Briefings in Bioinformatics, 2022, 1–17

https://doi.org/10.1093/bib/bbac590

Problem Solving Protocol

SimCH: simulation of single-cell RNA sequencing data by modeling cellular heterogeneity at gene expression level

Lei Sun, Gongming Wang and Zhihua Zhang

Corresponding authors: Lei Sun, School of Information Engineering, Yangzhou University, Yangzhou, No.196 Huayang West Road, Hanjiang District, Yangzhou 225127, China. Tel.: +86-158-5287-8867. E-mail: sunlei@yzu.edu.cn; Zhihua Zhang, CAS Key Laboratory of Genome Sciences and Information, Beijing Institute of Genomics, Chinese Academy of Sciences, and China National Center for Bioinformation, No.1, Beichen West Road, Chaoyang District, Beijing 100101, China. School of Life Science, University of Chinese Academy of Sciences, Beijing, China. Tel.: +86-152-0129-8782; Fax: +86-10-84097720. E-mail: zhangzhihua@big.ac.cn

Abstract

Single-cell ribonucleic acid (RNA) sequencing (scRNA-seq) has been a powerful technology for transcriptome analysis. However, the systematic validation of diverse computational tools used in scRNA-seq analysis remains challenging. Here, we propose a novel simulation tool, termed as Simulation of Cellular Heterogeneity (SimCH), for the flexible and comprehensive assessment of scRNA-seq computational methods. The Gaussian Copula framework is recruited to retain gene coexpression of experimental data shown to be associated with cellular heterogeneity. The synthetic count matrices generated by suitable SimCH modes closely match experimental data originating from either homogeneous or heterogeneous cell populations and either unique molecular identifier (UMI)-based or non-UMI-based techniques. We demonstrate how SimCH can benchmark several types of computational methods, including cell clustering, discovery of differentially expressed genes, trajectory inference, batch correction and imputation. Moreover, we show how SimCH can be used to conduct power evaluation of cell clustering methods. Given these merits, we believe that SimCH can accelerate single-cell research.

Keywords: single-cell, heterogeneity, generative model, gene coexpression

Introduction

Single-cell ribonucleic acid (RNA) sequencing (scRNA-seq) powerfully captures gene expression from both heterogeneous and homogeneous cell populations [1–3]. Constrained by relevant technologies, experimental scRNA-seq data (count matrix) may contain various kinds of technical noise, which can be alleviated by batch correction [4–6], imputation [7] and spike-in control [8, 9]. After preprocessing, several types of downstream analysis, e.g. cell clustering [10], testing differential expression (DE) genes [7], trajectory inference (TI) [11] and gene coexpression analysis [7, 12, 13], can be conducted to achieve significant biological discoveries.

In spite of the boom in the scRNA-seq analysis, computational methods must still be benchmarked comprehensively on both experimental (real) and simulated (synthetic) data [14]. Some experimental datasets can serve as reliable ground truth (gold standard), but they are quite rare and might not be suited to specific area of research [10]. To address this challenge, simulation can provide a fast, economical and scalable way to assess computational methods [1, 15], especially when the gold standard is lacking. Moreover, simulation can generate different sizes of sample data to evaluate the power of specific analytic methods, which can be useful for guiding the scRNA-seq experimental design [16–18].

Current simulators for scRNA-seq can be mainly divided into three classes. The first aims to model the sequencing process, e.g. Minnow [19]. The second models the statistical readout of raw sequencing data, e.g. Splat [15], SPARSim [20], SPsimSeq [21], bayNorm [22], scDesign [17], cscGAN [23], POWSC [18], muscat [24] and scDesign2 [25]. The third models biological processes, such as gene regulation networks (GRNs) and/or lineages, e.g. SERGIO [26], BEELINE [27], PROSSTT [28], SymSim [29] and ESCO [30]. Compared to the first and the third classes, simulators in the second class are based on simpler assumptions that make them faster and more flexible. However, it is still challenging to make the simulation comprehensive, even for the second class of simulators [14]. This can be attributed to the difficulty of modeling the high-dimensional and heterogeneous experimental data [30], maintaining gene coexpression information [31], modeling dropout events [32] and benchmarking for diverse computational methods. To address these problems, we herein propose a simulation tool, termed as Simulation of Cellular Heterogeneity (SimCH), based on a count matrix generative framework that can support reliable and unbiased benchmarking and evaluation of scRNA-seq computational methods, including unsupervised cell clustering, differentially expressed genes discovery, TI, batch correction and imputation.

We assessed the performance of SimCH on several types of datasets and demonstrated its remarkable power to benchmark and evaluate the scRNA-seq computational methods. By introducing the Copula framework, SimCH retains the gene coexpression

Lei Sun is an associate professor at Yangzhou University and is a postdoctoral researcher at the Beijing Institute of Genomics, Chinese Academy of Sciences, and China National Center for Bioinformation. His research interests include bioinformatics and systems biology.

Gongming Wang was formerly a Master's degree student in signal and information processing at Yangzhou University. He is now an engineer at the China Unicom Software Research Institute Jinan Branch. His research interests include mathematical modeling and data mining.

Zhihua Zhang is a professor at the Beijing Institute of Genomics, Chinese Academy of Sciences, and China National Center for Bioinformation. His current research interests include developing novel computational methods to reveal the structure and function of genome 3D architecture in mammals.

Received: July 25, 2022. **Revised:** November 8, 2022. **Accepted:** December 2, 2022

information of experimental data. We highlight that SimCH can model dropout events embedded in either unique molecular identifier (UMI)-based or non-UMI data by negative binominal (NB) distribution and NB followed by zero-inflation (NBZI), respectively. Furthermore, SimCH can simulate gene DE, differentiation paths, batches or loss of gene expression, all presumed to be parameters forming ground truth for benchmarking the computational methods noted above. Moreover, SimCH can generate varying magnitude of synthetic data for the power evaluation of specific computational methods.

Results

Framework of SimCH simulation

SimCH consists of three basic modes, i.e. SimCH-flex, SimCH-fit and SimCH-copula, as well as an extended mode, SimCH-ext (Figure 1), providing flexible magnitude configuration, good fit to experimental gene expression, gene coexpression preservation and complex simulation, respectively. Depending on the purpose of study, users may choose one of the three basic modes to estimate basic parameter settings from a homogeneous dataset (Figure 1).

The SimCH-flex mode can generate simulated data with varying gene number, cell number and sequencing depth (Methods) [15]. The flexibility of SimCH-flex is achieved by building two logarithmic Gaussian mixture model (GMM) distributions to model gene mean expression and size factor, respectively, and a scaled inverse chi-square distribution to model the biological coefficient of variation (BCV). The SimCH-fit mode can generate data to mimic the gene expression distribution of experimental data with varying cell number and sequencing depth (Methods). The SimCH-copula mode aims to retain the coexpression pattern among genes in the experimental data by the Gaussian Copula framework (Methods). Importantly, a feature in SimCH modes allows users to set zero-inflation, or not, after NB modeling. This feature was inspired by the increasing number of studies showing that data sourced from UMI-based protocols (e.g. Droplet-based sequencing) can be modeled very well using NB distribution without zero-inflation [32].

The SimCH-ext mode is designed to perform complex simulation for benchmarking the computational tools of cell clustering, DE gene detection, batch correction and TI (Figure 1, Methods). Users can reset several parameters of SimCH-ext to mimic multiple groups, batch effects and differentiation paths, respectively. For the multi-group simulation, it is assumed that multiple cell groups evolve from an ancestral cell group estimated from the experimental data with a proportion of DE genes having mean expression shift controlled by multiplicative factors (MFs). To simulate batch effects, the mean expression of all genes in the same batch of cells is assumed to shift with the same variation. When simulating differentiation paths, DE genes are randomly classified into linear and nonlinear categories, which evolve along differentiation paths by linear and nonlinear styles, respectively (Methods).

Meanwhile, basic simulation can optionally generate a synthetic 'true' count matrix without loss of gene expression (e.g. dropouts), as well as the count matrix with expression loss modeled by Poisson sampling, which provides a reference for users to benchmark imputation methods.

Benchmarking results could guide users to select a suitable tool or pipeline to achieve significant biological discoveries (Figure 1). In addition, users can assess the performance of specific methods with varying magnitude (i.e. cell number and sequencing depth). Such power evaluation can guide users toward choosing suitable cell number and sequencing depth in scRNA-seq experimental design to capture the biological signals more accurately (Figure 1).

SimCH gives a true representation of scRNA-seq data

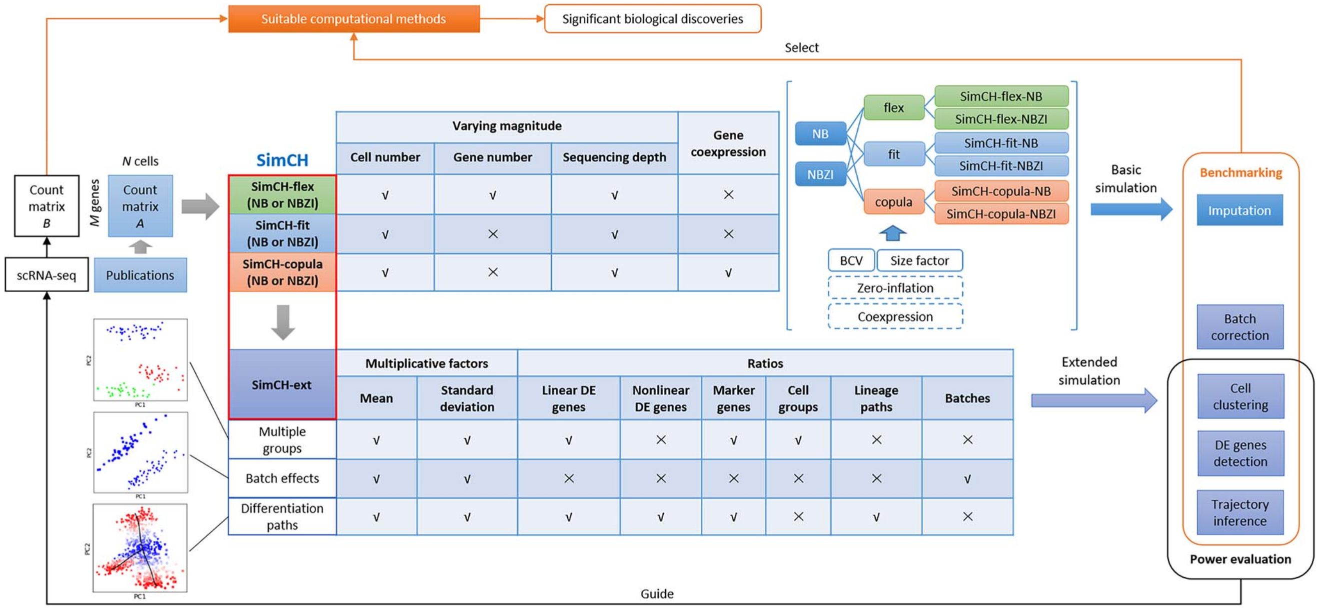
To confirm the reliability of SimCH in recovering properties of experimental data, we compared six SimCH sub-modes, namely SimCH-flex-NB, SimCH-fit-NB, SimCH-copula-NB, SimCH-flex-NBZI, SimCH-fit-NBZI and SimCH-copula-NBZI, with two state-of-the-art tools, namely Splat, v1.13.0 (splat, splat-dropout and kersplat) and scDesign2, v0.1.0, on 13 datasets. The evaluation metrics include mean, variance and mean–variance relationship of the gene expression, library size and dropouts (zeros per cell, zeros per gene and mean-zeros per gene) and Pearson correlation coefficient (PCC) matrix of coexpression genes. Performance was indexed by mean absolute error (MAE) (Figure 2A and B).

In all datasets we tested, the SimCH and scDesign2 simulators performed better than Splat in all eight metrics. In terms of retaining the gene coexpression, the Copula-based simulators, i.e. SimCH-copula-NB, SimCH-copula-NBZI and scDesign2, ranked at the top by PCC for all datasets (Figure 2A, and Additional file 1: Supplementary Figures S1–S3). SimCH outperformed scDesign2 on UMI-based datasets, while scDesign2 performed better on non-UMI datasets, except for the metrics of mean–variance relationship and library size (Figure 2A and B) [25]. In comparison within the six SimCH sub-modes, the three NB-based simulators outperformed the NBZI-based simulators in most metrics on UMI-based datasets (Tung, Haber and Zeisel), while the latter simulators outperformed the former simulators on non-UMI datasets (Ziegenhain_SmartSeq2, Camp and Li), which is consistent with the assumption that NB-based simulators are more suitable for modeling UMI-based data, while NBZI-based simulators are better for non-UMI data. Specifically, dropouts (zeros per gene and zeros per cell) are modeled very well by the NB and NBZI models of SimCH (except for SimCH-flex), corresponding to UMI-based and non-UMI data, respectively. Since the SimCH-flex sub-modes (SimCH-flex-NB and SimCH-flex-NBZI) randomly generate genes, as well as their expression, without corresponding to the genes of the experimental data, they represented poorer fitting results to the experimental data in comparison with the other SimCH sub-modes and scDesign2. Nevertheless, users can flexibly set the number of synthetic genes when using SimCH-flex.

Last, we assessed the central processing unit (CPU) time needed for all simulators on several datasets (Methods). We found that all simulators could finish simulation within hours (Additional file 1: Supplementary Figure S4).

Performing SimCH for benchmarking imputation methods

Imputation methods are utilized to recover missing gene expression, including dropouts of the experimental count matrix. Previous benchmarking work [34,35] mainly focused on evaluating how the imputation methods preserve statistical distribution and biological structures (e.g. cell clusters and DE genes) of synthetic data generated by simulators, such as Splat [15] and powsimR [16], which, however, lacked gene coexpression information. Here, we illustrate how to use SimCH to benchmark two imputation methods, namely DeepImpute [36] (v1.2) and SAVER [37] (v1.1.2), to preserve coexpression (Figure 3A). DeepImpute is a deep neural network-based imputation method [36], while SAVER is a method that takes advantage of the gene–gene relationship to recover the



SimCH	Varying magnitude			Gene coexpression
	Cell number	Gene number	Sequencing depth	
SimCH-flex (NB or NBZI)	√	√	√	×
SimCH-fit (NB or NBZI)	√	×	√	×
SimCH-copula (NB or NBZI)	√	×	√	√

SimCH-ext	Multiplicative factors		Ratios					
	Mean	Standard deviation	Linear DE genes	Nonlinear DE genes	Marker genes	Cell groups	Lineage paths	Batches
Multiple groups	√	√	√	×	√	√	×	×
Batch effects	√	√	×	×	×	×	×	√
Differentiation paths	√	√	√	√	√	×	√	×

Figure 1. Framework of SimCH simulation. SimCH can simulate on an scRNA-seq count matrix *A* of *M* genes × *N* cells sourced from publications. Basic simulation can first be performed on a homogeneous dataset by using one of three basic modes of SimCH (i.e. SimCH-flex, SimCH-fit and SimCH-copula), depending on the settings of varying magnitude and whether to retain gene coexpression. Specifically, each basic mode contains two sub-modes built on the underlying NB and NBZI models, respectively. Simulation results can be directly applied to benchmark imputation methods. Based on the parameters estimated from the basic modes, users can further run SimCH-ext with prespecified parameters to simulate multiple groups, batches or differentiation paths to benchmark computational methods for cell clustering, DE genes detection, TI and batch correction. Based on benchmarking results, a suitable method can be selected and applied to count matrix *B* of a specific scRNA-seq experiment to achieve significant biological discoveries. In addition, SimCH can be used to perform power evaluation on specific analytic methods to guide the experimental design of scRNA-seq.

true expression [37]. First, SimCH was used to generate synthetic 'true' and noisy count matrices before and after gene expression loss based on the Tung dataset (864 cells and 19 027 genes) [38]. Since the dataset is UMI based, the basic simulation was conducted by the SimCH-copula-NB sub-mode which models the gene expression loss by Poisson sampling without zero-inflation. Meanwhile, the gene–gene correlation matrix of experimental data was retained during simulation as the 'true' coexpression pattern. As a result, the 'true' gene coexpression pattern was blurred by the loss of gene expression, as seen from either PCC or Kendall rank coefficient of gene expression (first two columns of Figure 3A). Then, the matrix with gene expression loss was processed by DeepImpute and SAVER with default parameters, respectively. After that, coexpression of the two imputed matrices was compared with that of synthetic 'true' count matrix, respectively, as evaluated by the mean squared error (MSE). Results of this experiment demonstrate that SAVER outperformed DeepImpute in the recovery of gene coexpression on the homogeneous Tung dataset (Figure 3A). Moreover, users can use similar SimCH-based strategies to benchmark imputation methods systematically on different types of scRNA-seq datasets, including homogeneous and heterogeneous cell populations. In addition, SimCH can benchmark imputation methods in terms of preserving statistical distribution and biological structures of experimental data, and affecting downstream analyses, as performed in previous benchmarking studies [34,37].

Simulation for benchmarking cell clustering methods

In addition to biological ways of tagging cell identities [39], simulation methods can generate scRNA-seq data with prespecified cell groups with more flexibility [40]. Then, cell clustering methods can be evaluated in terms of recovering the prespecified cell groups using such metrics as adjusted Rand index (ARI) [41], adjusted mutual information (AMI) [42], Shannon entropy [43], normalized mutual information [44], Fowlkes-Mallows index and Jaccard index [45]. Here, we demonstrate how to benchmark two well-known cell clustering methods, SC3 [46] (v1.21.0) and Seurat [47] (v4.0.3), by SimCH (Figure 3B). First, the basic parameters were estimated from the Klein dataset (239 cells and 25 435 genes) [48] using SimCH-copula-NB. Second, we constructed two reference datasets with different multi-group settings (groups-A and groups-B) by SimCH-ext. Specifically, groups-A and groups-B both had the same DE genes ratio (0.2), mean of MF (0), marker genes ratio (0%) and three cell groups (group1, group2 and group3) where cell counts were set as 74, 79 and 86, respectively. Standard deviations (SDs) of MF of groups-A and groups-B were set as 0.2 and 0.3, respectively, for simulating different levels of group differentiation. Third, synthetic data with prespecified cell groups were input to SC3 and Seurat, respectively. SC3 was performed with default parameters, while several parameters (normalization.method = 'LogNormalize', scale.factor = 10,000, selection.method = 'vst', nfeatures = 2000, dims = 1:15 and resolution = 0.95) were set for Seurat. Finally, the performance of SC3 and Seurat in recovering the prespecified groups of the two scenarios was evaluated by ARI and AMI. Results show that SC3 outperformed Seurat when the SD of MF was small (e.g. 0.2). In contrast, both clustering methods achieved the same good performance (AMI = 1; ARI = 1) when the SD of MF was large enough (e.g. 0.3) (Figure 3B).

Performing SimCH for benchmarking DE analysis methods

Once the cell types of a cell population are determined, the next step is to identify the marker genes differentially expressed between cell groups [13]. Previous simulators that can benchmark

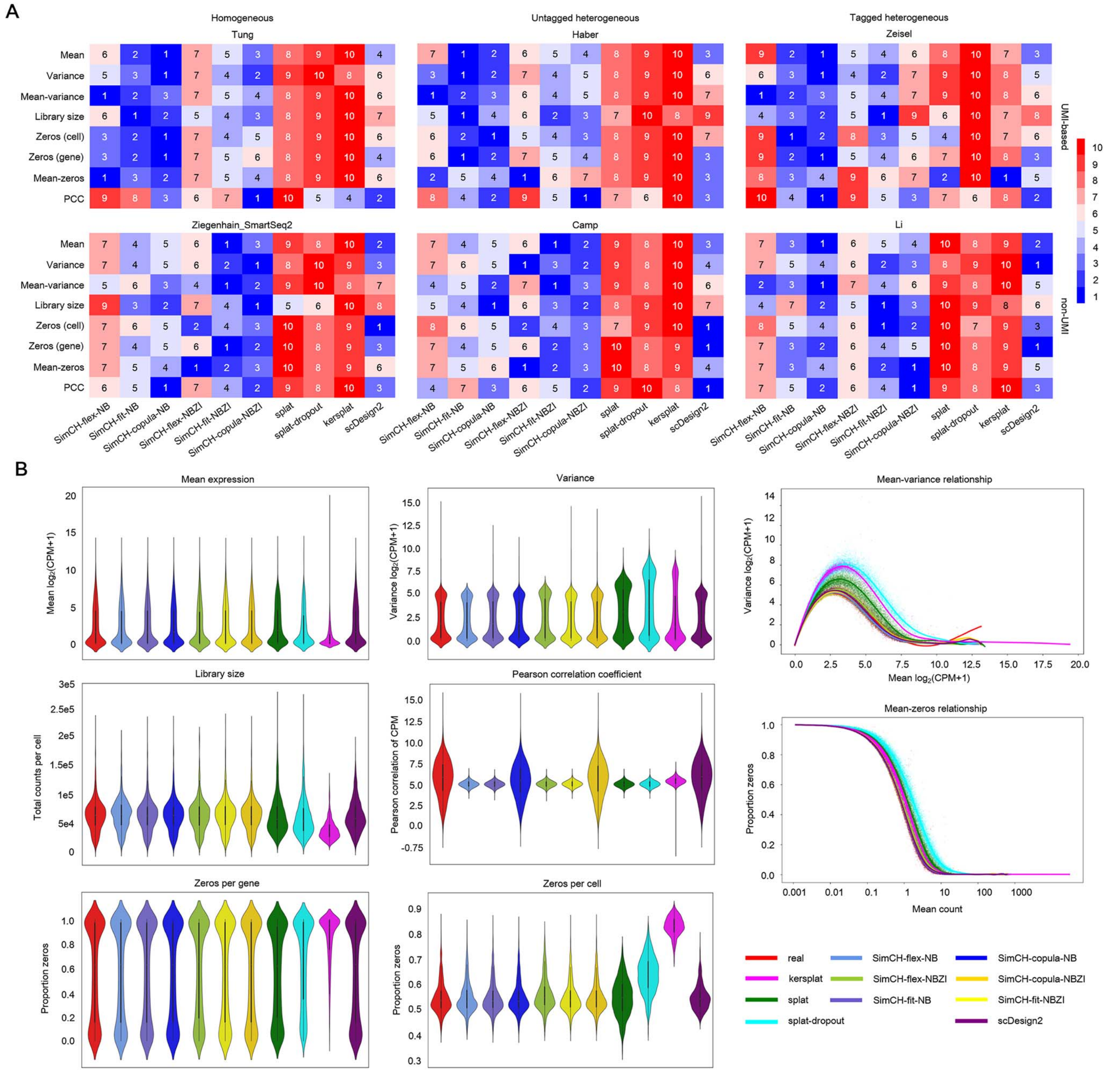


Figure 2. Comparison of 10 scRNA-seq simulators for count matrix generation. (**A**) Rank of 10 simulators evaluated by MAE of eight metrics on six representative datasets. The upper three plots are for UMI-based data, and the bottom plots are for non-UMI data. The column plots from left to right correspond to homogeneous, untagged heterogeneous and tagged heterogeneous populations, respectively. The performance of simulators in each metric is ranked from good to bad in numerical order (1–10). (**B**) Detailed comparison of marginal distribution among the 10 simulators in the eight metrics on the Tung dataset. CPM denotes counts per million.

DE testing methods include, for example, Splat [15], powsimR [16], SPsimSeq [21], POWSC [18] and scDD [49]. Similarly, SimCH-ext can simulate the datasets of two groups with prespecified DE genes for evaluating DE analytic tools. Evaluation metrics include receiver operating characteristic (ROC), true positive rates (sensitivities), false positive rates (FPRs), precision, recall, accuracy, F1 score and FPR. The current SimCH version just supports setting traditional DE genes [49]. Here, we demonstrate how to benchmark two DE gene detection methods, DESeq2 [50] (v1.33.2) and edgeR [51] (v3.34.0), based on SimCH (Figure 3C). First, the basic parameters were still estimated from the Klein dataset (239 cells and 25 435 genes) [48]. Second, we constructed two reference datasets with different DE settings (DE-A and DE-B). Specifically, both DE-A and DE-B had the same DE genes ratio (0.2), mean of MF (0), marker genes ratio (0%) and two cell groups (group1 and group2) with 101 and 138 cells, respectively. SDs of MF of DE-A and DE-B were set as 0.2 and 0.3, respectively. Third, the simulated data generated by SimCH-ext were fed to DESeq2 and edgeR, respectively, for DE genes discovery. Then, the predicted DE genes with varying thresholds (P-values) were compared to the prespecified DE genes and evaluated by the ROC curves as well as area under the curve (AUC) scores (Figure 3C). As a result, edgeR presented better performance than DESeq2 on either DE-A or DE-B datasets (Figure 3C).

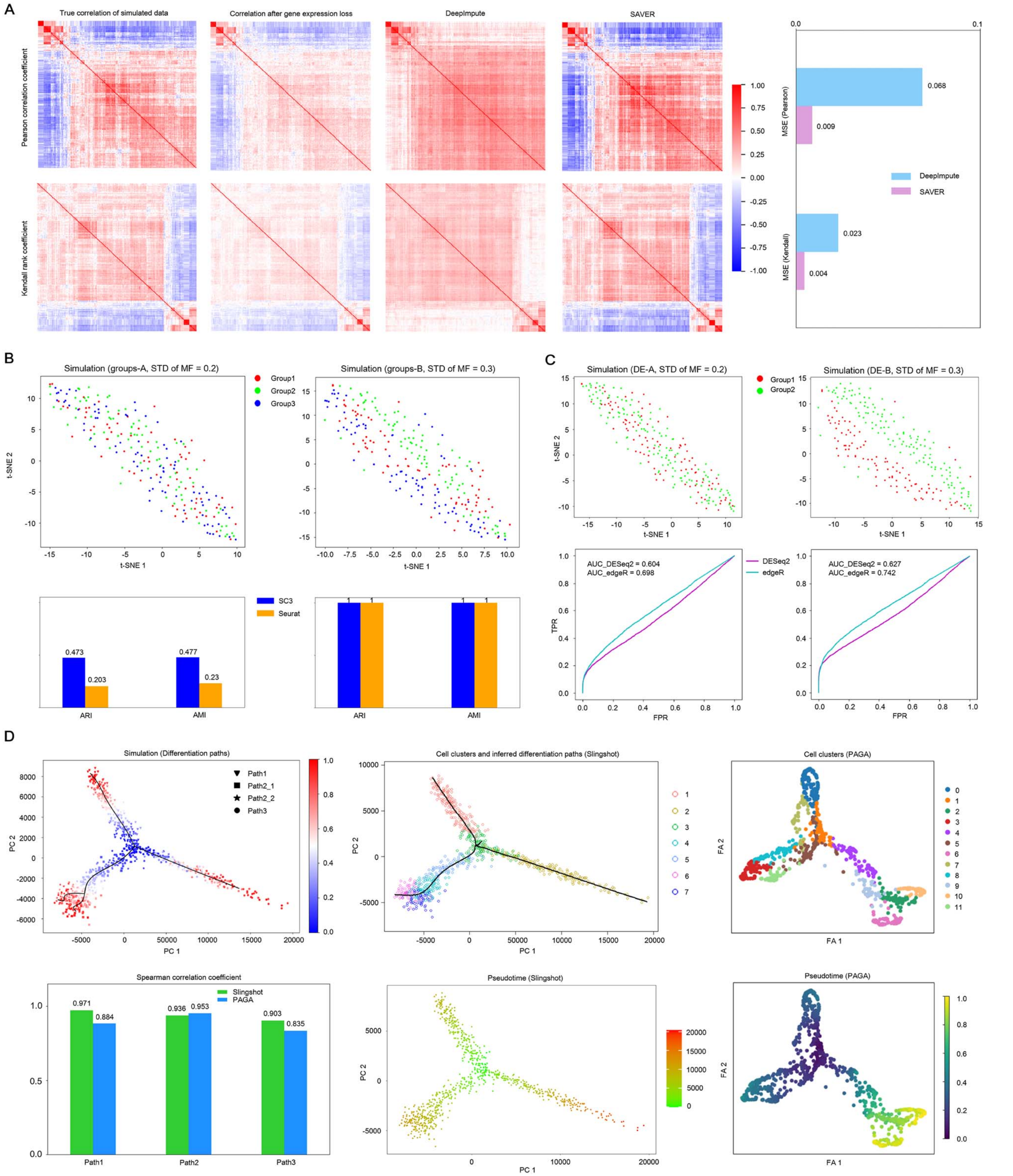


Figure 3. Benchmarking of scRNA-seq computational methods by SimCH. (**A**) Benchmarking of imputation methods (e.g. DeepImpute and SAVER). SimCH was used to generate two count matrices before and after gene expression loss based on the Tung dataset. DeepImpute and SAVER (third and fourth columns) were compared in terms of recovering the true gene–gene correlation (first column) from the noisy matrix with gene expression loss (second column). MSEs between the simulated 'true' correlation (Pearson or Kendall) and that of the imputed data (third and fourth columns) were calculated for the evaluation (bar plot on the right). (**B**) Benchmarking of cell clustering methods (e.g. SC3 and Seurat). SimCH was used to simulate three cell groups (group1, group2 and group3) with two different multi-group settings (SD of MF = 0.2 for group-A and SD of MF = 0.3 for group-B) based on the Klein dataset (upper *t*-SNE plots). Evaluation metrics are ARI and AMI (bottom two bar plots). (**C**) Benchmarking of DE detection methods (e.g. DESeq2 and edgeR). SimCH was used to simulate two cell groups (group1 and group2) with two different DE settings (SD of MF = 0.2 for DE-A and SD of MF = 0.3 for DE-B) based on the Klein dataset (upper *t*-SNE plots). Evaluation metric is the ROC curve with AUC score (bottom plots). (**D**) Benchmarking of TI methods (e.g. Singshot and PAGA). SimCH was used to simulate bilevel differentiation paths based on the Klein dataset (upper left). The trajectories and pseudotime reconstructed by Slingshot and PAGA are shown in the second and the third columns of the panel, respectively. The performance of each path is evaluated by the Spearman correlation coefficient (bottom left bar plot). FA denotes ForceAtlas2 [33].

Performing SimCH for benchmarking TI methods

In developmental biology, it is challenging to reconstruct cellular lineage along time, namely TI. Based on the cell identities dissected by cell clustering methods, TI methods are further used to reconstruct cell differentiation paths depicting the transition between cell identities by low-dimensional surfaces, and then arranging cells along paths with a timescale, termed as pseudotime. Previous simulation-based TI benchmarking [11] used such simulators as dyngen [52], PROSSTT [28] and Splat [15]. Benchmarking metrics included Hamming–Ipsen–Mikhailov, F1 score between branch assignments and cell position (correlation between geodesic distances) [11]. Here, we demonstrate how to benchmark two TI methods, Singshot [53] (v2.1.0) and PAGA [54] (Scanpy, v1.8.1), by SimCH (Figure 3D). First, basic parameters were still estimated from the Klein dataset [48]. Second, SimCH-ext was used to simulate the datasets of bilevel differentiation paths with several parameters (DE genes ratio = 0.2, marker genes ratio = 0, ratios of path1, path2_1, path2_2 and path3 = 0.3, 0.2, 0.2 and 0.3). Path2_1 and path2_2 are two sub-level paths of path2. Third, the simulated data were preprocessed and then input to Slingshot and PAGA, respectively, for TI. Fourth, Singshot and PAGA were compared in recovering the prespecified trajectories, as well as the pseudotime, and the results were evaluated by the Spearman correlation coefficients of the mapped cells between each pair of the prespecified path and the inferred path (Figure 3D). Our evaluation showed that Slingshot presented better performance than PAGA in reconstructing the unbranched paths (path1 or path3), while the result was reversed when inferring the bifurcated trajectory (path2).

Performing SimCH for benchmarking batch correction methods

As another type of confounding factor, batch effects sourced from different time points or places of library preparation or sequencing may introduce serious variation to gene expression quantification [7]. To tackle this problem, several batch correction methods, such as ComBat [4], svaseq [5] and mnnCorrect [6], were proposed. Previous benchmarking of batch correction methods used such simulators as Proper [55] and Splat. SimCH can also be used to benchmark the batch correction methods (Figure 1). Based on the basic parameters estimated from a homogeneous count matrix, users can perform SimCH-ext with batch-related parameters. Then, different batch correction methods can be used to correct the batch effects embedded in synthetic data. Batch correction methods can be further evaluated using such metrics as Silhouette [56] and FPR for highly variable genes [57].

Performing SimCH for power evaluation

In addition to the benchmarking to select a suitable method for specific processing or analysis, SimCH can evaluate the statistical power of a specific method, thereby guiding scRNA-seq design (Figure 1). Here, we demonstrate the use of SimCH to evaluate the power of two cell clustering methods (SC3 [46] and Seurat [47]) in dissecting cell types based on the Zeisel dataset [58] (Figure 4). Since the cell population of Zeisel data has been annotated into nine groups [58], two simulation strategies, namely extended and independent multi-group simulation, are available for this evaluation. For extended multi-group simulation, each point of the trends, shown in Figure 4A, was sourced from SimCH-ext with basic parameters estimated from the third group of the Zeisel dataset by SimCH-copula-NB. Simulated multi-group data had the same number of cell groups (nine) and population ratio (0.0965:0.1298:0.3155:0.2729:0.0326:0.0582:0.0659:0.0087:0.0199) as the Zeisel data. The ratio of DE genes was set as 0.7, and the SD of MF was set as 0.2. For independent multi-group simulation, each point of the trends, shown in Figure 4B, was sourced from the basic simulation of SimCH-copula-NB with parameters estimated from each sub-group of the nine Zeisel groups. Specifically, the Leiden clustering method [59] was used to classify each of the nine Zeisel groups into sub-groups, which were then simulated independently by SimCH-copula-NB and were subsequently combined into nine groups as the synthetic data. When evaluating ARI trends with the number of cells (left side of Figure 4A and B), the average count per cell was fixed at 13 950. When evaluating ARI trends with the average total count per cell (right side of Figure 4A and B), the cell number was fixed at 3005. Meanwhile, we also evaluated the ARI trends of SC3 and Seurat based on randomly sampled cells of the experimental data (dotted lines on the left side of Figure 4B).

Results show that SC3 and Seurat could present similar performance trends in cell clustering (Figure 4). Given the fixed cell number, their performance increased according to count per cell (right side of Figure 4). Although ARI trends based on the extended simulation presented higher correlation between the count per cell and clustering performance, ARI trends based on the independent simulation are likely to peak when climbing as the sequencing depth (right side of Figure 4B). The performance of SC3 and Seurat also increased according to cell number in the extended simulation (left side of Figure 4A), while the trend slightly fluctuated based on either independent multi-group simulation or sampled real data (left side of Figure 4B). In other words, the left plot of Figure 4B shows that cell number might not affect clustering performance, given likely sufficient sequencing depth (average count per cell = 13 950). In contrast, extended simulation based on the third group of the Zeisel dataset gave an increasing trend according to cell number (left side of Figure 4A). However, it is worth noting that extended simulation based on the third group only preserved partial information of the whole Zeisel data. Meanwhile, the nine cell groups extended from a small number of cells still had tight coupling, making it difficult to classify them in the beginning of Figure 4A (left). In such circumstance, cell number increment could provide more information for clustering the cells into distinct groups, as represented by the increasing curves on the left side of Figure 4A. Therefore, clustering methods could present different performance trends if users conduct different simulation strategies, which should be noted in the power evaluation. Given the limited budget, we envision a tradeoff between sequencing more cells and increasing sequencing depth per cell. Both simulation strategies shown here agree with increasing the sequencing depth in order to improve clustering performance (right side of Figure 4). However, whether increasing cell number can achieve the same goal remains controversial.

To evaluate the facticity and feasibility of the two simulation strategies, we added ARI scores, based on real Zeisel data, to the comparison (stars in Figure 4). Obviously, extended multi-group simulation may inflate the statistical power of clustering methods under specific cell number or sequencing depth as real performance is below the trend lines (Figure 4A). In contrast, independent multi-group simulation can generate more realistic data (Figure 4B), but the contradiction is that we cannot commonly obtain the cell-type composition as prior knowledge for performing the independent multi-group simulation.

Gene coexpression is associated with cellular heterogeneity

Even if a few simulators added gene coexpression of the experimental data into their simulation, they did not comprehensively give the reason for keeping such information. Here, we demonstrate the importance of retaining gene coexpression, which could encode the heterogeneous structures of cells at the gene expression level. Intrinsically, GRNs mostly determine the dynamic characteristics of gene expression in each cell, while a group of homogeneous cells may share the same GRNs underlying the steady stages of gene expression [26]. Therefore, we hypothesized that a heterogeneous cell population may have different GRNs corresponding to different cell groups. Since gene coexpression, when captured by scRNA-seq, is a snapshot of the outcome of GRNs, its association with the heterogeneity of different cell groups is expected to be observed. To prove our arguments, we first obtained gene coexpression patterns of 200 representative genes (top expressed genes) for each cell group of the Zeisel dataset. Results show that these cell groups had different coexpression patterns (Figure 5A), implying that the gene expression of every cell group could be regulated by specific GRNs. Furthermore, we designed an experiment to explore the correlation between gene coexpression and cellular heterogeneity via SimCH simulation. Specifically, we increasingly added the coexpressed genes from 0 to 4000 in the simulation and investigated the marginal distribution of real and simulated data simultaneously in the reduced space by principal component analysis (PCA) [60] and *t*-distributed stochastic neighbor embedding (t-SNE) [61], respectively. Results show that the similarity between the real and simulated data increases according to the preservation of gene coexpression (Figure 5B and C), which was evaluated by the median of local inverse Simpson's index (LISI) [62]. Meanwhile, the results strongly manifest that our SimCH-copula, which maintain gene coexpression, can preserve the heterogeneous structure of the real cell population.

Discussion

With so many competitive computational tools available, benchmarking becomes important in selecting a suitable method for specific scRNA-seq data analysis. Although several simulators based on generative models have been proposed recently, none can meet all the functional criteria noted at the beginning of this paper. In practice, users have to choose different simulators for different benchmarking scenarios [63], e.g. cell grouping, DE analysis, TI and imputation. In this paper, we focused on the statistical simulation of scRNA-seq count matrix without presuming complex biological processes.

Current scRNA-seq technologies can be classified into UMI-based and non-UMI categories, or full-length and tag-based or droplet-based and plate-based. To the best of our knowledge, most full-length protocols are non-UMI, while tag-based protocols are UMI-based [64], and droplet-based protocols are UMI-based. Plate-based protocols can be UMI-based or non-UMI methods. Previous studies demonstrated that various scRNA-seq protocols differ from each other in sensitivity, accuracy, precision, power and efficiency [65] and that different computational methods were recommended for processing and analyzing the sequencing data sourced from different protocols [7]. To assist the neutral and reliable benchmarking of computational methods, we have proposed SimCH to generate simulated data. SimCH can faithfully mimic several marginal distributions from different types of experimental data (homogeneous or heterogeneous, UMI-based or non-UMI), which guarantees the reliability of further extended simulation. To reliably learn parameters from different types of scRNA-seq data, SimCH provides users with several simulation sub-modes, namely SimCH-flex-NB, SimCH-fit-NB and SimCH-copula-NB, for the UMI-based data, and Sim-flex-NBZI, SimCH-fit-NBZI and SimCH-copula-NBZI, for the non-UMI data. If users want to retain gene coexpression information of experimental data, they can selectively use the SimCH-copula-NB or SimCH-copula-NBZI sub-modes. In addition, users can flexibly set the magnitudes of synthetic data (number of cells, number of genes and sequencing depth) as well as other modeling parameters in SimCH.

To compare different simulators, the similarity between the experimental and simulated data based on several characteristics was chosen as the evaluation metrics. Comparison results demonstrate that SimCH outperformed other simulators in recovering several statistics of either homogeneous or heterogeneous, UMI-based or non-UMI experimental data, while scDesign2, another simulator, presented competitive performance on non-UMI datasets. In fact, the performance of Splat is acceptable in practice. In our comparison, the performance of Splat ranked last as we did not add other simulators with lower ranking than Splat in the benchmark. And this metric has little impact on its wide use because of its modularity, ease of use, shorter running time and complete functions. Nonetheless, Splat cannot preserve gene coexpression. scDesign2 presented competitive performance in comparison to SimCH based on its flexible model-selection mechanism, which could dramatically improve the fitting results by selecting an optimum model from a set of models for each gene. However, its higher performance in the eight metrics we specified sacrifices its running time (Additional file 1: Supplementary Figure S4). Despite the flexibility of its adaptive model-selection, this same mechanism prevents the simulator from revealing the relationship between UMI-based data and the NB model, making its statistical models less interpretable. As illustrated in the original paper describing scDesign2 and our results, scDesign2 also had poor performance in mean–variance relationship and library size. Besides, scDesign2 can only be used to evaluate cell clustering methods without other functions, which will limit its use. Even though we discussed much about the fitting results, it should be noticed that a good fit to the experimental data at the marginal distribution level may not imply that a simulator can capture the underlying structure of scRNA-seq data since the experimental data are just a snapshot of the dynamic processes of biological variation and technical noises at the gene expression level [66]. Moreover, the impact of detailed experimental characteristics (e.g. different protocols) specific to the simulators will be evaluated systematically in the future.

From the basic parameters estimated from a homogeneous dataset, the SimCH-ext mode can perform extended simulation, cell groups, batches and differentiation paths, which can then be used to benchmark computational methods, including cell clustering, DE gene testing, TI, batch correction and imputation. The availability of SimCH for benchmarking was assessed by several cases presented above. However, in addition to preparing a reliable synthetic dataset, other parts of benchmarking work must be carefully designed and implemented to provide accurate, unbiased and informative results [10], the elaboration of which is, however, beyond the scope of this study.

Previous simulators like Splat, as noted above, do not consider gene coexpression in the simulation, which would affect

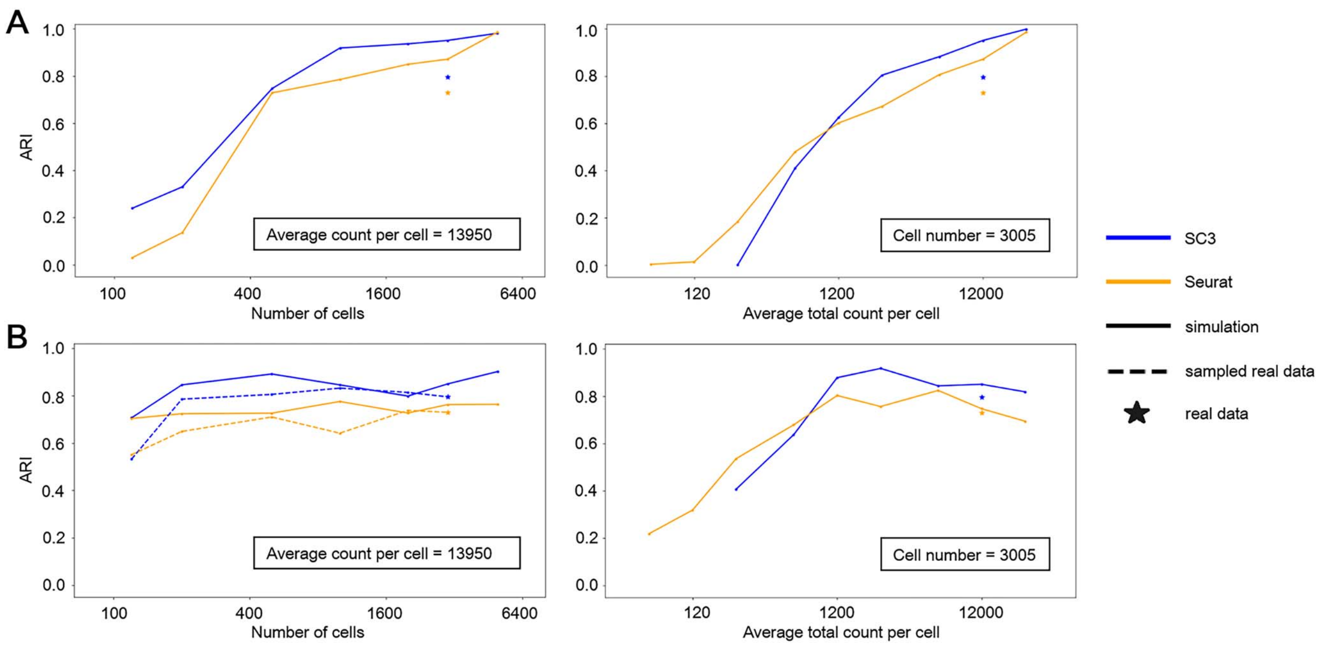


Figure 4. Power evaluation of SC3 and Seurat for cell clustering methods via SimCH. (**A**) Clustering power changes as cell number (left) and average total count per cell (right) change, using the extended multi-group simulation strategy. The simulation was extended from the third group of the Zeisel dataset. (**B**) Clustering power changes as cell number (left) and average total count per cell (right) change based on the independent multi-group simulation strategy. The simulation first classified nine groups of the Zeisel data into sub-groups, and SimCH-copula-NB simulation was conducted on each one. ARI scores of Seurat and SC3 based on experimental data are denoted by stars.

the benchmarking computational methods. To explain, simulated data without gene coexpression might be less reliable and informative for downstream analysis, such as dimensionality reduction and cell clustering, which are commonly affected by the coexpression information. Heterogeneous data may have clear gene coexpression patterns, but homogeneous data also exhibit coexpression patterns in the context of intrinsic GRNs (Figure 5). Therefore, users would find it helpful to set gene coexpression parameters in the extended simulation, especially for connecting cell identities, gene coexpression and gene regulatory networks in a framework, all of which will be explored in our future work. However, it is worth noting that the gene coexpression information extracted from the experimental data by SimCH-copula and other simulators may not be the true coexpression signals as a consequence of some level of gene expression loss, such as dropouts embedded in the original count matrix produced by current scRNA-seq technologies, as illustrated in Figure 3A. Thus, Copula-based simulators just preserve the observed gene coexpression of experimental data. It would be interesting to simulate on an imputed matrix to obtain more realistic simulation data, but the imputation method performed should be evaluated in advance.

Although the researchers can choose a suitable computational tool or pipeline for specific scRNA-seq analysis by benchmarking with such simulators as SimCH, we argue that it would be much more economical and reasonable to perform power evaluation on specific analytical methods or pipelines using SimCH before the scRNA-seq experiment. Through power evaluation, researchers can predict the theoretically available magnitude of scRNA-seq experiments with budgetary constraints. However, it should be noted that different evaluation strategies may produce different conclusions, which are suggested to be considered comprehensively. If researchers use the extended simulation for power evaluation, it is advisable to perform an initial deeper sequencing than the depth predicted by the performance trend as shown on the right of Figure 4A, followed by sequencing as many cells as possible, especially for those investigations focusing on rare or new cell types.

Methods

Calculating size factors

Size factors are important variables with which to model the varying library size affected by mRNA capture rate, amplification efficiency and sequencing biases across sample cells. Size factor S_j of the jth cell can be calculated from the experimental count matrix C^o (M genes × N cells) by

$$S_j = \frac{L_j}{\text{Med}\left(L_j\right)},$$

where L_j represents the library size of the jth cell by summing all counts of the cell, namely $L_j = \sum_i C_{i,j}$, and Med(L_j) denotes the median library size of all cells.

Normalization

To accurately conduct parameter estimation, the digital count $C^o_{i,j}$ at position (i, j) of C^o is first corrected by removing the size factor effect by

$$X_{i,j} = \frac{C^o_{i,j}}{S_j},$$

where $X_{i,j}$ denotes the normalized count.

SimCH-flex mode

Procedure of SimCH-flex simulation

As shown in Figure 6, the gene mean expression μ_i and size factor S_j are modeled by two logarithmic GMM distributions, respectively, while the BCV Φ_i is modeled by a scaled inverse chi-square distribution. After parameter estimation, μ_i can be generated from

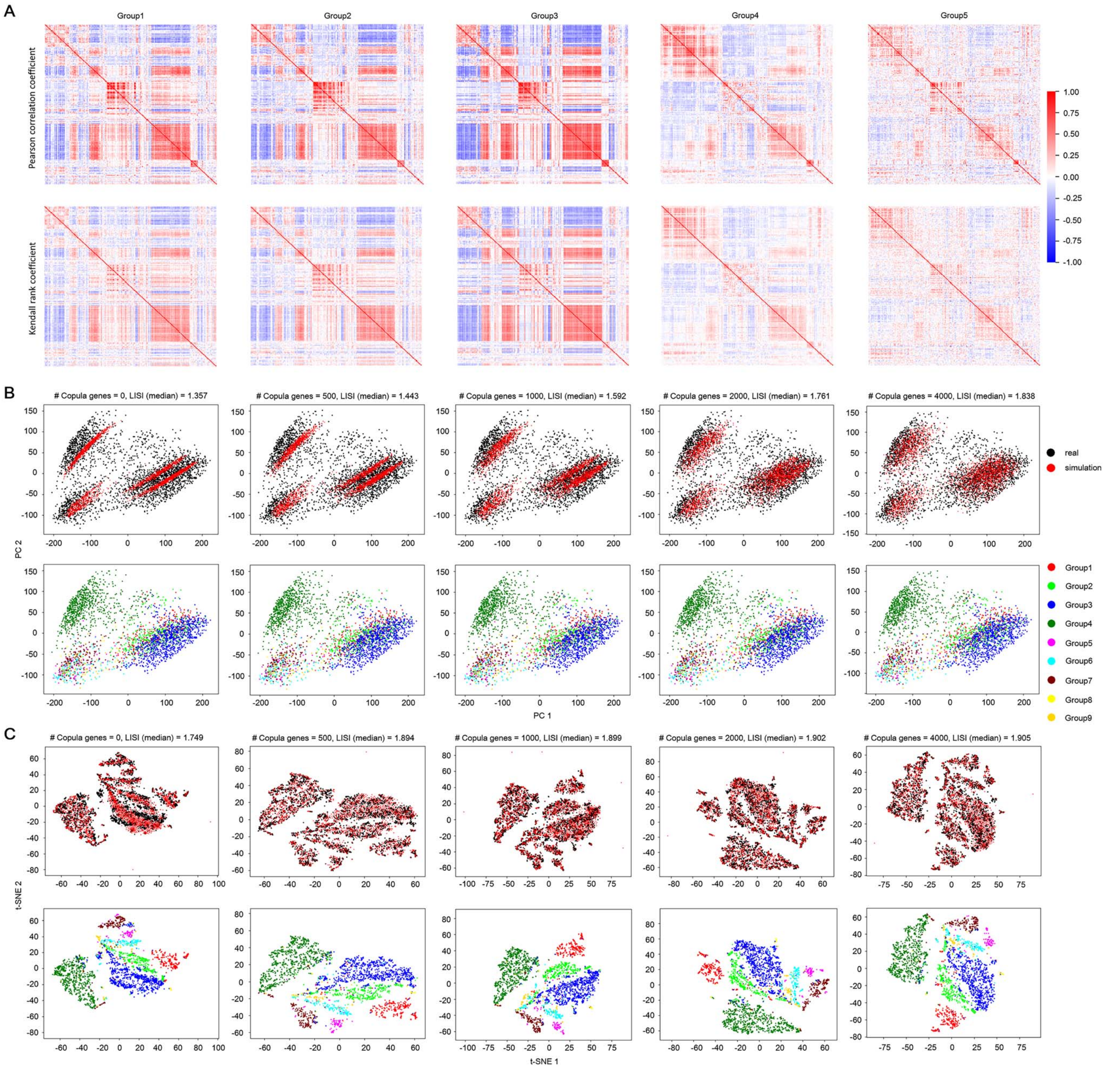


Figure 5. Gene coexpression association with cellular heterogeneity. (**A**) Coexpression patterns of the top 200 genes from groups 1–5 of the Zeisel dataset. Coexpression was evaluated by PCC and Kendall rank coefficient, respectively. (**B**) Correlation between gene coexpression and cellular heterogeneity, as depicted by PCA. The upper plots show that the similarity between simulation and real data increases as the number of coexpressed genes increase from 0 to 4000. The median of LISI was used to evaluate the similarity between simulation and real data. Meanwhile, the bottom plots show the cell group composition of real data. (**C**) Correlation between gene coexpression and cellular heterogeneity, as depicted by *t*-SNE. The description is similar to that of panel **B**.

GMM_5, and $\boldsymbol{\Phi}_{\mathrm{i}}$ can be generated from the scaled inverse chi-square model. The first gene-cell mean expression $\mu'_{\mathrm{i},j}$ is sampled from the Gamma distribution with the parameters μ_{i} and $\boldsymbol{\Phi}_{\mathrm{i}}$. After sampling S_j from GMM_3, the second gene-cell mean expression $\mu_{\mathrm{i},j}$ is calculated by the product of $\mu'_{\mathrm{i},j}$ and S_j. The last step of SimCH-flex can be slightly different, depending on whether to set zero-inflation or not. If zero-inflation is not set (SimCH-flex-NB, recommended for UMI-based data), $\mu_{\mathrm{i},j}$ is directly used to build a Poisson model for generating synthetic count $\mathrm{C}_{\mathrm{i},j}$. If zero-inflation is set (SimCH-flex-NBZI is recommended for non-UMI data), a zero-inflation model is built to generate zero-inflation probabilities, which are used to add additional zeros to the count matrix sampled from the Poisson models.

Statistical models of SimCH-flex

Gaussian mixture models

The logarithmic GMMs (GMM_5 and GMM_3) for modeling μ_{i} and S_j are shown as follows:

$$\ln\left(\mu_{\mathrm{i}}\right) \sim \sum_{k=1}^{5} \mathrm{P}_k N\left(\mu^k, \delta^k\right), \qquad \sum_{k=1}^{5} \mathrm{P}_k = 1,$$

$$\ln\left(\mathrm{S}_j\right) \sim \sum_{s=1}^{3} \mathrm{P}_s N\left(\mu^s, \delta^s\right), \qquad \sum_{s=1}^{3} \mathrm{P}_s = 1,$$

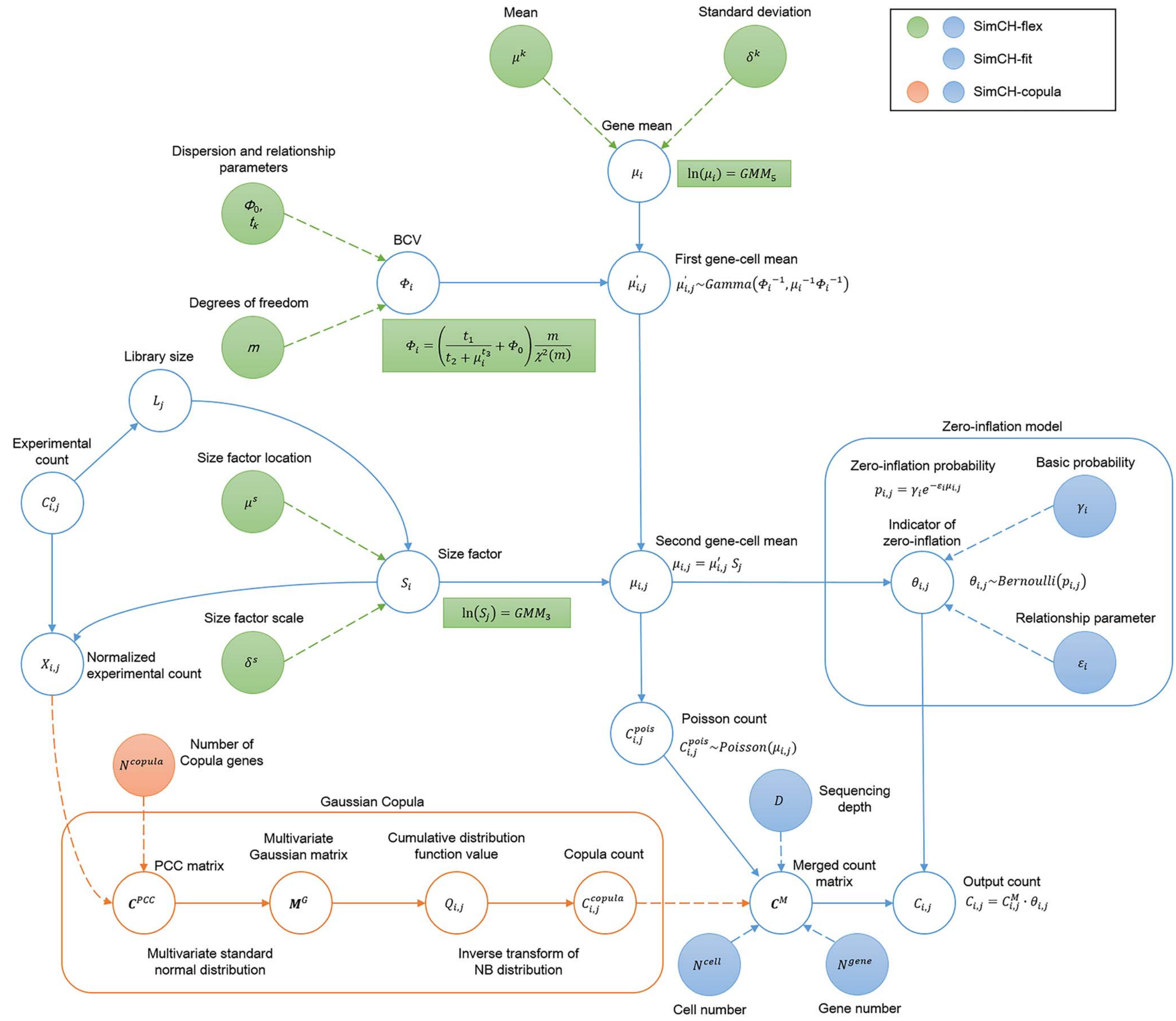


Figure 6. Underlying models of SimCH basic simulation. The SimCH basic simulation includes three modes, namely SimCH-flex, SimCH-fit and SimCH-copula, all of which share several underlying models. The backbone (blue graphics) of SimCH models comprises normalization, Gamma-Poisson (NB) distribution and the zero-inflation model with basic settings (cell number, gene number and sequencing depth), which are also the main components of SimCH-fit. Based on the backbone models, SimCH-flex (blue and green graphics) integrates two GMMs to model gene mean expression μ_i and size factor S_j, respectively, and uses a scaled inverse chi-square distribution to model BCV. In the SimCH-copula mode (blue and orange graphics), coexpression of the top expressed genes is preserved by the Gaussian Copula framework, while the other gene expression is learned by SimCH-fit. The final output of SimCH-copula is a merged count matrix of Copula counts and Poisson counts with zero-inflation, or not. All parameters can be calculated or estimated from the experimental data, and those in the green or blue filled circles can be reset by users.

where P_k, μ^k and δ^k denote the ratios, means and standard deviations of the kth Gaussian distribution for building GMM$_5$. Likewise, P_s, μ^s and δ^s are the parameters for building GMM$_3$.

Scaled inverse chi-square model

BCV is used to model a strong mean–variance trend in RNA-seq data, where low-expression genes are more variable, and high-expression genes are more consistent. Here, BCV is modeled by the scaled inverse chi-square model as follows:

$$\Phi_i = \left(\frac{t_1}{t_2 + \mu_i^{t_3}} + \Phi_0\right)\frac{m}{\chi^2(m)},$$

where m denotes the degrees of freedom of the inverse chi-square model, and t_1, t_2, t_3 and the initial BCV Φ_0 are the parameters that correct the inverse chi-square model.

Gamma-Poisson model (NB)

The Gamma-Poisson/NB model is first used to generate the mean expression $\mu_{i,j}$ of the element at the ith gene and jth cell from the Gamma distribution, and then the count is sampled from the Poisson distribution, which approximates the NB model.

$$\mu_{i,j} \sim \text{Gamma}\left(\frac{1}{\Phi_i}, \frac{1}{S_j\mu_i\Phi_i}\right),$$

$$C_{i,j} \sim \text{Poisson}\left(\mu_{i,j}\right),$$

where $1/\Phi_i$ and $1/(S_j\mu_i\Phi_i)$ denote the shape and the rate of Gamma distribution, respectively. This Gamma model is equivalent to the models for generating the first and second means. An alternate form of the Gamma distribution, $S_j\mu_i$ denotes the mean, and Φ_i is the dispersion parameter.

Zero-inflation model (NBZI)

In the zero-inflation model (Figure 6), the probability of zero inflation is proportional to the exponential transformation of the mean $\mu_{i,j}$ sampled from the Gamma distribution mentioned above. We call it as NBZI.

$$p_{i,j} = \gamma_i e^{-\varepsilon_i \mu_{i,j}} \ (0 < \gamma_i < 1, \varepsilon_i > 0),$$

$$\theta_{i,j} \sim \text{Bernoulli}\left(p_{i,j}\right),$$

$$C_{i,j} \sim \text{Poisson}\left(\mu_{i,j}\right) \bullet \theta_{i,j},$$

where $\theta_{i,j}$ is an indicator of zero-inflation ($\theta_{i,j} = 0$) at the position of (i, j) of the count matrix generated by $Poisson(\mu_{i,j})$, γ_i denotes the maximum value of the zero-inflation probability of the ith gene when $\mu_{i,j} = 0$ and ε_i is the relationship coefficient multiplied by $\mu_{i,j}$ controlling the changing rate of zero-inflation probability. If zero-inflation is not set, $\theta_{i,j}$ is a constant 1.

Parameter estimation of SimCH-flex

SimCH-flex-NB

Parameter estimation of SimCH-flex-NB is the same as that of SimCH-fit-NB. In addition, the parameters t_1, t_2, t_3 and Φ_0, are estimated by the least square method, and m is estimated by the maximum likelihood method.

SimCH-flex-NBZI

In the SimCH-flex-NBZI mode, γ and ε are set to constants, which are calculated by taking the median of γ_i and ε_i estimated from the top 5% genes ranked by mean expression. Then, the estimation method is the same as that performed in SimCH-fit-NBZI.

SimCH-fit mode

Procedure of SimCH-fit simulation

As shown in Figure 6, the main structure of SimCH-fit is similar to that of SimCH-flex, but several parameters, including the gene mean expression μ_i, BCV Φ_i, size factor S_j, basic probability γ_i, and relationship parameters ε_i, are directly estimated from the experimental data.

Statistical models of SimCH-fit

Gamma-Poisson model (NB)

The Gamma-Poisson model (NB) is the same as SimCH-flex. Here, GMMs are not required for modeling gene mean expression and size factors, which are directly estimated from the experimental data.

Zero-inflation model (NBZI)

This is the same as SimCH-flex.

Parameter estimation of SimCH-fit

The parameters of the gene-wise count distribution are estimated from the experimental data.

SimCH-fit-NB

First, the sample mean of each gene of the normalized data is used to estimate μ_i.

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} X_{i,j}.$$

As in bayNorm [22], a hybrid method integrating maximum likelihood estimation and moment estimation is used to estimate Φ_i. As we found that the results of maximum likelihood estimation and moment estimation have a certain deviation, we made adjustments to the estimated value of maximum likelihood by a nonlinear fitting function (a hybrid method).

(i) Moment estimation

$$E\left({C_{i,j}}^2\right) = Var\left(C_{i,j}\right) + E^2\left(C_{i,j}\right)$$

$$= \int_0^{\infty} \frac{\left(\Phi_i S_j \mu_i\right)^{{\Phi_i}^{-1}}}{\Gamma\left({\Phi_i}^{-1}\right)} x^{{\Phi_i}^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \left(x + x^2\right) dx$$

$$= \left(\Phi_i + \frac{1}{S_j \mu_i}\right)\left(S_j \mu_i\right)^2 + \left(S_j \mu_i\right)^2,$$

$$\Phi_{i1} = \frac{\sum_{j=1}^{n} {C_{i,j}}^2 - \mu_i \sum_{j=1}^{n} S_j}{\sum_{j=1}^{n}\left(S_j \mu_i\right)^2} - 1.$$

(ii) Maximum likelihood estimation

$$L\left(\Phi_i\right) = \prod_{j=1}^{n} \int_0^{\infty} \frac{\left(S_j \mu_i \Phi_i\right)^{-{\Phi_i}^{-1}} x^{-{\Phi_i}^{-1}-1} e^{-\left(S_j \mu_i \Phi_i\right)^{-1} x}}{\Gamma\left({\Phi_i}^{-1}\right)} \bullet \frac{x^{C_{i,j}}}{C_{i,j}} e^{-x} dx$$

$$= \prod_{j=1}^{n} \left(S_j \mu_i \Phi_i\right)^{-{\Phi_i}^{-1}} \left[\left(S_j \mu_i \Phi_i\right)^{-1} + 1\right]^{-{\Phi_i}^{-1} - C_{i,j}}$$

$$\bullet \frac{\Gamma\left({\Phi_i}^{-1} + C_{i,j}\right)}{\Gamma\left({\Phi_i}^{-1}\right)\Gamma\left(C_{i,j} + 1\right)},$$

$\ln\left(L(\Phi_i)\right) = \sum_{j=1}^{n}\left[-{\Phi_i}^{-1}\ln\left(S_j\mu_i\Phi_i\right) - \left({\Phi_i}^{-1} + C_{i,j}\right)\ln\left(\left(S_j\mu_i\Phi_i\right)^{-1} + 1\right) + \Gamma\ln\left({\Phi_i}^{-1} + C_{i,j}\right) - \Gamma\ln\left({\Phi_i}^{-1}\right) - \Gamma\ln\left(C_{i,j} + 1\right)\right]$.

The gradient descent method is used to find the optimal Φ_{i2} for maximizing the likelihood function.

(iii) A hybrid method

A nonlinear transformation is performed on Φ_{i2} for estimating Φ_i

$$\Phi_i = \left(\frac{p_1}{p_2 + {\mu_i}^{p_3}} + p_4\right) \bullet \Phi_{i2}.$$

Then, the parameters are fitted by the least square method based on

$$\left(\mu_i, \Phi_{i1}/\Phi_{i2}\right).$$

SimCH-fit-NBZI

To estimate μ_i, Φ_i, γ_i and ε_i of the ith gene, the theoretical and actual zero probability, mean, variance and high-order moments are used to construct an objective function. First, the theoretical values are calculated by

$$P\left(C_{i,j} = 0\right) = \int_0^{\infty} \frac{\left(\Phi_i S_j \mu_i\right)^{{\Phi_i}^{-1}}}{\Gamma\left({\Phi_i}^{-1}\right)} x^{{\Phi_i}^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \gamma e^{-\varepsilon x} dx$$

$$+ \int_0^{\infty} \frac{\left(\Phi_i S_j \mu_i\right)^{{\Phi_i}^{-1}}}{\Gamma\left({\Phi_i}^{-1}\right)} x^{{\Phi_i}^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \left(1 - \gamma e^{-\varepsilon x}\right) \bullet e^{-\left(S_j \mu_i\right)} dx$$

$$= \left(1 + \Phi_i S_j \mu_i\right)^{{\Phi_i}^{-1}} \left[1 - \gamma\left(\frac{\varepsilon}{\left(\Phi_i S_j \mu_i\right)^{-1} + 1} + 1\right)^{-{\Phi_i}^{-1}}\right]$$

$$+ \gamma\left(1 + \varepsilon \Phi_i S_j \mu_i\right)^{-{\Phi_i}^{-1}}.$$

Theoretical zero probability: $P(C_i = 0) = \frac{1}{n}\sum_{j=1}^{n} P(C_{i,j} = 0)$,

$$E\left(X_{i,j}\right) = \frac{1}{S_j}\int_0^\infty \frac{\left(\Phi_i S_j \mu_i\right)^{\Phi_i^{-1}}}{\Gamma\left(\Phi_i^{-1}\right)} x^{\Phi_i^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \left(1-\gamma e^{-\varepsilon x}\right) \bullet x dx$$

$$= \frac{\Phi_i \mu_i \Gamma\left(\Phi_i^{-1}+1\right)}{\Gamma\left(\Phi_i^{-1}\right)} \left[1-\gamma\left(1+\varepsilon \Phi_i S_j \mu_i\right)^{-\Phi_i^{-1}-1}\right].$$

Theoretical mean: $E(X_i) = \frac{1}{n}\sum_{j=1}^{n} E(X_{i,j})$,

$$E\left(X_{i,j}^2\right) = \frac{1}{S_j^2}\int_0^\infty \frac{\left(\Phi_i S_j \mu_i\right)^{\Phi_i^{-1}}}{\Gamma\left(\Phi_i^{-1}\right)} x^{\Phi_i^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \left(1-\gamma e^{-\varepsilon x}\right) \bullet \left(x+x^2\right) dx$$

$$= \frac{\Phi_i \mu_i \Gamma\left(\Phi_i^{-1}+1\right)}{S_j \Gamma\left(\Phi_i^{-1}\right)} \left[1-\gamma\left(1+\varepsilon \Phi_i S_j \mu_i\right)^{-\Phi_i^{-1}-1}\right]$$

$$+ \frac{(\Phi_i \mu_i)^2 \Gamma\left(\Phi_i^{-1}+2\right)}{\Gamma\left(\Phi_i^{-1}\right)} \left[1-\gamma\left(1+\varepsilon \Phi_i S_j \mu_i\right)^{-\Phi_i^{-1}-2}\right].$$

Theoretical variance: $\mathbf{Var}(X_i) = \frac{1}{n}\sum_{j=1}^{n}[X_{i,j} - E(X_i)]^2 = \frac{1}{n}\sum_{j=1}^{n} [(X_{i,j}{}^2) - E(X_i)^{2.}]$.

Theoretical third-order central moment:

$$E\left[\left(X_{i,j} - E\left(X_i\right)\right)^3\right] = E\left(X_{i,j}{}^3\right) - 3E\left(X_i\right)E\left(X_{i,j}{}^2\right) + 3E(X_i)^2 E\left(X_{i,j}\right) + E(X_i)^3$$

$$= \frac{1}{S_j^3}\int_0^\infty \frac{\left(\Phi_i S_j \mu_i\right)^{\Phi_i^{-1}}}{\Gamma\left(\Phi_i^{-1}\right)} x^{\Phi_i^{-1}-1} e^{-\Phi_i S_j \mu_i x} \bullet \left(1-\gamma e^{-\varepsilon x}\right) \bullet \left(x+3x^2+x^3\right) dx$$

$$-3E\left(X_i\right)E\left(X_{i,j}{}^2\right) + 3E(X_i)^2 E\left(X_{i,j}\right) + E(X_i)^3$$

$$= \frac{(\mu_i \Phi_i)^3 \Gamma\left(\Phi^{-1}+3\right)}{\Gamma\left(\Phi^{-1}\right)} \bullet \left[1-\gamma\left(1+\Phi_i \mu_i S_j\right)^{-\Phi^{-1}-3}\right]$$

$$+3\, E\left(X_{i,j}{}^2\right)\left(1-E\left(X_i\right)\right) - 2E\left(X_{i,j}\right) + 3E(X_i)^2 E\left(X_{i,j}\right) + E(X_i)^3.$$

Zero probability, mean, variance and the third-order central moment of the experimental data are denoted by A, B, C and D, respectively. Then, we construct an objective function as follows:

If $A \neq 0$:

$$f\left(\mu_i, \Phi_i, \gamma_i, \varepsilon_i\right) = \left(\frac{|P\left(C_i = 0\right) - A|}{A}\right)^{0.5} + \frac{|E\left(X_i\right) - B|}{B} + \left(\frac{|Var\left(X_i\right) - C|}{C}\right)^2 + \left(\frac{\left|\frac{1}{n}\sum_{j=1}^{n} E\left[\left(X_{i,j} - E\left(X_i\right)\right)^3\right] - D\right|}{D}\right)^3$$

else:

$$f\left(\mu_i, \Phi_i, \gamma_i, \varepsilon_i\right) = \frac{|E\left(X_i\right) - B|}{B} + \left(\frac{|Var\left(X_i\right) - C|}{C}\right)^2 + \left(\frac{\left|\frac{1}{n}\sum_{j=1}^{n} E\left[\left(X_{i,j} - E\left(X_i\right)\right)^3\right] - D\right|}{D}\right)^3.$$

To obtain the optimum solution for the objective function, we constructed a simplified genetic algorithm, which does not perform operations, such as encoding and chromosome crossover, but instead does perform random walk at each individual to produce offspring. Initially, it randomly generates several top-level ancestors. Then every top-level ancestor produces a group of offspring (first generation) by random walk, each of which produces the next group of offspring (second generation). Then, the optimum offspring of each group of the second generation are selected to be alive for producing the next group of offspring (third generation). And, n generations ($n = 625$) are repeated to get the final optimum result.

SimCH-copula mode

As shown in Figure 6, the gene coexpression information of the top expressed genes (default number of Copula genes = 2000) of the experimental data is retained in the simulation under the Gaussian Copula framework, while the other real genes are fitted by the SimCH-fit models. First, the PCC matrix (C^{PCC}) between coexpressed genes is calculated from the normalized matrix of the experimental data, and then the matrix, as well as the multivariate standard normal distribution, is used to generate multivariate Gaussian data (M^G). For the (i–j)th element of M^G, we obtain its cumulative distribution function value $Q_{i,j}$, which is then inversely transformed to the count value by NB distribution. The count matrices generated by Gaussian Copula and SimCH-fit are merged to produce the final count matrix with zero-inflation, or not.

SimCH-ext mode

As shown in Figure 7, the SimCH-ext mode can be conducted to simulate multiple cell groups, batches and differentiation paths based on the parameters estimated from the basic modes of SimCH.

Simulation of cell groups

A given proportion (R^{DE}) of genes are randomly selected as DE genes, and cells are randomly grouped according to the group ratio (R^P). Then mean expression (μ_i) of each DE gene of group p is adjusted by multiplying its MF $\pi_i{}^{(p)}$ (0–∞), which is generated by a log-normal distribution with parameters μ^a and σ^a. For any two groups, SimCH-ext generates two distinctive mean values on at least one gene. In addition, we also expand the cell grouping function by grouping sub-groups of cells to build a multi-layer tree structure of the multiple groups. Moreover, users can set the ratio of marker genes, which are significantly expressed in only one group.

Simulation of batches

In the simulation of batches via SimCH-ext, cells are randomly classified into different batches according to batch ratio (R^K). Then mean expression (μ_i) of all genes of batch k is adjusted by multiplying MF $\pi^{(k)}$ (0–∞), which are generated by a log-normal distribution with parameters μ^b and σ^b. Batches and groups can, when taken together, affect the final mean expression of genes.

Simulation of differentiation paths

In the simulation of differentiation paths, a proportion (R^{NLDE}) of genes are randomly selected as nonlinearly changed genes along the path, and the other DE genes are linearly changed. Cells are then randomly allocated to multiple paths with random steps x ($0 \le x \le 1$). For any gene, the initial MF $\pi_i{}^{(p)}$ is first generated from the log-normal distribution for the multi-group simulation. Calculation of the MF is different for the linear and the nonlinear genes. For each linear gene in a cell at step x of path p, its MF is a linear function of x with $\pi_i{}^{(p)}$ as a parameter, named with $g_i{}^{(p)}(x)$.

$$g_i^{(p)}(x) = 1 + \left(\pi_i^{(p)} - 1\right)x.$$

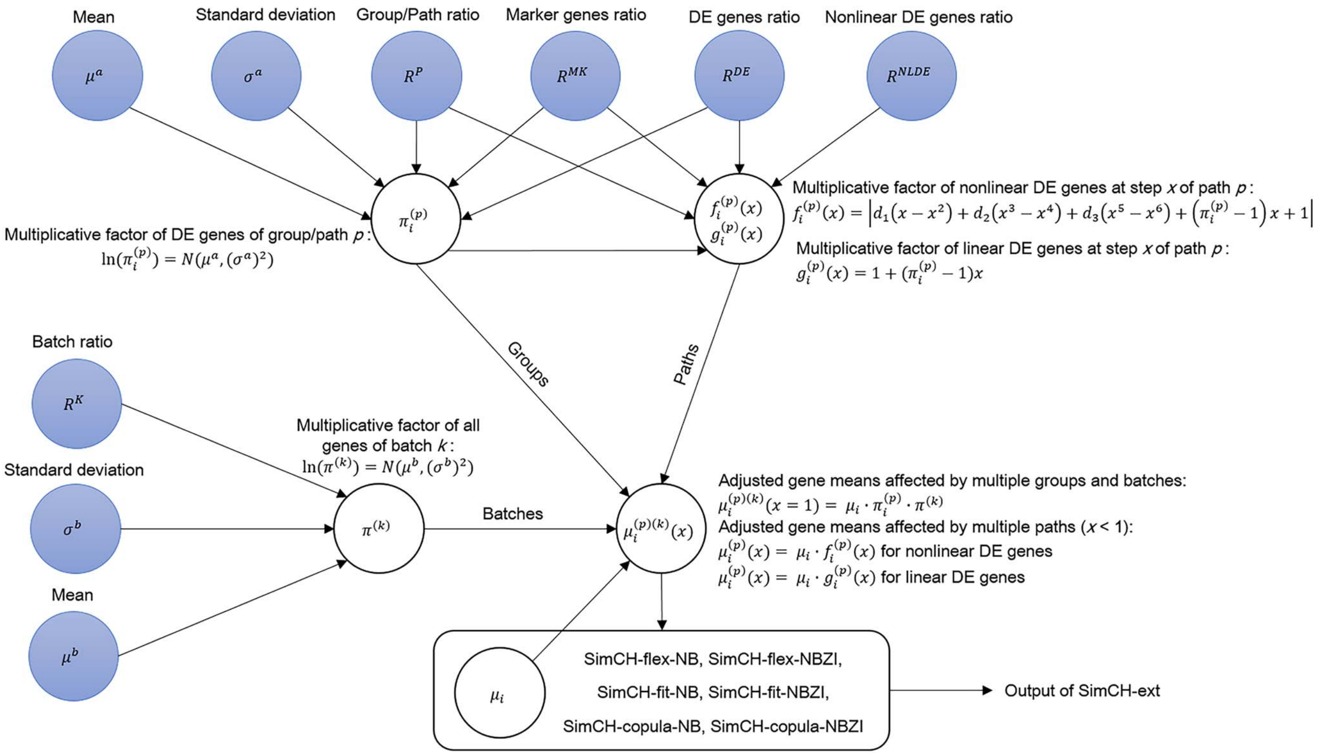

Mean
μ^a
Standard deviation
σ^a
Group/Path ratio
R^P
Marker genes ratio
R^{MK}
DE genes ratio
R^{DE}
Nonlinear DE genes ratio
R^{NLDE}
$\pi_i^{(p)}$
Multiplicative factor of DE genes of group/path p:
$\ln(\pi_i^{(p)}) = N(\mu^a, (\sigma^a)^2)$
$f_i^{(p)}(x)$
$g_i^{(p)}(x)$
Multiplicative factor of nonlinear DE genes at step x of path p:
$f_i^{(p)}(x) = \left|d_1(x - x^2) + d_2(x^3 - x^4) + d_3(x^5 - x^6) + (\pi_i^{(p)} - 1)x + 1\right|$
Multiplicative factor of linear DE genes at step x of path p:
$g_i^{(p)}(x) = 1 + (\pi_i^{(p)} - 1)x$
Groups
Paths
Batch ratio
R^K
Standard deviation
σ^b
Mean
μ^b
Multiplicative factor of all genes of batch k:
$\ln(\pi^{(k)}) = N(\mu^b, (\sigma^b)^2)$
$\pi^{(k)}$
Batches
$\mu_i^{(p)(k)}(x)$
Adjusted gene means affected by multiple groups and batches:
$\mu_i^{(p)(k)}(x = 1) = \mu_i \cdot \pi_i^{(p)} \cdot \pi^{(k)}$
Adjusted gene means affected by multiple paths (x < 1):
$\mu_i^{(p)}(x) = \mu_i \cdot f_i^{(p)}(x)$ for nonlinear DE genes
$\mu_i^{(p)}(x) = \mu_i \cdot g_i^{(p)}(x)$ for linear DE genes
μ_i
SimCH-flex-NB, SimCH-flex-NBZI,
SimCH-fit-NB, SimCH-fit-NBZI,
SimCH-copula-NB, SimCH-copula-NBZI
Output of SimCH-ext

Figure 7. Underlying models of SimCH extended simulation. Based on the parameters estimated from the basic simulation, SimCH-ext can perform extended simulation with prespecified multiple groups, batches or differentiation paths. The parameters of SimCH-ext are estimated from the experimental data or can be initially set in the program, and those in the purple filled circles can be reset by users.

Then, the mean expression μ_i of the nonlinear gene is adjusted by multiplying $g_i^{(p)}(x)$, and the adjusted mean expression $\mu_i^{(p)}(x)$ of the gene is linearly changed along the path. For each nonlinear gene in the path, we define a nonlinear function for its MF as follows:

$$f_i^{(p)}(x) = \left|d_1\left(x - x^2\right) + d_2\left(x^3 - x^4\right) + d_3\left(x^5 - x^6\right) + \left(\pi_i^{(p)} - 1\right)x + 1\right|.$$

The function, including parameters d_1, d_2 and d_3, is obtained by fitting the data points generated by the Brown Bridge. The starting point of the Brown Bridge is 1, and the end point is $\pi_i^{(p)}$. Then, the mean expression μ_i of the nonlinear gene is adjusted by multiplying $f_i^{(p)}(x)$, and the adjusted mean expression $\mu_i^{(p)}(x)$ of the gene is nonlinearly changed along the path. Similar to multi-group simulation, we further expand the path simulation function. SimCH can achieve simulation of the multi-layer tree structure by repeating the above operation.

Simulation on tagged and untagged heterogeneous data

We used two different strategies for simulation on the tagged heterogeneous data. For simulators (e.g. SimCH-fit-NB, SimCH-copula-NB, SimCH-fit-NBZI, SimCH-copula-NBZI and scDesign2) that can preserve the real genes, we split tagged heterogeneous data by groups (or tags), each of which was simulated independently, namely independent multi-group simulation used in the power evaluation. Then, the simulated data of the groups were combined with the preserved genes for further analysis. On the other hand, simulators that cannot preserve real genes, including SimCH-flex-NB, SimCH-flex-NBZI, splat, splat-dropout and kersplat, were directly performed on the count matrix mixing all groups. Most simulators presume the data where they learn are homogeneous. However, simulators are directly performed on both untagged heterogeneous data and homogeneous data.

Comparison of simulation spending time

To compare the spending time of the simulators, we constructed several datasets to test running time as well as its change according to the sequencing depth or cell number on the simulation stages of parameters estimation and simulated data generation, respectively. To test runtime change according to sequencing depth, four datasets were constructed by randomly sampling 100 cells with 18 000 genes from four experimental datasets (Zeisel, Tung, Darmains and Li) with different sequencing depths. To test change in runtime according to cell number, four datasets were constructed by randomly sampling 200, 400, 800 and 1600 cells from the Zeisel dataset with replacement.

Simulation for benchmarking imputation methods

To benchmark DeepImpute and SAVER in terms of gene coexpression preservation on the Tung dataset, we ran SimCH-copula-NB to generate the synthetic 'true' and noisy count matrices. The 'true' count matrix Λ was produced by Gamma distribution by multiplying size factors with parameters estimated from the Tung data, while the noisy count matrix Y was simulated by Poisson sampling from Λ.

Evaluation metrics

It is assumed that the simulation of data should represent similar distribution or features in some dimensionalities, similar to the experimental data where it estimates parameters. Common metrics for evaluating simulators in recovering properties of experimental data include the mean expression, variance, mean–variance relationship, library size, zeros per cell, zeros per gene, mean-zeros per gene and PCC matrix of coexpression genes. Thus, the quality of different simulators can be evaluated and compared by fitting to the same experimental data after calculating the MAEs of the eight metrics. Theoretically, MAE is the average of all absolute difference between the real values x_i and the predicted ones y_i; that is, MAE = (1/*K*) Σ|x_i − y_i|. This expression can be slightly different when calculating each of the eight metrics, but the concept is the same. For example, x_i can be the mean value of the ith gene of the real count matrix, while y_i is the mean value of the ith gene for the simulated matrix. However, the simulated genes from Splat and SimCH-flex are randomly sampled from statistical models without corresponding to real genes line by line, which makes it unreasonable to perform the MAE calculation mentioned above. Thus, we modified the MAE calculation by sorting all values of each of the metrics before further calculation, which can be expressed as MAE = (1/K) Σ|x_i’ − y_i’|. In the new expression, x_i’ is the ith value of the sorted metric elements of the real matrix X, while y_i’ is for the sorted metric elements of the simulated matrix Y.

Specifically, we first conducted CPM normalization on both real and simulated count matrices, except for the evaluation of library size. Then, we calculated the MAEs of the eight metrics as follows:

(i) Mean expression: Sorted gene mean values of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of sorted gene means.

(ii) Variance: Sorted gene variance of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of sorted gene variance.

(iii) Mean–variance: Sorted gene mean values of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of corresponding gene variance between real and simulated data.

(iv) Library size: Sorted library size of real and simulated matrices, respectively, and then calculated the MAE of sorted library size.

(v) Zeros per cell: Sorted zero proportions of cells of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of zero proportions.

(vi) Zeros per gene: Sorted zero proportions of genes of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of zero proportions.

(vii) Mean-zeros: Sorted gene mean values of real and simulated matrices normalized by CPM, respectively, and then calculated the MAE of corresponding zeros per gene between real and simulated data.

(viii) Similarity between gene coexpression matrices (PCC): Calculated the PCC matrix (PCC_matrix_real) of counts between the top 500 highly expressed genes (gene set A) of real data; calculated the PCC matrix (PCC_matrix_SimCH-fit or PCC_matrix_scDesign2) of counts between the 500 simulated genes generated by SimCH-fit or scDesign2, corresponding to gene set A, and then calculated the MAE between PCC_matrix_real and PCC_matrix_SimCH-fit/PCC_matrix_scDesign2; calculated the PCC matrix (PCC_matrix_SimCH-flex or PCC_matrix_Splat) of counts between the top 500 highly expressed genes generated by SimCH-flex and Splat, respectively, and then calculated the MAE between sorted PCC_matrix_real and the sorted PCC_matrix_SimCH-flex/PCC_matrix_Splat.

After calculating all eight MAEs for all simulators, their performance was then ranked according to MAE values such that the smaller the value, the better the performance is, as shown in Figure 1.

Key Points

- We propose a flexible generative framework to simulate scRNA-seq data for benchmarking and evaluating different types of computational methods.
- We recommend the NB-based sub-modes for simulating UMI-based data and the NBZI-based sub-modes for simulating non-UMI data.
- Gene coexpression extracted from the experimental data by SimCH-copula and other simulators may not be the true coexpression signals as the Copula-based simulators just preserve the observed gene coexpression of experimental data.
- It is recommended that power evaluation can be performed on specific analytic methods or pipelines using SimCH before the scRNA-seq experiment.

Supplementary Data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Acknowledgements

The authors thank Dr Bingxiang Xu and Xiaomeng Gao for useful discussion.

Funding

National Nature Science Foundation of China (91940304, 31871331, 31671342 to Z.Z.).

Data availability

- The scRNA-seq data used in this manuscript are all publicly available, and they are summarized in Additional file 1: Supplementary Figure S4, Supplementary Tables S1 and S2. All datasets preprocessed by the analysis in this manuscript are freely available at https://github.com/SIRG-YZU/SimCH/blob/main/datasets.zip. The initial datasets of Camp, Klein, Zeisel and Tung used in our analysis are available at https://github.com/Oshlack/splatter-paper/blob/master/data.tar.gz published with Splatter’s paper [15].
- Camp [67] (734 untagged heterogeneous human whole brain organoids, 60 575 genes in total, non-UMI): The Camp dataset was generated by deleting the first and second rows and columns 2–6 of the Camp data prepared by Splatter [15] and then assigning cell IDs to columns of the count matrix.
- Klein [48] (239 homogeneous human K562 erythroleukemia cells, 25 435 genes in total, UMI-based): The Klein dataset in TXT format was also converted form the Klein data in CSV format generated by Splatter [15].
- Zeisel [58] [3005 tagged heterogeneous mouse cortex and hippocampus cells (nine groups), 19 972 genes in total, UMI-based]: The Zeisel dataset was also converted from Splatter’s datasets by only keeping the cell IDs and count matrix. The cell groups annotated by Splatter were stored in a single TXT file.

- Avraham [68] (96 untagged heterogeneous mouse macrophages cells, 37 315 genes in total, non-UMI): The Avraham dataset named 'GSE65528_s121_p1_Counts_table.txt' was downloaded from the Gene Expression Omnibus (GEO) [69] (accession GSE65528).
- Darmanis [70] (466 untagged heterogeneous cells of human brain cortex tissue, 22 085 genes in total, non-UMI): The Darmanis dataset was created by merging the counts of all CSV files downloaded from GEO (accession GSE67835).
- Engel [71] (203 untagged heterogeneous mouse natural killer T cells, 23 337 genes in total, non-UMI): The Engel dataset was created by merging the counts of all CSV files downloaded from GEO (accession GSE74596).
- Tung [38] (864 homogeneous human induced pluripotent stem cells, 19 027 genes in total, UMI-based): We directly used the Tung dataset generated by Splatter [15].
- Li [72] (460 tagged heterogeneous cells of human primary colorectal tumors, 55 186 genes in total, non-UMI): The Li dataset was converted from 'GSE81861_Cell_Line_COUNT.csv' downloaded from GEO (accession GSE81861). Two cell groups named H1_B1 and GM12878_B1 were removed for excluding batch effects.
- Haber [73] (4700 untagged heterogeneous cells of mouse intestinal epithelial tissue, 16 164 genes in total, UMI-based): The Haber dataset (GSE92332_FAE_UMIcounts.txt) was downloaded from GEO (accession GSE92332).
- Ziegenhain_SCRBseq [65] (45 homogeneous mouse embryonic stem cells, 39 016 genes in total, UMI-based): The Ziegenhain_SCRBseq dataset, including cells with prefix SCRBseqB, was extracted from the complete Ziegenhain data downloaded from GEO (accession GSE75790).
- Ziegenhain_CELseq2 [65] (37 homogeneous mouse embryonic stem cells, 39 016 genes in total, UMI-based): The Ziegenhain_CELseq2 dataset, including cells with prefix CELseq2B, was extracted from the Ziegenhain data.
- Ziegenhain_SmartSeq [65] (69 homogeneous mouse embryonic stem cells, 39 016 genes in total, non-UMI-based): The Ziegenhain_SmartSeq dataset, including cells with prefix SmartSeqA, was extracted from the Ziegenhain data.
- Ziegenhain_SmartSeq2 [65] (77 homogeneous mouse embryonic stem cells, 39 016 genes in total, non-UMI-based): The Ziegenhain_SmartSeq2 dataset, including cells with prefix SmartSeq2B, was extracted from the Ziegenhain data. Details of above datasets are summarized in Additional file 1: Supplementary Table S1.
- Software and code: SimCH is freely available as a Python package from the following GitHub repository (https://github.com/SIRG-YZU/SimCH).

Authors' contributions

L.S. and Z.Z. initiated the project. L.S. designed the simulation framework and analysis. G.W. built the simulation models and performed the analysis. Z.Z. supervised the project. L.S., G.W. and Z.Z. contributed to the manuscript. All authors read and approved the final manuscript.

References

1. Lähnemann D, Köster J, Szczurek E, *et al.* Eleven grand challenges in single-cell data science. *Genome Biol* 2020;**21**:31.
2. Potter SS. Single-cell RNA sequencing for the study of development, physiology and disease. *Nat Rev Nephrol* 2018;**14**:479–92.
3. Chen S, Rivaud P, Park JH, *et al.* Dissecting heterogeneous cell populations across drug and disease conditions with PopAlign. *Proc Natl Acad Sci* 2020;**117**:28784–94.
4. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**:118–27.
5. Leek JT. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res* 2014;**42**:e161.
6. Haghverdi L, Lun ATL, Morgan MD, *et al.* Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**:421–7.
7. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* 2019;**15**:e8746.
8. Brennecke P, Anders S, Kim JK, *et al.* Accounting for technical noise in single-cell RNA-seq experiments. *Nat Methods* 2013;**10**:1093–5.
9. Lun A, Calero-Nieto FJ, Haim-Vilmovsky L, *et al.* Assessing the reliability of spike-in normalization for analyses of single-cell RNA sequencing data. *Genome Res* 2017;**27**:1795–806.
10. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat Rev Genet* 2019;**20**:273–82.
11. Saelens W, Cannoodt R, Todorov H, *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019;**37**:547–54.
12. Rostom R, Svensson V, Teichmann SA, *et al.* Computational approaches for interpreting scRNA-seq data. *FEBS Lett* 2017;**591**:2213–25.
13. Hwang B, Lee JH, Bang D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med* 2018;**50**:1–14.
14. Weber LM, Saelens W, Cannoodt R, *et al.* Essential guidelines for computational method benchmarking. *Genome Biol* 2019;**20**:125.
15. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol* 2017;**18**:174.
16. Vieth B, Ziegenhain C, Parekh S, *et al.* powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics* 2017;**33**:3486–8.
17. Li WV, Li JJ. A statistical simulator scDesign for rational scRNA-seq experimental design. *Bioinformatics* 2019;**35**:i41–50.
18. Su K, Wu Z, Wu H. Simulation, power evaluation and sample size recommendation for single-cell RNA-seq. *Bioinformatics* 2020;**36**:4860–8.
19. Sarkar H, Srivastava A, Patro R. Minnow: a principled framework for rapid simulation of dscRNA-seq data at the read level. *Bioinformatics* 2019;**35**:i136–44.
20. Baruzzo G, Patuzzi I, Di Camillo B. SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics* 2019;**36**:1468–75.
21. Assefa AT, Vandesompele J, Thas O. SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**:3276–8.
22. Tang W, Bertaux F, Thomas P, *et al.* bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**:1174–81.
23. Marouf M, Machart P, Bansal V, *et al.* Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nat Commun* 2020;**11**:166.
24. Crowell HL, Soneson C, Germain P-L, *et al.* Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nat Commun* 2020;**11**:6077.

25. Sun T, Song D, Li WV, *et al.* scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome Biol* 2021;**22**:163.
26. Dibaeinia P, Sinha S. SERGIO: a single-cell expression simulator guided by gene regulatory networks. *Cell Syst* 2020;**11**:252–271.e11.
27. Pratapa A, Jalihal AP, Law JN, *et al.* Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat Methods* 2020;**17**:147–54.
28. Papadopoulos N, Gonzalo PR, Söding J. PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes. *Bioinformatics* 2019;**35**:3517–9.
29. Zhang X, Xu C, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. *Nat Commun* 2019;**10**:2611.
30. Tian J, Wang J, Roeder K. ESCO: single cell expression simulation incorporating gene co-expression. *Bioinformatics* 2021;**37**: 2374–81.
31. Megan, Crow, Jesse et al. Co-expression in single-cell analysis: Saving grace or original sin?, *Trends Genet* 2018;**34**:823–31.
32. Svensson V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol* 2020;**38**:147–50.
33. Jacomy M, Venturini T, Heymann S, *et al.* ForceAtlas2, a continuous graph layout algorithm for Handy network visualization designed for the Gephi software. *PLoS One* 2014;**9**:e98679.
34. Zhang L, Zhang S. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**17**:376–89.
35. Hou W, Ji Z, Ji H, *et al.* A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biol* 2020;**21**:218.
36. Arisdakessian C, Poirion O, Yunits B, *et al.* DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biol* 2019;**20**:211.
37. Huang M, Wang J, Torre E, *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nat Methods* 2018;**15**:539–42.
38. Tung PY, Blischak JD, Hsiao CJ, *et al.* Batch effects and the effective design of single-cell gene expression studies. *Sci Rep* 2017;**7**:39921.
39. Freytag S, Tian L, Lönnstedt I, *et al.* Comparison of clustering tools in R for medium-sized 10x genomics single-cell RNA-sequencing data. *F1000research* 2018;**7**:1297.
40. Duò A, Robinson MD, Soneson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000research* 2018;**7**:1141.
41. Hu Be Rt L, Arabie P. Comparing partitions. *J Classif* 1985;**2**: 193–218.
42. Nguyen V, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J Mach Learn Res* 2010;**11**:2837–54.
43. Shannon CE. A mathematical theory of communication. *Bell Syst Tech J* 1948;**27**:379–423.
44. Studholme C, Hill DLG, Hawkes DJ. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognit* 1999;**32**:71–86.
45. Wagner S, Wagner D. *Comparing Clusterings - An Overview*. Fakultät für Informatik: Universität Karlsruhe, 2007.
46. Kiselev VY, Kirschner K, Schaub MT, *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**:483–6.
47. Satija R, Farrell JA, Gennert D, *et al.* Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;**33**:495–502.
48. Klein AM, Linas M, Ilke A, *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 2015;**161**: 1187–201.
49. Korthauer KD, Chu L-F, Newton MA, *et al.* A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol* 2016;**17**:222.
50. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 2014;**15**:550.
51. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 2010;**26**:139–40.
52. Cannoodt R, Saelens W, Deconinck L, *et al.* Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat Commun* 2021;**12**:3942.
53. Street K, Risso D, Fletcher RB, *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genom* 2018;**19**:477.
54. Wolf FA, Hamey FK, Plass M, *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* 2019;**20**: 59.
55. Wu H, Wang C, Wu Z. PROPER: comprehensive power evaluation for differential expression using RNA-seq. *Bioinformatics* 2014;**31**: 233–41.
56. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**: 53–65.
57. Büttner M, Miao Z, Wolf FA, *et al.* A test metric for assessing single-cell RNA-seq batch correction. *Nat Methods* 2019;**16**: 43–9.
58. Zeisel A, Muñoz-Manchado AB, Codeluppi S, *et al.* Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 2015;**347**:1138–42.
59. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 2019;**9**: 5233.
60. Pearson K. On lines and planes of closest fit to systems of points in space. *Lond Edinb Dublin Philos Mag J Sci* 1901;**2**:559–72.
61. Van der Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.
62. Korsunsky I, Millard N, Fan J, *et al.* Fast, sensitive and accurate integration of single-cell data with harmony. *Nat Methods* 2019;**16**:1289–96.
63. Cao Y, Yang P, Yang JYH. A benchmark study of simulation methods for single-cell RNA sequencing data. *Nat Commun* 2021;**12**:6911.
64. Chen G, Ning B, Shi T. Single-cell RNA-Seq technologies and related computational data analysis. *Front Genet* 2019;**10**: 317.
65. Ziegenhain C, Vieth B, Parekh S, *et al.* Comparative analysis of single-cell RNA sequencing methods. *Mol Cell* 2017;**65**:631–43.e634.
66. Weinreb C, Wolock S, Tusi BK, *et al.* Fundamental limits on dynamic inference from single-cell snapshots. *Proc Natl Acad Sci* 2018;**115**:E2467–76.
67. Camp JG, Badsha F, Florio M, *et al.* Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proc Natl Acad Sci* 2015;**112**:15672–7.
68. Avraham R, Haseley N, Brown D, *et al.* Pathogen cell-to-cell variability drives heterogeneity in host immune responses. *Cell* 2015;**162**:1309–21.
69. Barrett T, Wilhite SE, Ledoux P, *et al.* NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Res* 2012;**41**:D991–5.

70. Darmanis S, Sloan SA, Zhang Y, *et al.* A survey of human brain transcriptome diversity at the single cell level. *Proc Natl Acad Sci* 2015;**112**:7285–90.

71. Engel I, Seumois G, Chavez L, *et al.* Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. *Nat Immunol* 2016;**17**:728–39.

72. Li H, Courtois ET, Sengupta D, *et al.* Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nat Genet* 2017;**49**: 708–18.

73. Haber AL, Biton M, Rogel N, *et al.* A single-cell survey of the small intestinal epithelium. *Nature* 2017;**551**:333–9.