Springer Handbooks of Computational Statistics

Henry Horng-Shing Lu
Bernhard Schölkopf
Martin T. Wells
Hongyu Zhao *Editors*

# Handbook of Statistical Bioinformatics

*Second Edition*

Springer

# Springer Handbooks of Computational Statistics

Henry Horng-Shing Lu • Bernhard Schölkopf •
Martin T. Wells • Hongyu Zhao
Editors

# Handbook of Statistical Bioinformatics

Springer

*Editors*
Henry Horng-Shing Lu
Institute of Statistics
National Yang Ming Chiao Tung University
Hsinchu, Taiwan, ROC

Bernhard Schölkopf
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Martin T. Wells
Department of Statistics and Data Science
Cornell University
Ithaca, NY, USA

Hongyu Zhao
Department of Biostatistics
Yale University
New Haven, CT, USA

ISSN 2197-9790 ISSN 2197-9804 (electronic)
Springer Handbooks of Computational Statistics
ISBN 978-3-662-65901-4 ISBN 978-3-662-65902-1 (eBook)
https://doi.org/10.1007/978-3-662-65902-1

# Preface

Numerous fascinating and important breakthroughs in biotechnology have generated massive volumes of high throughput data with diverse types that demand novel developments of efficient and appropriate tools in computational statistics that are integrated with biological knowledge and computational algorithms. This updated volume collects contributed chapters from leading researchers to survey many recent active research topics that have developed since the previous edition of the Handbook of Statistical Bioinformatics. This updated handbook is intended to serve as both an introductory and reference monograph for students and researchers who are interested in learning the state-of-the-art developments in computational statistics as applied to computational biology.

This collection of articles, from the leading scholars in the field, is primarily a monograph which will be of interest to the educational, academic, and professional organizations related to statisticians, computer scientists, biological and biomedical researchers with strong interests in computational biology. Although there are other volumes available for computational statistics and bioinformatics on the market, there are few books such as this that focus on the interface between computational statistics and cutting-edge developments in computational biology. Seeing this need, this completely updated collection is aimed to establish this bridge. This handbook covers many significant up-to-date topics in probabilistic and statistical modeling as well as the analysis of massive data sets generated from modern biotechnology. These methods and technologies will change the perspectives of biology, healthcare, and medicine in the twenty-first century! This collection is an extended version of the previous edited handbook. The advanced research topics cover statistical methods for single-cell analysis, network analysis, and systems biology.

During the editing process of this handbook, the world has been upended by the massive influence of COVID-19 pandemic and other challenges. The editors would like to thank the contributing authors, Springer management team members,

supporting colleagues and family members for their incredible support and patience during this challenging time period in order for this handbook to be made available to the related scholarly communities!

Hsinchu, Taiwan, ROC — Henry Horng-Shing Lu
Tübingen, Germany — Bernhard Schölkopf
Ithaca, NY, USA — Martin T. Wells
New Haven, CT, USA — Hongyu Zhao
May 8, 2022

# Contents

**Part I Single-Cell Analysis**

**Computational and Statistical Methods for Single-Cell RNA Sequencing Data** ..... 3
Zuoheng Wang and Xiting Yan

**Pre-processing, Dimension Reduction, and Clustering for Single-Cell RNA-seq Data** ..... 37
Jialu Hu, Yiran Wang, Xiang Zhou, and Mengjie Chen

**Integrative Analyses of Single-Cell Multi-Omics Data: A Review from a Statistical Perspective** ..... 53
Zhixiang Lin

**Approaches to Marker Gene Identification from Single-Cell RNA-Sequencing Data** ..... 71
Ronnie Y. Li, Wenjing Ma, and Zhaohui S. Qin

**Model-Based Clustering of Single-Cell Omics Data** ..... 85
Xinjun Wang, Haoran Hu, and Wei Chen

**Deep Learning Methods for Single-Cell Omics Data** ..... 109
Jingshu Wang and Tianyu Chen

**Part II Network Analysis**

**Probabilistic Graphical Models for Gene Regulatory Networks** ..... 135
Zhenwei Zhou, Xiaoyu Zhang, Peitao Wu, and Ching-Ti Liu

**Additive Conditional Independence for Large and Complex Biological Structures** ..... 153
Kuang-Yao Lee, Bing Li, and Hongyu Zhao

**Integration of Boolean and Bayesian Networks** ..... 173
Meng-Yuan Tsai and Henry Horng-Shing Lu

**Computational Methods for Identifying MicroRNA-Gene Regulatory Modules** ..... 187
Yin Liu

**Causal Inference in Biostatistics** ..... 209
Shasha Han and Xiao-Hua Zhou

**Bayesian Balance Mediation Analysis in Microbiome Studies** ..... 237
Lu Huang and Hongzhe Li

**Part III Systems Biology**

**Identifying Genetic Loci Associated with Complex Trait Variability** ..... 257
Jiacheng Miao and Qiongshi Lu

**Cell Type-Specific Analysis for High-throughput Data** ..... 271
Ziyi Li and Hao Wu

**Recent Development of Computational Methods in the Field of Epitranscriptomics** ..... 285
Zijie Zhang, Shun Liu, Chuan He, and Mengjie Chen

**Estimation of Tumor Immune Signatures from Transcriptomics Data** .... 311
Xiaoqing Yu

**Cross-Linking Mass Spectrometry Data Analysis** ..... 339
Chen Zhou and Weichuan Yu

**Cis-regulatory Element Frequency Modules and their Phase Transition across Hominidae** ..... 371
Lei M Li, Mengtian Li, and Liang Li

**Improved Method for Rooting and Tip-Dating a Viral Phylogeny** ..... 397
Xuhua Xia

# Part I
# Single-Cell Analysis

# Computational and Statistical Methods for Single-Cell RNA Sequencing Data

**Zuoheng Wang and Xiting Yan**

**Abstract** In recent years, advances in droplet-based technology have boosted the popularity of using single-cell RNA sequencing (scRNA-seq) technology to investigate transcriptomic and cell population composition changes in various tissues and diseases. Despite the potential of these technologies in understanding disease pathogenesis and developing novel personalized therapeutics, analyses of the generated scRNA-seq data are challenging, mainly due to high noise level, prevalent dropout events, heterogeneous sources of variation confounding phenotype of interest, and so on. In this chapter, we introduce these challenges in analyses of scRNA-seq data and the corresponding computational and statistical methods developed to address them. The topics include data preprocessing, data normalization, dropout imputation, and differential expression analysis.

## 1 Introduction

Gene expression profiling measures levels of mRNA to understand transcriptomic changes due to disease, treatment, environment, time, and so on. Traditional bulk RNA gene expression profiling using microarrays and RNA sequencing pools RNAs from a large population of cells consisting of various and often unknown cell types. It measures the average expression profile in mixed cell populations with unknown contribution from different cells or cell types. Thus, bulk RNA gene expression data is unable to precisely identify the cellular source of transcriptomic changes of interest, especially when high cell-to-cell heterogeneity exists [1–5]. To investigate transcriptomic changes at single-cell resolution, two major challenges exist including (1) isolating cells from each other without strong perturbations to cells that lead to systematic transcriptomic changes and (2) amplification of extremely low

Z. Wang · X. Yan (✉)
Yale University, New Haven, CT, USA
e-mail: zuoheng.wang@yale.edu; xiting.yan@yale.edu



H. H.-S. Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics,
https://doi.org/10.1007/978-3-662-65902-1_1

amount of mRNAs from each cell. To address these challenges, multiple types and generations of single-cell transcriptomic technology, including single-cell qPCR [6–11], single-cell microarray [12–14], and single-cell RNA sequencing [15–18], have been developed. Major differences between these technologies exist in the steps of single-cell capturing, cDNA amplification, and cDNA profiling. There are mainly five ways to capture single cell, including micropipetting micromanipulation, laser capture microdissection, fluorescence-activated cell sorting (FACS), microfluidics, and microdroplets [19]. The early-staged micropipetting micromanipulation and laser capture microdissection that capture low number of cells are time consuming and require microliter volumes of specimen. FACS and microfluidics can both capture hundreds of cells and are fast although microfluidics requires nanoliter volumes. Microdroplets, the most popular cell capturing method, can capture the largest number of cells (currently from thousands to tens of thousands), are fast, and require nanoliter volumes. After cells are captured at single-cell resolution, mRNAs are reversed transcribed into cDNAs and further amplified using PCR, which may have amplification bias leading to uneven amplification across different genes. Some of the single-cell sequencing technologies reduce or remove this amplification bias by in vitro transcription (IVT) or unique molecular identifier (UMI). The amplified cDNAs will further be profiled using qPCR, microarrays, or RNA sequencing with RNA sequencing being the most popular due to its unbiasedness in gene capturing and the existence of nonspecific probe binding in microarrays. Taken together, the single-cell transcriptomics field has moved from capturing a few targeted genes in less than 100 cells by single-cell qPCR to whole-transcriptome profiling in hundreds of thousands of cells in an unbiased style by droplet-based single-cell RNA sequencing.

Instead of pooling RNAs from all cells together, droplet-based single-cell RNA (scRNA-seq) sequencing technologies isolate cells in oil droplets and measure transcriptome-wide mRNA expression levels in each single cell separately. Despite differences in protocols, each scRNA-seq technology follows a similar basic strategy. As an example, we demonstrate the workflow of 10x Genomics Chromium Single Cell 3′ v3 assay in Fig. 1. First, organ or tissue samples are processed to generate a single-cell suspension in which cells are separated. Most of the time, this process involves usage of proteases to digest attachments between cells, especially for solid tissues. Due to different perturbations the dissociation step could have on different cell types, tissue dissociation needs to be optimized to balance between releasing cell types that are difficult to dissociate and avoiding damage to fragile cell types. Second, cells are co-encapsulated with distinctively barcoded microparticle (bead) in oil droplet. Ideally, each droplet contains only one cell and one bead. Cells are lysed in droplet. Third, mRNAs in each droplet are reverse transcribed into full-length cDNAs, during which oligo primers on beads are ligated onto cDNAs. Each oligo primer consists of sequencing primer, cell barcode, UMI, and poly(dT). Cell barcode is the same across all oligo primers on the same bead but UMI is distinctively unique. As a result, cell barcodes and UMIs can identify cell origin and transcript origin of each cDNA, respectively. Finally, the full-length cDNAs from different droplets are pooled, amplified, and fragmented into smaller cDNA

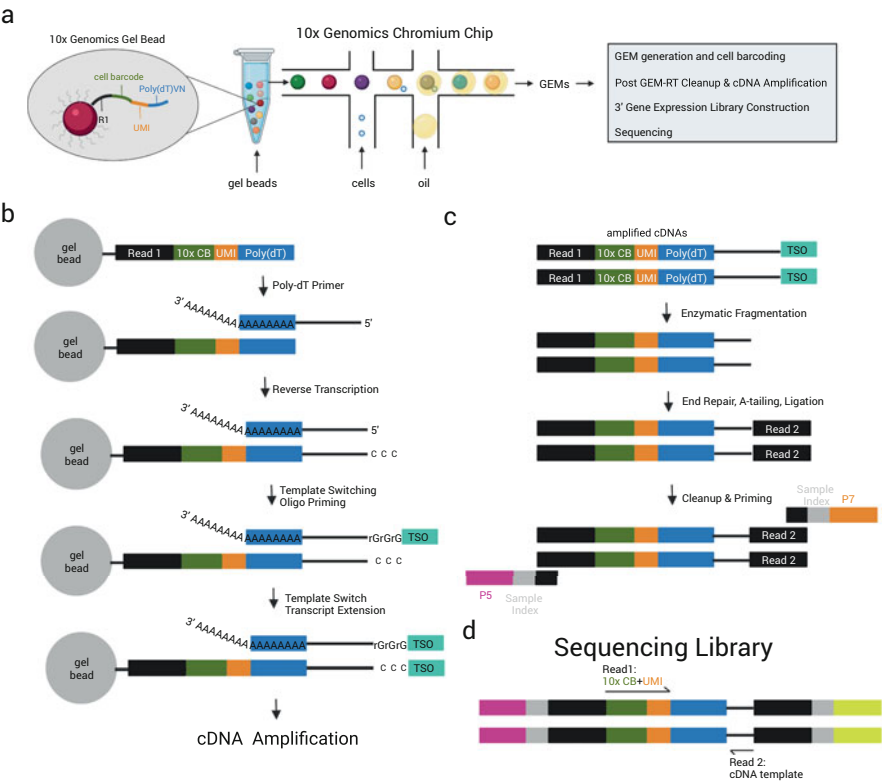


**Fig. 1** Workflow of 10x Genomics Chromium 3′ V3 chip. (**a**) Structure of the oligo primer on the gel beads. (**b**) For each GEM, steps to capture mRNAs with poly(A) tails and reverse transcribe them into cDNAs for amplification. (**c**) Steps to fragment the amplified cDNAs into small pieces and ligating sample index, P5 and P7 for sequencing. (**d**) Structure of the final cDNA templates in the library for sequencing. Read 1 and Read 2 are copies of cDNA template from the corresponding location, representing the cell barcode+UMI and a small fragment of cDNA from the 3′ end of the transcript. Created with BioRender.com

inserts for sequencing using enzymes. The fragmented inserts are cleaned up to only keep those with oligo primers, which are from the 3′ end of cDNAs with polyA tails. In each of these inserts, one end has oligo primer, and the other end has sequence from the cDNA template. Both ends are sequenced using a pair of sequencing reads (Read 1 and Read 2). Read 1 sequences cell barcode and UMI to determine the cell origin and remove PCR duplication. Read 2 measures the sequence content of a small fragment of the transcript close to the 3′ end, which can be mapped to human genome to determine the gene origin of the mRNA. In this way, sequencing reads can get demultiplexed into different cells and different transcripts to enable single-cell transcriptome profiling and PCR amplification bias reduction.

To date, many on-market scRNA-seq platforms are mainly different in the total number of captured cells, whether full-length cDNAs are profiled, and whether

UMI or IVT is used to reduce PCR amplification bias. Applications of scRNA-seq technologies in different human diseases and tissues have revealed potential disease-associated rare cell types, cell population composition changes, and cell type-specific transcriptomic changes [20–24]. These scRNA-seq datasets also have the potential to provide information on disease-associated in vivo cell-to-cell communication in different tissues. Moreover, scRNA-seq technology has also served as a base for development of single-nucleus RNA sequencing (snRNA-seq) and spatial single-cell transcriptomic technology. The snRNA-seq measures transcriptome in single nucleus, and the spatial single-cell transcriptomics measure single-cell transcriptome together with spatial location of each cell in intact tissue. The extra information gained through these technologies has further boosted our understanding of single-cell biology in challenging tissues, spatial structure of tissues at single-cell resolution, and in vivo cell-to-cell communications.

## 2 Data Preprocessing

Raw scRNA-seq data are sequencing reads in FASTQ or BAM formatted files that need to be preprocessed and quantified for downstream analysis. In this section, we describe the preprocessing of scRNA-seq data from 10x Genomics Chromium platform, which is currently the most popular scRNA-seq platform. Preprocessing of data from other scRNA-seq technologies should follow the same principle with small variations. Multiple tools have been developed to preprocess 10x Genomics scRNA-seq data, including the Cell Ranger pipeline from 10x Genomics, STARsolo [25], Alevin/Alevin-fry [26], Kallisto-bustools [27], UMI-tools [28], and zUMI [29]. Despite differences across these methods, key common steps in these methods include (1) reads mapping, (2) cell barcodes demultiplexing with or without error correction, (3) UMI deduplication with or without error correction, and (4) cell barcodes selection. These methods have been previously reviewed and compared [30, 31]. The output of data preprocessing is a matrix of counts, in which rows are genes, columns are cells, and each entry is the number of UMIs of the corresponding gene in the corresponding cell.

### *2.1 Reads Mapping*

The first key preprocessing step is to map the reads from cDNA templates back to the target genome or transcriptome to identify their transcript origin. There are mainly two types of aligners used in the existing methods. One category maps reads back to the genome. STAR [32] is the most popular method in this category due to its high mapping accuracy and its capacity in identifying novel exons, constructing splice junction libraries based on the data and providing the two-pass mapping option for more accurate mapping results. Other aligners in this category used in the existing

scRNA-seq data preprocessing pipelines include BWA [33], Tophat2 [34], Subread [35], and Bowtie2 [36]. The other category maps reads to transcriptome instead of genome, including RapMap [37] and kallisto [38] used in Alevin and Kallisto-bustools, respectively. These mappers are lightweight and very efficient in both memory usage and speed. However, the results strongly depend on the transcriptome annotation. Potential incomplete annotation of exons and splicing junctions could lead to inaccurate mapping results.

## *2.2 Cell Barcodes Demultiplexing*

The second key step is to correct for sequencing errors in cell barcodes so that reads with the same cell barcodes can be assigned to the same cell. For 10x Genomics Chromium platform, a barcode whitelist is provided which contains all known barcode sequences included in the assay kit. Under perfect scenario, all observed cell barcodes can be compared to this list to split reads into different cells. However, sequencing errors cause the observed cell barcodes to be slightly different from the true cell barcodes. A common approach to correct these sequencing errors is to consider cell barcodes within a given Hamming distance as the same cell barcode. The Hamming distance-based approach completely relies on the number of base differences between the observed cell barcodes and the whitelist barcodes, which may be inaccurate due to varying sequencing quality of different bases in the observed cell barcodes. To address this, the Cell Ranger pipeline estimates the posterior probability of an observed cell barcode originating from a given whitelisted cell barcode based on sequencing quality score and the number of reads exactly matching the whitelist barcode. For technologies without a manufacturer provided cell barcode whitelist, Alevin and STARsolo provide the option to run one pass of the cell barcode without cell barcode correction and use the uncorrected cell barcodes from the first pass as the "whitelist" for the second pass of demultiplexing with correction.

## *2.3 UMI Collapsing*

The third key preprocessing step is to deduplicate UMIs with or without error correction. Ideally, reads with the same UMI and cell barcodes originate from the same transcript and therefore should be counted as one single UMI. However, in real data, sequencing errors (nucleotide substitutions, nucleotide miscalling, insertion, deletion, and recombination) in both UMI and cDNA read cause reads originating from the same transcript to have slightly different UMIs leading to overestimated number of UMIs and to be mapped to different genes or transcripts leading to multigene or multi-transcript UMIs. To correct for errors in UMIs, considering that miscalling during sequencing is the most prevalent error, both zUMI and

Cell Ranger pipeline correct errors in UMI by collapsing UMIs within a given Hamming distance. UMI-tools [28] implemented two previously proposed methods, unique and percentile, and developed three network-based UMI error correction methods including cluster, adjacency, and directional. The directional method was shown to have the highest accuracy and robustness in both simulated data and real data. STARsolo provides both options to use Hamming distance and directional method for UMI error correction. Kallisto-bustools reported low percentage of reads recovered by UMI error correction (0.5 and 0.6% for 10-base-pair and 12-base-pair UMIs, respectively) and therefore does not perform UMI error correction. Alevin does not correct for errors in UMI either. To resolve the multigene UMIs, Alevin utilizes transcript-level information and a parsimonious UMI graph (PUG) to find a minimal set of transcripts to cover the PUG and split the multigene or multi-transcript UMIs. Both STARsolo and Cell Ranger pipeline compare the number of reads supporting the multiple genes a UMI is associated with and keep the gene with the largest number of supporting reads. In addition, STARsolo provides options to filter out all multigene UMIs, to uniformly distribute the multigene UMIs to all genes, to distribute multigene UMIs among all genes using maximum likelihood estimation (MLE) that consider other UMIs from the same cell, and to distribute multigene UMIs to their gene set proportionally to the sum of the number of unique-gene UMIs and uniformly distributed multigene UMIs in each gene. Kallisto-bustools performs naïve collapsing based on its report of low percentage of lost counts (0.4 and 0.17% for 10xv2 and 10xv3 dataset, respectively).

## *2.4 Cell Barcodes Selection*

The previous key steps generate the raw gene × cell barcode count matrix, which includes cell barcodes from empty droplets containing ambient RNAs as well as target cells. Since usually the empty droplets have significant lower RNA content, Cell Ranger pipeline v2.2 [17] simply kept cell barcodes with total number of UMIs (nUMI) higher than 10% of the robust maximum count defined as the 99-th percentile of the largest $N$ UMI counts where $N$ is the expected number of cells to be captured in the experiment. This approach is similar to the knee-point thresholding approach [18] that searches for an inflection point or "knee" in the cumulative frequency of total nUMI per barcode and filters out barcodes with nUMI lower than the identified knee point. The most recent and popular method is the EmptyDrops approach [39]. It estimates the profile of cell barcodes containing ambient RNAs and test each cell barcode for deviations from the estimated profile using a Dirichlet-multinomial model of UMI count sampling. Barcodes with significant deviations are considered as cells and included for downstream analysis. This approach allows inclusion of cells with low total RNA content and thus small total nUMIs. Different versions of Cell Ranger pipelines provide different cell barcode selection approaches but covered all the three methods described above. STARsolo provides options for both the Cell Ranger v2.2 approach and EmptyDrops. Alevin conducts

the knee-based approach at the beginning of the pipeline and a naïve Bayes classifier [40] to differentiate between high- and low-quality cells at the end of the pipeline.

### *2.5 Summary*

In general, STARsolo provides the most comprehensive options to implement different approaches for each key data preprocessing step and can be applied to data generated by different platforms. STARsolo also provides the flexibility of using exonic reads only (gene), exonic and intronic reads together (pre-mRNA), or annotated and novel spliced junctions. All these options have made STARsolo the most popularly used scRNA-seq data preprocessing pipeline in addition to Cell Ranger pipelines so far.

## 3 Data Normalization and Visualization

### *3.1 Background*

Preprocessing of scRNA-seq data generates a matrix of nUMIs, in which rows are genes, columns are cells, and each entry is the number of UMIs of each gene in each cell. The UMI count of the same gene from different cells is not directly comparable due to cell-to-cell technical variations associated with different technical factors, including sequencing depth, cell lysis, reverse transcription efficiency, molecular sampling during sequencing, and so on. Although the utilization of UMI removes variations associated with PCR amplification bias, there are still substantial technical variations in the UMI count data that need to be corrected before downstream analysis. Therefore, normalization is critically important for scRNA-seq data analysis to make the data from difference cells and samples comparable, which was also shown to have the largest impact on performance of downstream analyses [41] compared to data preprocessing and the choice of downstream analytical method. Many different normalization methods have been developed, which can be roughly divided into two groups: global scaling normalization approaches and probabilistic model-based normalization approaches.

### *3.2 Global Scaling Normalization for UMI Data*

The global scaling normalization methods estimate a global "size factor" to represent the technical variation in each cell. The UMI count of all genes in each cell is then divided by the estimated size factor for the same cell to scale the data

for normalization. Note that nUMI of different genes in the same cell are scaled by the same factor. Many normalization methods designed for bulk RNA sequencing data normalization, including TPM, TMM, DESeq2, and edgeR, fit well into this category and therefore have been used in some scRNA-seq studies. The other global scaling methods designed for scRNA-seq data include library size normalization, BASiCS [42], scran [43], census [44], and PsiNorm [45], among which BASiCS is the only method requiring spike-in controls.

To better explain different normalization methods, we use the uniform notation as follows. Suppose in total, there are $q$ genes measured in $n$ cells. Let $X_{ij}$ denote the nUMI of gene $i$ in cell $j$ as a random variable and $x_{ij}$ denote the observed realization of $X_{ij}$. The library size normalization scales the nUMI of all genes by dividing them by the total number of UMIs $\left(L_j = \sum_{i=1}^{q} X_{ij}\right)$ in cell $j$. This approach is one of the most used methods and is implemented in the Seurat R package [46].

BASiCS requires the data to have nonbiological spike-in genes, which are added into the lysis buffer at known concentration levels and therefore present at the same level in every cell. These spike-in genes provide information for BASiCS to quantify technical variation and separate it from the biological variation in the data. Suppose the first $q_0$ $(i = 1, \cdots, q_0)$ genes are biological genes and the remaining genes $(i = q_0 + 1, \cdots, q)$ are spike-in controls. BASiCS models the UMI counts in each cell $j$ using the following hierarchical model:

$$X_{ij} \mid \mu_i, v_j \sim \begin{cases} Poisson\left(\phi_j v_j \mu_i \rho_{ij}\right), i = 1, \cdots, q_0 \\ Poisson\left(v_j \mu_i\right), i = q_0 + 1, \cdots, q \end{cases} \tag{1}$$

$$\text{with } v_j \mid s_j, \theta \sim Gamma\left(\frac{1}{\theta}, \frac{1}{s_j \theta}\right), \quad \rho_{ij} \mid \delta_i \sim Gamma\left(\frac{1}{\delta_i}, \frac{1}{\delta_i}\right),$$

where $\mu_i$ is the true normalized expression level of gene $i$ in the cells, $\phi_j$ represents the differences in total mRNA content of the cells, and $v_j$ and $\rho_{ij}$ are independent random effects representing the cell-to-cell technical variability with a mean of $s_j$ and variance of $s_j^2\theta$ and the gene-specific biological cell-to-cell variability with a mean of 1 and variance of $\delta_i$, respectively. Because $\mu_{q_0+1}, \cdots, \mu_q$ are known from the spike-in genes' experimental design, $s_j$'s can be identified. $\delta_i$'s and $\theta$ can also be identified based on the variance of the biological and technical expression counts. However, because the scale of $\phi_j$'s is arbitrary, restriction is needed to make the model identifiable. This can be done by assuming that $n^{-1}\sum_{j=1}^{n}\phi_j = \phi_0$ or by reparametrizing the model in terms of $\kappa_1, \cdots, \kappa_n$ so that

$$\phi_j = \phi_0 \frac{e^{\kappa_j}}{\sum_{l=1}^{n} e^{\kappa_l}}, \quad \kappa_1 = 0. \tag{2}$$

All model parameters are assumed to have independent prior with a flat non-informative prior for the normalized expression levels $\mu_1, \cdots, \mu_{q_0}$ and conjugate informative prior for all other model parameters including $s_j$'s, $\theta$'s, $\delta_i$'s, and $\kappa_j$'s.

Bayesian inference is implemented using an adaptive Metropolis (AM) within Gibbs sampling (GS) algorithm. The estimations of $\phi_j$ and $s_j$ are eventually used to calculate the scaling factor for cell $j$.

Scran first generates pools of cells to calculate the pool-based size factors, which are then deconvoluted to yield the cell-based size factors to scale the data. Scran assumes that $E(X_{ij}) = \theta_j \lambda_{i0}$, where $\theta_j$ describes cell-specific bias and $\lambda_{i0}$ is the expected UMI count of gene $i$. So $\theta_j$ can serve as the scaling size factor for cell $j$. Define $Z_{ij} = X_{ij}/t_j$, where $t_j$ is the library size of cell $j$. We have that $E(Z_{ij}) = \theta_j \lambda_{i0}/t_j$. Consider an arbitrary set of cells $S_k$. Define $V_{ik} = \sum_{j \in S_k} Z_{ij}$ so we have $E(V_{ik}) = \lambda_{i0} \sum_{j \in S_k} \theta_j t_j^{-1}$. Also define $U_i = \sum_{j=1}^{n} Z_{ij}/n$ so we have $E(U_i) = \lambda_{i0} \sum_{j=1}^{n} \theta_j t_j^{-1}/n$. Define $R_{ik} = V_{ik}/U_i$ so we have

$$E(R_{ik}) \approx \frac{E(V_{ik})}{E(U_i)} = \frac{\sum_{j \in S_k} \theta_j t_j^{-1}}{n^{-1} \sum_{j=1}^{n} \theta_j t_j^{-1}} = \frac{\sum_{j \in S_k} \theta_j t_j^{-1}}{C} \tag{3}$$

where $C$ is a constant independent of genes, cells, or pool of cells $S_k$ and can be set to 1 since it does not affect the differences in size factor $\theta_j$. Denote the realizations of $V_{ik}$, $U_i$, and $R_{ik}$ as $v_{ik}$, $u_i$, and $r_{ik}$. Based on Eq. (3), we have that $r_{ik} = \sum_{j \in S_k} \theta_j t_j^{-1}$ for each $S_k$. By constructing different pools of cells, we can have an overdetermined system of linear equations in which $\theta_j t_j^{-1}$ for cell $j$ is represented at least once. This cell pool construction was achieved by ordering cells based on their total nUMI and divide all cells into two groups with odd and even ranking, respectively. These cells are arranged in a ring with odd ranking cells on the left and even ranking cells on the right. Starting at the 12 o'clock on the ring, a sliding window of a given size moves clockwise cell-by-cell across the ring so that each window contains the same number of cells. Cells in each window will be used to define one pool $S_k$. This cell pool construction strategy will obtain cell pools with similar library size to provide robustness to estimation errors for small $\theta_j t_j^{-1}$. Although the estimation steps of scran seem to be circuitous, the summation across cells from the constructed pools reduces the number of stochastic zeros that cause problems in some other existing normalization methods.

Census considers the relative abundance of genes on the TPM scale. The generative model of scRNA-seq predicts that when a small portion of transcripts in a cell can be captured, the signal from most detectable genes will originate from a single mRNA. Therefore, the TPMs of these genes will be very similar. Based on this prediction, Census first identifies the TPM value $x_j^*$ defined by the mode of log-transformed TPM distribution for cell $j$. Genes with detectable TPM smaller than $x_i^*$ correspond to genes whose signal originates from a single transcript. Therefore, the total number of mRNAs captured for cell $j$ is calculated as

$$M_j = \frac{1}{\theta} \cdot \frac{n_j}{F_{X_j}\left(x_j^*\right) - F_{X_j}(\epsilon)} \tag{4}$$

where $\theta$ is the expected number of cDNA molecules generated from each RNA molecule or simply the capture rate, $F_{X_j}$ is the cumulative distribution function of TPMs in cell $j$, $\epsilon$ is a TPM value below which no mRNA is believed to be present (default $\epsilon = 0.1$), and $n_j$ is the number of genes with TPM between $\epsilon$ and $x_j^*$. The capture rate $\theta$ is unknown a priori and it is highly protocol dependent and has little dependence on cell type or state. Based on estimations from existing data with spike-in controls, Census sets $\theta = 0.25$ by default. Taken together, $M_j$ is taken as the scaling size factor for cell $j$ and the Census normalized count for gene $i$ in cell $j$ is

$$\hat{Y}_{ij} = TPM_{ij} \cdot \frac{M_j}{10^6} \tag{5}$$

SCnorm does not model the cell-specific technical variations. It directly estimates the relationship between the observed un-normalized UMI counts and sequencing depth using quantile regression. Let $S_j$ denote the sequencing depth of cell $j$ and $Y_{ij}$ denote the log nonzero UMI count for gene $i$ in cell $j$. SCnorm divides the genes into $K$ different groups with substantially different UMI count-depth relationship. Within each group, the overall relationship between log un-normalized UMI count and log sequencing depth for all genes is estimated via the following quantile regression:

$$Q^{\tau_k, d_k}\left(Y_j | S_j\right) = \beta_0^{\tau_k} + \beta_1^{\tau_k} S_j + \cdots + \beta_{d_k}^{\tau_k} S_j^{d_k}$$

where $\tau_k$ and $d_k$ are chosen to minimize $\left|\hat{\eta}_1^{\tau_k} - \text{mode}_g\left(\hat{\beta}_{g,1}\right)\right|$ in which $\hat{\eta}_1^{\tau_k}$ describes the UMI count-depth relationship between the predicted expression values estimated by median quantile regression using a first-degree polynomial: $Q^{0.5}\left(\hat{Y}_j^{\tau_k} | S_j\right) = \eta_0^{\tau_k} + \eta_1^{\tau_k} S_j$.The scaling factor for cell $j$ is then defined as

$$SF_j = e^{\hat{Y}_j^{\tau_k, d_k}} / e^{Y^{\tau_k}}$$

where $Y^{\tau_k}$ is the $\tau_k$th quantile of expression counts in the $k$th group of genes. The normalized count of gene $i$ in cell $j$ is given by $Y'_{ij} = X_{ij} / SF_j$.

PsiNorm assumes that the UMI count follows the Pareto distribution, based on which the PsiNorm normalized counts of cell $j$ is

$$\tilde{x}_j = x_j \cdot \frac{q}{\sum_{i=1}^{q} \log\left(x_{ij} + 1\right)}.$$

In general, global scaling normalization methods are computational efficient and highly scalable. However, it assumes that the technical variations are cell-specific and uniform across different genes. Although UMI-based protocols in principle remove PCR amplification biases and sequencing depth, the assumption is true only if all the cDNAs are sequenced, namely, it reaches the sequencing saturation. When the sequencing is not saturated, some UMI-tagged transcripts will be lost

and systematic differences between nUMI of these lost transcripts will emerge. In addition, the UMI tags were added onto the cDNAs during reverse transcription. So, they cannot address the differences in capture efficiency before the reverse transcription or differences in the amount of mRNA content.

### *3.3 Probabilistic Model-Based Normalization for UMI Data*

Normalization methods in this section build probabilistic model for the observed UMI count data, which adopt for gene-specific technical variations and high sparsity of the scRNA-seq UMI count data. The most popular distribution used by these methods is the negative binomial distribution, which can accommodate for the overdispersion in the data. There are mainly two methods in this group: sctransform [47] and ZINB-WaVE [48]. For notations, $X_{ij}$ denotes the observed nUMI of gene $i$ in cell $j$ as a random variable and $x_{ij}$ denotes the realization of $X_{ij}$. In total, we assume that there are $q$ genes measured in $n$ cells.

Sctransform assumes that the UMI counts of each gene follow a negative binomial (NB) distribution $NB(\mu_i, \theta_i)$, for which the log-transformed mean is decided by a linear function of the sequencing depth:

$$\log\left(E\left(x_i\right)\right) = \beta_{0\mathrm{i}} + \beta_{1i}\log(m)$$

where $x_i$ is the expression of gene $i$ in all cells and $m$ is the vector of sequencing depth for all cells. Fitting this model for different gene separately results in overfitting. So after fitting this model, sctransform estimates the relationship between the estimated model parameter values and the mean gene expression across all genes using kernel regression. Based on the kernel regression curve, the model parameter estimations are then regularized and re-estimated. Let $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\theta}$ be the regularized estimation; the normalized UMI counts are calculated as

$$z_{ij} = \frac{x_{ij} - \hat{\mu}_{ij}}{\hat{\sigma}_{ij}},$$

where $\hat{\mu}_{ij} = \exp\left(\hat{\beta}_{0i} + \hat{\beta}_{1i}\log\left(m_j\right)\right)$ and $\hat{\sigma}_{ij} = \sqrt{\hat{\mu}_{ij} + \hat{\mu}_{ij}^2/\hat{\theta}_i}$.

Due to the low amount of RNAs in a single cell and the low sequencing depth per cell, some genes, especially the lowly expressed genes, may fail to be detected even if they are being expressed in the cell. This causes an excessive number of zeroes in the UMI count data and challenges in removing unwanted technical variations in the data. ZINB-WaVE [48] models the UMI count using a zero-inflated negative binomial distribution:

$$f_{ZINB}\left(x_{ij}; \mu_{ij}, \theta_{ij}, \pi_{ij}\right) = \pi_{ij}\delta_0\left(x_{ij}\right) + \left(1 - \pi_{ij}\right) f_{NB}\left(x_{ij}; \mu_{ij}, \theta_{ij}\right), \qquad (6)$$

where $\pi$ is the probability of the observed count being 0 instead of the actual count, $\delta_0(x)$ is an indicator function of whether $x$ is zero, and $\mu$ and $\theta$ are the mean and dispersion of the negative binomial distribution describing the actual count distribution. To consider various technical and biological effects, sctransform considers the following regression models:

$$\ln\left(\mu_{ij}\right) = \left(C_\mu \beta_\mu + \left(V\gamma_\mu\right)^T + W\alpha_\mu + O_\mu\right)_{ij}^T,$$

$$logit\left(\pi_{ij}\right) = \left(C_\pi \beta_\pi + \left(V\gamma_\pi\right)^T + W\alpha_\pi + O_\pi\right)_{ij}^T,$$

$$\ln\left(\theta_{ij}\right) = \zeta_i,$$

where $C_\mu$ and $C_\pi$ are known $n \times M$ matrices representing $M$ cell-level covariates, $V_\mu$ and $V_\pi$ are known $q \times L$ matrices representing $L$ gene-specific covariates, $W$ is an unobserved $n \times K$ matrix representing $K$ unknown cell-level covariates, $O_\mu$ and $O_\pi$ are known $n \times q$ matrices of offsets, and $\zeta_i$ is the gene-specific dispersion. The parameters of this model are inferred by maximizing the following penalized likelihood function to reduce overfitting:

$$\max_{\beta,\gamma,W,\alpha,\zeta} \left\{ l\left(\beta, \gamma, W, \alpha, \zeta\right) - \frac{\epsilon_\beta}{2}\left\|\beta^0\right\|^2 - \frac{\epsilon_\gamma}{2}\left\|\gamma^0\right\|^2 - \frac{\epsilon_W}{2}\|W\|^2 - \frac{\epsilon_\alpha}{2}\|\alpha\|^2 - \frac{\epsilon_\zeta}{2}\mathrm{var}\left(\zeta\right)\right\},$$

where $l(\beta, \gamma, W, \alpha, \zeta)$ is the likelihood function of the model in Eq. (6), $\beta^0$ contains coefficients for columns in $C_\mu$ and $C_\pi$ that are not constant column of ones, and $\|\cdot\|$ is the Frobenius matrix norm.

### *3.4 Dimension Reduction and Cell Clustering*

Normalized scRNA-seq data serve as basis for many downstream analyses. The first two analyses, which are also the must to-do analyses, are dimension reduction and unsupervised clustering of cells. Dimension reduction reduces noise level and helps identify outliers and understand systematic differences and variations in the data. Unsupervised clustering of cells helps identify groups of cells that are potentially different cell types or even cell subtypes within a given cell type.

Principal component analysis (PCA) has been successfully and commonly used for dimension reduction in microarray expression data, bulk RNA-seq data, and genome-wide genotyping data. However, PCA was shown to have poor performance when applied to scRNA-seq data by multiple studies [49, 50] possibly due to linear nature of PCA, excessive number of zeroes, and high technical and biological

variations in the data. Multiple methods have been developed and designed for dimension reduction in scRNA-seq data, including canonical correlation analysis (CCA) [51], independent components analysis (ICA) [52], Laplacian eigenmaps [53, 54], t-distributed stochastic neighbor embedding (t-SNE) [18, 55, 56], and uniform manifold approximation and projection (UMAP) [57–60]. Among these methods, t-SNE and UMAP are the most popular methods with UMAP preserving the global distances and t-SNE preserving local distances. Although distortions of distance exist in both methods due to representing the data using low dimensions (two to three dimensions) [61], in common practice, highly variable genes are selected, and PCA is conducted on these genes to select the top PCs with significant variations. Then t-SNE and UMAP are applied to the PCA pre-conditioned data to reduce the dimension for data visualization. Note that dimension reduction discussed here is only for data visualization and cell clustering. Many of the downstream analyses, including data imputation and differential expression analysis, are still conducted on normalized data or even the un-normalized UMI count data with their original dimensions.

Unsupervised clustering of cells is usually conducted on the reduced dimensional space or by using highly variable genes. Different types of clustering methods have been applied to scRNA-seq data, including the traditional k-means clustering and hierarchical clustering. These traditional unsupervised clustering methods are limited when applied to scRNA-seq data due to their poor scalability to the total number of cells in terms of required computational time and memory, sensitivity to outlying cells or cell clusters, and bias to identify equal-sized clusters mixing rare cell types in a larger cluster [62, 63]. Other types of methods have been developed to address these issues, including mixture model-based clustering, density-based clustering, neural network clustering, and affinity propagation clustering [63]. Among these methods, the community detection-based approaches have gained the most popularity due to their scalability and robustness to noise in the data. Instead of clustering cells close to each other based on chosen distance, community detection identifies groups of cells that are densely connected based on a k-nearest-neighbors graph constructed using the PCA reduced dimensional space or highly variable genes. The number of clusters is affected by the number of nearest neighbors in the constructed k-nearest-neighbors graph and indirect resolution parameters. Although the Louvain algorithm [46, 64, 65] is currently the most widely used approach for scRNA-seq data, there are many other community detection approaches [66] available, and some of them have demonstrated better performance in benchmarking studies [67, 68].

# 4 Dropout Imputation

## *4.1 Background*

Analysis of scRNA-seq data can be challenging due to low library size, high technical noise, and prevalent dropout events [49, 69, 70]. In scRNA-seq data, due to the tiny amount of mRNAs in each cell, some mRNAs may be totally missed during the reverse transcription and cDNA amplification step, thus cannot be detected in the sequencing step. This phenomenon is referred to as dropout event for which a given gene is observed at a moderate expression level in one cell but is not detected in another cell of the same type from the same sample, thus generating an increased sparsity in single-cell data, especially for genes with low or moderate expression [71]. These observed zero values can be the biological variation in actual expression levels among cells or the technical imperfect measure on small numbers of molecules. Dropouts lead to inaccurate assessment of gene expression levels that may mislead downstream analyses such as cell clustering and differential expression analysis, and cell trajectory inference [72]. To alleviate the increased sparsity observed in scRNA-seq data, many data imputation methods have been developed and compared [73, 74]. They can be classified into four categories [75].

## *4.2 Cell-Cell Similarity-Based Imputation*

The first category of methods evaluates cell-cell similarities and imputes dropouts in each cell using information from cells that are similar to the cell to be imputed, including kNN-smoothing [76], MAGIC [77], scImpute [78], drImpute [79], and VIPER [80]. Specifically, kNN-smoothing imputes dropouts by aggregating information from the $k$ closest neighboring cells of each cell using the stepwise k-nearest neighbors approach [76]. MAGIC constructs a cell-cell affinity matrix based on their expression profiles across genes and diffuses the gene expression values in cells with similar expression profiles for imputation [77]. scImpute infers dropout events based on the dropout probability estimated from a Gamma-Gaussian mixture model and only imputes these events by combining information from similar cells within cell clusters identified by spectral clustering [78]. drImpute defines similar cells using k-means clustering and performs imputation by averaging the gene expression values in cells within the same cluster [79]. While improving the quality of scRNA-seq data to some extent, the above methods were found to eliminate the natural cell-to-cell stochasticity which is an important piece of information available in scRNA-seq data compared to bulk RNA-seq data [80]. Instead, VIPER overcomes this limitation through selecting a sparse set of neighboring cells for imputation to preserve variation in gene expression across cells [80]. In general, the first category of imputation methods that borrow information across similar cells tends to intensify

subject variation in scRNA-seq datasets with multiple subjects, resulting in cells from the same subject to be more similar than those from different subjects.

### *4.3 Gene-Gene Similarity-Based Imputation*

The second category of methods relies on the gene-gene similarities for imputation, including SAVER [81], G2S3 [82], netNMF-sc [83], and netSmooth [84]. SAVER borrows information across similar genes instead of cells to impute gene expression using a penalized regression model [81]. G2S3 recovers gene expression by borrowing information from adjacent genes in a sparse gene graph learned from gene expression profiles across cells using graph signal processing [82]. netNMF-sc uses network-regularized nonnegative matrix factorization to leverage gene-gene interactions for imputation [83]. netSmooth smooths gene expression values by incorporating protein-protein interaction networks [84]. Both netNMF-sc and netSmooth require prior information on gene-gene interactions from RNA-seq or microarray studies of bulk tissue.

### *4.4 Gene-Gene and Cell-Cell Similarity-Based Imputation*

The third category of methods leverages information from both genes and cells. For example, ALRA imputes gene expression using low-rank matrix approximation [85], and scTSSR uses two-side sparse self-representation matrices to capture gene-gene and cell-cell similarities for imputation [86].

### *4.5 Deep Neural Network-Based Imputation*

The last category consists of machine learning-based methods, such as autoImpute [87], DCA [88], deepImpute [89], and SAUCIE [90], that use deep neural network to impute for dropout events. While computationally more efficient, these methods were found to generate false-positive results in differential expression analyses [91]. Recently, an ensemble approach, EnImpute, was developed to integrate results from multiple imputation methods using weighted trimmed mean [92].

### *4.6 G2S3*

In this section, we give a detailed presentation on the imputation method G2S3 developed by our group. G2S3 uses graph signal processing to learn a sparse gene

graph from scRNA-seq data and imputes dropouts by borrowing information from nearby genes in the graph. G2S3 first constructs a sparse graph representation of gene network under the assumption that expression values change smoothly between closely connected genes. Suppose $X = [x_1, x_2, \ldots, x_m] \in \mathbb{R}^{n \times m}$ is the observed transcript counts of $m$ genes in $n$ cells, where the column $x_j \in \mathbb{R}^n$ represents the expression vector of gene $j$, for $j = 1, \ldots, m$. We consider a weighted gene graph $G = (V, E)$, in which each vertex $V_j$ represents gene $j$ and the edge between genes $j$ and $k$ is associated with a weight $W_{jk}$.

The gene graph is determined by the weighted adjacency matrix $W \in \mathbb{R}_+^{m \times m}$. Assuming signals on the graph are smooth and sparse, G2S3 searches for an optimal adjacency matrix $W$ from the space $\mathcal{W} = \left\{W \in \mathbb{R}_+^{m \times m} : W = W^T, \mathrm{diag}(W) = 0\right\}$. To accomplish this, we optimize the objective function adapted from Kalofolias's model [93]:

$$\min_{W \in \mathcal{W}} \|W \circ Z\|_{1,1} - \mathbf{1}^T \log(W\mathbf{1}) + \frac{1}{2}\|W\|_F^2, \tag{7}$$

where $Z \in \mathbb{R}_+^{m \times m}$ is the pairwise Euclidean distance matrix of genes, defined as $Z_{jk} = \|x_j - x_k\|^2$, $\mathbf{1}$ is a vector of ones, $\|\cdot\|_{1,1}$ is the elementwise L − 1 norm, $\circ$ is the Hadamard product, and $\|\cdot\|_F$ is the Frobenius norm. In Eq. (1), the first term is equivalent to $2\mathrm{tr}(X^T L X)$ that quantifies how smooth the signals are on the graph, where $L$ is the graph Laplacian and $\mathrm{tr}(\cdot)$ is the trace of a matrix. This term penalizes edges between distant genes, so it favors a sparse set of edges between the nodes with a small distance in $Z$. The second term represents the node degree such that the degree of each gene is positive to improve the overall connectivity of the gene graph. The third term controls graph sparsity to penalize large edges between genes.

Equation (1) can be optimized via primal dual techniques [94] by rewriting it as

$$\min_{w \in \omega} \mathbb{I}_{\{w \geq 0\}} + 2w^T z - \mathbf{1}^T \log(d) + \|w\|^2, \text{ where } \omega = \left\{ w \in \mathbb{R}_+^{\frac{m(m-1)}{2}} \right\}, \tag{8}$$

where $w$ and $z$ are vector forms of $W$ and $Z$, respectively; $\mathbb{I}_{\{\cdot\}}$ is the indicator function that takes value 0 when the condition in the brackets is satisfied, infinite otherwise; $d = Kw \in \mathbb{R}^m$; and $K$ is the linear operator that satisfies $W\mathbf{1} = Kw$. After obtaining the optimal $W$, a lazy random walk matrix can be constructed on the graph as $M = (D^{-1}W + I)/2$, where $D$ is an $m$-dimensional diagonal matrix with $D_{jj} = \sum_k W_{jk}$, the degree of gene $j$, and $I$ is the identity matrix. We then obtain the imputed count matrix $X_{\text{imputed}}$ by taking a $t$-step random walk on the graph $X_{\text{imputed}}^T = M^t X^T$.

By default, G2S3 takes a one-step random walk ($t = 1$) to avoid over-smoothing. Adapted from a diffusion-based imputation method [95], we also implement hyperparameter tuning based on an objective function that minimizes the mean squared error (MSE) between the imputed and observed data, i.e., $t^* = \underset{t}{\mathrm{argmin}} \left\| M^t X^T - X^T \right\|$. A good imputation method is not expected to

deviate too far away from the raw data structure in the process of denoising. This criterion enables us to denoise the observed gene expression through attenuating noise due to technical variation while preserving biological structure and variation.

Like other diffusion-based methods, G2S3 spreads out counts while keeping the sum constant in the random walk step. This results in the average value of nonzero matrix entry decreasing after imputation. To match the observed expression at the gene level, we rescale the values in $X_{\text{imputed}}$ so that the mean expression of each gene in the imputed data matches that of the observed data. The pseudo-code for G2S3 is given in Algorithm 1.

**Algorithm 1:** Pseudo-code of G2S3

1: **Input**: $X$

2: **Result**: $X_{imputed} = \text{G2S3}(X)$

3: $Z = \text{distance}(X)$

4: $W = \min_{w \in \mathbb{R}_+^{m(m-1)/2}} \mathbb{I}_{\{w \geq 0\}} + 2w^T z - \mathbf{1}^T \log(d) + \|w\|^2$

5: $D = \text{degree}(W)$

6: $M = (D^{-1}W + I)/2$

7: $t^* = \underset{t}{\text{argmin}} \left\| M^t X^T - X^T \right\|$

8: $X^T_{imputed} = M^{t^*} X^T$

9: $X_{rescaled} = \text{rescale}(X_{imputed})$

10: $X_{imputed} = X_{rescaled}$

11: **End**

## *4.7 Methods Evaluation and Comparison*

In this section, we evaluated and compared the performance of 11 imputation methods, kNN-smoothing, MAGIC, scImpute, VIPER, SAVER, G2S3, ALRA, scTSSR, DCA, SAUCIE, and EnImpute, in recovering gene expression using three unique molecular identifier (UMI)-based datasets. The three datasets are the Reyfman dataset from human lung tissue [21], the peripheral blood mononuclear cell (PBMC) dataset from human peripheral blood [17], and the Zeisel dataset from the mouse cortex and hippocampus [55]. In Reyfman, the raw data include 33,694 genes and 5437 cells. We selected cells with a total number of UMIs greater than 10,000 and genes that have nonzero expression in more than 20% of cells. This resulted in 3918 genes and 2457 cells as the reference dataset. The PBMC dataset was downloaded from 10x Genomics website (https://support.10xgenomics.com/single-cell-gene-expression/datasets). The raw data include 33,538 genes and 7865 cells. We selected cells with a total number of UMIs greater than 5000 and genes that have nonzero expression in more than 20% of cells. This resulted in 2308 genes and 2081 cells as the reference dataset. In Zeisel, the raw data include 19,972 genes and 3005 cells. We selected cells with a total number of UMIs greater than 10,000

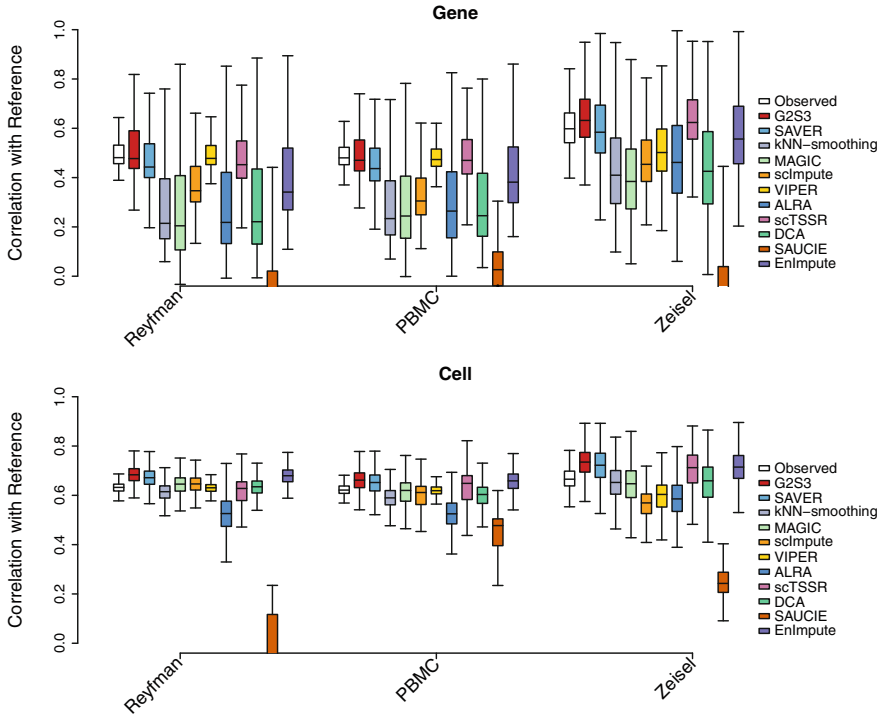


**Fig. 2** Evaluation of expression data recovery of all imputation methods by down-sampling. Performance of imputation methods measured by correlation with reference data from the first category of datasets, using gene-wise (top) and cell-wise (bottom) correlation. Box plots show the median (centerline), interquartile range (hinges), and 1.5 times the interquartile (whiskers)

and genes that have nonzero expression in more than 40% of cells. This resulted in 3529 genes and 1800 cells as the reference dataset.

In each of the three scRNA-seq datasets, the reference dataset was treated as the true expression. Down-sampling was performed to generate benchmarking observed datasets. We performed random binary masking of UMIs in the reference datasets to mimic the inefficient capturing of transcripts in dropout events. The binary masking process masked out each UMI independently with a given probability. In each reference dataset, we randomly masked out 80% of UMIs to create the down-sampled observed dataset. All imputation methods were applied to each down-sampled dataset to generate imputed data separately. We performed library size normalization on all imputed data. Figure 2 shows the gene-wise Pearson correlation and cell-wise Spearman correlation between the imputed and reference data from each dataset. The correlation between the observed data without imputation and reference data was set as a benchmark. In all datasets, G2S3 consistently achieved the highest correlation with the reference data at both gene and cell levels;

SAVER and scTSSR had slightly worse performance. EnImpute had comparable performance to G2S3 based on the cell-wise correlation but performed worse than G2S3, SAVER, and scTSSR based on the gene-wise correlation. VIPER performed well in the Reyfman and PBMC datasets but not in the Zeisel dataset based on the gene-wise correlation, although the cell-wise correlations were much lower than G2S3, SAVER, scTSSR, and EnImpute in all datasets. The other methods, kNN-smoothing, MAGIC, scImpute, ALRA and DCA, did not have comparable performance, especially based on the gene-wise correlation. SAUCIE did not have comparable performance to the other methods in all datasets. To quantify the performance improvement of G2S3, one-sided t-test was applied to compare the gene-wise and cell-wise correlations of G2S3 to those of the other methods. G2S3 had significantly higher correlations than all the other methods across three datasets for both gene-wise and cell-wise correlations ($p < 0.05$, Table 1). Overall, G2S3 provided the most accurate recovery of gene expression levels.

## 5 Differential Expression Analysis

### *5.1 Background*

Although aims vary widely across different scRNA-seq studies, one common task is to identify disease−/phenotype-associated genes [96] within each identified cell type, which provides a potential list of candidate genes for further therapeutic development and a better understanding of the disease pathogenesis. However, this task is challenging due to prevalent dropout events and substantial subject effect, or so-called between-replicate variation [97], in scRNA-seq data. We have described dropout events in Sect. 4. For subject effect, many studies have consistently shown that within the same cell type, cells of the same subject cluster together but separate well from cells of other subjects regardless of the phenotype of subjects [21, 98, 99]. For example, Fig. 3 shows a good separation between cells from the same subject in both alveolar macrophages and nature killer cells from patients with idiopathic pulmonary fibrosis (IPF) [98]. This suggests that the across-subject variation is dominant and much higher than the within-subject variation across cells, possibly due to heterogeneous genetic backgrounds or environmental exposures. DE analysis of scRNA-seq data is severely confounded by this dominant subject effect because the across-subject difference driving genes are likely to be significantly different between two groups of subjects [97, 100, 101]. In summary, it is critical to dissect subject effect from disease effect with considerations of dropout events in the DE analysis of scRNA-seq data with multiple subjects.

Sometimes, subject effect can be easily confused with technical batch effect because early-stage scRNA-seq datasets profiled freshly collected samples and thus each sample forms a separate batch. Since transcriptomic data is known to be sensitive to batches, one possible explanation for the observed large variation

**Table 1** Comparison of the gene-wise and cell-wise correlations of G2S3 and other methods in down-sampling experiments. P-values of testing the difference of correlations of G2S3 and other methods with the reference data

| Gene-wise correlation | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Datasets | SAVER | kNN-smoothing | MAGIC | scImpute | VIPER | ALRA | scTSSR | DCA | SAUCIE | EnImpute |
| Reyfman | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $2.3 \times 10^{-10}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ |
| PBMC | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $6.3 \times 10^{-4}$ | $<2.2 \times 10^{-16}$ | $1.4 \times 10^{-4}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ |
| Zeisel | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $8.1 \times 10^{-3}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ |
| Cell-wise correlation | | | | | | | | | | |
| Datasets | SAVER | kNN-smoothing | MAGIC | scImpute | VIPER | ALRA | scTSSR | DCA | SAUCIE | EnImpute |
| Reyfman | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $1.5 \times 10^{-6}$ |
| PBMC | $1.7 \times 10^{-10}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $2.4 \times 10^{-2}$ |
| Zeisel | $6.1 \times 10^{-9}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ | $<2.2 \times 10^{-16}$ |

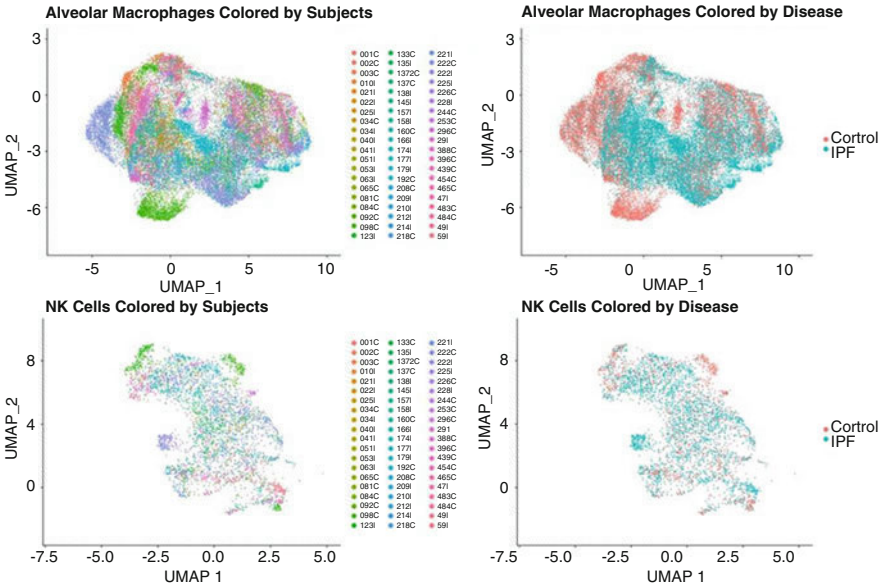


**Fig. 3** UMAPs of the alveolar macrophages (top row) and nature killer cells (bottom row) demonstrate a dominant subject effect in both cell types. In each row, figure on the left and the right show UMAPs of cells colored by subjects and disease status, respectively

across subjects is batch effect. Recent advances in preserving cells using dimethyl sulfoxide (DMSO) enabled processing multiple samples from different subjects in the same batch [102]. In the scRNA-seq data of sputum samples from patients with asthma (data unpublished), comparison of scRNA-seq data from the same sputum sample with and without DMSO preservation showed no significant difference between the fresh and DMSO data, but significant separation between different subjects was still present. This confirmed that the dominant between-subjects variation was a real biological subject effect instead of a technical batch effect. Therefore, it is inadequate to remove the across-subjects variation using batch effect adjustment tools. More importantly, removing the across-subject variations using batch effect adjustment tools will also remove the disease effect of interest because subject effect confounds with disease effect. Therefore, DE analysis of scRNA-seq data does not use the data adjusted to remove batch effect using batch effect removal tools including the integrated analysis in Seurat. All DE analysis of scRNA-seq data methods use either the normalized UMI counts or the un-normalized UMI counts.

Many DE analysis methods have been developed and compared for scRNA-seq data [103–105] although not all of them consider subject effects or dropout events. There are mainly two categories of methods depending on whether subject effect is considered.