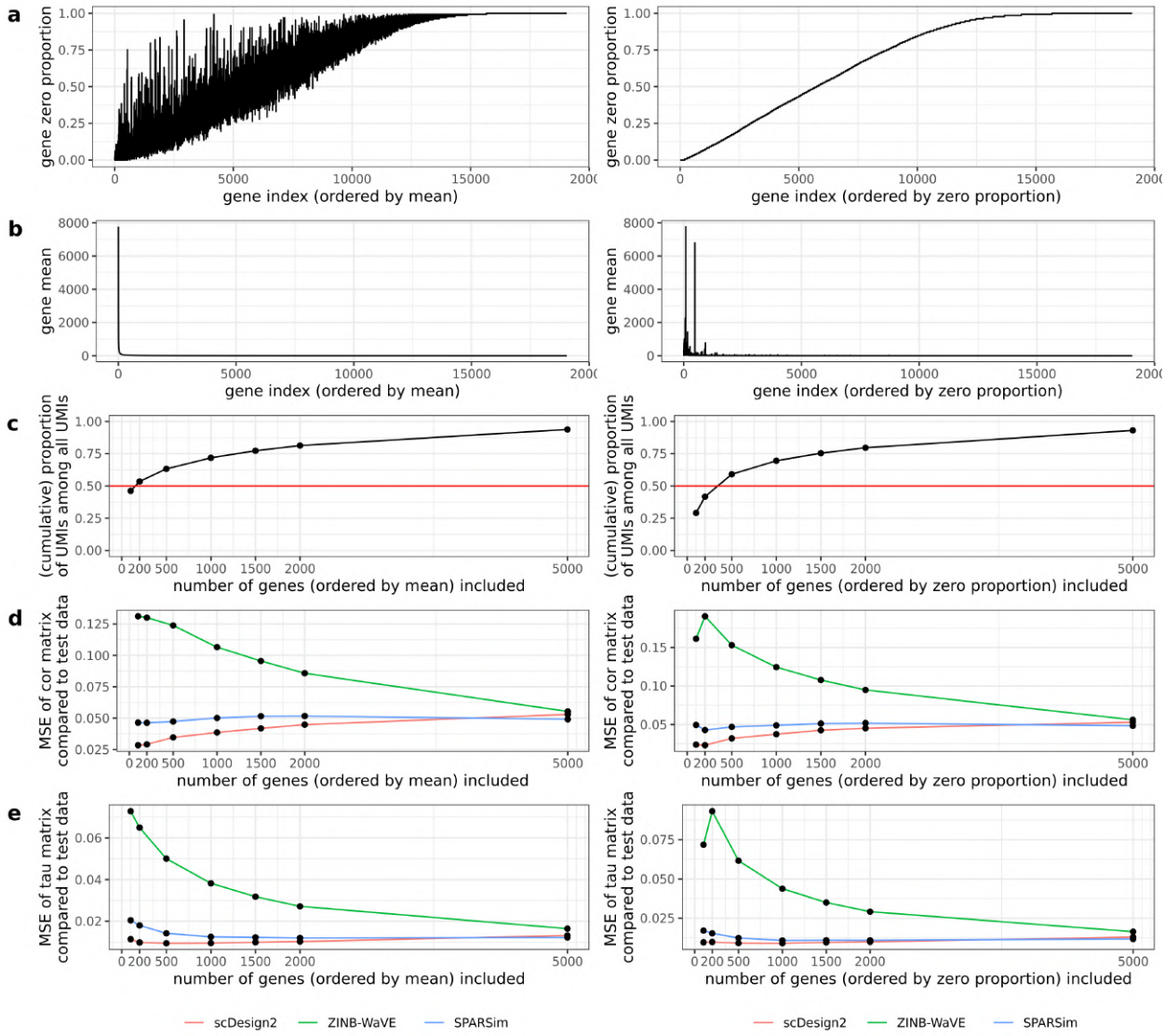

Figure 3.32: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the CEL-Seq2 acinar cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

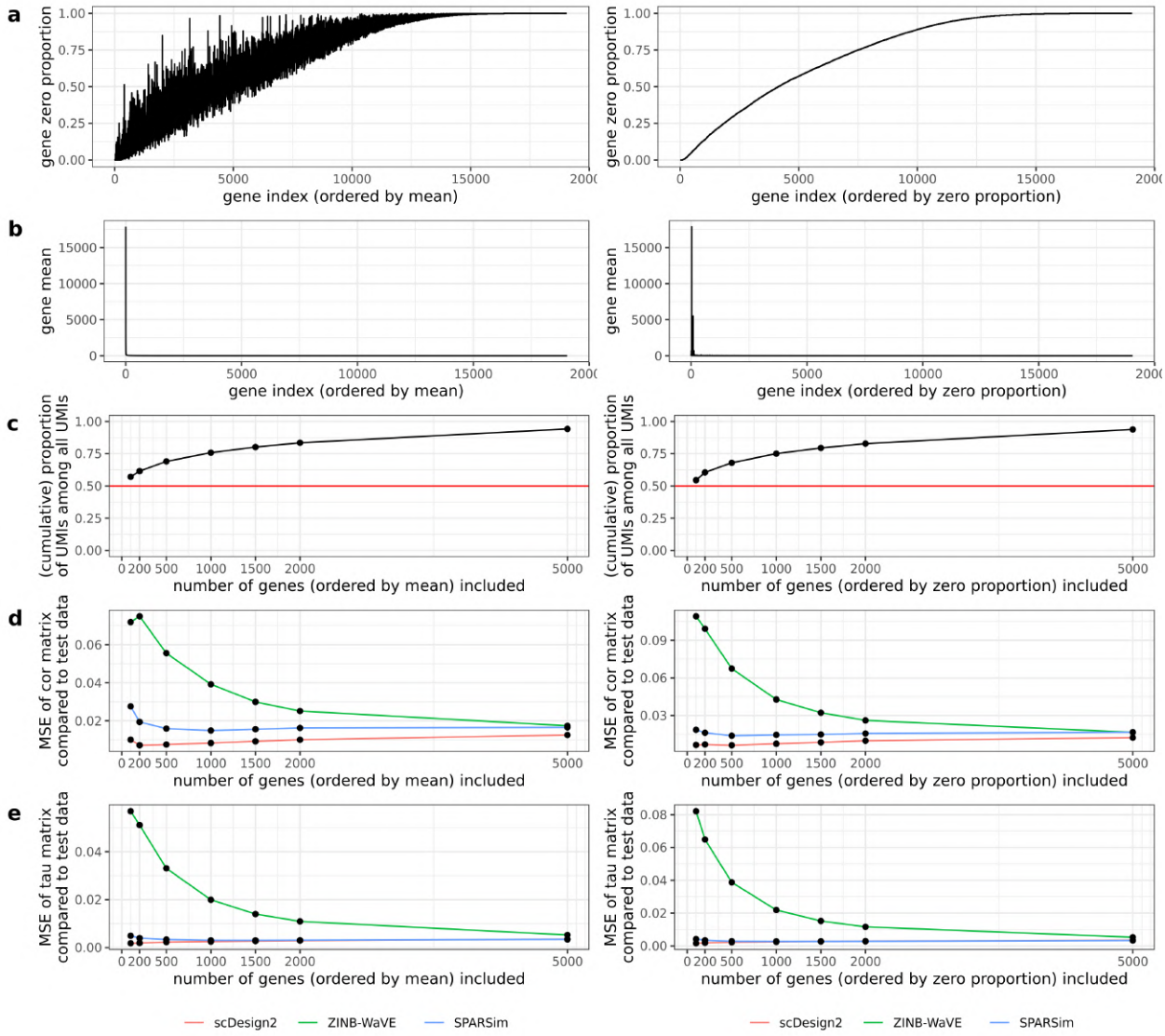

Figure 3.33: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the CEL-Seq2 alpha cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

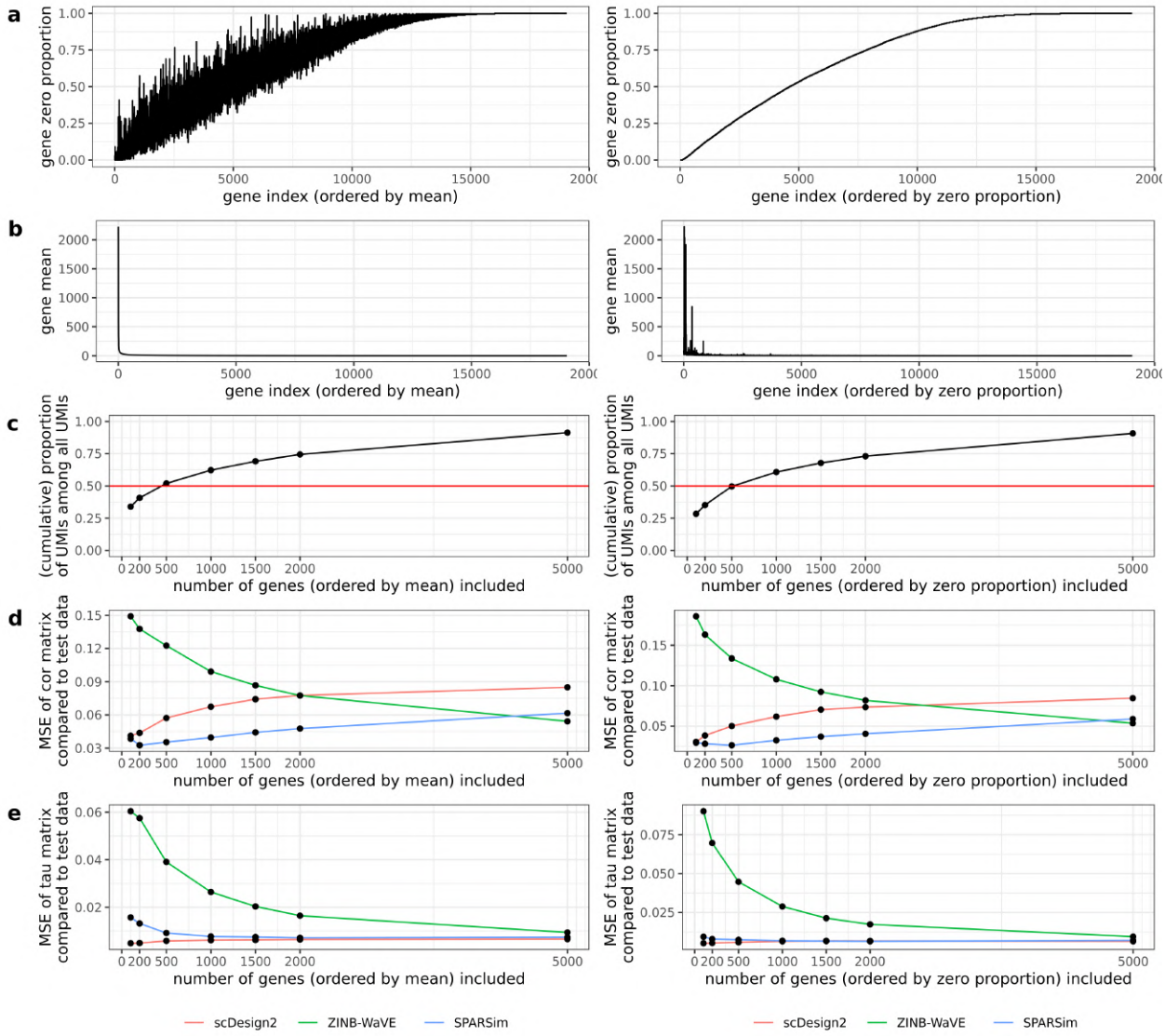

Figure 3.34: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the CEL-Seq2 beta cell data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

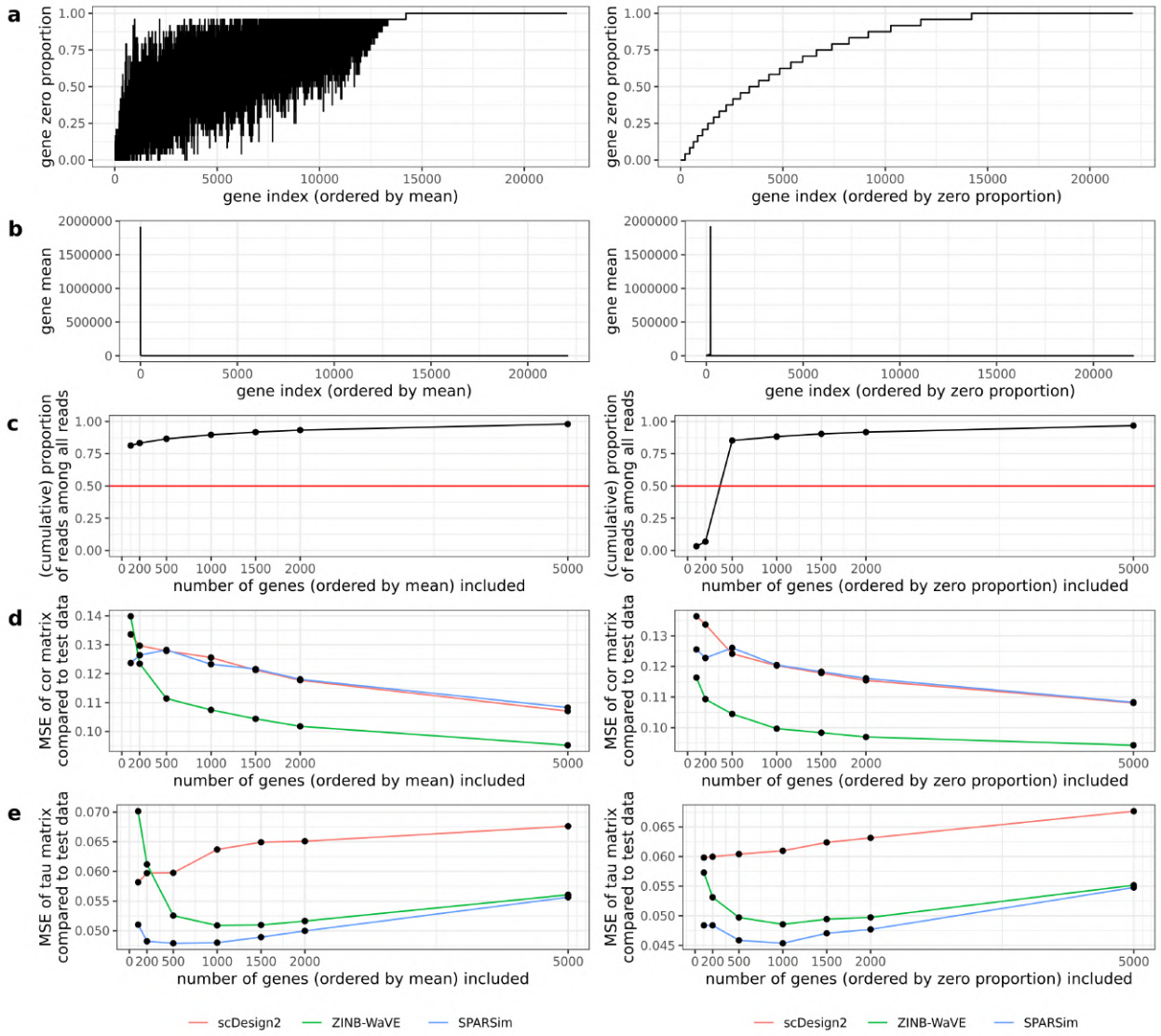

Figure 3.35: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Fluidigm C1 astrocytes data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

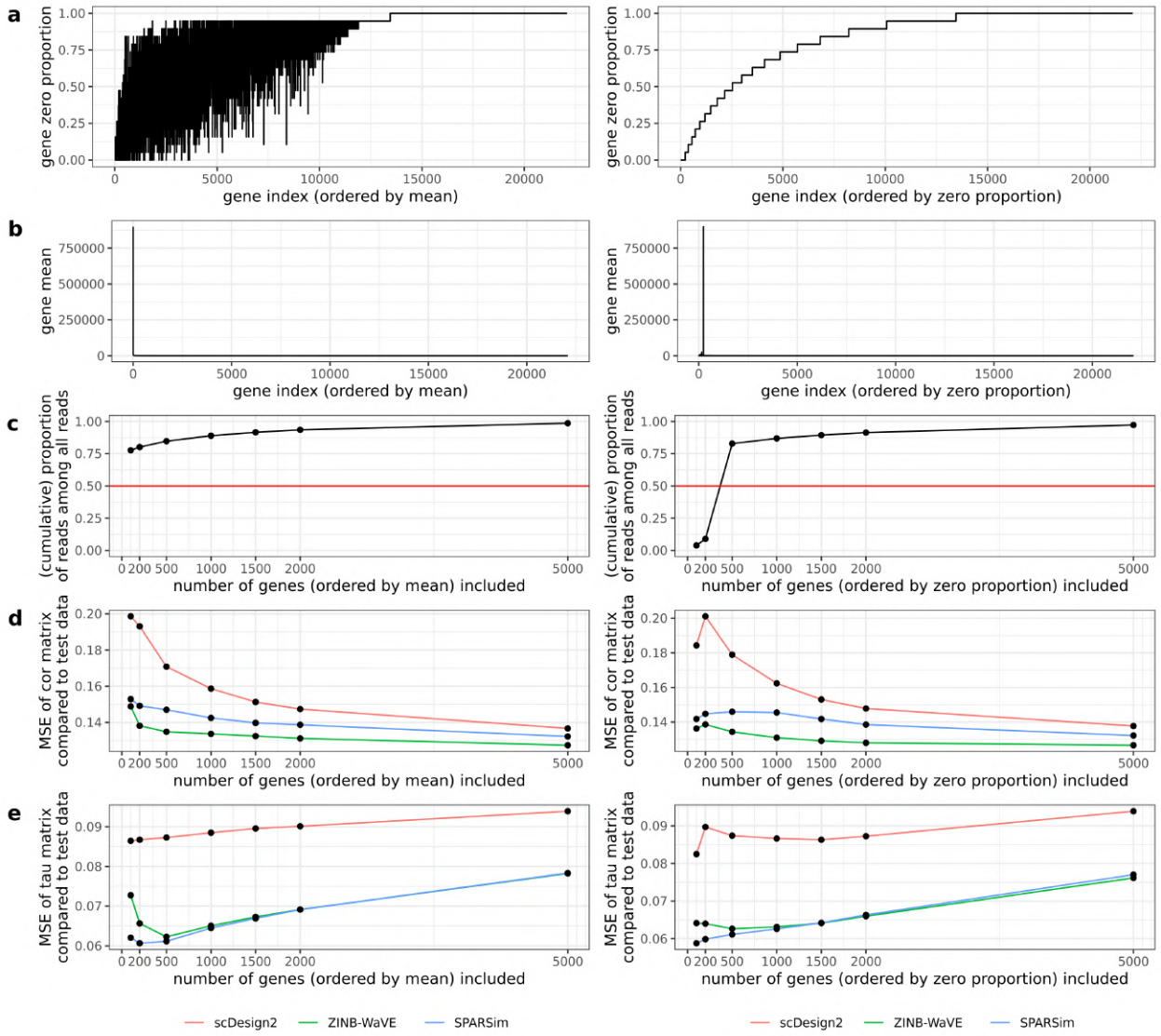

Figure 3.36: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Fluidigm C1 oligodendrocytes data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

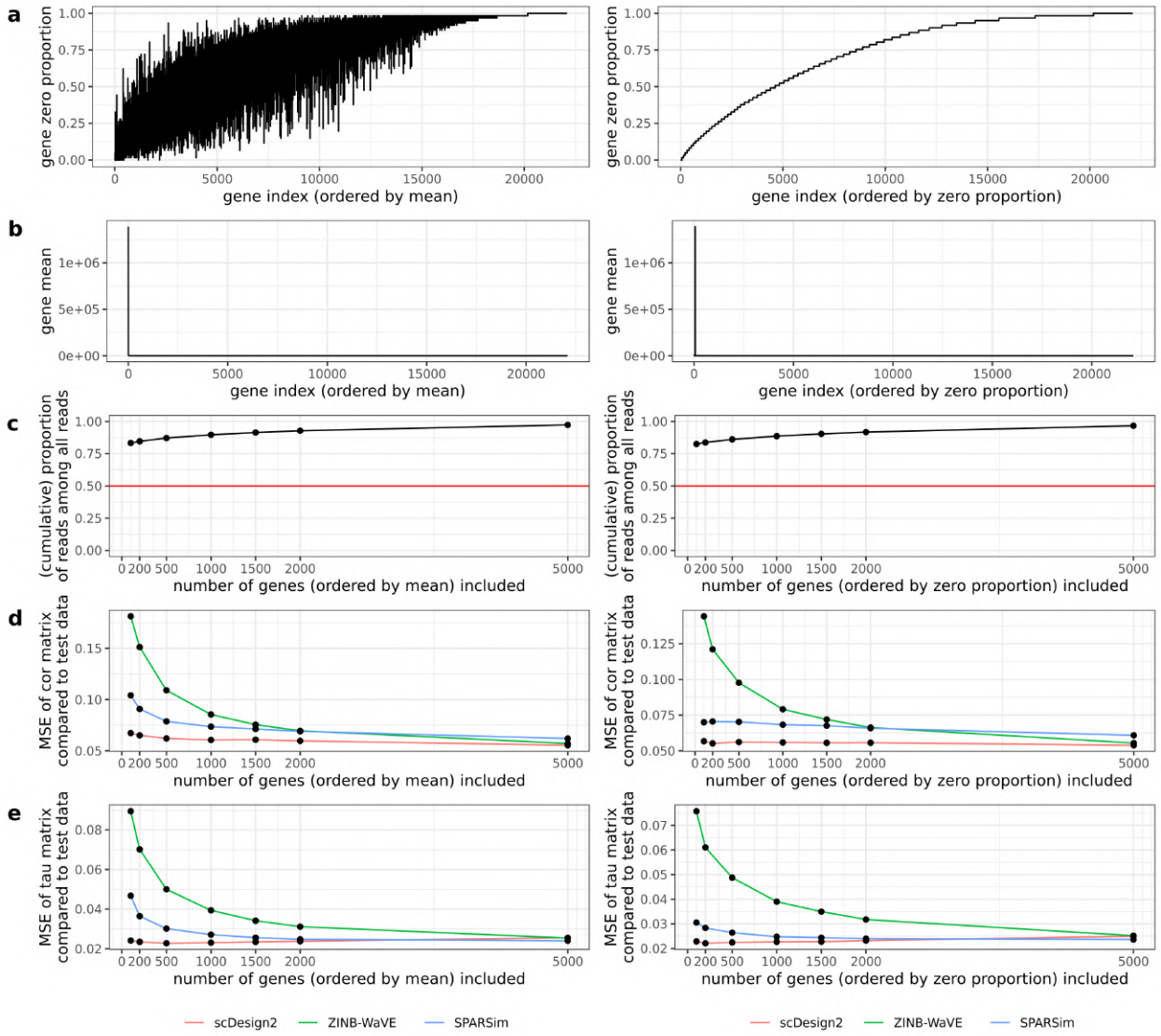

Figure 3.37: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Fluidigm C1 neurons data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

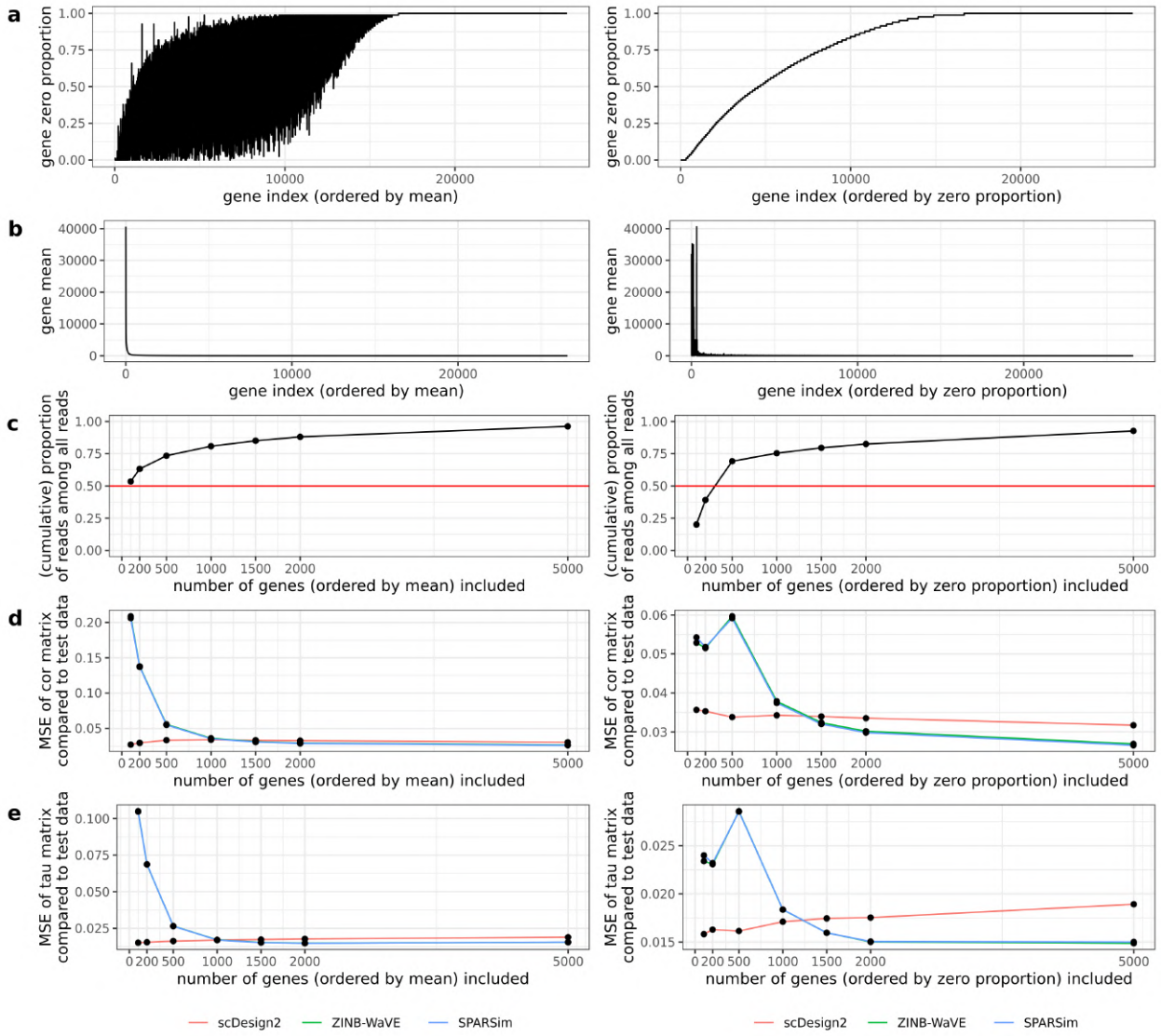

Figure 3.38: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Smart-Seq2 dendrocyte (subtype 1) data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

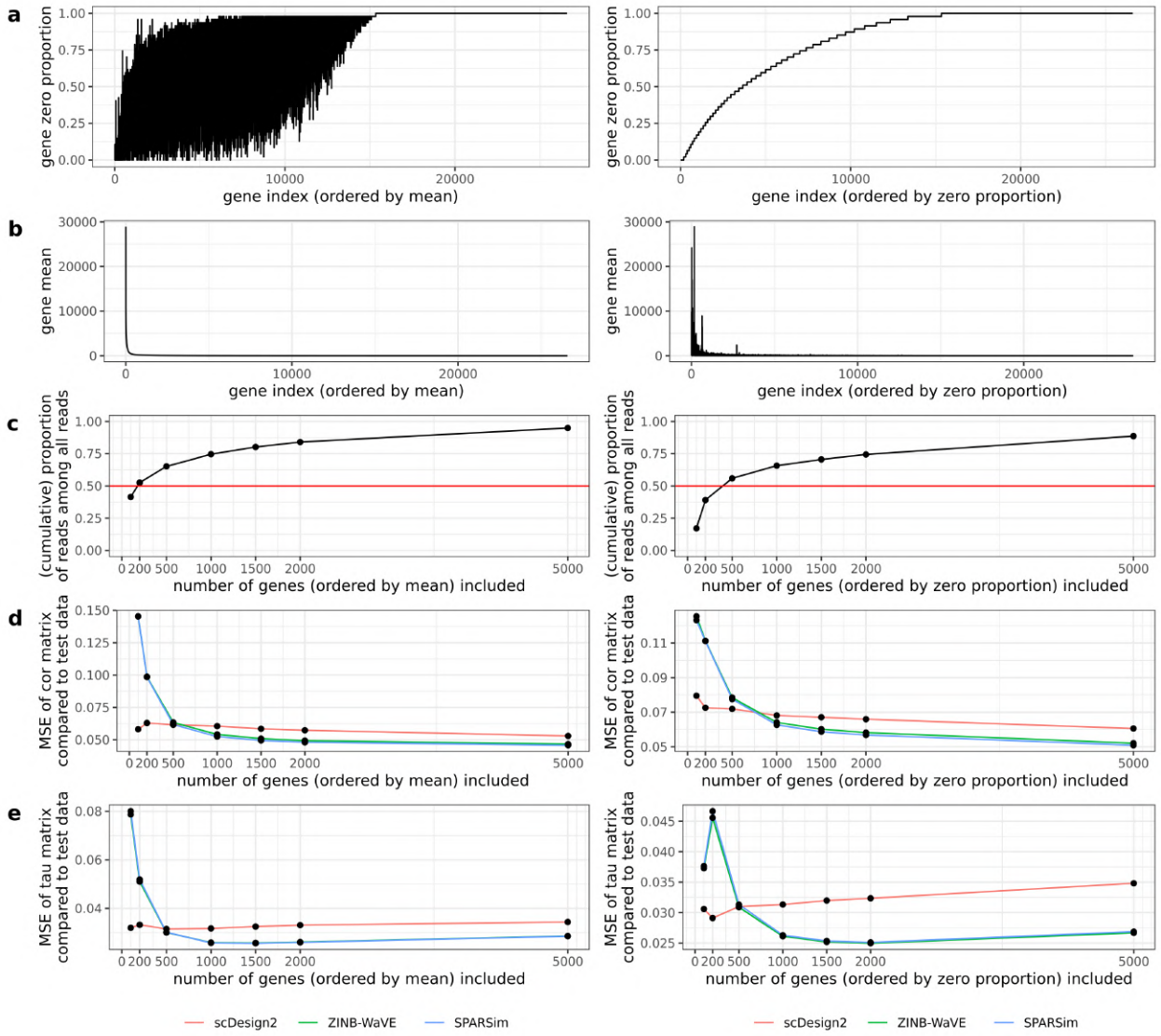

Figure 3.39: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Smart-Seq2 dendrocyte (subtype 2) data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

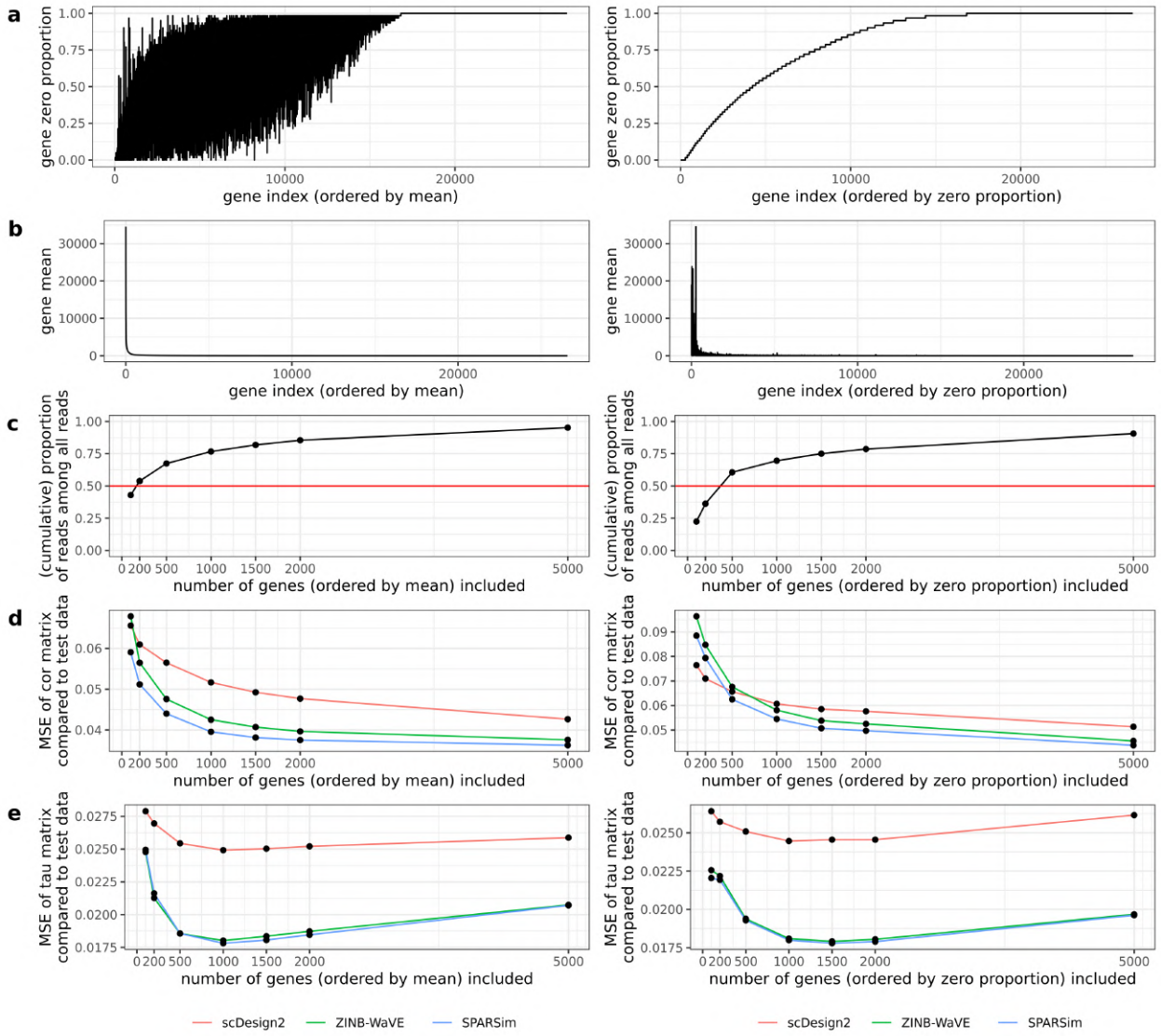

Figure 3.40: Relationships of the mean squared error (MSE) vs. the dimension (i.e., number of genes) of the (Pearson or Kendall's tau) gene correlation matrices, which are estimated from the synthetic data generated by three simulators trained on the Smart-Seq2 monocyte (subtype 2) data.

The genes are ordered in two ways, by their mean expression from high to low (left) or by their zero proportion from low to high (right). The plots are shown for the top 100, 200, 500, 1000, 1500, and 2000 genes ordered in either way. (a) The zero proportion of each top gene. (b) The mean expression of each top gene. (c) The relationship of the cumulative proportion of the top genes' UMIs among all UMIs vs. the number of top genes. (d)–(e) The MSE is defined as the average per-entry squared difference between the correlation matrices estimated from each synthetic dataset and the test data. Each plot shows the relationship of the MSE of (d) the Pearson correlation matrix or (e) the Kendall's tau matrix for the top genes estimated from each simulator's synthetic data vs. the number of top genes.

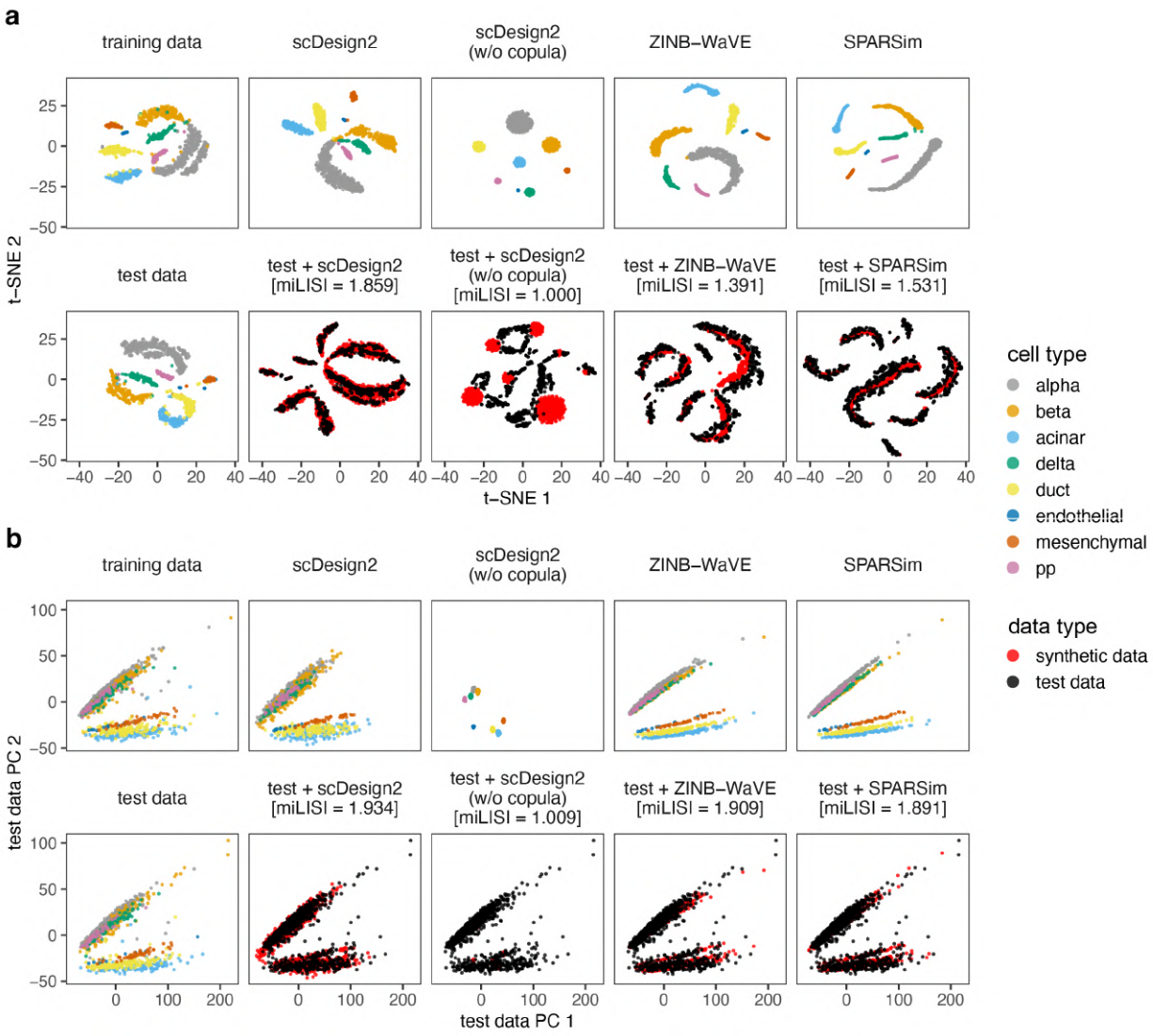

Figure 3.41: Comparison of CEL-Seq2 data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction. We find that the synthetic data generated by scDesign2 most resemble the test data.

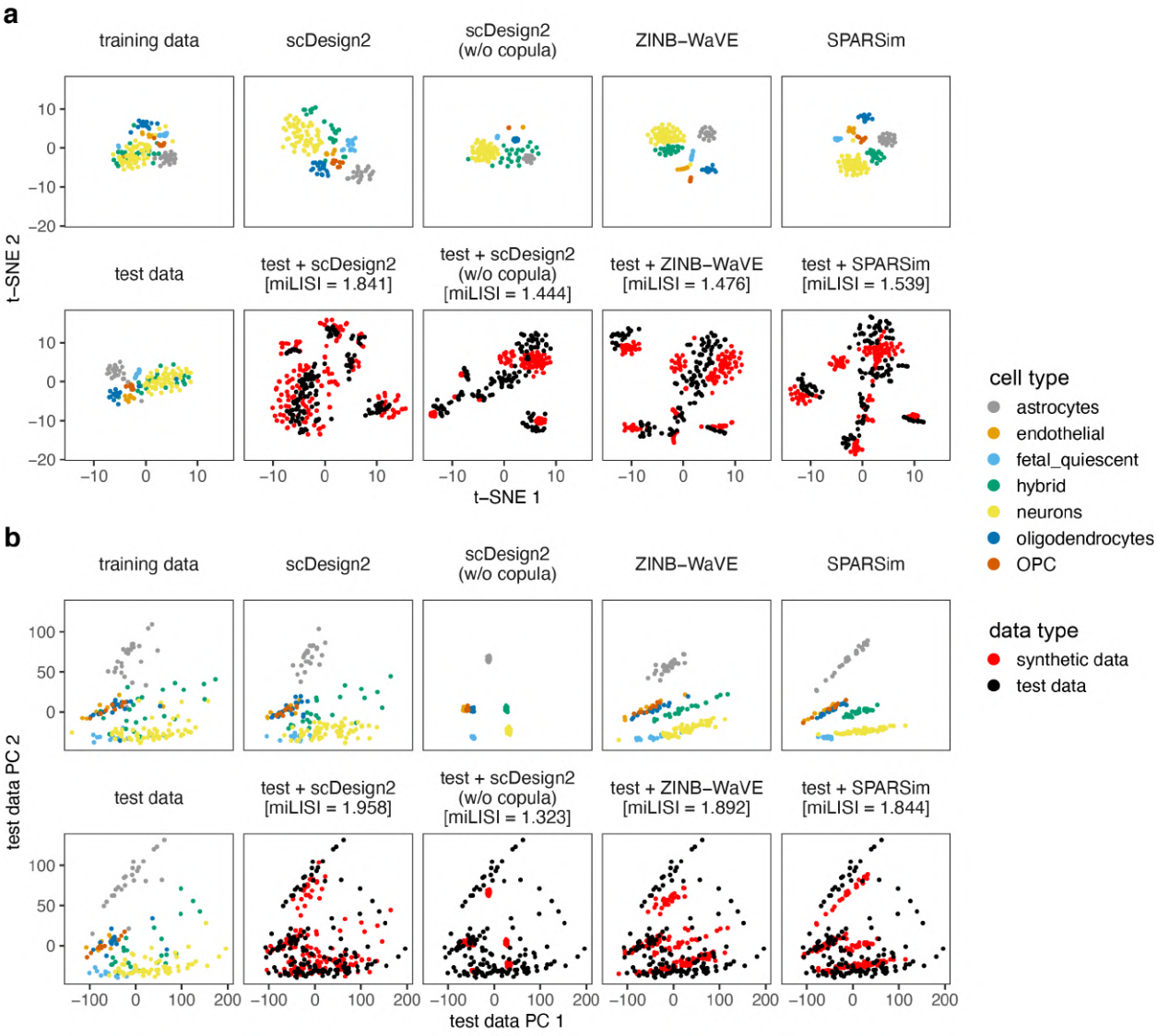

Figure 3.42: Comparison of Fluidigm C1 (SMARTer) data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction. We find that the synthetic data generated by scDesign2 most resemble the test data.

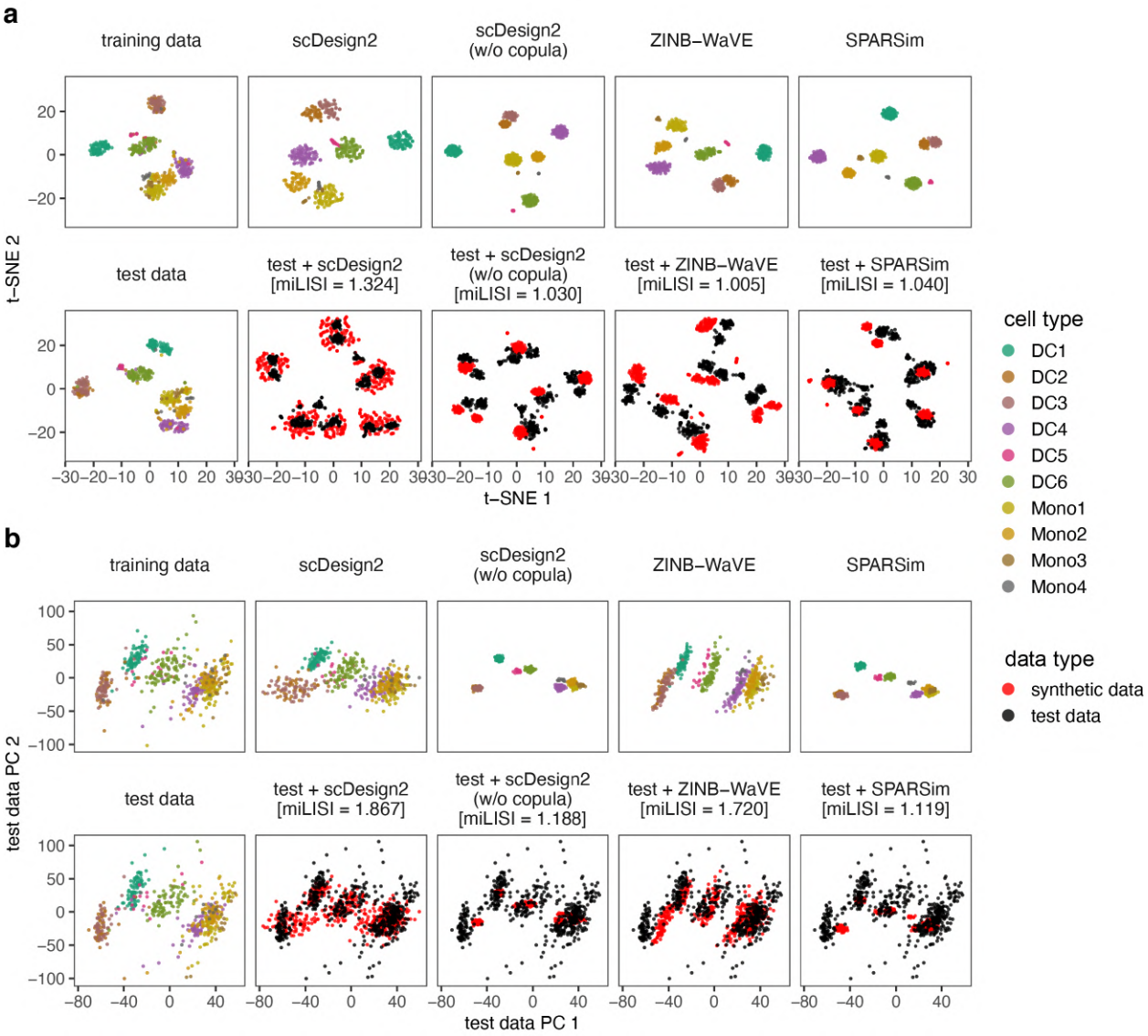

Figure 3.43: Comparison of Smart-Seq2 data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction. We find that the synthetic data generated by scDesign2 most resemble the test data.

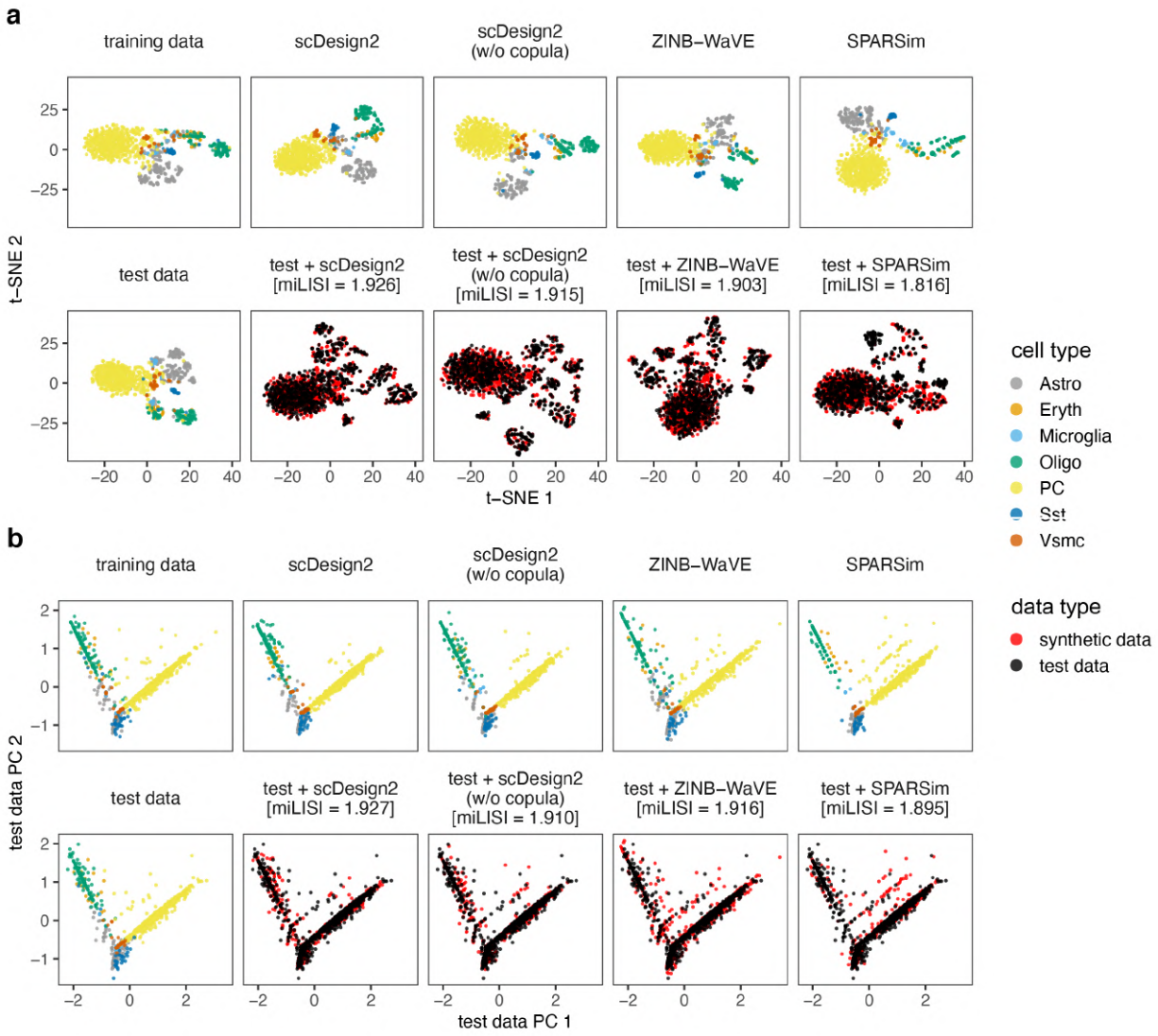

Figure 3.44: Comparison of pciSeq data and synthetic data generated by scDesign2, its variant without copula, ZINB-WaVE, and SPARSim in 2D visualization.

(a) t-SNE plots and (b) principal component (PC) plots of training data, test data, synthetic data generated by each simulator, and combinations of test data and each synthetic dataset. Gene expression counts are transformed as log(1 + count) before dimensionality reduction. miLISI is short for median integration local inverse Simpson's Index, a higher value of which indicates that the simulated data mix better with the test data in the 2D visualization plot. By visually inspecting the patterns in these plots as well as comparing the miLISI values, we find that the synthetic data generated by scDesign2 most resemble the test data.

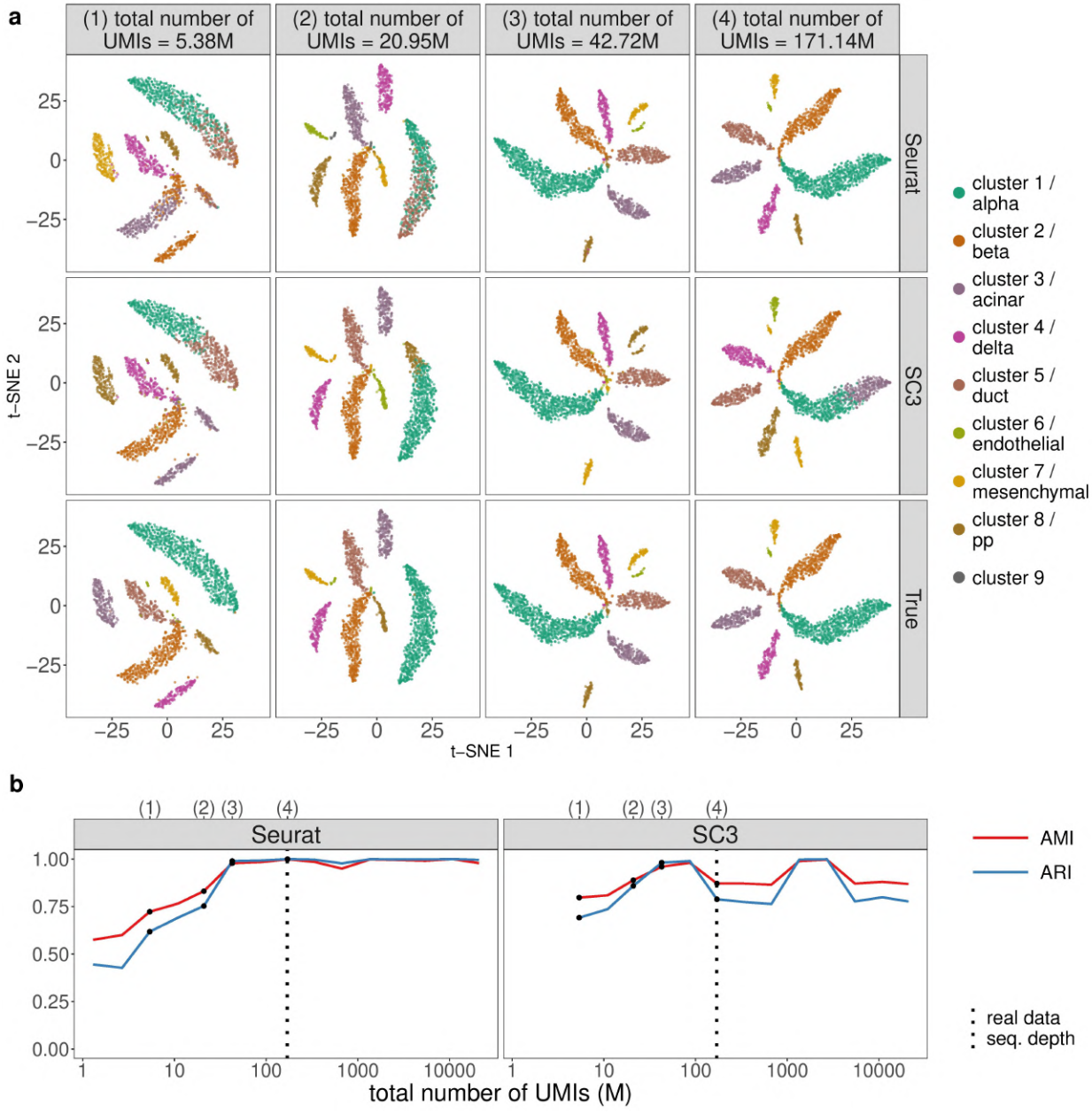

Figure 3.45: scDesign2 guides the choice of sequencing depth in cell clustering.

scDesign2 generates synthetic CEL-Seq2 data with fifteen sequencing depths. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four sequencing depths in (a) are marked as dots and in the top, and the sequencing depth of the real dataset [120] is marked as vertical dashed lines.

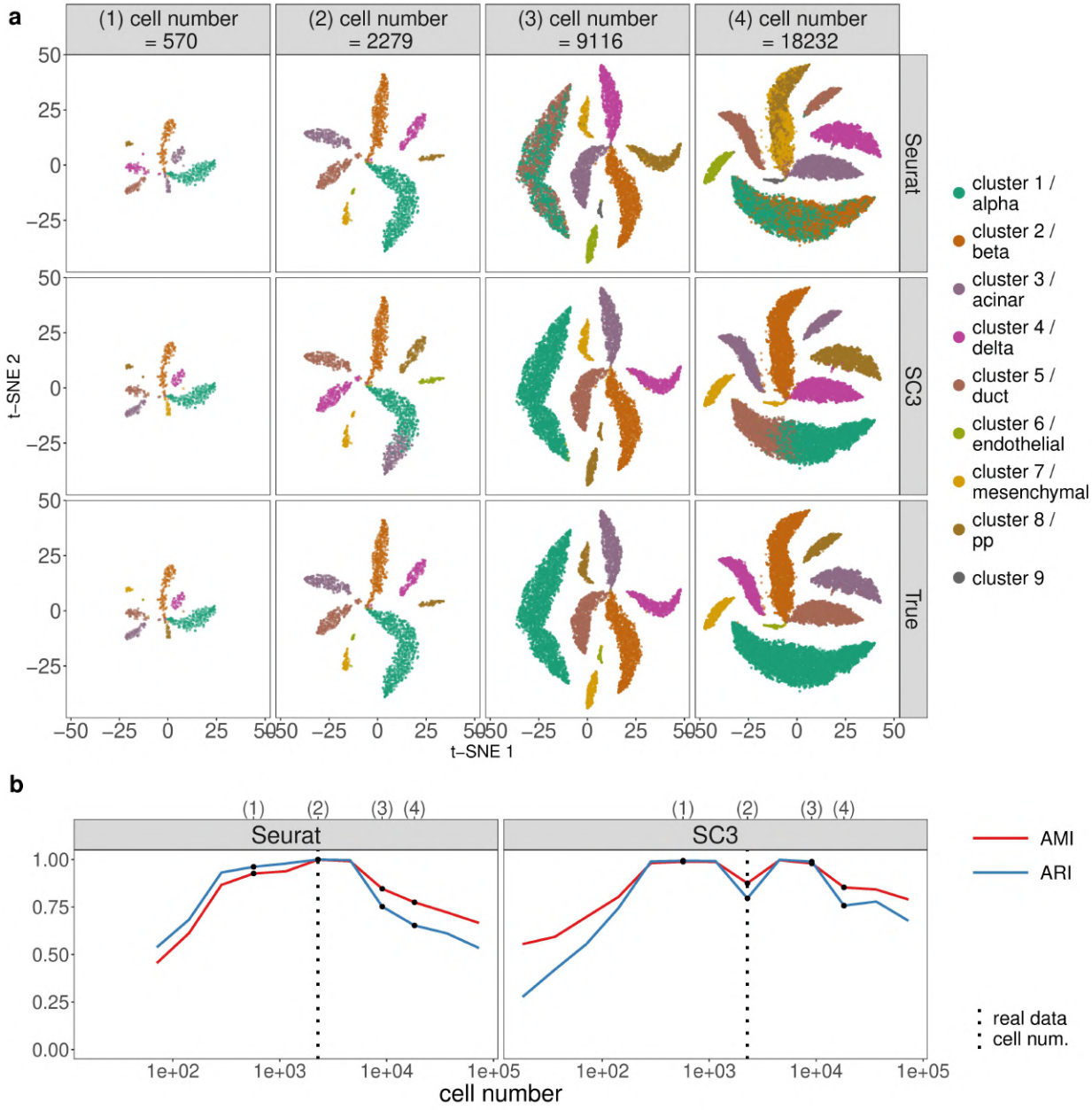

Figure 3.46: scDesign2 guides the choice of cell number in cell clustering, in the case where the total sequencing depth is kept as fixed.

scDesign2 generates synthetic CEL-Seq2 data with thirteen cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [120] is marked as vertical dashed lines.

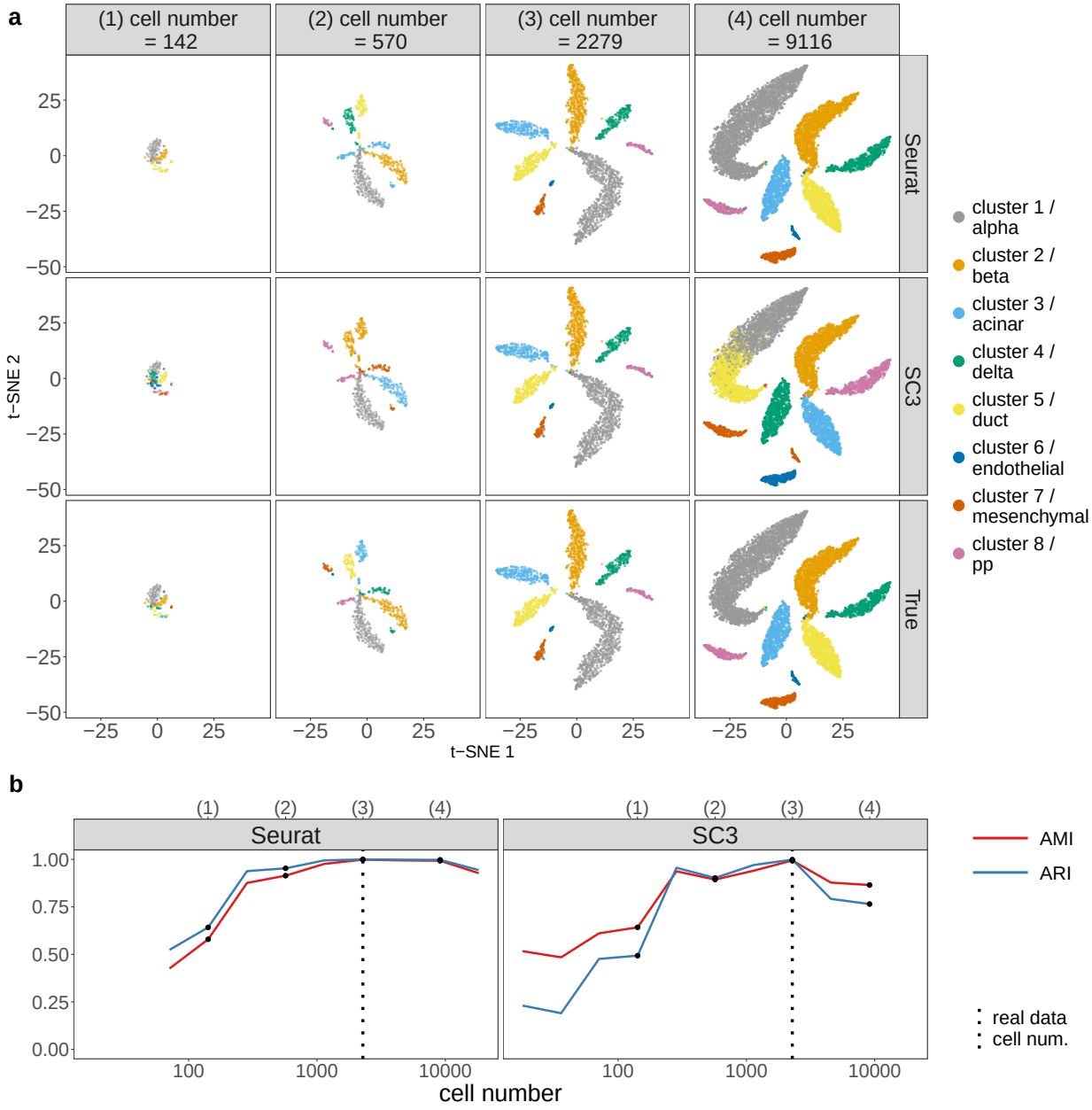

Figure 3.47: scDesign2 guides the choice of cell number in cell clustering, in the case where the average sequencing depth is kept as fixed.

scDesign2 generates synthetic CEL-Seq2 data with ten cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [120] is marked as vertical dashed lines.

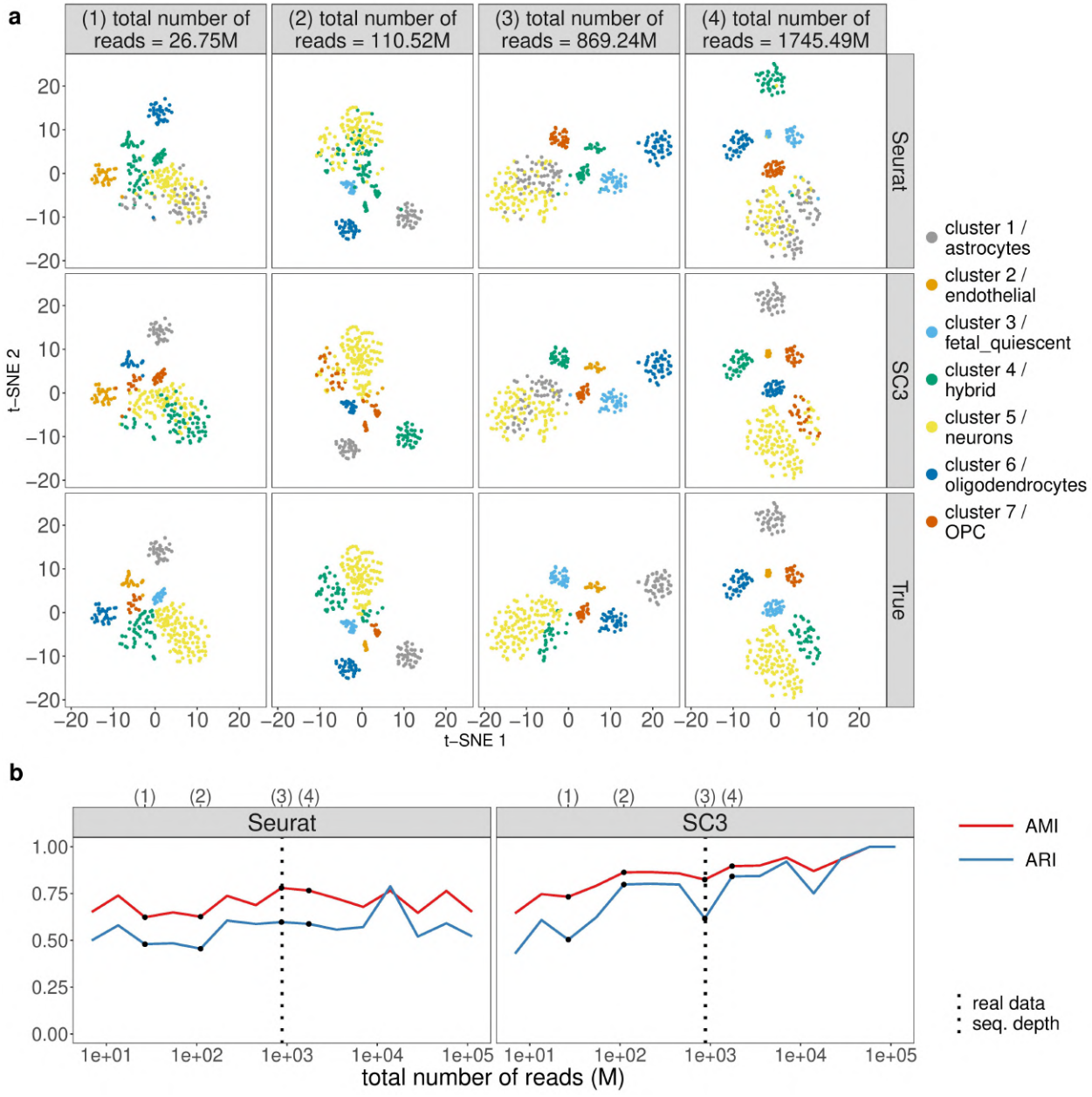

Figure 3.48: scDesign2 guides the choice of sequencing depth in cell clustering.

scDesign2 generates synthetic Fluidigm C1 (SMARTer) data with fifteen sequencing depths. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four sequencing depths in (a) are marked as dots and in the top, and the sequencing depth of the real dataset [121] is marked as vertical dashed lines.

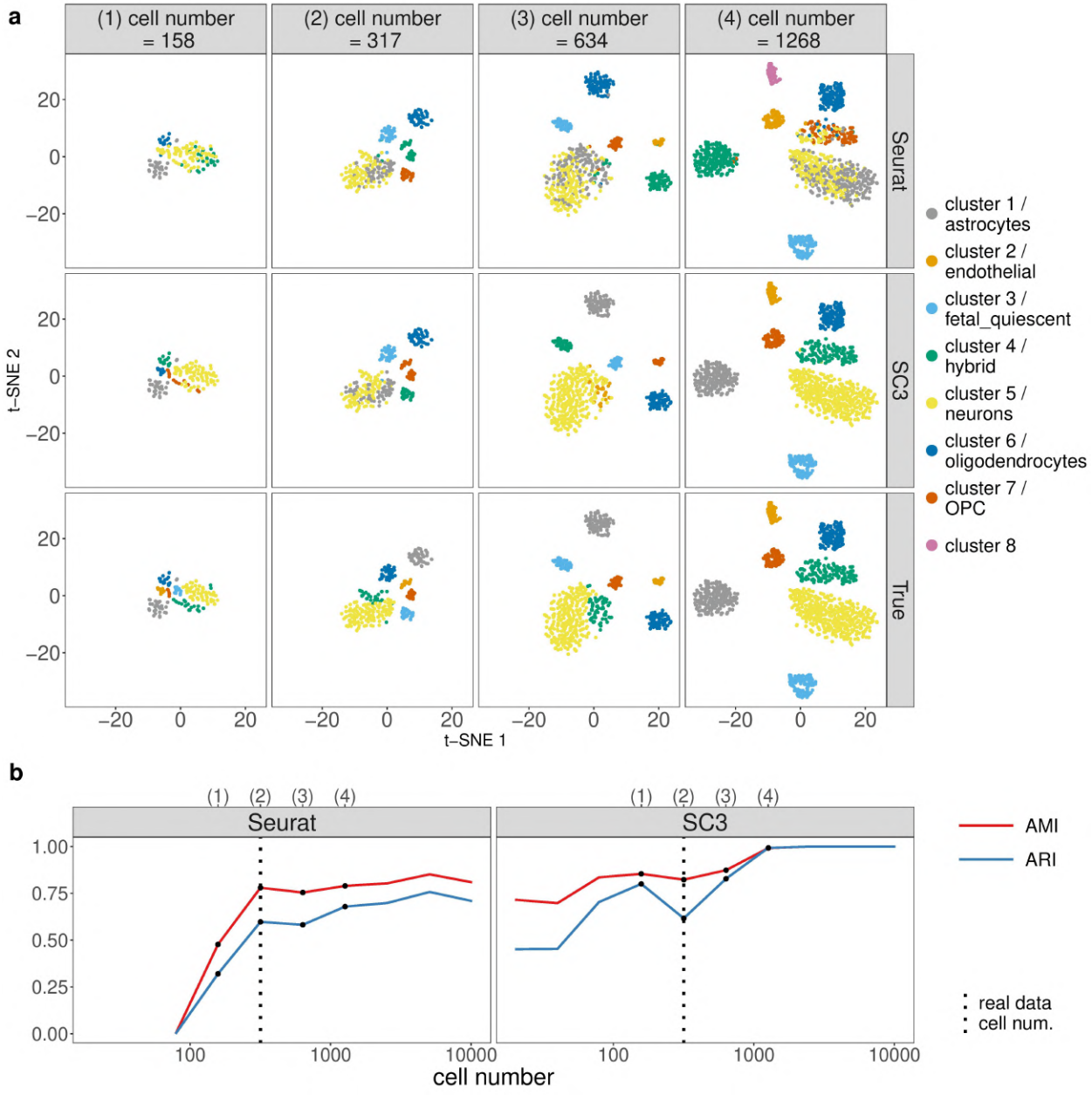

Figure 3.49: scDesign2 guides the choice of cell number in cell clustering, in the case where the total sequencing depth is kept as fixed.

scDesign2 generates synthetic Fluidigm C1 (SMARTer) data with ten cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [121] is marked as vertical dashed lines.

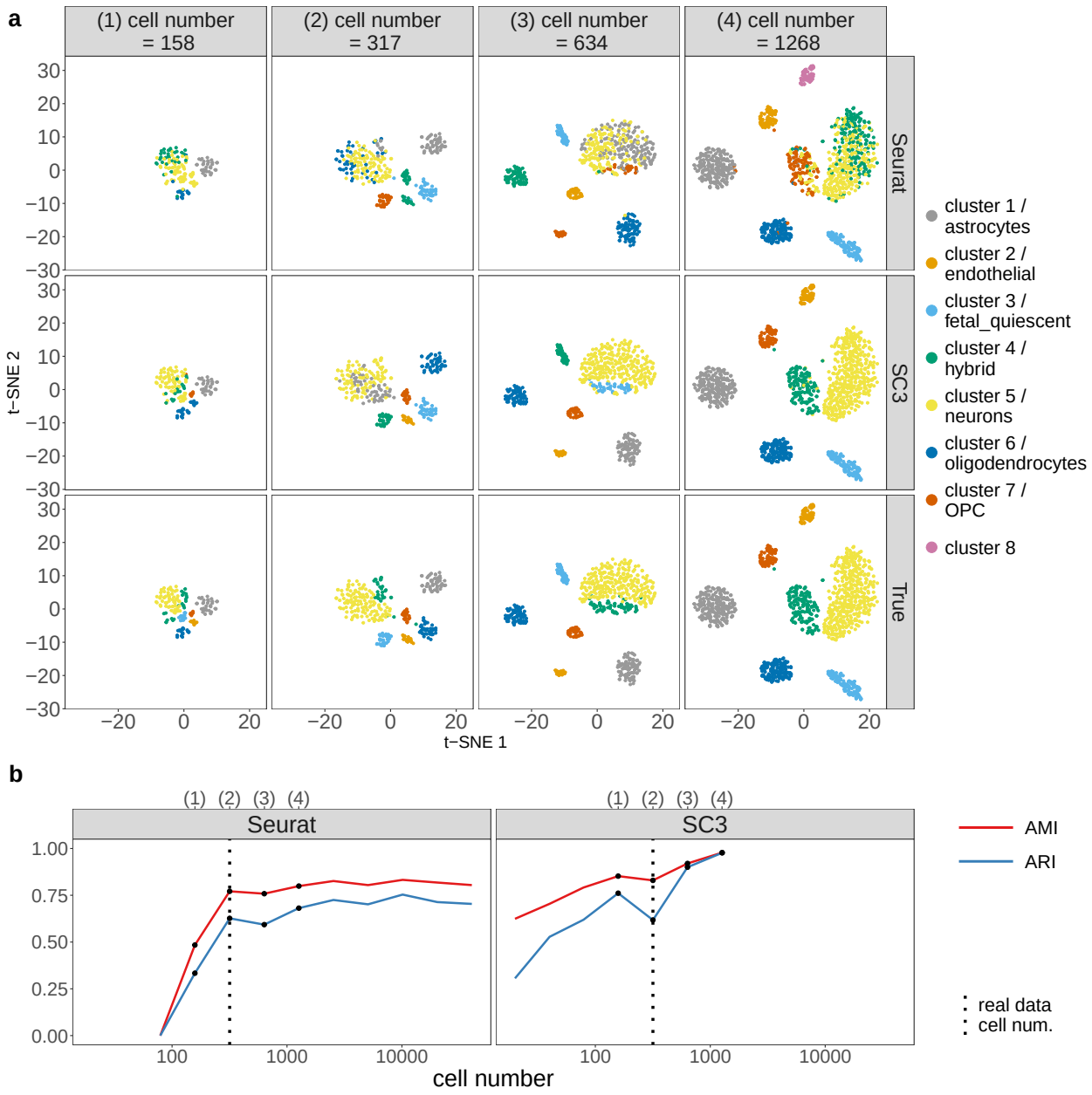

Figure 3.50: scDesign2 guides the choice of cell number in cell clustering, in the case where the average sequencing depth is kept as fixed.

scDesign2 generates synthetic Fluidigm C1 (SMARTer) data with twelve cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [121] is marked as vertical dashed lines.

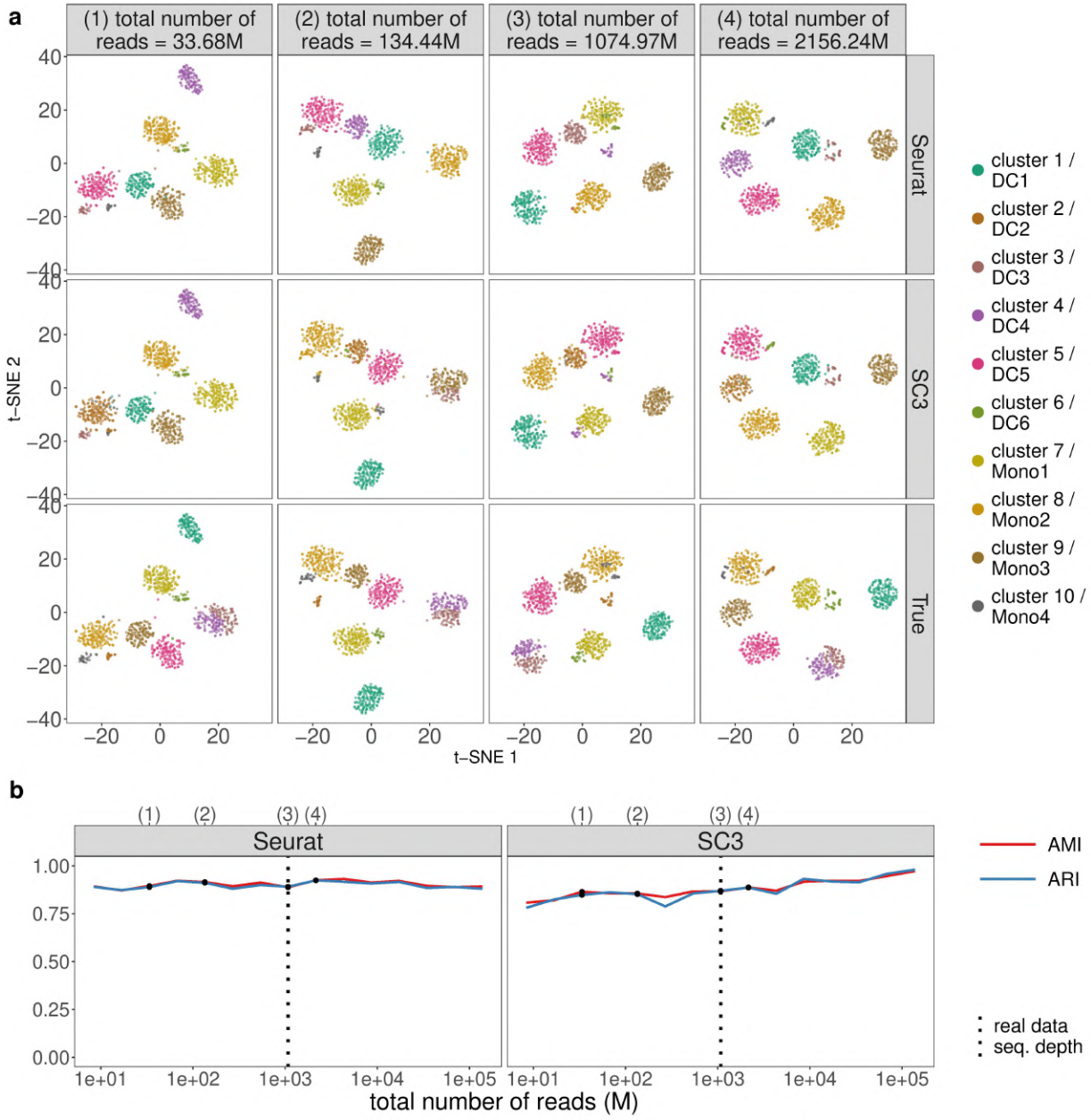

Figure 3.51: scDesign2 guides the choice of sequencing depth in cell clustering.

scDesign2 generates synthetic Smart-Seq2 data with fifteen sequencing depths. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. sequencing depth; left: Seurat; right: SC3. In (b), the results of the four sequencing depths in (a) are marked as dots and in the top, and the sequencing depth of the real dataset [122] is marked as vertical dashed lines.

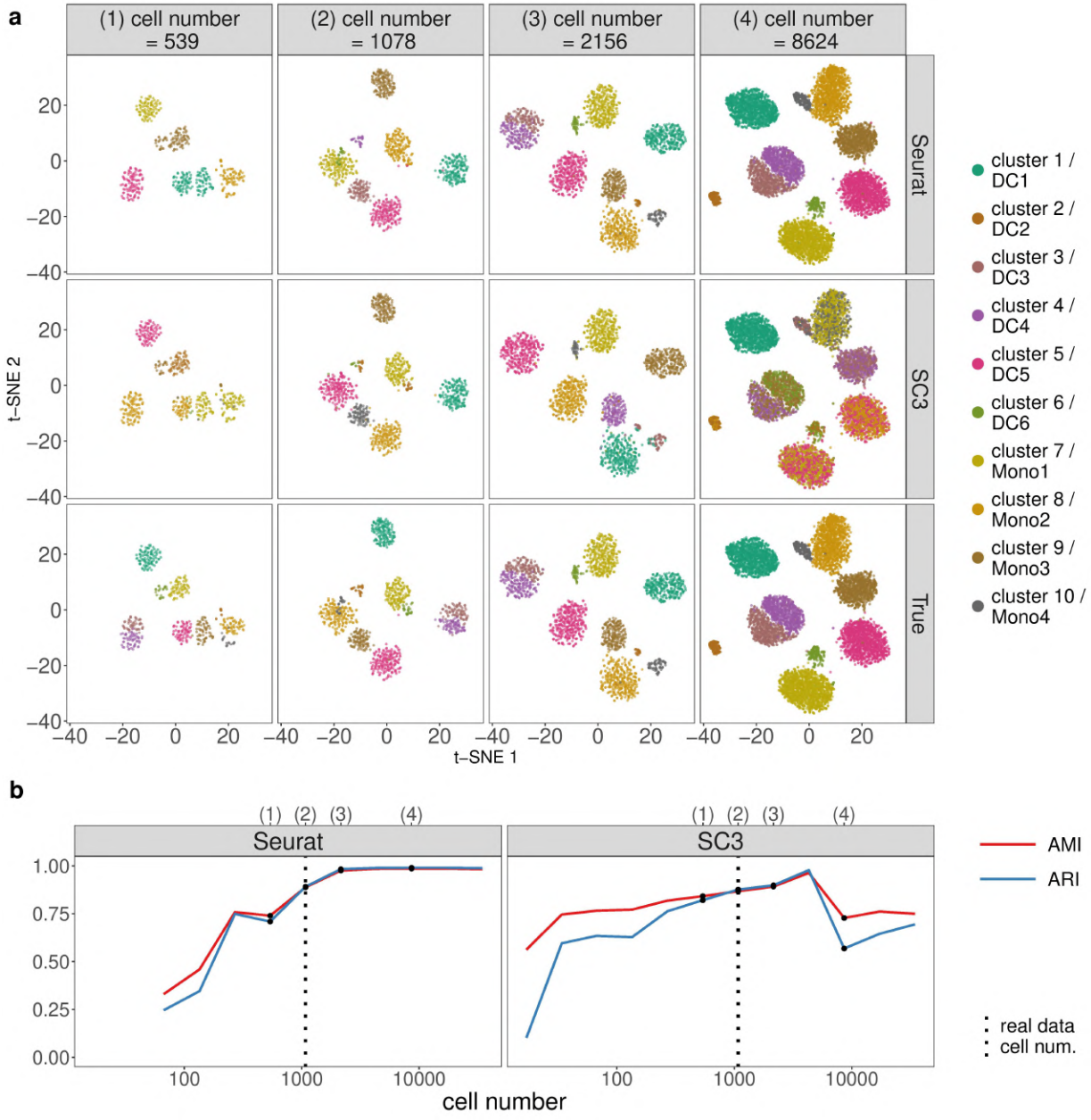

Figure 3.52: scDesign2 guides the choice of cell number in cell clustering, in the case where the total sequencing depth is kept as fixed.

scDesign2 generates synthetic Smart-Seq2 data with twelve cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [122] is marked as vertical dashed lines.

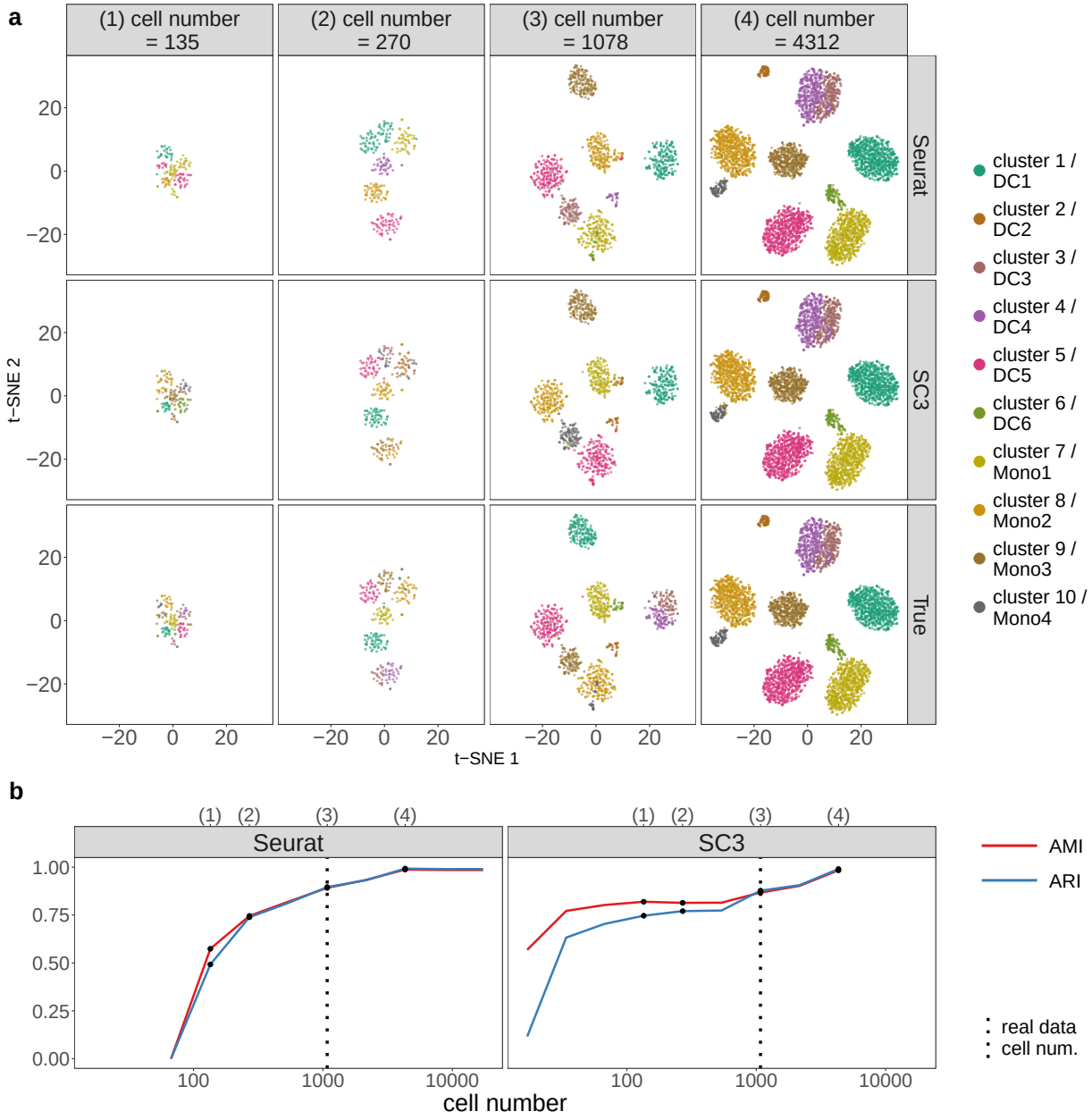

Figure 3.53: scDesign2 guides the choice of cell number in cell clustering, in the case where the average sequencing depth is kept as fixed.

scDesign2 generates synthetic Smart-Seq2 data with eleven cell numbers. Two cell clustering methods—Seurat and SC3—are applied to each synthetic dataset to partition cells into cell clusters. (a) t-SNE visualization of four synthetic datasets, where cells are labelled by Seurat clusters (top), SC3 clusters (middle), and annotated cell types (bottom). (b) Two clustering accuracy measures (AMI and ARI) vs. cell number; left: Seurat; right: SC3. In (b), the results of the four cell numbers in (a) are marked as dots and in the top, and the cell number of the real dataset [122] is marked as vertical dashed lines.

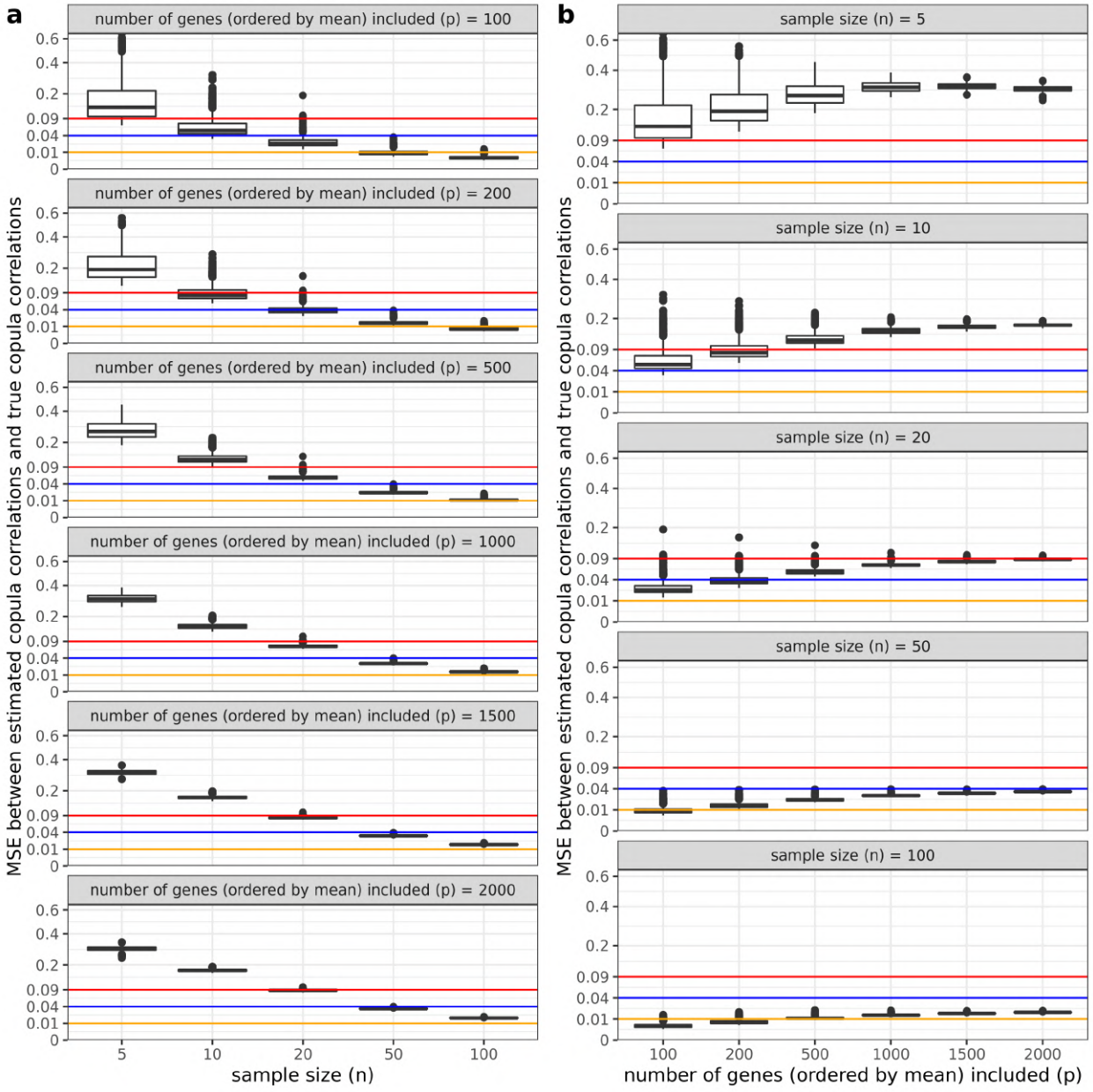

Figure 3.54: The effects of n (the sample size, i.e., the number of cells) and p (the number of top highly expressed genes) on the estimation of the copula correlation matrix in the context of 10x Genomics stem cell data.

The mean squared error (MSE) between the estimated copula correlations and the true copula correlations are calculated. For each sample size n, 1000 random samples are simulated from a known Gaussian copula model, which is fitted to the 10x Genomics stem cell data, and each sample is then used to estimate the copula correlation matrix. For computational efficiency, we estimate copula correlations using the formula $\rho_z = \sin(\frac{\pi}{2}\tau)$ by plugging in the sample tau values. (a) The relationship between MSE and n, for each p varying from 100 to 2000. (b) The relationship between MSE and p, for each n varying from 5 to 100. The vertical axes are on the square-root scale.

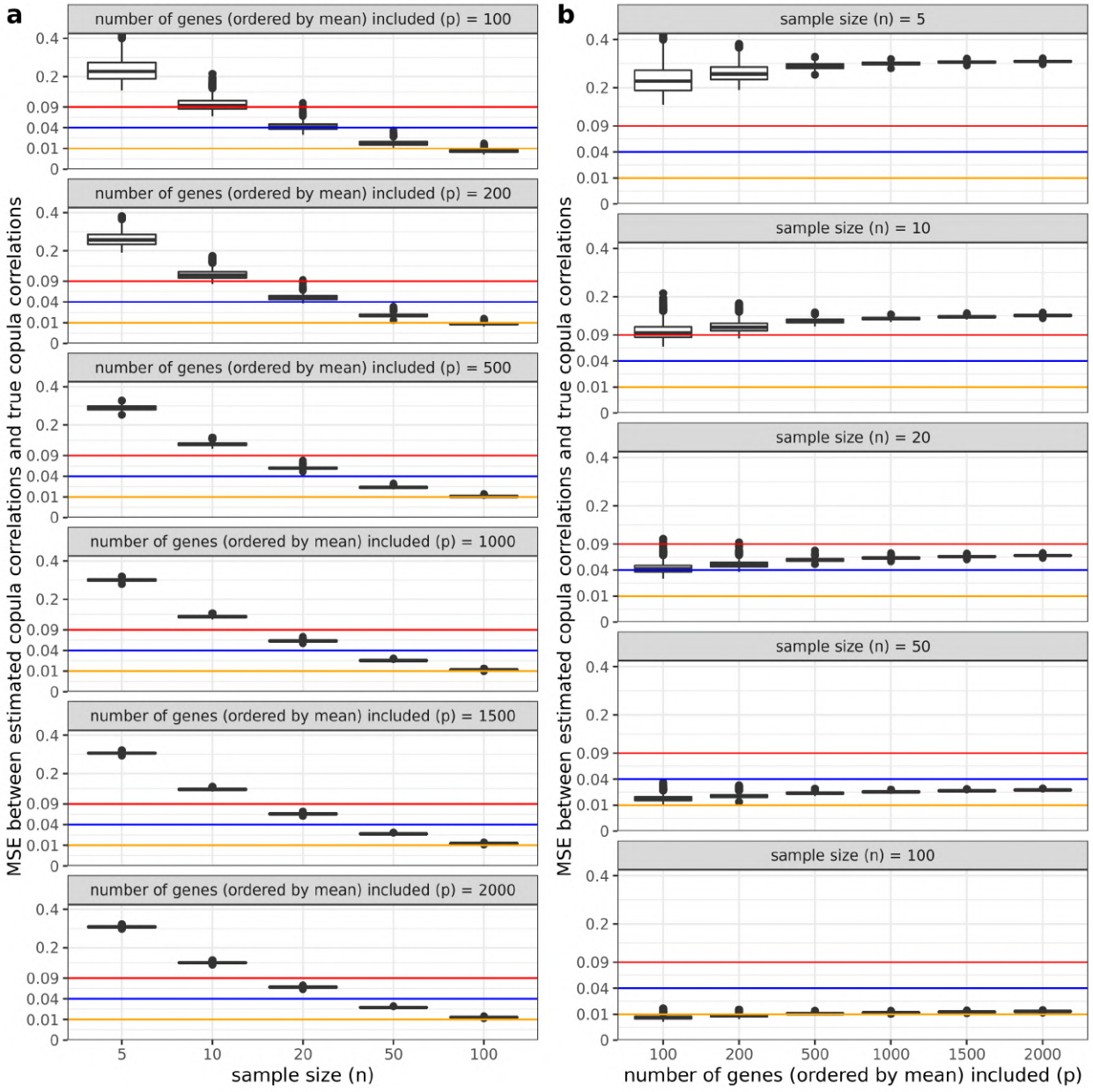

Figure 3.55: The effects of n (the sample size, i.e., the number of cells) and p (the number of top highly expressed genes) on the estimation of the copula correlation matrix in the context of Smart-Seq2 dendrocytes (subtype 1) data.

The mean squared error (MSE) between the estimated copula correlations and the true copula correlations are calculated. For each sample size n, 1000 random samples are simulated from a known Gaussian copula model, which is fitted to the Smart-Seq2 dendrocytes (subtype 1) data, and each sample is then used to estimate the copula correlation matrix. For computational efficiency, we estimate copula correlations using the formula $\rho_z = \sin\left(\frac{\pi}{2}\tau\right)$ by plugging in the sample tau values. (a) The relationship between MSE and n, for each p varying from 100 to 2000. (b) The relationship between MSE and p, for each n varying from 5 to 100. The vertical axes are on the square-root scale.

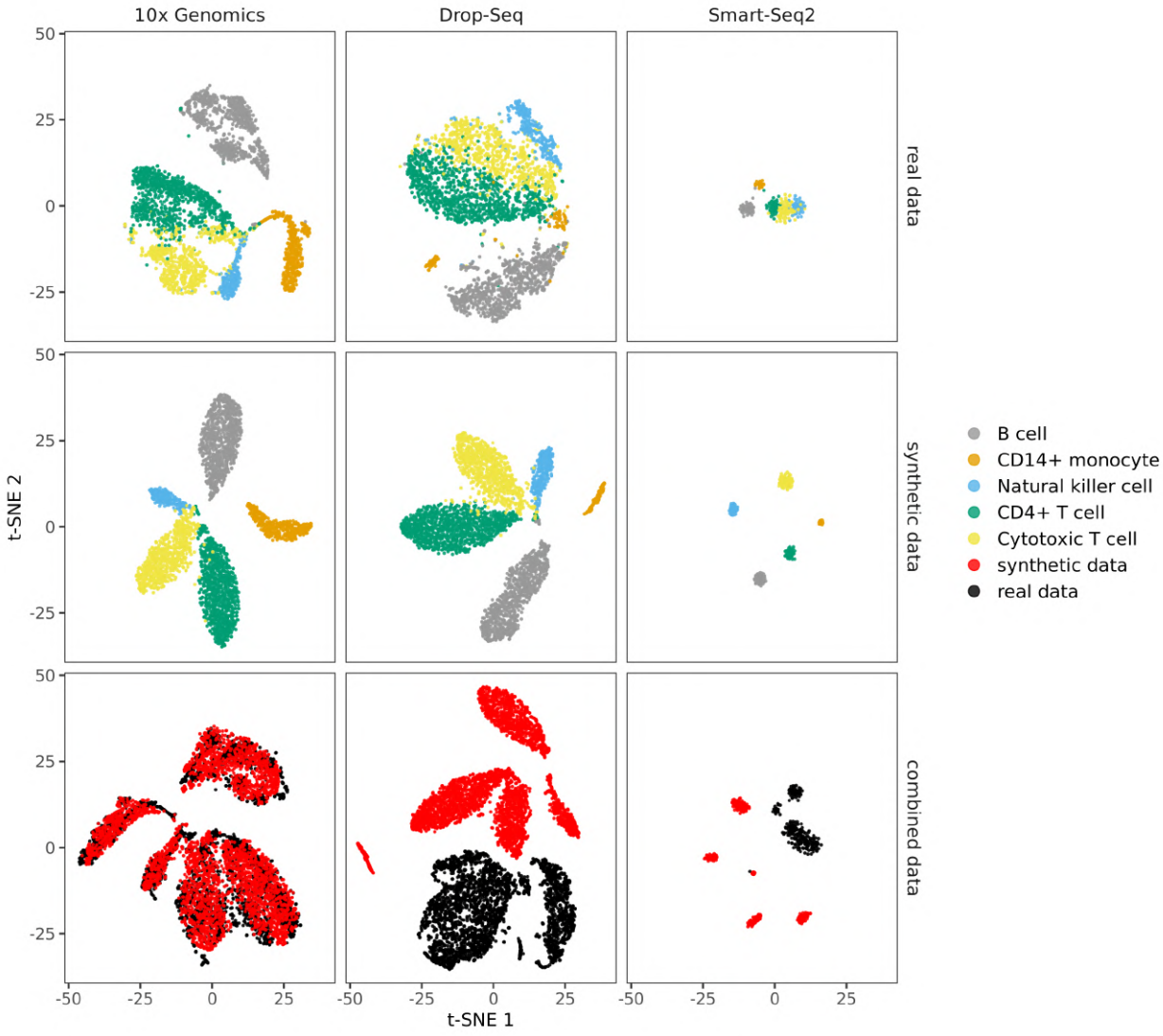

Figure 3.56: 2D t-SNE visualization of the results of a cross-platform simulation experiment.

We use a multi-protocol dataset of peripheral blood mononuclear cells (PBMCs) generated for benchmarking purposes [74]. We select data of five cell types measured by three protocols, 10x Genomics, Drop-Seq, and Smart-Seq2, and we train scDesign2 on the 10x Genomics data. Then we adjust the fitted scDesign2 model for the Drop-Seq and Smart-Seq2 protocols by rescaling the mean parameters in the fitted model (see Methods for details). After the adjustment, we use the model for each protocol to generate synthetic data. The 2D t-SNE visualization plot is generated for the real data, the synthetic data, and the combination of the real data and the synthetic data, for each of the three protocols. From the combination plot, we can see that the synthetic cells do not mix well with the real cells for the two cross-protocol scenarios; only for 10x Genomics, the same-protocol scenario, the synthetic cells mix well with the real cells. As a quantitative comparison, the median integration local inverse Simpson's Index (miLISI) for the three combination plots are 1.756, 1.000, and 1.000, from left to right, with the largest value indicating the best mixing for the same-protocol scenario, confirming our conclusion.

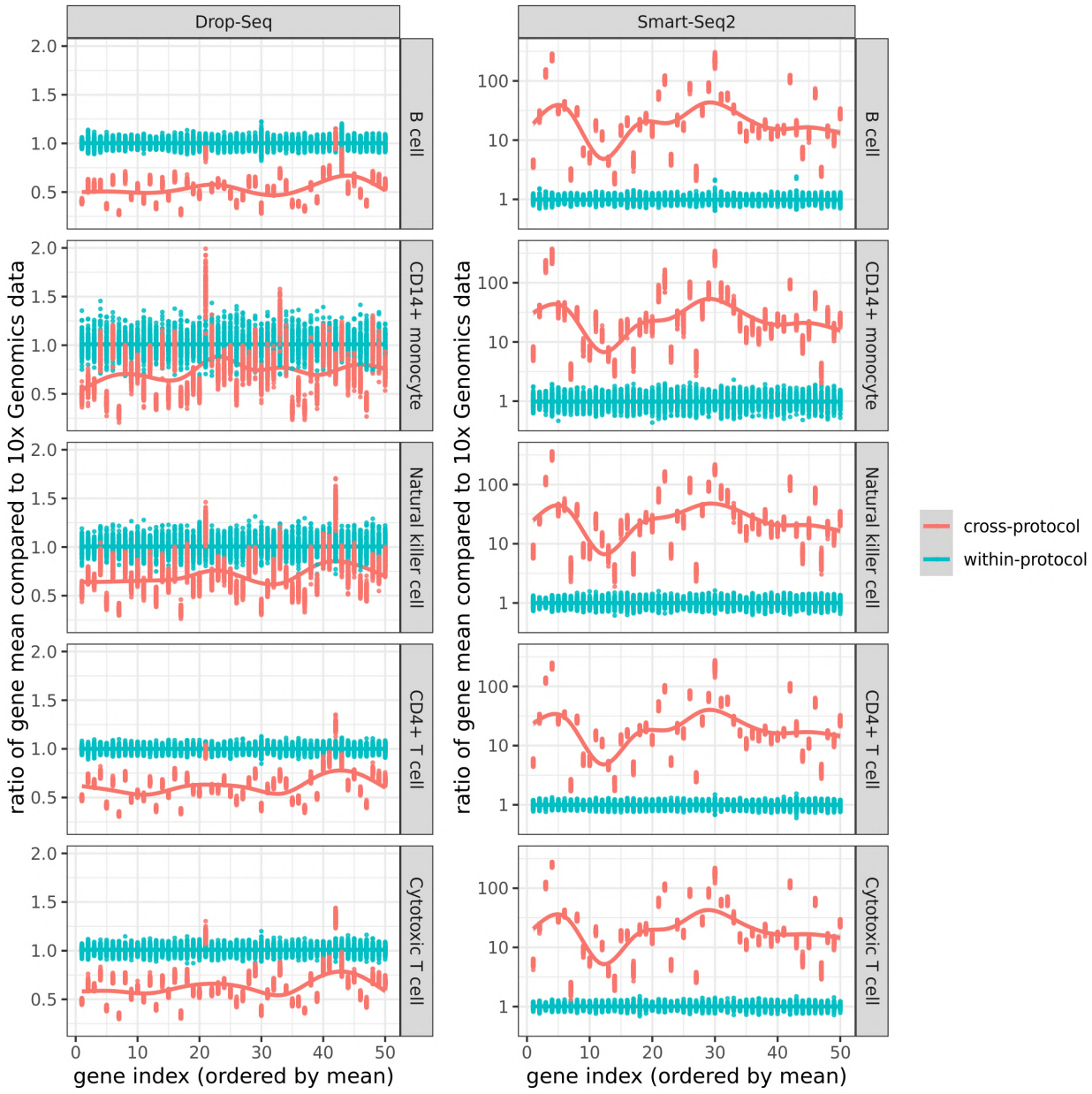

Figure 3.57: The cross-protocol and within-protocol ratios of genes' mean expression levels between a target protocol (Drop-Seq or Smart-Seq2) and the reference protocol (10x Genomics) in five cell types.

The results are shown for the top 50 highly expressed genes across the five cell types in the reference protocol, ordered by mean expression from high to low. In each cell type (row) and each target protocol (column), a gene has 100 cross-protocol ratios and 100 within-protocol ratios as a result of random partitioning. To illustrate the trends of the ratios, smoothed curves are added by the R function `geom_smooth()`. The calculation detail is as follows. For each cell type, suppose the target protocol has m cells, and the reference protocol has n cells. In each random partition, we first randomly select $\frac{\min(m,n)}{2}$ cells from the target protocol and the reference protocol each, and compute a gene's two means respectively. Second, for the remaining cells in the reference protocol, we randomly select $\frac{n}{2}$ cells and compute the same gene's mean as a reference. Third, we calculate the cross-protocol ratio and the within-protocol ratio by dividing the first two means by the reference mean. We repeat the above three steps for 100 times for every gene in each cell type and each target protocol. From the plot, we can see that, unlike the within-protocol ratios, which center around the constant value of 1, the cross-protocol ratios fluctuate between genes, and do not center around a constant value. This shows that there does not exist a single scaling factor to convert all genes' expression levels from one protocol to another.

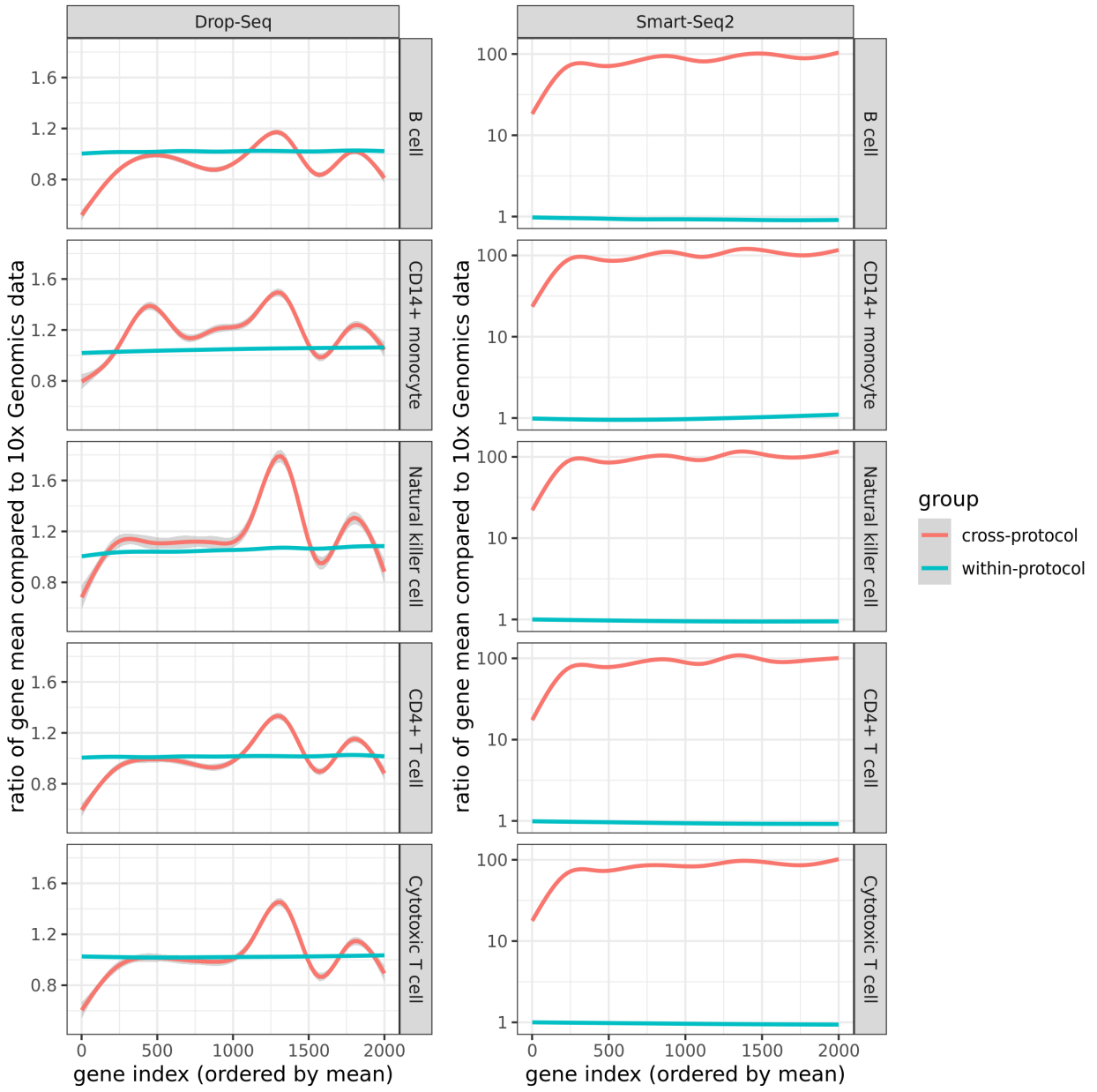

Figure 3.58: The cross-protocol and within-protocol ratios of genes' mean expression levels between a target protocol (Drop-Seq or Smart-Seq2) and the reference protocol (10x Genomics) in five cell types.

The results are computed for the top 2000 highly expressed genes in each cell type in the reference protocol, ordered by mean expression from high to low. In each cell type (row) and each target protocol (column), a gene has 100 cross-protocol ratios and 100 within-protocol ratios as a result of random partitioning. For clarity of illustration, only the trends of the ratios are shown, which are smoothed curves, added by the R function `geom_smooth()`. The calculation detail is as follows. For each cell type, suppose the target protocol has m cells, and the reference protocol has n cells. In each random partition, we first randomly select $\frac{\min(m,n)}{2}$ cells from the target protocol and the reference protocol each, and compute a gene's two means respectively. Second, for the remaining cells in the reference protocol, we randomly select $\frac{n}{2}$ cells and compute the same gene's mean as a reference. Third, we calculate the cross-protocol ratio and the within-protocol ratio by dividing the first two means by the reference mean. We repeat the above three steps for 100 times for every gene in each cell type and each target protocol. From the plot, we can see that, unlike the within-protocol ratios, which center around the constant value of 1, the cross-protocol ratios fluctuate between genes, and do not center around a constant value. This shows that there does not exist a single scaling factor to convert all genes' expression levels from one protocol to another.

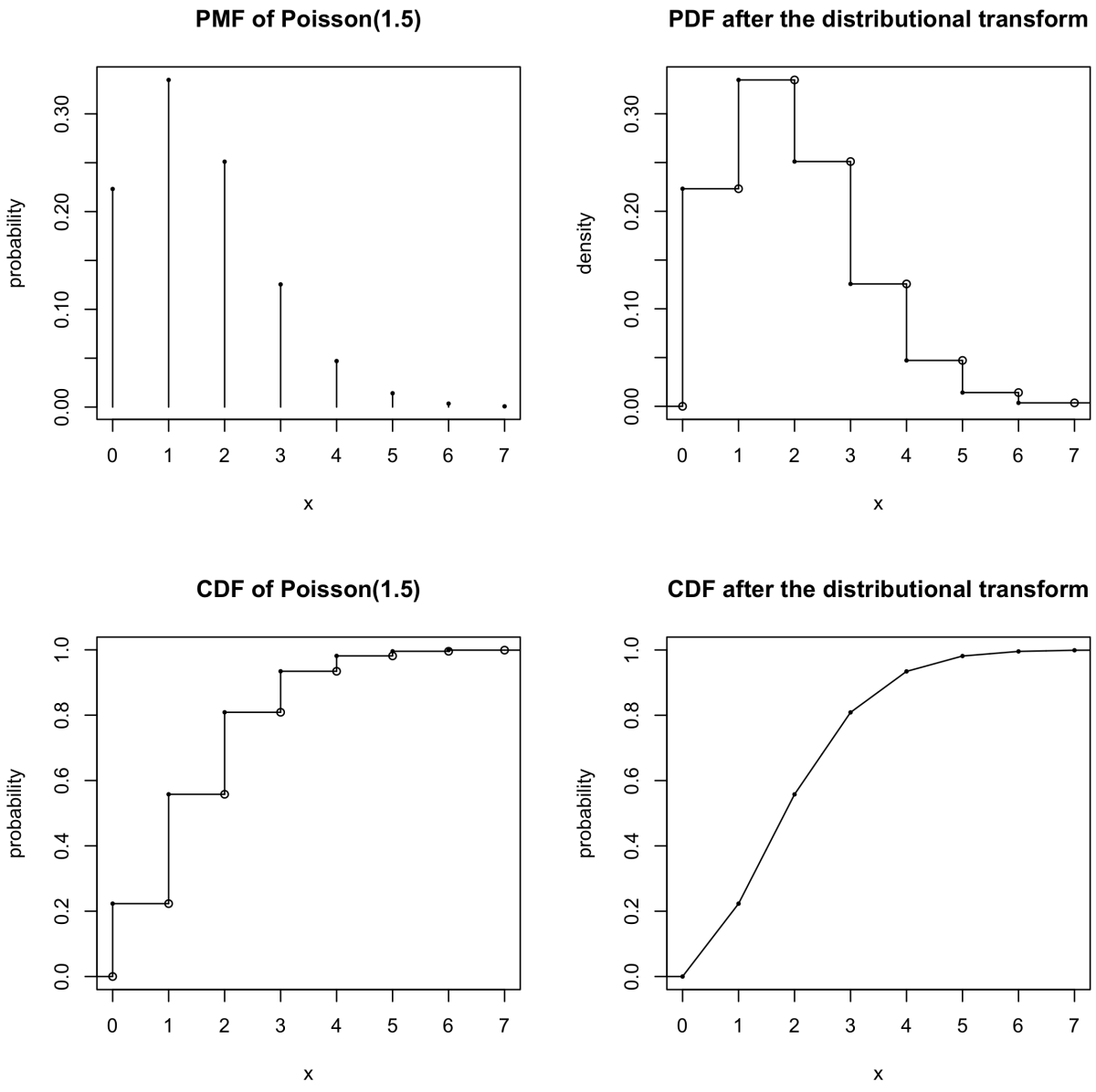

Figure 3.59: A toy example showing the effect of the distributional transform.

For a discrete random variable X, the distributional transform maps its probability mass at each value x, which has a non-zero probability mass, uniformly to the interval $[x, x+1)$, thus converting X to a continuous random variable. The top left and right panels show the probability mass function (PMF) before the transform and the probability density function (PDF) after the transform; the bottom left and right panels show the cumulative distribution functions (CDFs) before and after the transform.

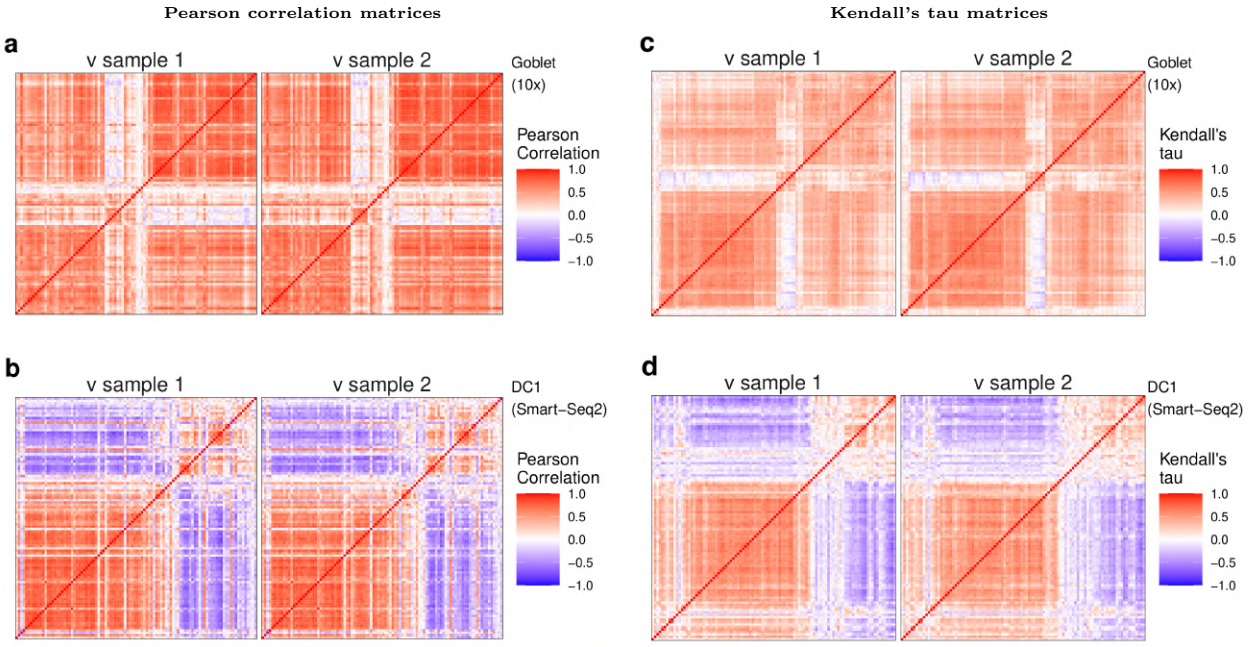

Figure 3.60: Heatmaps of gene correlation matrices estimated from synthetic data generated by scDesign2, with $\widehat{\mathbf{R}}$ estimated under two different random samples of v_{ij}^*.

(a)-(b) Pearson correlation matrices; (c)-(d) Kendall's tau matrices. In (a) and (c), the scDesign2 model is fitted with goblet cells measured by 10x Genomics [119]; In (b) and (d), the scDesign2 model is fitted with cells of dendrocytes subtype 1 (DC1) measured by Smart-Seq2 [122]. For each cell type, the Pearson correlation matrices and Kendall's tau matrices are shown for the 100 genes with the highest mean expression values in the test data; the rows and columns (i.e., genes) of all the matrices are ordered by the complete-linkage hierarchical clustering of genes (using Pearson correlation as the similarity in (a)-(b) and Kendall's tau in (c)-(d)) in the test data. We find that the effect of the sampling of v_{ij}^* on the estimated gene correlation matrices of the synthetic data is negligible, since the two matrices in each panel are very similar.

CHAPTER 4

scStructure: latent structure type selection for single-cell RNA-seq datasets

4.1 Introduction

In recent years, single-cell RNA-sequencing (scRNA-seq) has become an important technology in biology research [57, 58], with numerous protocols developed [14, 15, 76, 78] and new knowledge gained [9–13]. Compared to bulk RNA-seq, scRNA-seq has the advantage of profiling the transcriptome of individual cells, thus revealing cell-to-cell heterogeneity in a biological system of interest. In particular, two different types of computational tasks, clustering and trajectory inference, can be performed to illustrate two different types of cellular heterogeneity. With clustering, the cells will be partitioned into cell types, representing discrete cell states [25–27]. With trajectory inference, the cells will be connected as single-branch or multiple-branch trajectories, representing cell states that change on a continuum [28, 29, 157, 158]. These two different tasks will give different interpretations of the data. However, as of now, there is no principled way to tell which of these two interpretations is more reasonable for a given scRNA-seq dataset.

In this manuscript, we show two different approaches that aim to solve the cluster-type vs. trajectory-type latent structure selection problem for scRNA-seq datasets. The first approach is based on the eigenvalue properties of the covariance matrix of a scRNA-seq dataset, drawing inspiration from random matrix theory (RMT). In particular, we train classifiers based on real data of cluster type and trajectory type and then make predictions on new data. The second approach is based on comparing the similarity of real data and simulated data generated by assuming the cell latent structure as clusters or a trajectory.

While both approaches have limitations, we show that the second approach gives more promising results and has room for further improvements.

4.2 Eigenvalue-classification-based approach (Approach 1)

4.2.1 Existing applications of random matrix theory in single-cell RNA-seq data analysis

A random matrix is simply a matrix whose elements are random variables. When these random variables follow certain types of distributions, probabilistic conclusions can be made about the (asymptotic) distributions of the eigenvalues, spacing between eigenvalues and eigenvectors of the original matrix [159–161]. Random matrix theory (RMT) refers to the collection of such mathematical results. The application of RMT was mainly in the field of nuclear physics [159], which later expanded to other fields [162, 163]. Since the data of scRNA-seq can be represented by a gene-by-cell matrix, the application of RMT is possible, and a number of papers have used random matrix theory to analyze single-cell RNA-seq data.

In [164], the authors consider a simple model for single-cell lineage progression. The model describes a bifurcating process of cell differentiation, where the expression profile of p genes is approximated as a binary vector, with entries taking values of "ON" or "OFF". The bifurcating process takes a total of b rounds, within each of which an m-step process of a randomly chosen gene switching its state (from ON to OFF or vice versa) occurs, before the whole vector duplicates itself. Under this model, for the terminal cells, the eigenvalue distribution of the covariance matrix of its gene expression matrix "has a power law structure, $\lambda \sim r^{-\log(2\alpha)/\log(2)}$, for each eigenvalue λ, in which $\alpha = \exp(-4m/p)$ and $r > 1$ is the eigenvalue rank" [164, 165]. The authors then demonstrate using simulated data and real data, that only in datasets of developmental or differentiation processes, the power law structure would occur.

Although this paper has shown some interesting results, it has some limitations. For

example, in most real scRNA-seq data matrices, the gene expression entries are not binary values, but rather numerical values. Moreover, the power law structure only applies to terminal cells. Therefore, the approach developed in [164] could not be used in datasets where researchers perform trajectory and pseudotime inference, as the cells on trajectories would naturally represent a continuous change of cell states, not just terminal cells.

In [166] and [167], the authors have developed a data denoising method and a clusterability measure respectively, based on a set of different and more well-known results from the random matrix theory. In particular, they consider a type of random matrix whose elements are independent and identically distributed (i.i.d.) with zero mean and finite variance. Then, when the number of rows and the number of columns are large, the eigenvalue distribution of its covariance matrix can be approximated by the Marchenko–Pastur (MP) distribution [168]. Moreover, the norm of each of its eigenvectors is approximately equally distributed across the entries, a property called "delocalization" [166].

To apply these interesting results to analyze scRNA-seq data, we can assume that the gene expression matrix can be decomposed into the sum of a noise matrix and a low-rank signal matrix (Fig. 4.1). Then we can analyze the eigenvalues and eigenvectors of its covariance matrix. Finally, deviation from the MP distribution, or equivalently the presence of large eigenvalues, indicates the presence of biological signals, e.g., the presence of cell clusters. Alternatively, the biological signal can be detected from the delocalization to localization transition of eigenvectors.

In [166], the authors further consider another source of signal that comes from the typical sparsity present in a scRNA-seq data matrix. Therefore, their data-denoising approach would decompose the information from a scRNA-seq data matrix into three sources — noise, sparsity, and true biological signal. After the data is denoised, only the true biological signal will be left and analyzed. In [167], the authors construct a numerical measure ϕ_{clust} that quantifies how well the observed data matrix aligns with the underlying signal matrix. A higher ϕ_{clust} value indicates a stronger biological signal, or as the authors of [167] conclude, a more confident conclusion on the presence of clusters.

However, this immediately leads to a limitation of their study, because the presence of a low-rank signal matrix could indicate not only the presence of clusters, but also the presence of trajectories. In Fig. 4.1, we construct two simulated data examples to demonstrate this, where the top row represents the case of a cluster-type dataset, while the bottom a trajectory-type dataset. In both cases, the measured gene expression matrix could be decomposed into the sum of a low-rank signal matrix and a noise matrix. Further, if we analyze the eigenvalue distributions, both follow the pattern where the small eigenvalues can be approximated by the MP distribution, while a few large eigenvalues indicate the presence of biological signals. However, in the top row, the cluster structure appears because cells form three groups, each having its own set of marker genes. Or equivalently, it appears because genes' expression changes discretely across cells. In the bottom row, the trajectory structure appears because genes' expression changes continuously across cells.

The above artificially constructed example is simple, but it motivates us to explore whether there are distinguishable differences in the eigenvalue distributions of the gene expression covariance matrix for cluster-type and trajectory-type datasets. Due to the popularity of the scRNA-seq field, numerous datasets of cluster type and trajectory type have been generated. Therefore, they serve as valuable empirical resources and machine learning approaches can be naturally used to study this problem.

4.2.2 Approach 1 results

For our proposed Approach 1, we aim to distinguish cluster-type vs. trajectory-type scRNA-seq datasets using the eigenvalue properties from the covariance matrices. In particular, we first collect a number of trustworthy cluster-type datasets and trajectory-type datasets (summarized in Table 4.1 and Table 4.2). Then, for each collected dataset, we compute the eigenvalues of its gene covariance matrix and extract different types of features from the eigenvalues. Finally, we train a number of SVM classification models with linear kernels based on the extracted features. We evaluate the model performances using the average classification accuracy (mean ACC) of a 10-fold cross-validation.

Dataset Group	# Single Datasets	Description	Ref.
DuoClustering	4	Datasets selected from a clustering methods benchmark paper.	[36]
Hemberg	13	Datasets selected from a clustering parameters benchmark paper.	[37]
TabulaSapiens	56	From the Tabula Sapiens consortium, which profiles the single cell transcriptome of multiple tissues in individual humans. Cells from a single tissue type of one individual form a single dataset.	[169]
ZebrafishClust1	37	Datasets from a study of zebrafish heart regeneration. Cells from one zebrafish sample form a single dataset. All cell types are selected based on Figure 2 of the original manuscript [170].	[170]
Hppu	6	Datasets from a study of adult human kidney. Cells from one human sample form a single dataset.	[171]
Cbec	6	Datasets from a study of breast epithelial cells. Cells from one human sample form a single dataset.	[172]
Hkch	5	Datasets from a study of normal adult human prostate and prostatic urethra. Cells from one human sample form a single dataset.	[173]

Table 4.1: A summary of the cluster-type datasets that were used for training the classifier. The total number of single datasets is 127.

Dataset Group	# Single Datasets	Description	Ref.
BenchRealGold	27	Datasets selected from a trajectory inference methods benchmark paper. These are real datasets whose reference trajectories were "not extracted from the expression data itself, such as via cellular sorting or cell mixing" [38].	[38]
BenchRealSilver	83	Datasets selected from a trajectory inference methods benchmark paper. These are real datasets whose reference trajectories were "extracted from the expression data itself" [38].	[38]
ZebrafishTraj1	37	Datasets from a study of zebrafish heart regeneration. Cells from one zebrafish sample form a single dataset. Cell types are selected based on Figure 4 of the original manuscript [170].	[170]
ZebrafishTraj2	37	Datasets from a study of zebrafish heart regeneration. Cells from one zebrafish sample form a single dataset. Cell types are selected based on Figure 5 of the original manuscript [170].	[170]

Table 4.2: A summary of the trajectory-type datasets that were used for training the classifier. The total number of single datasets is 184.

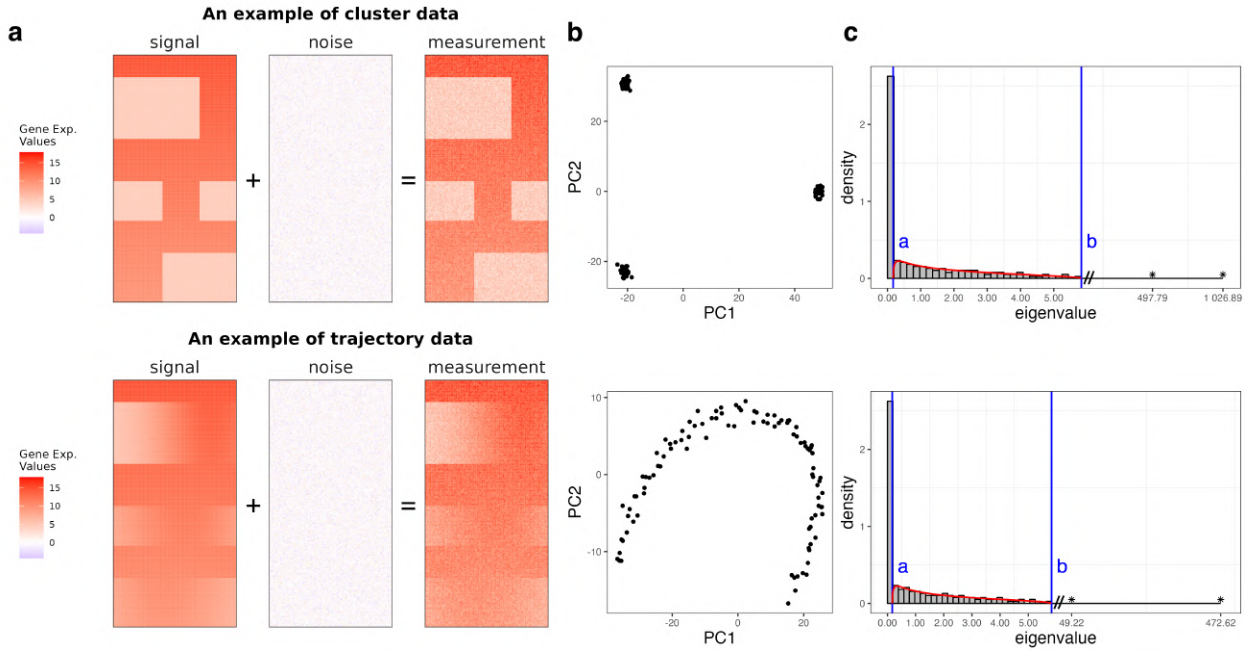

Figure 4.1: Two simulated dataset examples demonstrating how the signal-noise decomposition could occur for both cluster-type data and trajectory-type data.

(a) The measured gene expression matrix could be decomposed as the sum of a low-rank signal matrix and a noise matrix. (b) The first two PCs of the measured gene expression matrix. (c) The eigenvalue distribution of the covariance matrix of the measured gene expression matrix. The blue lines represent the lower and upper bounds (values a and b) of the MP distribution. The red line represents the MP distribution density. The large eigenvalues outside the MP distribution are marked separately.

4.2.2.1 Eigenvalue universality theorems

Here, we briefly summarize the key eigenvalue universality results that are used in this study. The following two theorems are directly quoted from the supplementary material of [167].

Theorem 1. *(Marchenko-Pastur) Let Y be a $M \times N$ matrix with entries that are independent identically distributed (i.i.d.), mean 0 and variance $\nu^2 < \infty$. The corresponding Wishart matrix is defined as $W = \frac{1}{M} Y^T Y$. For $N \to \infty$, $M \to \infty$ and $0 < c < 1$, where c is defined as $\frac{M}{N}$, the distribution of the eigenvalues λ of W is given by*

$$\mu(\lambda) = \frac{\sqrt{(b-\lambda)(\lambda-a)}}{2\pi c \lambda \nu^2} d\lambda \text{ if } a \leqslant \lambda \leqslant b.$$

For $c > 1$ the distribution has an additional number of 0 eigenvalues:

$$\mu(\lambda) = \frac{\sqrt{(b-\lambda)(\lambda-a)}}{2\pi c \lambda \nu^2} \mathbb{1}_{[a,b]} + (1 - \frac{1}{c}\delta_0(\lambda))$$

with

$$a, b = \nu^2[1 \pm \sqrt{c}]^2.$$

$\delta_0(\lambda)$ *is the Dirac delta function, which is* 1 *if* $\lambda = 0$ *and* 0 *otherwise.*

Theorem 2. *(Tracy-Widom) For empirical correlation matrices of size* $N \times N$ *of i.i.d. random variables with a finite fourth moment, the distance between the upper edge of the spectrum of the MP distribution* b *and the largest eigenvalue* $\lambda_{\max}$ *converges towards the Tracy-Widom distribution*

$$\text{Prob}(\lambda_{\max} \leqslant b + \gamma N^{-2/3} u) = F_1(u),$$

where γ *in this case is given by* $\gamma = \sqrt{c} b^{2/3}$ *and* $F_1(u)$ *is the TW distribution. We denote* $b + \gamma N^{-2/3} u$ *as* ulim_TW*.*

4.2.2.2 Data preprocessing

In order to apply Theorem 1 and Theorem 2 to our study, we need to preprocess the input gene expression matrix. We assume it is organized such that rows represent genes and columns cells. Then we perform the following procedure:

1. For each cell, we perform library normalization to the original count values rescaled by the median library size of all cells.
2. Perform $\log(1 + \cdot)$ transform to the matrix values.
3. Retain as many as 2000 genes with the highest mean expression.
4. Perform gene-wise standardization.
5. Perform cell-wise standardization.

The gene filtering step simplifies the analysis to focus on more informative genes. The gene-wise and cell-wise standardization steps are performed to approximate the equal vari-

ance condition for the comparison of the empirical eigenvalue distribution and the theoretical MP distribution.

4.2.2.3 Features based on the eigenvalue distribution directly

After the preprocessing steps, we denote the resulting gene expression matrix as X with N rows of genes and M columns of cells. To apply Theorem 1 and Theorem 2, we can let $Y = X^T$ and compute the key quantities of a, b, $\mu(\lambda)$, and ulim_TW. We also compute the eigenvalues of X. Based on these calculations, we can construct features for the SVM classification models.

The first set of features is constructed based on the comparison of the empirical eigenvalue distribution and the theoretical MP distribution. We first consider the distribution of small eigenvalues. In particular, for eigenvalues in a certain range (e.g., $(a, \text{ulim\_TW})$), we create a histogram with 30 bins and compute the difference between $\mu(\lambda)$ and the empirical eigenvalue distribution at the center of each of the 30 bins. In this way, we obtain a feature vector of length 30, which we denote as curve_diff. Under these input features, the mean ACC's of the trained SVM models are summarized in Table 4.3. As can be seen, under different choices of the lower and upper limits of the range of eigenvalues, the mean ACC fluctuates between 0.843 to 0.904.

Since the mean ACC is the highest when eigenvalues are in the range $(a, \text{ulim\_TW})$, we compute a single-value predictor in the case. In particular, we compute the mean squared summary of curve_diff (mse). As can be seen, using this single predictor, we can achieve a mean ACC as high as 0.859.

Features	ev_llim	ev_ulim	mean ACC
curve_diff	0	2.5 × ulim_TW	0.849
	0	ulim_TW	0.868
	a	2.5 × ulim_TW	0.843
	a	ulim_TW	0.904
mse	a	ulim_TW	0.859

Table 4.3: The classification performances when using the difference between the empirical eigenvalue distribution and the theoretical MP distribution as input features, focusing on the small eigenvalues.

Next, we consider the distribution of large eigenvalues, i.e., eigenvalues that exceed ulim_TW. We construct three different types of features: (1) n_above: the number of large eigenvalues, (2) dist_ev_above: the distribution of large eigenvalues constructed from a 30-bin histogram, and (3) curve_diff + dist_ev_above: combining the curve_diff features constructed for eigenvalues in the range $(a, \text{ulim\_TW})$ and dist_ev_above. The results are summarized in Table 4.4. We can see that n_above as a single predictor has good performance, while combining curve_diff and dist_ev_above gives worse performance than only using curve_diff. The latter could be due to model overfitting from the relatively high combined feature dimensions of $30 + 30 = 60$.

Features	mean ACC
n_above	0.782
dist_ev_above	0.727
curve_diff + dist_ev_above	0.881

Table 4.4: The classification performances when considering the distribution of large eigenvalues.

4.2.2.4 Features based on other summary statistics of the eigenvalue distribution

In this part, we construct other summary statistics of the eigenvalue distribution. In particular, we compute the entropy (ent), standard deviation (sd), skewness (sk), and kurtosis (kur). We compute these statistics for all the eigenvalues as well as only the large eigenvalues. The results are summarized in Table 4.5. We can see that the entropy of all the eigenvalues, as well as the skewness and kurtosis for the large eigenvalues are the better ones.

Finally, we select the single predictors mse, n_above, ent_ev_all, sk_ev_large, and kur_ev_large, since they have good marginal classification performances, and train an SVM model combining all five of them. This classification model achieves a mean ACC of 0.875, which is higher than each of the single predictors by itself and is comparable to the performance using the higher-dimensional features of curve_diff.

Features	ev_range	mean ACC
ent	all	0.800
	large	0.595
sd	all	0.592
	large	0.592
sk	all	0.684
	large	0.762
kur	all	0.665
	large	0.765
mse + n_above + ent_ev_all + sk_ev_large + kur_ev_large	—	0.875

Table 4.5: The classification performances using single-value predictors as input features. The last row shows the performance combining selected good single-value predictors.

4.2.3 Approach 1 problems

In the previous section, we can see that using certain features of the eigenvalue distribution as input, we can build classification models and achieve relatively good cross-validation performance (mean ACC > 0.85) on classifying whether an input dataset is cluster-type or trajectory-type. However, it turns out that there are some crucial issues with these constructed models, which can be seen by testing their performances on the following simulated data examples.

In Fig. 4.2, we construct a set of simulated datasets whose structures change from trajectory-type to cluster-type. These simulated datasets are generated using the scRNA-seq data simulator scDesign3 [174] as follows: We first obtain a real pancreas dataset whose selected cell types form a trajectory [63, 175]. Then we use the computational tool Slingshot [28] to infer pseudotime values for the real cells, which are further normalized into the interval [0, 1]. We then apply scDesign3 to fit one multivariate probabilistic model for all the cells using the normalized pseudotime values as the cell covariates. Finally, we generate the simulated datasets using the scDesign3 fitted model with cell pseudotime values sampled from beta distributions with different parameters, where we keep $\alpha = \beta$ and gradually decrease their value from 1 to 0.002. This will make the pseudotime distribution change from uniform to being more concentrated on two modes and thus make the simulated cells change from forming a trajectory to forming two clusters.

For the simulated datasets in Fig. 4.2, a well-functioned classification model should output cluster-type probabilities that gradually increase from close to 0 to close to 1. However, as shown in Table 4.6, when we take some of the good-performing classification models from Tables 4.3 - 4.5, all of them outputs consistently high or low probabilities, not reflecting the change in structure that is present in Fig. 4.2. Table 4.6 also shows that if we apply the selected classification models to the two toy examples in Fig. 4.1, they will even output identical or almost identical cluster-type probabilities. This is in contrast to the obvious patterns of cluster-type vs. trajectory-type data in Fig. 4.1.

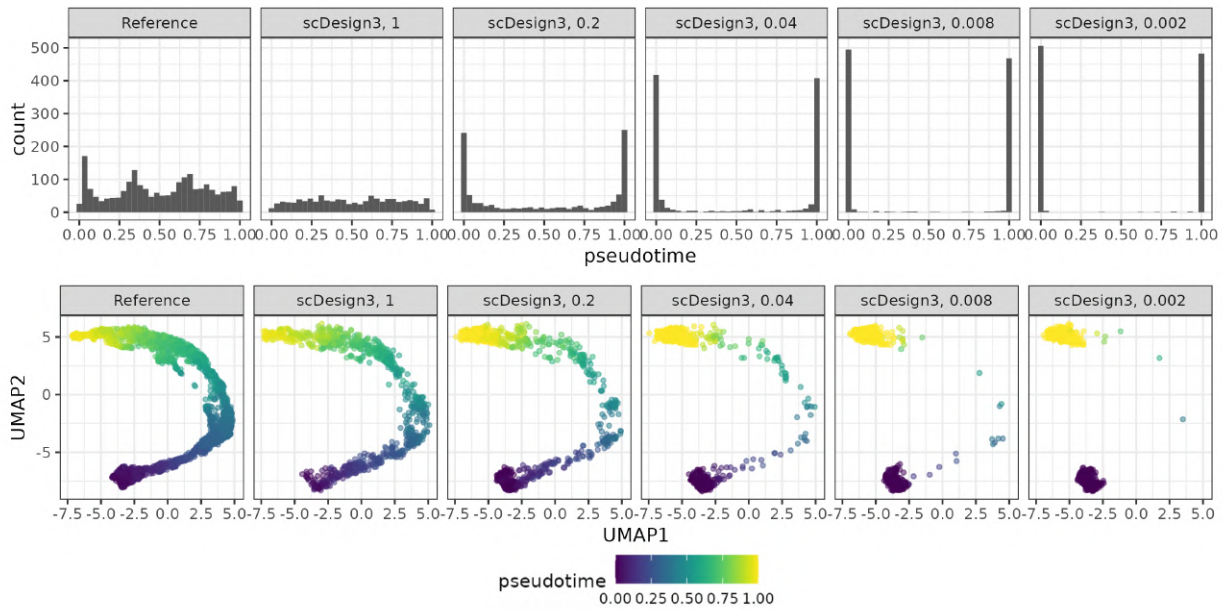

Figure 4.2: A real trajectory-type dataset (Reference) and a set of simulated datasets whose structure changes from trajectory-type to cluster-type.

The simulated datasets are generated using the scRNA-seq data simulator scDesign3. (a) The pseudotime values inferred for the Reference datasets and sampled for the simulated datasets. (b) The UMAP visualization plots of the Reference dataset and the simulated datasets.

4.2.3.1 Testing classifiers with permuted dataset partitions

To investigate the reason why the trained classifiers fail on the simulated data examples in Fig. 4.1 and Fig.4.2, we obtain the mean ACC for some different partitions of the datasets. In particular, we construct two sets of permuted datasets (Permuted1 and Permuted2) and 1 set of mixed datasets (Mixed). For the permuted datasets, we select three groups of datasets

Data \ Features	curve_diff	curve_diff + dist_ev_above	mse	n_above	ent_all	mse + n_above + ent_ev_all + sk_ev_large + ku_ev_large
Reference	1.000	1.000	1.000	0.062	0.059	1.000
scDesign3(1)	1.000	1.000	1.000	0.053	0.045	1.000
scDesign3(0.2)	1.000	1.000	1.000	0.053	0.144	1.000
scDesign3(0.04)	1.000	1.000	1.000	0.045	0.170	1.000
scDesign3(0.008)	1.000	1.000	1.000	0.039	0.212	1.000
scDesign3(0.002)	1.000	1.000	1.000	0.039	0.200	1.000
toy_ex_cluster	0.055	0.085	0.16	0.033	0.741	0.239
toy_ex_trajectory	0.116	0.087	0.16	0.033	0.741	0.240

Table 4.6: SVM model prediction probabilities that the input dataset is cluster-type, under different input features.

from the original seven cluster-type dataset groups and two groups from the original four trajectory-type dataset groups to form a pseudo-class 1, leaving the rest of the datasets to form the other pseudo-class 2. For the mixed datasets, each of the two pseudo-classes contains half of all the cluster-type datasets and half of the trajectory-type datasets. We then train SVM models using the top-performing features from Tables 4.3 - 4.5 and obtain the average 10-fold cross-validation accuracies. The results are summarized in Table 4.7.

We can see that although the mean ACC's of the permuted partitions are not as high as the original partitions, they are also not as low as the mixed partitions (close to 0.5). The gain in mean ACC between the two permuted dataset groups and the mixed dataset groups suggests the presence of intrinsic "batch effects" due to the grouping of the datasets. This "batch effect" will inflate the true classification accuracies. Therefore, the trained classifiers may not have reliable prediction results on new datasets.

	mean ACC			
Feature(s)	Original	Permuted1	Permuted2	Mixed
curve_diff	0.904	0.629	0.682	0.565
curve_diff + dist_ev_above	0.881	0.677	0.675	0.502
mse	0.859	0.649	0.669	0.476
n_above	0.782	0.552	0.604	0.469
ent_all	0.800	0.668	0.678	0.511
mse + n_above + ent_ev_all + skewness_ev_large + kurtosis_ev_large	0.875	0.652	0.694	0.544

Table 4.7: The comparison of classification performances for permuted and mixed datasets vs. datasets from the original partition.

There are some other issues with Approach 1. For example, even if we take the classifier with the highest possible mean ACC of 0.904, and treat this value as is, still about 10% of the time, the classifier will not work. And it is difficult to tell when that will happen.

Due to these limitations of Approach 1, we will try another approach to distinguish cluster-type and trajectory-type datasets.

4.3 Data-simulation-based approach (Approach 2)

4.3.1 Methods

Fig. 4.3 shows the diagram of the new approach, which is based on synthetic data simulation. To be more specific, for an input query dataset for which we would like to distinguish whether its underlying structure is cluster-type or trajectory-type, we split half of it as a training dataset and the other half as a test dataset. Next, the training dataset goes through two branches. On one branch, we apply Seurat clustering [25, 26] and obtain cell cluster labels. On the other branch, we apply Slingshot trajectory inference [28] and obtain cell pseudotime and branch values. Then, given the same training dataset and different fitted cell covariates (cluster labels vs. pseudotime and branch values), we fit two different probabilistic generative models and simulate two different datasets of different latent structures, using the all-in-one single-cell data simulation tool scDesign3 [174]. Finally, we compare each of the two simulated datasets with the test dataset. The final output of the selected underlying structure of the input query dataset will be from the simulated dataset that is more similar to the test dataset.

One thing to note about this approach is that for the trajectory inference branch, scDesign3 does not directly use the fitted cell pseudotime values after obtaining them from Slingshot trajectory inference. Rather, we apply kernel density estimation to the fitted pseudotime values and then sample a new set of pseudotime values based on the smoothed pseudotime density. This is because as cell covariates, pseudotime values intrinsically carry more information than cluster labels. Without the smoothing step, the simulated data based on the

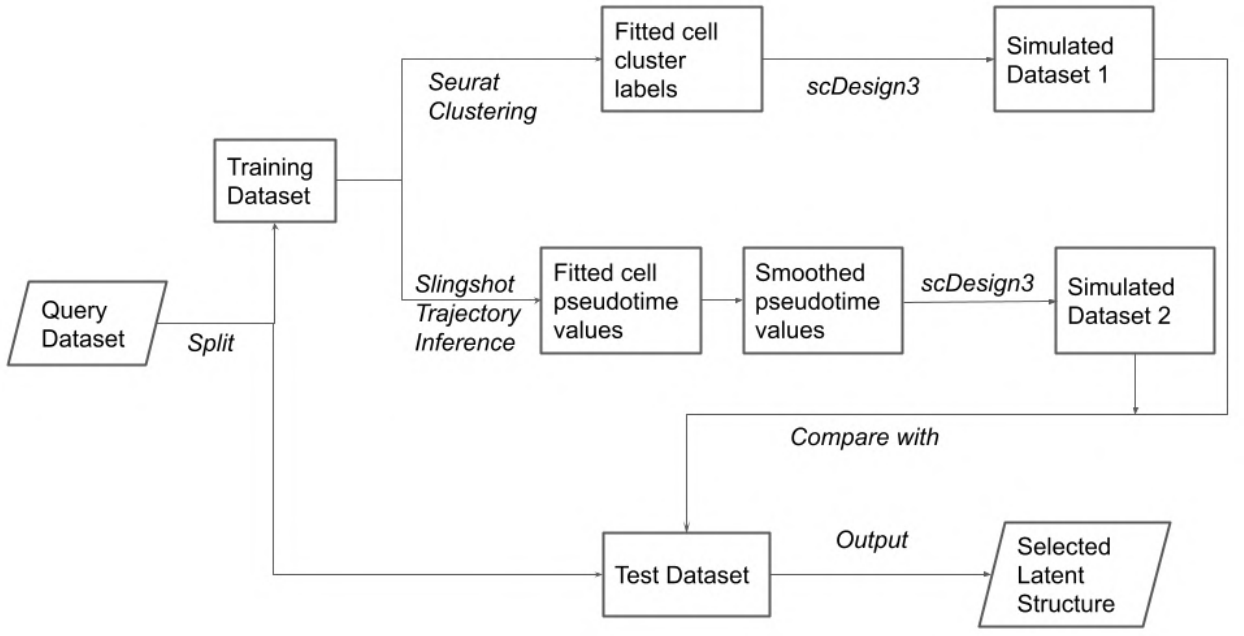

Figure 4.3: The diagram of the data-simulation-based approach.

more complicated cell covariates will always be more similar to the test data compared to the alternative.

Finally, in order to compare the two simulated datasets to the test dataset, we need a measure of similarity. For the results in this section, we tried three different summary statistics based on the Local Inverse Simpson's Index (LISI) [123] of cells in the 2D visualization plots (PCA and UMAP [21]) where we project the simulated cells to the space of the test cells. For a given cell, its LISI value varies from 1 to 2, with a higher value indicating a better mixing of cells from two categories (simulated vs. test) in its neighborhood. Therefore, meaningful summary statistics of this value can reflect the overall quality of the mixing of simulated data and test data and thus their similarity.

The three statistics we choose are (1) qLISI difference: the difference between the lower 10% quantiles of the LISI values between the cluster-simulated dataset and the trajectory-simulated dataset, (2) ECDF area difference: the signed area difference of the two LISI empirical cumulative distribution functions (ECDF) between the cluster-simulated dataset and the trajectory-simulated dataset, and (3) AUROC: the area under the ROC curve when choos-

ing different LISI thresholds to classify cluster-simulated dataset and trajectory-simulated dataset. The benefit of qLISI is its ease of computing, while the benefit of the latter two is that they make use of all the LISI values.

4.3.2 Approach 2 results

We first test the performance of Approach 2 on a typical trajectory-type dataset and another typical cluster-type dataset. The trajectory-type dataset is the single trajectory real/reference dataset from Fig. 4.2 [175] and the cluster-type dataset is constructed with four selected cell types from the data generated in [3]. As shown in Fig. 4.4, by comparing the mixing patterns of the simulated data and the test data in the 2D test data PC space, the correct underlying structure can be selected. Quantitatively, this can be seen from the higher qLISI values.

Next, we test the performance of this approach on the reference and the simulated datasets in Fig. 4.2. The results are summarized in Fig. 4.5. We first compute the qLISI values and use their difference between the cluster-simulated dataset and the trajectory-simulated dataset to select the underlying latent structure. As shown in Fig. 4.5a, this difference works correctly for the reference dataset as well as the first and last two simulated datasets. However, for the simulated dataset in the middle that represents the intermediate state between the trajectory type and cluster type transition, the difference in qLISI is not close to zero, but rather a value indicating the cluster type. Moreover, the change of qLISI differences across the simulated datasets of different parameters is not monotonic, which is a major limitation.

To see if we can overcome this limitation and obtain quantitatively correct results, we compute the ECDF area difference and the AUROC from all the LISI values. The results are shown in Fig. 4.5bc. We can see that similar to the qLISI difference, these two statistics can give qualitatively correct results for the reference dataset and the two simulated datasets on the two ends of the trajectory to cluster transition spectrum. However, they do not produce close to 0 (for ECDF area difference) or 0.5 (for AUROC) values for the intermediate dataset

and the changes of these two values are also not monotonic across the transition spectrum.

In addition to PCA 2D plots, we also obtained results from UMAP 2D plots. The results are summarized in Supplementary Fig. 4.6 with the same conclusions.

4.4 Discussion

In this project, we have explored two computational approaches to decide latent structure types for scRNA-seq datasets. Compared with Approach 1, Approach 2 has a more interpretable procedure and gives better results for the trajectory-type transition to cluster-type simulated datasets example. Approach 2's LISI based summary statistics can give qualitatively correct results for typical trajectory-type and cluster-type datasets, but do not do well for intermediate datasets nor are they quantitatively meaningful. To obtain better results, other types of similarity measures can be considered. For example, we can mix the simulated dataset with the test dataset and train a classification model to separate them. Then a worse classification accuracy would represent better similarity. Alternatively, we can compute the Wasserstein distance between the simulated dataset and the test dataset using optimal transport [176]. Compared to the LISI values in the 2D space, these two measures have the advantage of being computed in the original dimensions of the datasets, losing less information.

In addition to challenges with improving the statistical techniques of Approach 2, we also face the challenge of the complication of real biological datasets. Supplementary Fig. 4.7 shows the PCA visualizations of some of the datasets used in the trajectory inference methods benchmark paper [38]. These datasets are trajectory datasets with "gold standard" labels, meaning that their "reference trajectory was not extracted from the expression data itself, such as via cellular sorting or cell mixing" [38]. Here, we have shown the results of the first four PCs. However, the "overall shape" of the datasets can be seen from just the first two PCs. From visual inspection, although the two datasets in panels (a) and (b) show trajectory-type patterns, it is hard to determine for datasets in panels (c) - (f). For the dataset in panel (c), the pattern formed by the datasets from each time point appears more

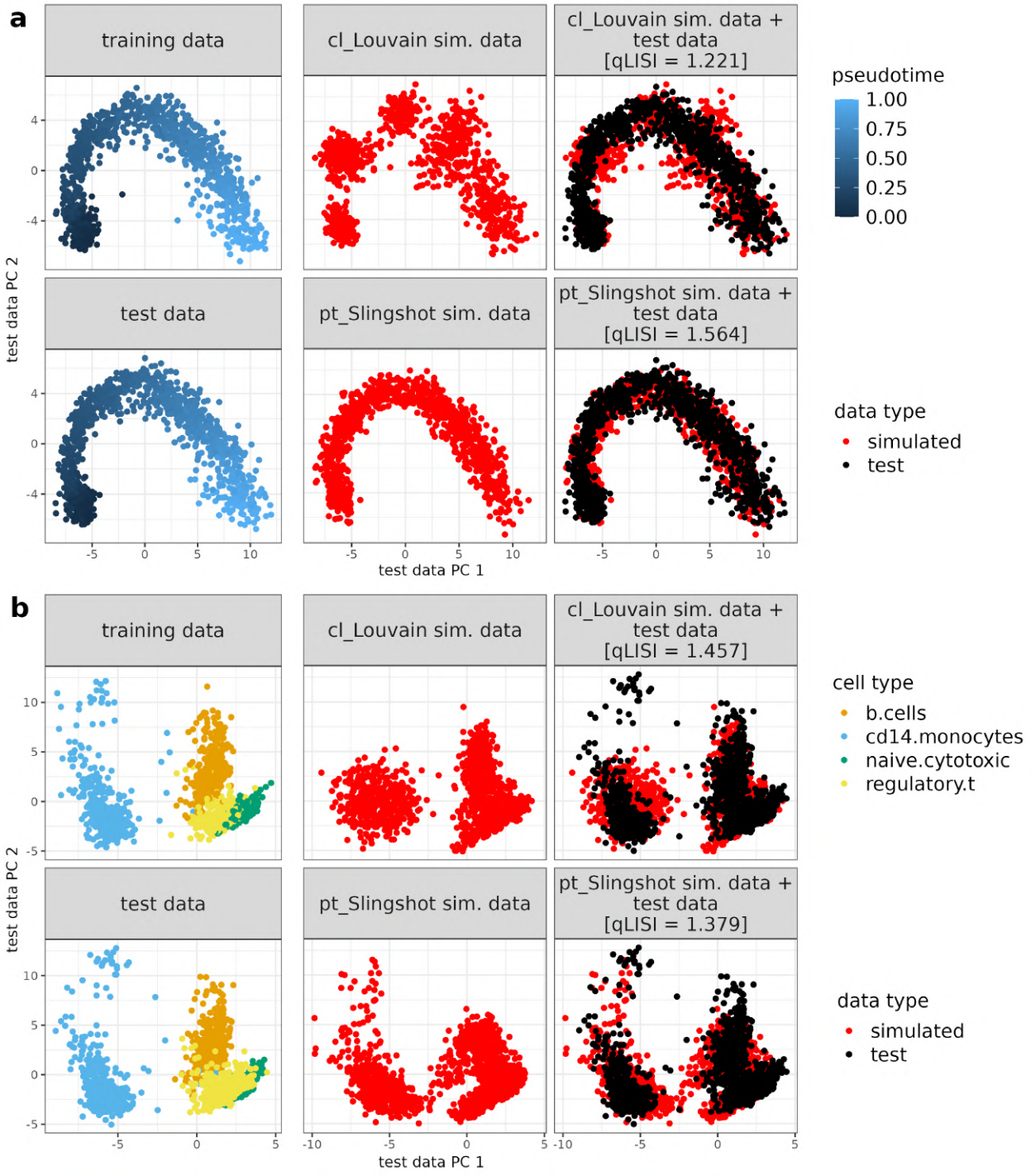

Figure 4.4: 2D principal component (PC) plots of the data-simulation-based approach for two datasets.

(a) Results for a typical trajectory-type dataset. (b) Results for a typical cluster-type dataset. Gene expression counts are transformed as $\log(1+\text{count})$ before dimensionality reduction. qLISI is the lower 10% quantile of the cell local inverse Simpson's Index, a higher value of which indicates that the simulated data mix better with the test data in the 2D visualization plot. By visually inspecting the patterns in these plots as well as comparing the qLISI values, we find that the simulated dataset generated under the correct latent structure type mixes better with the test dataset in these plots and has higher qLISI values.

a

qLISI	Reference	scDesign3, 1	scDesign3, 0.2	scDesign3, 0.04	scDesign3, 0.008	scDesign3, 0.002
Clusters (Louvain)	1.221	1.303	1.172	1.679	1.718	1.660
Trajectories (Slingshot)	1.564	1.715	1.498	1.352	1.318	1.258
Difference	-0.343	-0.412	-0.326	0.327	0.400	0.402

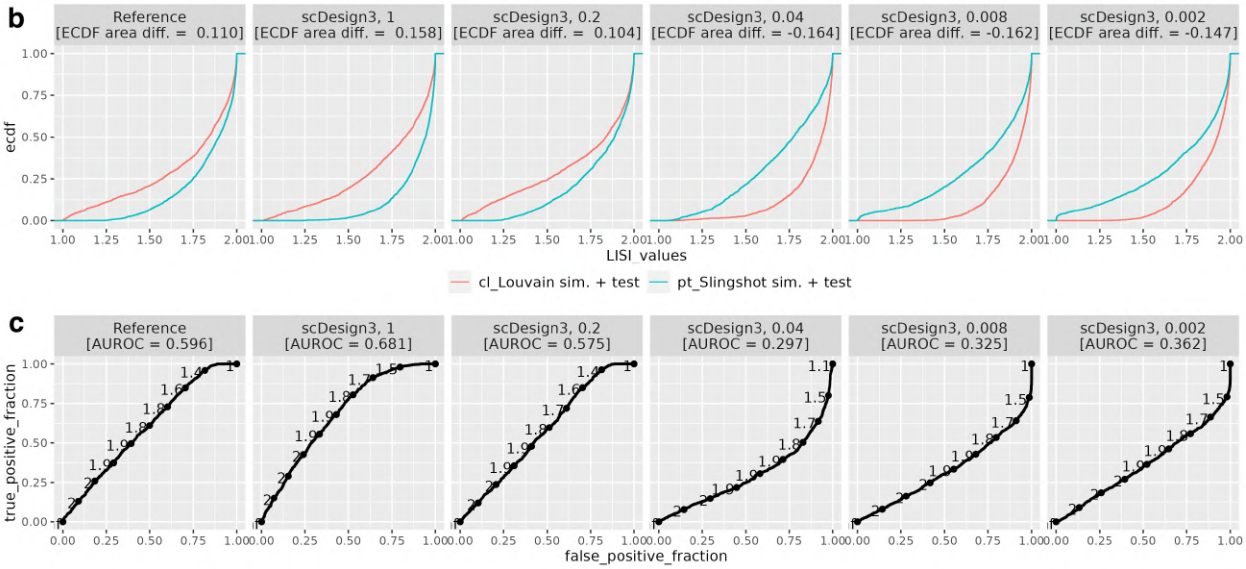

Figure 4.5: LISI-value-based results summary for the real and simulated datasets in Fig. 4.2.

The LISI values are computed for simulated and test cells in the 2D principal component (PC) space of the test cells. (a) Table summary of the qLISI values. (b) Plot summary for the ECDF area differences. (c) Plot summary for the AUROC values. All three panels show that these LISI-value-based summary statistics can give qualitatively correct conclusions, but do not do well for the intermediate dataset or monotonically change across the trajectory-type to cluster-type transition.

like clusters than a trajectory. The dataset in panel (d) appears like only one data cloud, but not a trajectory. The dataset in panel (e) appears like two clusters weakly connected with each other. In each of the three cases of (c) - (e), it is hard to justify a trajectory direction that aligns with the true time values. The dataset in panel (f) again appears like a few clusters weakly connected. Also, even if a trajectory can be created for it, it is not possible to determine the start and ends of the trajectory based on data points alone. These examples demonstrate the limitations of solving the latent structure type selection problem based on the gene expression count matrix alone.

4.5 Acknowledgements

I would like to thank my advisor, Dr. Jingyi Jessica Li, for the overall guidance and Dongyuan Song for the discussion of this project. We also appreciate the comments and feedback from the members of the Junction of Statistics and Biology at UCLA (http://jsb.ucla.edu).

S4.6 Supplementary Figures

a

qLISI	Reference	scDesign3, 1	scDesign3, 0.2	scDesign3, 0.04	scDesign3, 0.008	scDesign3, 0.002
Clusters (Louvain)	1.334	1.367	1.481	1.655	1.636	1.619
Trajectories (Slingshot)	1.674	1.709	1.633	1.442	1.490	1.571
Difference	-0.34	-0.342	-0.152	0.213	0.146	0.048

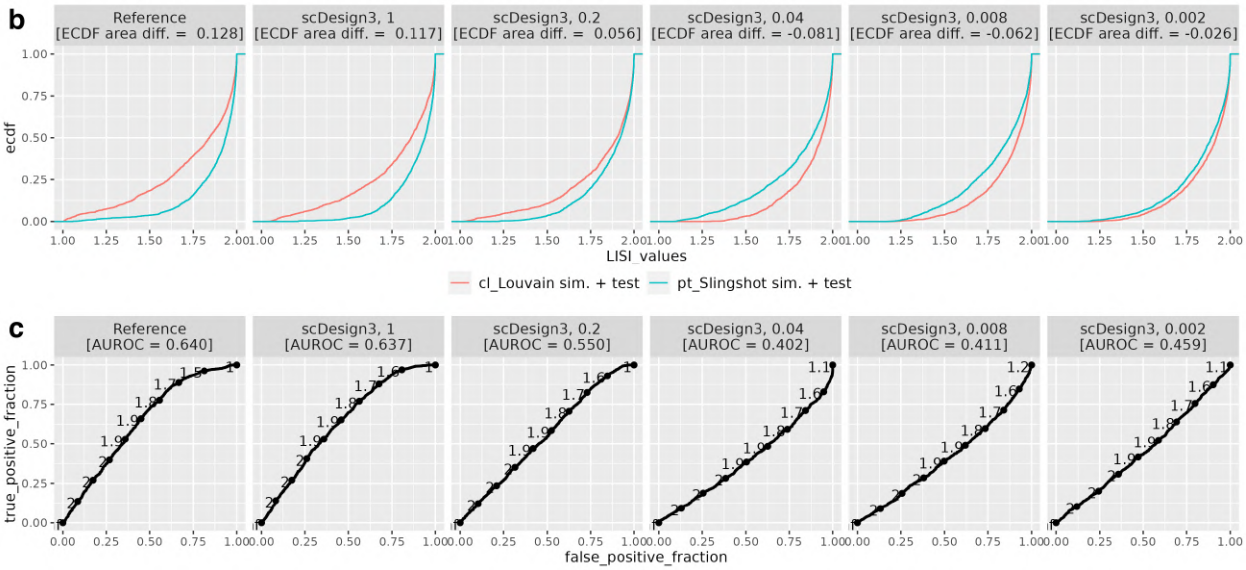

Figure 4.6: LISI-value-based results summary for the real and simulated datasets in Fig. 4.2.

The LISI values are computed for simulated and test cells in the 2D UMAP space of the test cells. (a) Table summary of the qLISI values. (b) Plot summary for the ECDF area differences. (c) Plot summary for the AUROC values. All three panels show that these LISI-value-based summary statistics can give qualitatively correct conclusions, but does not do well for the intermediate dataset or monotonically change across the trajectory-type to cluster-type transition.

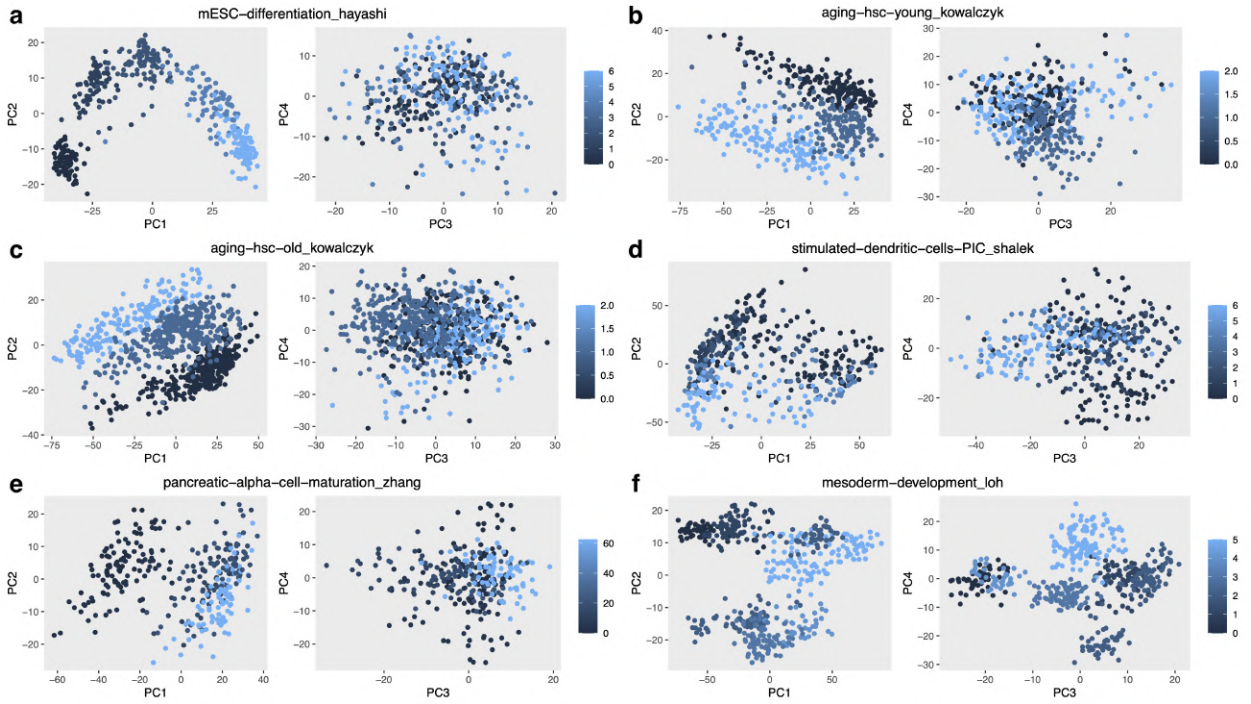

Figure 4.7: 2D principal component analysis (PCA) plots of selected real "gold standard" datasets used in the trajectory inference methods benchmark paper [38].

Gene expression counts are transformed as $\log(1 + \text{count})$ before dimensionality reduction and results of the first four PCs are shown. The cells are colored by their associated time values. Although the data patterns in (a) and (b) can be visually determined as trajectory-type relatively easily, this is not the case for the datasets in (c) - (f).

Bibliography

[1] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-cell RNA sequencing technologies and bioinformatics pipelines". In: *Experimental & molecular medicine* 50.8 (2018), pp. 1–14.

[2] Dragomirka Jovic et al. "Single-cell RNA sequencing technologies and applications: A brief overview". In: *Clinical and Translational Medicine* 12.3 (2022), e694.

[3] Grace XY Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature communications* 8.1 (2017), p. 14049.

[4] Pavithra Kumar, Yuqi Tan, and Patrick Cahan. "Understanding development and stem cells using single cell-based analyses of gene expression". In: *Development* 144.1 (2017), pp. 17–32.

[5] Ugomma C Eze et al. "Single-cell atlas of early human brain development highlights heterogeneity of human neuroepithelial cells and early radial glia". In: *Nature neuroscience* 24.4 (2021), pp. 584–594.

[6] Ethan J Armand et al. "Single-cell sequencing of brain cell transcriptomes and epigenomes". In: *Neuron* 109.1 (2021), pp. 11–26.

[7] Fengying Wu et al. "Single-cell profiling of tumor heterogeneity and the microenvironment in advanced non-small cell lung cancer". In: *Nature communications* 12.1 (2021), p. 2540.

[8] Yijie Zhang et al. "Single-cell RNA sequencing in cancer research". In: *Journal of Experimental & Clinical Cancer Research* 40 (2021), pp. 1–17.

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

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

[11] Orit Rozenblatt-Rosen et al. "The Human Cell Atlas: from vision to reality". In: *Nature* 550.7677 (2017), pp. 451–453.

[12] Aviv Regev et al. "The human cell atlas". In: *elife* 6 (2017), e27041.

[13] Yoshinari Ando, Andrew Tae-Jun Kwon, and Jay W Shin. "An era of single-cell genomics consortia". In: *Experimental & Molecular Medicine* 52.9 (2020), pp. 1409–1418.

[14] Simone Picelli et al. "Full-length RNA-seq from single cells using Smart-seq2". In: *Nature Protocols* 9.1 (Jan. 2014), pp. 171–181. DOI: 10.1038/nprot.2014.006.

[15] Grace X. Y. Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature Communications* 8.1 (Jan. 2017). DOI: 10.1038/ncomms14049.

[16] Malte D Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular systems biology* 15.6 (2019), e8746.

[17] Christoph Hafemeister and Rahul Satija. "Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression". In: *Genome biology* 20.1 (2019), p. 296.

[18] Nicholas Lytal, Di Ran, and Lingling An. "Normalization methods on single-cell RNA-seq data: an empirical survey". In: *Frontiers in genetics* 11 (2020), p. 41.

[19] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[20] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[21] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[22] Etienne Becht et al. "Dimensionality reduction for visualizing single-cell data using UMAP". In: *Nature biotechnology* 37.1 (2019), pp. 38–44.

[23] Emma Pierson and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis". In: *Genome Biology* 16.1 (Nov. 2015).

[24] Davide Risso et al. "A general and flexible method for signal extraction from single-cell RNA-seq data". In: *Nature Communications* 9.1 (2018). DOI: 10.1038/s41467-017-02554-5.

[25] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008). DOI: 10.1088/1742-5468/2008/10/p10008.

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

[27] Vladimir Yu Kiselev et al. "SC3: consensus clustering of single-cell RNA-seq data". In: *Nature Methods* 14.5 (Mar. 2017), pp. 483–486. DOI: 10.1038/nmeth.4236.

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

[29] 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.

[30] 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), p. 6077.

[31] Liang He et al. "NEBULA is a fast negative binomial mixed model for differential or co-expression analysis of large-scale multi-subject single-cell data". In: *Communications biology* 4.1 (2021), p. 629.

[32] Greg Finak et al. "MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA-seq data." In: (2015).

[33] Mengqi Zhang and F Richard Guo. "BSDE: barycenter single-cell differential expression for case–control studies". In: *Bioinformatics* 38.10 (2022), pp. 2765–2772.

[34] 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.

[35] Luke Zappia and Fabian J Theis. "Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape". In: *Genome biology* 22 (2021), pp. 1–18.

[36] 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).

[37] Monika Krzak et al. "Benchmark and parameter sensitivity analysis of single-cell RNA sequencing clustering methods". In: *Frontiers in genetics* 10 (2019), p. 1253.

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

[39] Nan Miles Xi and Jingyi Jessica Li. "Benchmarking computational doublet-detection methods for single-cell RNA sequencing data". In: *Cell systems* 12.2 (2021), pp. 176–194.

[40] 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), p. 6911.

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

[42] J. Peccoud and B. Ycart. "Markovian Modeling of Gene Product Synthesis". In: *Theoretical Population Biology* 48.2 (1995), pp. 222–234.

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

[44] Evan Z. Macosko et al. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets". In: *Cell* 161.5 (2015), pp. 1202–1214.

[45] Allon M. Klein et al. “Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells”. In: *Cell* 161.5 (2015), pp. 1187–1201.

[46] 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.

[47] Wei Vivian Li and Jingyi Jessica Li. “An accurate and robust imputation method scImpute for single-cell RNA-seq data”. In: *Nature Communications* 9.1 (2018).

[48] Zhun Miao et al. “DEsingle for detecting three types of differential expression in single-cell RNA-seq data”. In: *Bioinformatics* (2017).

[49] Michael B. Elowitz et al. “Stochastic Gene Expression in a Single Cell”. In: *Science* 297.5584 (Aug. 2002), pp. 1183–1186.

[50] Jonathan M. Raser and Erin K. O’Shea. “Control of Stochasticity in Eukaryotic Gene Expression”. In: *Science* 304.5678 (June 2004), pp. 1811–1814.

[51] Jonathan R. Chubb et al. “Transcriptional Pulsing of a Developmental Gene”. In: *Current Biology* 16.10 (May 2006), pp. 1018–1025.

[52] Arjun Raj et al. “Stochastic mRNA Synthesis in Mammalian Cells”. In: *PLoS Biology* 4 (Oct. 2006).

[53] Jong Kim and John C Marioni. “Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data”. In: *Genome Biology* 14.1 (Jan. 2013).

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

[55] Mihails Delmans and Martin Hemberg. “Discrete distributional differential expression (D3E) - a tool for gene expression analysis of single-cell RNA-seq data”. In: *BMC Bioinformatics* 17.1 (Feb. 2016).

[56] Mads Kærn et al. “Stochasticity in gene expression: from theories to phenotypes”. In: *Nature Reviews Genetics* 6.6 (June 2005), pp. 451–464.

[57] 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.

[58] 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.

[59] Alexandra-Chloé Villani et al. “Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors”. In: *Science* 356.6335 (2017).

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

[61] Maximilian Strunz et al. “Alveolar regeneration through a Krt8+ transitional stem cell state that persists in human lung fibrosis”. In: *Nature communications* 11.1 (2020), pp. 1–20.

[62] Loukia G Karacosta et al. “Mapping lung cancer epithelial-mesenchymal transition states and trajectories with single-cell resolution”. In: *Nature communications* 10.1 (2019), pp. 1–15.

[63] Volker Bergen et al. “Generalizing RNA velocity to transient cell states through dynamical modeling”. In: *Nature Biotechnology* (2020), pp. 1–7.

[64] 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.

[65] Li-Fang Chu et al. “Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm”. In: *Genome biology* 17.1 (2016), p. 173.

[66] Nathan G Skene et al. “Genetic identification of brain cell types underlying schizophrenia”. In: *Nature genetics* 50.6 (2018), pp. 825–833.

[67] Qingyun Li et al. “Developmental heterogeneity of microglia and brain myeloid cells revealed by deep single-cell RNA sequencing”. In: *Neuron* 101.2 (2019), pp. 207–223.

[68] Itay Tirosh et al. "Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq". In: *Science* 352.6282 (2016), pp. 189–196.

[69] Woosung Chung et al. "Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer". In: *Nature communications* 8.1 (2017), pp. 1–12.

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

[71] Aleksandra A Kolodziejczyk et al. "The technology and biology of single-cell RNA sequencing". In: *Molecular cell* 58.4 (2015), pp. 610–620.

[72] Xiannian Zhang et al. "Comparative analysis of droplet-based ultra-high-throughput single-cell RNA-seq systems". In: *Molecular cell* 73.1 (2019), pp. 130–142.

[73] Geng Chen, Baitang Ning, and Tieliu Shi. "Single-cell RNA-seq technologies and related computational data analysis". In: *Frontiers in genetics* 10 (2019), p. 317.

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

[75] Tamar Hashimshony et al. "CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq". In: *Genome Biology* 17.1 (Apr. 2016). DOI: 10.1186/s13059-016-0938-8.

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

[77] Todd M Gierahn et al. "Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput". In: *Nature methods* 14.4 (2017), pp. 395–398.

[78] Alex A Pollen et al. "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex". In: *Nature Biotechnology* 32.10 (Aug. 2014), pp. 1053–1058. DOI: 10.1038/nbt.2967.

[79] Kuanwei Sheng et al. "Effective detection of variation in single-cell transcriptomes using MATQ-seq". In: *Nature methods* 14.3 (2017), pp. 267–270.

[80] Ashwinikumar Kulkarni et al. "Beyond bulk: a review of single cell transcriptomics methodologies and applications". In: *Current opinion in biotechnology* 58 (2019), pp. 129–136.

[81] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. "Exponential scaling of single-cell RNA-seq in the past decade". In: *Nature protocols* 13.4 (2018), pp. 599–604.

[82] Teemu Kivioja et al. "Counting absolute numbers of molecules using unique molecular identifiers". In: *Nature methods* 9.1 (2012), pp. 72–74.

[83] Valentine Svensson et al. "Power analysis of single-cell RNA-sequencing experiments". In: *Nature methods* 14.4 (2017), pp. 381–387.

[84] Christoph Ziegenhain et al. "Comparative analysis of single-cell RNA sequencing methods". In: *Molecular cell* 65.4 (2017), pp. 631–643.

[85] Alessandra Dal Molin and Barbara Di Camillo. "How to design a single-cell RNA-sequencing experiment: pitfalls, challenges and perspectives". In: *Briefings in bioinformatics* 20.4 (2019), pp. 1384–1394.

[86] Martin Jinye Zhang, Vasilis Ntranos, and David Tse. "Determining sequencing depth in a single-cell RNA-seq experiment". In: *Nature communications* 11.1 (2020), pp. 1–11.

[87] 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. DOI: 10.1093/bioinformatics/btz321.

[88] Wei Vivian Li and Jingyi Jessica Li. "An accurate and robust imputation method scImpute for single-cell RNA-seq data". In: *Nature communications* 9.1 (2018), pp. 1–9.

[89] Yungang Xu et al. "scIGANs: single-cell RNA-seq imputation using generative adversarial networks". In: *Nucleic Acids Research* (2020).

[90] Emma Pierson and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis". In: *Genome biology* 16.1 (2015), pp. 1–10.

[91] Shiquan Sun et al. "Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis". In: *Genome biology* 20.1 (2019), p. 269.

[92] Yuqi Tan and Patrick Cahan. "SingleCellNet: a computational tool to classify single cell RNA-Seq data across platforms and across species". In: *Cell systems* 9.2 (2019), pp. 207–213.

[93] Hannah A Pliner, Jay Shendure, and Cole Trapnell. "Supervised classification enables rapid annotation of cell atlases". In: *Nature methods* 16.10 (2019), pp. 983–986.

[94] Nelson Johansen and Gerald Quon. "scAlign: a tool for alignment, integration, and rare cell identification from scRNA-seq data". In: *Genome biology* 20.1 (2019), pp. 1–21.

[95] Daphne Tsoucas and Guo-Cheng Yuan. "GiniClust2: a cluster-aware, weighted ensemble clustering method for cell-type detection". In: *Genome biology* 19.1 (2018), p. 58.

[96] Aashi Jindal et al. "Discovery of rare cells from voluminous single cell expression data". In: *Nature Communications* 9.1 (Nov. 2018). DOI: 10.1038/s41467-018-07234-6.

[97] 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.

[98] 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.

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

[100] 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.

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

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

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

[104] Tianyu Wang et al. "Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data". In: *BMC bioinformatics* 20.1 (2019), p. 40.

[105] Wenpin Hou et al. "A Systematic Evaluation of Single-cell RNA-sequencing Imputation Methods". In: *bioRxiv* (2020).

[106] Laurens Van Der Maaten. "Accelerating t-SNE using tree-based algorithms". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 3221–3245.

[107] 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 (Sept. 2019), pp. 3517–3519. DOI: 10.1093/bioinformatics/btz078.

[108] Luke Zappia, Belinda Phipson, and Alicia Oshlack. "Splatter: simulation of single-cell RNA sequencing data". In: *Genome Biology* 18.1 (2017). DOI: 10.1186/s13059-017-1305-0.

[109] Xiuwei Zhang, Chenling Xu, and Nir Yosef. "Simulating multiple faceted variability in single cell RNA sequencing". In: *Nature Communications* 10.1 (2019). DOI: 10.1038/s41467-019-10500-w.

[110] Giacomo Baruzzo, Ilaria Patuzzi, and Barbara Di Camillo. "SPARSim single cell: a count data simulator for scRNA-seq data". In: *Bioinformatics* (2019). DOI: 10.1093/bioinformatics/btz752.

[111] 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 (Jan. 2020). DOI: 10.1038/s41467-019-14018-z.

[112] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nature Methods* 15.12 (2018), pp. 1053–1058. DOI: 10.1038/s41592-018-0229-2.

[113] Keegan D. Korthauer et al. "A statistical approach for identifying differential distributions in single-cell RNA-seq experiments". In: *Genome Biology* 17.1 (Oct. 2016). DOI: 10.1186/s13059-016-1077-y.

[114] Beate Vieth et al. "powsimR: power analysis for bulk and single cell RNA-seq experiments". In: *Bioinformatics* 33.21 (Nov. 2017), pp. 3486–3488. DOI: 10.1093/bioinformatics/btx435.

[115] 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.

[116] F. William Townes et al. "Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model". In: *Genome Biology* 20.1 (Dec. 2019).

[117] Abhishek K Sarkar and Matthew Stephens. "Separating measurement and expression models clarifies confusion in single cell RNA-seq analysis". In: *BioRxiv* (2020).

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

[119] Adam L Haber et al. "A single-cell survey of the small intestinal epithelium". In: *Nature* 551.7680 (2017), pp. 333–339.

[120] Mauro J. Muraro et al. "A Single-Cell Transcriptome Atlas of the Human Pancreas". In: *Cell Systems* 3.4 (Oct. 2016), pp. 385–394. DOI: 10.1016/j.cels.2016.09.002.

[121] Spyros Darmanis et al. "A survey of human brain transcriptome diversity at the single cell level". In: *Proceedings of the National Academy of Sciences* 112.23 (May 2015), pp. 7285–7290. DOI: 10.1073/pnas.1507125112.

[122] Alexandra-Chloé Villani et al. "Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors". In: *Science* 356.6335 (2017).

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

[124] Baolin Liu et al. "An entropy-based metric for assessing the purity of single cell populations". In: *Nature communications* 11.1 (2020), pp. 1–13.

[125] Valentine Svensson, Sarah A Teichmann, and Oliver Stegle. "SpatialDE: identification of spatially variable genes". In: *Nature methods* 15.5 (2018), pp. 343–346.

[126] Shiquan Sun, Jiaqiang Zhu, and Xiang Zhou. "Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies". In: *Nature methods* 17.2 (2020), pp. 193–200.

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

[128] Xiaoyan Qian et al. "Probabilistic cell typing enables fine mapping of closely related cell types in situ". In: *Nature methods* 17.1 (2020), pp. 101–106.

[129] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information theoretic measures for clusterings comparison". In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML 09* (2009). DOI: 10.1145/1553374.1553511.

[130] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

[131] Eliot T McKinley et al. "Optimized multiplex immunofluorescence single-cell analysis reveals tuft cell heterogeneity". In: *JCI insight* 2.11 (2017).

[132] Rui Dong and Guo-Cheng Yuan. "GiniClust3: a fast and memory-efficient tool for rare cell type identification". In: *BMC bioinformatics* 21 (2020), pp. 1–7.

[133] Jacob Bien and Robert J Tibshirani. "Sparse estimation of a covariance matrix". In: *Biometrika* 98.4 (2011), pp. 807–820.

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

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

[136] 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.

[137] Robrecht Cannoodt et al. "dyngen: a multi-modal simulator for spearheading new single-cell omics analyses". In: *BioRxiv* (2020).

[138] Samuel L. Wolock, Romain Lopez, and Allon M. Klein. "Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data". In: *Cell Systems* 8.4 (Apr. 2019), pp. 281–291. DOI: 10.1016/j.cels.2018.11.005.

[139] Christopher S. Mcginnis, Lyndsay M. Murrow, and Zev J. Gartner. "DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors". In: *Cell Systems* 8.4 (Apr. 2019). DOI: 10.1016/j.cels.2019.03.003.

[140] Marlon Stoeckius et al. "Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics". In: *Genome Biology* 19.1 (Dec. 2018). DOI: 10.1186/s13059-018-1603-1.

[141] Nan Miles Xi and Jessica Li. "Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data". In: *SSRN Electronic Journal* (July 2020). DOI: 10.2139/ssrn.3646565.

[142] David Lähnemann et al. "Eleven grand challenges in single-cell data science". In: *Genome biology* 21.1 (2020), pp. 1–35.

[143] Rachel Y. Wang et al. "Network modeling in biology: statistical methods for gene and brain networks". In: *Statistical Science* (2020), (in press). URL: https://www.e-publications.org/ims/submission/STS/user/submissionFile/42325?confirm=7b64374b.

[144] Ying Ma et al. "Integrative differential expression and gene set enrichment analysis using summary statistics for scRNA-seq studies". In: *Nature communications* 11.1 (2020), pp. 1–13.

[145] 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.

[146] Hoa Thi Nhu Tran et al. "A benchmark of batch-effect correction methods for single-cell RNA sequencing data". In: *Genome biology* 21.1 (2020), pp. 1–32.

[147] Andrew Butler et al. "Integrating single-cell transcriptomic data across different conditions, technologies, and species". In: *Nature biotechnology* 36.5 (2018), pp. 411–420.

[148] Abe Sklar. "Fonctions de répartition à n dimensions et leurs marges". In: *Publications de l'Institut de Statistique de l'Université de Paris* 8 (1959), pp. 229–231.

[149] Christian Genest and Johanna Nešlehová. "A primer on copula for count data". In: *ASTIN Bulletin: The Journal of the IAA* 37.2 (2007), pp. 475–515.

[150] David I Inouye et al. "A review of multivariate distributions for count data derived from the Poisson distribution". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 9.3 (2017), e1398.

[151] Ludger Rüschendorf. "Copulas, Sklar's theorem, and distributional transform". In: *Mathematical Risk Analysis*. Springer, 2013, pp. 3–34.

[152] Athanassios N Avramidis, Nabil Channouf, and Pierre L'Ecuyer. "Efficient correlation matching for fitting discrete multivariate distributions with arbitrary marginals and normal-copula dependence". In: *INFORMS Journal on Computing* 21.1 (2009), pp. 88–106.

[153] Regis Lebrun and Anne Dutfoy. "An innovating analysis of the Nataf transformation from the copula viewpoint". In: *Probabilistic Engineering Mechanics* 24.3 (2009), pp. 312–320.

[154] Soumyadip Ghosh and Shane G Henderson. "Behavior of the NORTA method for correlated random vector generation as the dimension increases". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 13.3 (2003), pp. 276–294.

[155] Nabil Channouf and Pierre L'Ecuyer. "A normal copula model for the arrival process in a call center". In: *International Transactions in Operational Research* 19.6 (2012), pp. 771–787.

[156] Aaron TL Lun and John C Marioni. "Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data". In: *Biostatistics* 18.3 (2017), pp. 451–464.

[157] Laleh Haghverdi et al. "Diffusion pseudotime robustly reconstructs lineage branching". In: *Nature methods* 13.10 (2016), pp. 845–848.

[158] F Alexander Wolf et al. "PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells". In: *Genome biology* 20 (2019), pp. 1–9.

[159] Eugene P Wigner. "Characteristic vectors of bordered matrices with infinite dimensions i". In: *The Collected Works of Eugene Paul Wigner: Part A: The Scientific Papers* (1993), pp. 524–540.

[160] M MacMahon, D Garlaschelli, and Garlaschelli D MacMahon M. "Community detection for correlation matrices." In: *Physical Review X* 5 (2015), p. 021006.

[161] Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. "Cleaning large correlation matrices: tools from random matrix theory". In: *Physics Reports* 666 (2017), pp. 1–109.

[162] Oriol Bohigas, Marie-Joya Giannoni, and Charles Schmit. "Characterization of chaotic quantum spectra and universality of level fluctuation laws". In: *Physical review letters* 52.1 (1984), p. 1.

[163] Marc Potters, Jean-Philippe Bouchaud, and Laurent Laloux. "Financial applications of random matrix theory: Old laces and new pieces". In: *arXiv preprint physics/0507111* (2005).

[164] Mor Nitzan and Michael P Brenner. "Revealing lineage-related signals in single-cell gene expression using random matrix theory". In: *Proceedings of the National Academy of Sciences* 118.11 (2021), e1913931118.

[165] Chongli Qin and Lucy J Colwell. "Power law tails in phylogenetic systems". In: *Proceedings of the National Academy of Sciences* 115.4 (2018), pp. 690–695.

[166] Luis Aparicio et al. "A random matrix theory approach to denoise single-cell data". In: *Patterns* 1.3 (2020).

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

[168] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. "Distribution of eigenvalues for some sets of random matrices". In: *Matematicheskii Sbornik* 114.4 (1967), pp. 507–536.

[169] Tabula Sapiens Consortium* et al. "The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans". In: *Science* 376.6594 (2022), eabl4896.

[170] Bo Hu et al. "Origin and function of activated fibroblast states during zebrafish heart regeneration". In: *Nature genetics* 54.8 (2022), pp. 1227–1237.

[171] Gervaise H Henry et al. "A cellular anatomy of the normal adult human prostate and prostatic urethra". In: *Cell reports* 25.12 (2018), pp. 3530–3542.

[172] Poornima Bhat-Nakshatri et al. "A single-cell atlas of the healthy breast tissues reveals clinically relevant clusters of breast epithelial cells". In: *Cell Reports Medicine* 2.3 (2021).

[173] Yoshiharu Muto et al. "Single cell transcriptional and chromatin accessibility profiling redefine cellular heterogeneity in the adult human kidney". In: *Nature communications* 12.1 (2021), p. 2190.

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

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

[176] Cédric Villani et al. *Optimal transport: old and new*. Vol. 338. Springer, 2009.