



RESEARCH

Open Access



The shaky foundations of simulating single-cell RNA sequencing data

Helena L. Crowell^{1,2} , Sarah X. Morillo Leonardo³, Charlotte Soneson^{1,2,4} and Mark D. Robinson^{1,2,*} 

*Correspondence:
mark.robinson@imls.uzh.ch

¹ Department of Molecular Life Sciences, University of Zurich, Zurich, Switzerland

² SIB Swiss Institute of Bioinformatics, University of Zurich, Zurich, Switzerland

³ ETH Zurich, Zurich, Switzerland

⁴ Current address: Friedrich Miescher Institute for Biomedical Research and SIB Swiss Institute of Bioinformatics, Basel, Switzerland

Abstract

Background: With the emergence of hundreds of single-cell RNA-sequencing (scRNA-seq) datasets, the number of computational tools to analyze aspects of the generated data has grown rapidly. As a result, there is a recurring need to demonstrate whether newly developed methods are truly performant—on their own as well as in comparison to existing tools. Benchmark studies aim to consolidate the space of available methods for a given task and often use simulated data that provide a ground truth for evaluations, thus demanding a high quality standard results credible and transferable to real data.

Results: Here, we evaluated methods for synthetic scRNA-seq data generation in their ability to mimic experimental data. Besides comparing gene- and cell-level quality control summaries in both one- and two-dimensional settings, we further quantified these at the batch- and cluster-level. Secondly, we investigate the effect of simulators on clustering and batch correction method comparisons, and, thirdly, which and to what extent quality control summaries can capture reference-simulation similarity.

Conclusions: Our results suggest that most simulators are unable to accommodate complex designs without introducing artificial effects, they yield over-optimistic performance of integration and potentially unreliable ranking of clustering methods, and it is generally unknown which summaries are important to ensure effective simulation-based method comparisons.

Keywords: Benchmarking, Simulation, Single-cell RNA-seq

Background

Single-cell RNA-sequencing (scRNA-seq) has become an established tool for studying the transcriptome at individual cell resolution. Since the first scRNA-seq study's publication in 2009 [1], there has been a rapid increase in the number of scRNA-seq datasets, number of cells, and samples per dataset [2], and a corresponding growth in the number of computational methods to analyze such data, with over one thousand tools catalogued to date [3, 4]. With the development of new methods comes the need to demonstrate



© The Author(s) 2023, corrected publication 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

their performance, and to consolidate the space of available methods through comprehensive and neutral benchmark studies [5–7].

In this context, simulations have become an indispensable tool, for example, to investigate how methods respond to varying parameter inputs and quantify their scalability in terms of computational cost, as well as ensuring that a method is performant across a range of scenarios and in comparison to other available tools. The attractiveness of simulation studies is largely due to being able to specify a ground truth, which is often challenging or infeasible to establish in experimental data [8]. For example, evaluation of methods to group cells into biologically meaningful subpopulations (clusters) relies on “true” labels to be compared against. While these may be attainable (e.g., through cell-sorting) or derived (e.g., manual annotation by an expert), simulations enable testing methods across a wide range of scenarios where the number of clusters, between-cluster (dis)similarity, and effects of other covariates can be deeply explored. As a result, simulations have been applied to benchmark methods across a wide range of tasks, including differential expression analysis [9–11], trajectory inference [12], and data integration [13, 14].

By definition, simulations generate *synthetic* data. On the one hand, conclusions drawn from simulation studies are frequently criticized, because simulations cannot completely mimic (real) experimental data. On the other hand, it is often too expensive, or even impossible, to generate experimental data that is suitable for formal performance comparison. Nonetheless, setting a high-quality standard for simulations is all the more important to ensure results based on them are transferable to corresponding experimental datasets.

Typically, new simulation methods come with minimal (non-neutral) benchmarks that focus on one-dimensional evaluations, i.e., how similarly a set of summaries is distributed between a reference and simulated dataset (e.g., Zappia et al. [15]). In some cases, two-dimensional relationships (e.g., gene expression mean-variance) are explored (e.g., Assefa et al. [16]). However, the faithfulness of the full complexity of the simulated scRNA-seq data, including batch effects and clusters, is rarely evaluated. To date, there has been only one neutral evaluation of how well scRNA-seq data simulators recapitulate key characteristics of the counts, sample- and subpopulation-effects present in real data [17]; in particular, they proposed a novel kernel density metric to evaluate similarity of real and simulated data summaries. However, to what extent simulators affect the results of method comparisons is not considered.

Methods for simulating scRNA-seq data may be categorized according to various factors. Most importantly, there is a dichotomy between methods that generate synthetic data *de novo* and those that rely on a reference dataset. The former depend on user-defined parameter inputs to generate counts, and introduce artificial effects between, e.g., different groups of cells or samples. Conversely, reference-based methods estimate parameters to mimic the gene expression profiles observed in the reference dataset. However, many methods employ a hybrid framework where, e.g., baseline parameters are estimated from a “singular” reference (i.e., a homogeneous group of cells), and additional layers of complexity (e.g., batch effects, multiple clusters, and/or experimental conditions) are added *post hoc*. Both strategies have their advantages and disadvantages: *de novo* simulators offer high flexibility in varying the strength and specificity of

different effects, but might not generate realistic data; in contrast, reference-based methods are limited to the complexity of the input data and consequently less flexible, but are by default more realistic. Taken together, there is a trade-off between how applicable methods are in benchmarking single-cell analysis tools across a wide range of scenarios versus whether simulation study results are directly transferable to real data.

Here, we evaluated 16 scRNA-seq simulation methods in their ability to replicate important aspects of a real reference dataset. We considered various global, gene- and cell-level summaries, and compared them between reference and simulated data, in both one- and two-dimensional settings. In addition to global distributions (i.e., across all cells), we made batch- and cluster-level comparisons to capture structural differences in the summaries.

Our results suggest that there is a noticeable shortage of simulators that can accommodate complex situations and that popular simulators do not adequately mimic real datasets. In particular, some current methods are able to simulate multiple groups of cells (e.g., batches and/or clusters), but do so in an ad hoc manner, e.g., by introducing arbitrary differences based on parameter inputs. Few methods attempt to estimate and mimic group effects from reference datasets, but this comes at a loss of supplying a ground truth.

Results

Benchmark design

We evaluated simulators based on 12 published datasets (Additional file 1: Table S1), from which we generated a variety of subsets that serve as references for simulation (Additional file 1: Table S2). We labeled references as one of three types according to their complexity: type n are “singular” references that contain cells from a single batch and cluster; type b contain cells from multiple batches; and type k contain cells from multiple clusters (Additional file 1: Fig. S1-3). Here, batches can be either biological or technical replicates; clusters refer to cell subpopulations or types as annotated in the original data; and groups can be either batches, clusters, or experimental conditions. In total, we used 10, 8, and 8 references of type n , b , and k , respectively.

To objectively cover the space of currently available simulators, we browsed the scRNA-seq tools database [3, 4], which, at the time of writing, catalogued over 1000 tools for analyzing scRNA-seq data, 65 of which hold a “simulation” tag. We included all methods that (i) could be installed and run after at most minor manual adjustment(s) and (ii) were reference-based, i.e., supported parameter estimation from a real dataset. We selected a total of 16 methods, 9/6 of which could accommodate batches/clusters (Table 1). A brief summary of each method’s model framework, capabilities and limitations, and the parameter settings used in this study is given under Methods.

Because different simulators can generate different levels of complexity (two groups, multiple clusters or batches, both or neither), we tagged each method according to their capabilities (type n , b and/or k). Each method was only run on corresponding reference datasets. A more detailed overview of the computational pipeline for this study is given in Fig. 1 (see also Additional file 1: Sec. 6). Notably, there are various methods aimed at simulating continuous scenarios (e.g., a trajectory or time-course [18–20]). Here, we limited comparisons to methods that generate a single group or multiple groups of cells,

Table 1 Overview of scRNA-seq simulators compared in this study. Methods are ordered alphabetically and annotated according to their (in)ability to accommodate multiple batches and/or clusters, support for parallelization (parameter estimation and data simulation, respectively), software availability, and publication year. ‘Type(s)’ column specifies which type of simulations can be produced (n: “singular” references: single batch or cluster; b: multiple batches; k: multiple clusters). ‘Cell #’ refers to whether the number of cells can be varied. Symbols: ✓ = yes, ✗ = no, (✓) = yes, but based on user input parameters, i.e., no support for parameter estimation, *requires random splitting of cells into two groups, †/‡ = internal/prior resampling from empirical parameter distribution, ◦ = no separate estimation step)

	Batches	Clusters	Type(s)	Cell #	Parallelization	Availability	Year	Model
BASICS [37]	✓	✗	b	✗	✓✗	R/Bioc	2015	NB
ESCO [38]	✓	✓	n,b,k	✓	✓✓	R/GitHub	2020	Gamma-Poisson
hierarchicell [39]	✓	✗	n,b	✓	✗✗	R/GitHub	2021	NB
muscat [40]	✓	✓	n,b,k	(✓)†	✗✗	R/Bioc	2020	NB
POWSC [41]	✗	✓	n,k	(✓)†	✗✗	R/Bioc	2020	zero-inflated, log-normal Poisson mixture
powsimR [42]	✗	(✓)	n*	(✓)†	✓✓	R/GitHub	2017	NB
scDD [43]	✗	✗	n*	✓	✓✓	R/Bioc	2016	Bayesian NB mixture model
scDesign [44]	✗	(✓)	n	✓	◦	R/GitHub	2019	Gamma-Normal mixture model
scDesign2 [45]	✗	✓	n,k	✓	✓✗	R/GitHub	2020	(zero-inflated) Poisson or NB + Gaussian copula for gene-gene correlations
SCRIP [46]	✓	✓	n,b,k	✓	✗✗	R/GitHub	2020	(Beta-)Gamma-Poisson
SPARSim [47]	✓	✗	n,b	(✓)‡	✗✗	R/GitLab	2020	Gamma-multivariate hypergeometric
splatter [15] (Splat model)	(✓)	(✓)	n	✓	✗✗	R/Bioc	2017	Gamma-Poisson
SPsimSeq [16]	✓	✗	n,b	✓	◦✗	R/Bioc	2020	log-linear model-based density estimation + Gaussian copula for gene-gene correlations
SymSim [48]	✓	✗	n,b	✓	✗✗	R/GitHub	2019	kinetic model using MCMC
ZINB-WaVE [49]	✓	✓	n,b,k	✗	✗✗	R/Bioc	2018	zero-inflated NB
zinger [50]	✗	✗	n	(✓)†‡	✗✗	R/GitHub	2017	zero-inflated NB

since current continuous simulators are fully or in part de novo, making it challenging to validate the faithfulness of the data they generate (see Discussion).

In order to investigate how widely and for what purpose different simulators are applied, we browsed the literature for benchmark studies that compare tools for a specific scRNA-seq analysis task. Depending on the task, such comparisons often rely on simulation studies where a ground truth is known (by design), or a combination of simulated and real data, where an experimental ground truth exists. For each benchmark, we summarized the task of interest and, if any, which simulator(s) are used. Across all considered benchmarks, these amounted to only five, namely *muscat* (1 [21]), *scDesign* (1

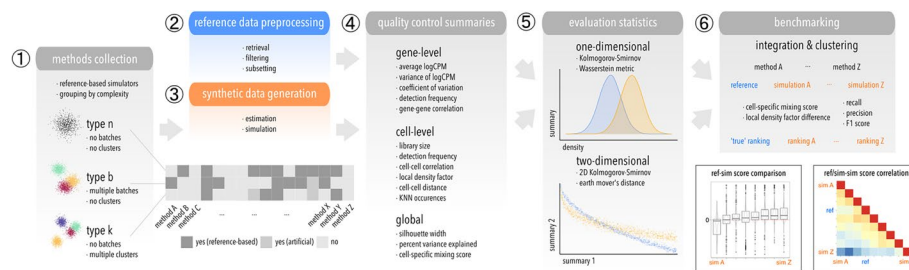


Fig. 1 Schematic of the computational workflow used to benchmark scRNA-seq simulators. (1) Methods are grouped according to which level of complexity they can accommodate: type *n* (“singular”), *b* (batches), *k* (clusters). (2) Raw datasets are retrieved reproducibly from a public source, filtered, and subsetted into various datasets that serve as reference for (3) parameter estimation and simulation. (4) Various gene-, cell-level, and global summaries are computed from reference and simulated data, and (5) compared in a one- and two-dimensional setting using two statistics each. (6) Integration and clustering methods are applied to type *b* and *k* references and simulations, respectively, and relative performances compared between reference-simulation and simulation-simulation pairs

[22]), *powsimR* (2 [10, 23]), *scDD* (3 [9, 11, 24]), and *splatter* (13 [12–14, 25–34]). Benchmark tasks included batch effects [13, 14], clustering [28, 31, 35], doublet detection [22], differential expression [9–11, 30], dimensionality reduction [29, 32], imputation [25, 34], isoform quantification [36], marker selection [24, 27], normalization [26], pipelines [21, 23], cell type assignment [33], and trajectory inference [12]. Yet, this listing of benchmarks and their use cases is not exhaustive; the frequency with which simulators are applied in benchmarks need not speak for or against their performance (e.g., long-lived and user-friendly methods might be favored), and because some simulators have not been applied to a given task does not mean they cannot be.

In order to summarize how well each simulator recapitulates key characteristics of the reference scRNA-seq dataset, we computed a range of gene- and cell-level summaries for both reference and simulated datasets. These include average and variance of log-transformed counts per million (CPM), coefficient of variation, gene detection frequency, gene-to-gene correlation, log-transformed library size (total counts), cell detection frequency, cell-to-cell correlation, local density factor, cell-to-cell distance, and *k*-nearest neighbor (KNN) occurrences (see Additional file 1: Sec. 2).

Since some summaries (e.g., detection frequency) can vary between batches (e.g., sequencing depths may vary between protocols) and clusters (e.g., different cell types may differ in their overall expression), we computed them globally, i.e., across all cells, as well as for each batch and cluster. Thus, for a given dataset with *B* batches and *K* clusters, we obtain 1 , $1 + B$, and $1 + K$ results per summary for type *n*, *b*, and *k*, respectively. Three additional summaries were computed across all cells – namely, the percent variance explained (PVE) at the gene-level (i.e., expression variance accounted for by batch/cluster for type *b/k*); and the silhouette width [51] and cell-specific mixing score (CMS) [52] at the cell-level (considering as group labels the batch/cluster for type *b/k*)—that aim to capture global batch or cluster effects on gene expression variability and cell-to-cell similarity, respectively.

To evaluate simulator performance, we compared summaries between reference and simulated data in one- and two-dimensional settings by computing the Kolmogorov-Smirnov (KS) distance [53] and Wasserstein metric for each summary (Additional file 1:

Fig. S5-7), and the KS distance and earth mover’s distance (EMD) for each relevant pair of summaries (Additional file 1: Fig. S8-10). In general, these metrics quantify how dissimilar a pair of (univariate or bivariate) distributions are (see Additional file 1: Sec. 4). Test statistics were generally consistent between KS test and Wasserstein metric, as well as KS test and EMD (Additional file 1: Fig. S4). Thus for brevity, method performances are hereafter reported as one- and two-dimensional KS statistics.

Simulators vary in their ability to mimic scRNA-seq data characteristics

Across all simulation types, cell-level quality control summaries were generally poorly recapitulated (Fig. 2a), with the largest deviation in cell-to-cell correlation. The silhouette width, CMS, and PVE gave among the highest KS distances for most methods, indicating that while group-level (i.e., within a batch or cluster) summaries might be preserved well during simulation, the global data structure (e.g., inter-group relations) is not. Despite its popularity, *splatter* ranked in the middle for the majority of summaries. *scDD* ranked poorly for most summaries, preceded by *hierarchicell* and *ESCO*. Considering all summaries, *ZINB-WaVE*, *scDesign2*, and *muscat* were among the best

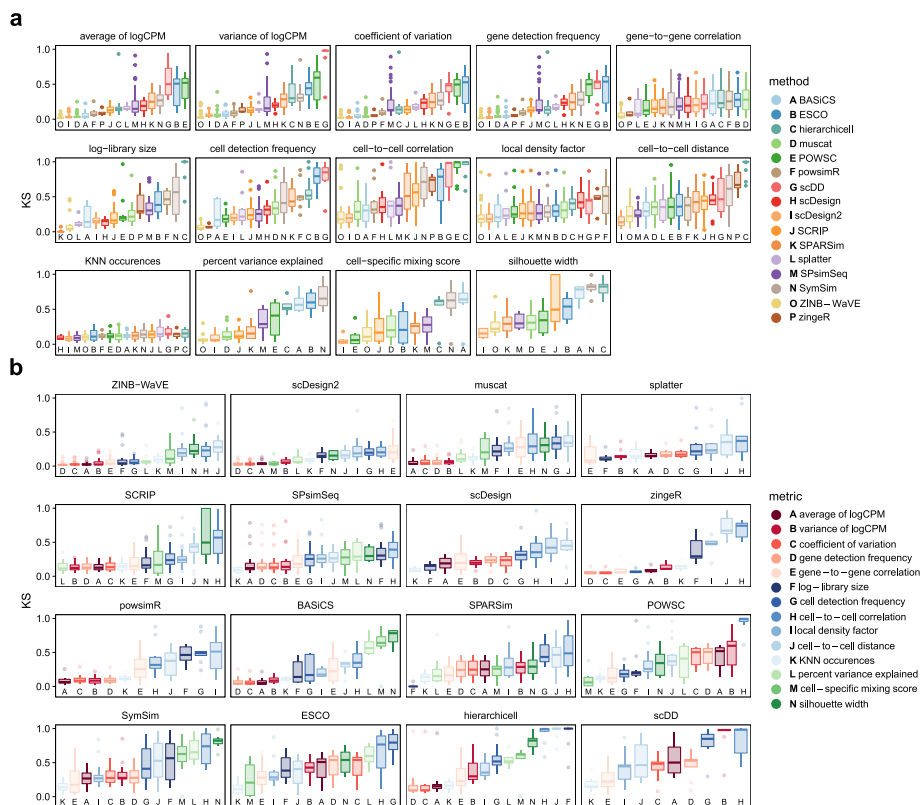


Fig. 2 Kolmogorov-Smirnov (KS) test statistics comparing reference and simulated data across methods and summaries. Included are datasets and methods of all types; statistics are from global comparisons for type n , and otherwise averaged across cluster-/batch-level results. **a** Data are colored by method and stratified by summary. For each summary (panel), methods (x-axis) are ordered according to their average. **b** Data are colored by summary and stratified by method. For each method (panel), metrics (x-axis) are ordered according to their average from best (small) to worst (large KS statistic). Panels (methods) are ordered by increasing average across all summaries

performing simulators, yielding low KS test statistics across a large number of metrics and datasets (Fig. 2b).

Finally, we ranked simulators according to their overall performance. In order to weight datasets equally and independently of the number of subsets drawn from them, we first averaged statistics across subsets, then datasets. Secondly, because simulators ranked similarly in both one- and two-dimensional comparisons, and performances were often linked for certain metrics, we limited rankings to one-dimensional evaluations only, and averaged across all gene- and cell-level metrics. This resulted in three independent rankings, one for each set of methods that can accommodate a given simulation type (Fig. 3). Notably, subpopulations (batches/clusters) may vary in size and complexity. Thus, for types other than *n*, we averaged across group-level results (instead of using global test results).

For type *n*, *ZINB-WaVE*, *scDesign2*, *muscat*, and *SPsimSeq* performed similarly well, with *POWSC*, *ESCO*, *hierarchicell*, and *scDD* ranking last across various summaries. *ZINB-WaVE* and *muscat* were also the most performant among type *b* and *k* simulators, joined by *SPARSim* and *scDesign2*, respectively. LDE, cell-to-cell distance and correlation (across all types), and global summaries (PVE and silhouette width for type *b* and *k*) were poorly recapitulated.

To measure the scalability of methods, we repeatedly timed estimation and simulation steps across varying numbers of genes and cells (see [Methods](#)). Runtimes varied across

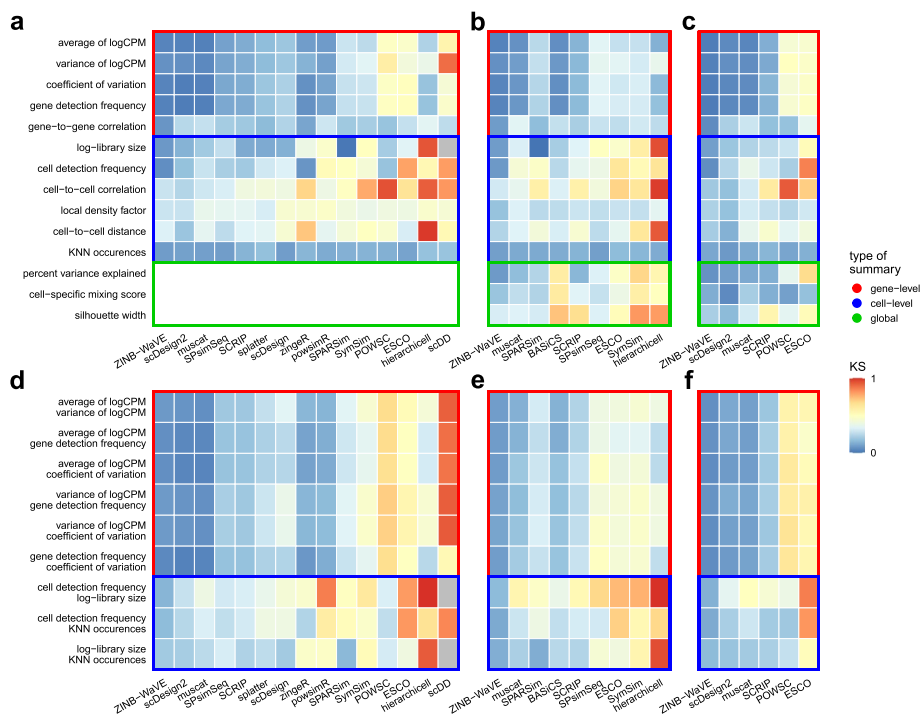


Fig. 3 Average performance in one- (upper row) and two-dimensional evaluations (bottom row) for (a, d) type *n*, (b, e) type *b*, and (c, f) type *k* simulations. For each type, methods (x-axis) are ordered according to their average performance across summaries in one-dimensional comparisons. Except for type *n*, batch- and cluster-level results are averaged across batches and clusters, respectively. Boxes highlight gene-level (red), cell-level (blue), and global summaries (green)

several orders of magnitude (Additional file 1: Fig. 11). Some methods did not offer separate estimation and data generation steps, while others can generate multiple simulations from one-time parameter estimates. Specifically, the estimation step of *scDesign2*, *ZINB-WaVE*, and *zingeR* was relatively slow, but data simulation was not. In contrast, *scDD* and *SymSim* took longer for simulation than estimation. Overall, *BASiCS* was by far the slowest. *SPsimSeq*, *SPARSim*, *SCRIP*, and *SymSim* were approximately tenfold faster. The remaining methods (*ESCO*, *hierarchicell*, *muscat*, *POWSC*, and *splatter*) were the fastest. Memory usage (Additional file 1: Fig. S12 and Additional file 1: Fig. S13) was similar across methods but exceptionally high for *SPsimSeq*. While some methods provide arguments for parallelization (see Table 1), all methods were run on a single core for comparability.

Batch simulators yield over-optimistic but faithful integration method performance

Ideally, benchmark results (i.e., the ranking of computational tools for a given task) should be the same for experimental and simulated data. In order to investigate how method comparison results are affected by simulation, we used the 8 type *b* references to compare 6 scRNA-seq batch correction methods. To evaluate method performances, we computed (i) cell-specific mixing scores (CMS) and (ii) difference in local density factors (Δ LDF) before and after integration [52]. In order to make metrics comparable across datasets, we zero-centered CMS (denoted CMS*), and zero-centered and range-one scaled Δ LDF (denoted Δ LDF*). Finally, we computed a batch correction score BCS = $|\text{CMS}^*| + |\Delta\text{LDF}^*|$, where small values indicates ideal mixing (CMS* of 0 on average) while retaining the data's internal structure (Δ LDF* centered at 0), and large values indicates batch-specific bias (high CMS* density at ± 1) and changes in overall structure (Δ LDF* non-symmetric).

Δ LDF* were largely consistent between references and simulations (Additional file 1: Fig. S14), whereas CMS* were much less correlated for most methods (Additional file 1: Fig. S15). BCSs were overall similar for simulated compared to reference data and well correlated between most reference-simulation and simulation-simulation pairs (Additional file 1: Fig. S16). Simulations from *SPsimSeq*, *ZINB-WaVE*, *SPARsim*, and *SCRIP* gave results most similar to real data, followed by *BASiCS*, and lastly *muscat* and *SymSim* (see also Additional file 1: Fig. S17-24).

Cluster simulators affect the performance of clustering methods

Secondly, we used the 8 type *k* references to evaluate 9 scRNA-seq clustering methods that were previously compared in Duó et al. [31]. To evaluate method performances, we computed cluster-level F1 scores, after using the Hungarian algorithm [54] to match cluster assignments to “true” labels.

Across all methods and datasets, F1 scores were consistently higher for simulated compared to real data (Fig. 4a-b). In addition, for similarly performant simulators, clustering method rankings were more dependent on the underlying reference dataset than the specific simulator used (Fig. 4c). And, some simulators (e.g., *SCRIP*) gave almost identical F1 scores and rankings for multiple references. Overall, method rankings (according to F1 scores) were lowly correlated between simulated and reference data, as

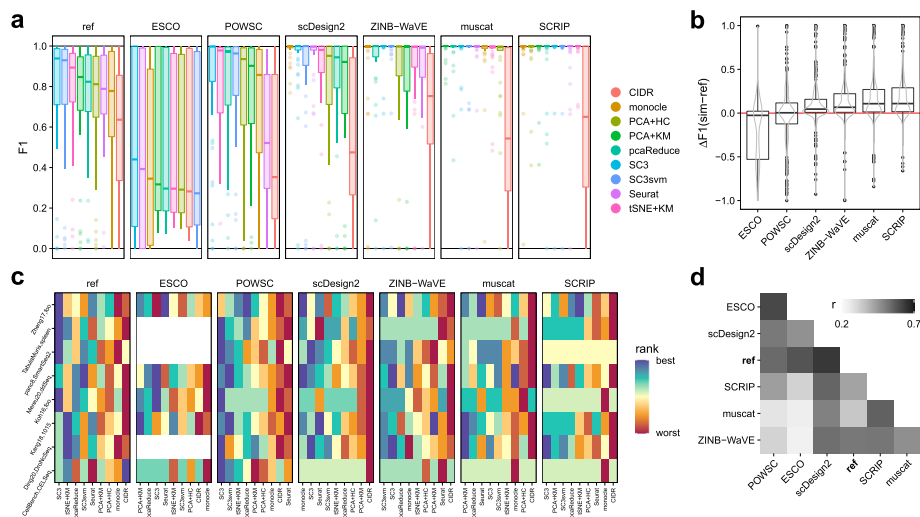


Fig. 4 Comparison of clustering results across (experimental) reference and (synthetic) simulated data. **a** Boxplot of F1 scores across all type k references, simulation and clustering methods. **b** Boxplot of difference (Δ) in F1 scores obtained from *reference* and *simulated* data. **c** Heatmap of clustering method (ρ) rankings across datasets (rows), stratified by simulator (panels). **d** Heatmap of Spearman's rank correlation (ρ) between F1 scores across datasets and clustering methods

well as between simulations (Fig. 4d), with *scDesign2* and *POWSC* giving the most, and *muscat* and *SCRIP* giving the least similar ranking, respectively.

Taken together, these results suggest that simulations do not achieve the same level of complexity in terms of intra- and inter-subpopulation effects (i.e., batches and clusters). Consequently, methods to correct of such effects (integration) or group together similar cells (clustering) perform over-optimistically in simulated data compared to more complex and noisy experimental data and are almost indistinguishable in their performance for 'simple' datasets.

Meta-analysis of summaries

Inevitably, summaries used to assess whether simulated data mimics real data may be redundant, and we expect that some summaries are more important than others. To quantify the relationship between summaries, we correlated all comparable summaries, i.e., gene- and cell-level summaries, respectively, excluding those that include sampling, i.e., correlations and cell-to-cell distances (Fig. 5a). Gene detection frequency and average expression were highly similar ($r \sim 1$) and correlated well with expression variance ($r > 0.5$). At the cell level, detection frequencies and library sizes were most similar.

Next, we correlated the KS test statistics obtained from comparing reference-simulation pairs of summaries across all datasets (Fig. 5b). Summaries grouped together according to their type (global, gene- or cell-level), indicating that simulators recapitulated each type of summary to a similar degree, and that one summary per type could be sufficient to distinguish between method performances.

To investigate the overall similarity of summaries, we performed multi-dimensional scaling (MDS) on KS statistics across methods and datasets (Fig. 5c and Additional file 1: Fig. S25). In line with the observed correlation structure, summaries grouped together by type, with gene-level summaries being most similar to one another.

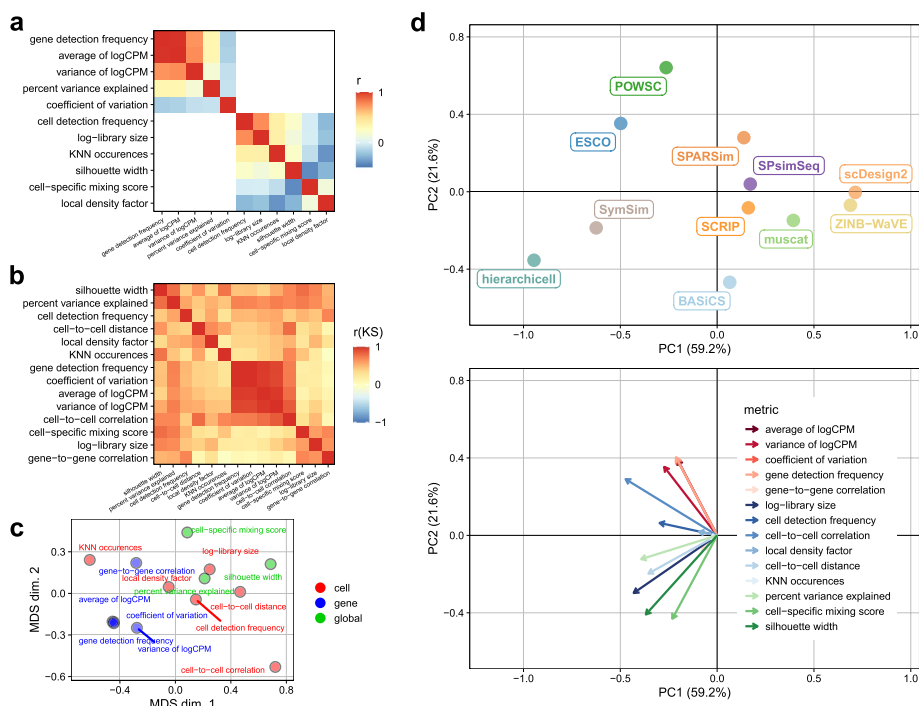


Fig. 5 Comparison of quality control summaries and KS statistics across datasets and methods. Spearman rank correlations (r) of **a** gene- and cell-level summaries across reference datasets, and **b** KS statistics across methods and datasets. **c** Multi-dimensional scaling (MDS) plot and **d** principal component (PC) analysis of KS statistics across all and type b/k methods, respectively, averaged across datasets

Next, we performed principal component analysis (PCA) on test statistics of summaries across methods and datasets (Fig. 5d and Additional file 1: Fig. S26-28), thus representing each method-dataset as a linear combination of statistics for each summary. For all types, the largest fraction of variance (PC1: >40%) was attributable to differences in overall method performance, followed by (PC2: >15%) differences in summary type-specific performance (global, gene-, cell-level).

Taken together, our analyses suggest that several summaries convey similar information, with gene-level summaries being particularly redundant, and global summaries the least redundant. Accordingly, simulator performance may be sufficiently quantifiable by a combination of one gene-level, few cell-level, and various global summaries. However, the number (and nature) of summaries to comprehensively cover the inherent data structure is dependent on its complexity (e.g., global summaries are void for type n , but all the more relevant for type b and k), and there may exist other informative summaries not considered here.

Discussion

In this study, we compared scRNA-seq data simulators in their ability to generate synthetic data that can recapitulate key characteristics of real reference datasets. We considered a range of gene- and cell-level summaries, as well as ones specific to capturing local and global group-effects (i.e., intra- and inter-variability of batches and clusters). By comparing the distribution of summaries as well as pairs thereof

between reference and simulated data, we evaluated how well simulations capture the structure exhibited by a given reference dataset. We ranked simulators by averaging their performance across summaries and simulations, thus evaluating each method across a multifaceted range of criteria (covering gene-, cell- and group-specific summaries) and datasets (from various tissues and technologies). Our results confirm some aspects of those in the study from Cao et al. [17] (our study includes 16 simulators, theirs covered 12, with an overlap of 10); for example, *ZINB-WaVE* stands out as performant in nearly all reference-simulation comparisons. However, we additionally investigated whether the choice of simulator affects method comparison results, included additional quality control summaries that capture group (batch/cluster) structure, and stratified simulators according to type (singular, batch, cluster).

Overall, simulations have proven paramount for the development of computational tools for the analysis of single-cell data. However, they are often implemented for a specific task, e.g., evaluating clustering, batch correction, or DE analysis methods; an “all-rounder” simulator is currently lacking, and most methods are limited to simple designs. The arguably most sobering observation of this study is that, without introducing arbitrary effects that depend on user inputs (e.g., the frequency of DE genes and magnitude of changes in their expression), the vast majority of methods can simulate only one group of cells (i.e., one batch, one cluster). As a result, most current simulators are rather limited in their applicability.

Both neutral benchmark studies as well as non-neutral comparisons performed alongside newly presented methods rely on a ground truth for evaluation. Thus, future work should be focused on the development of flexible, faithful simulation frameworks to fill this gap, especially in scenarios where an experimental ground truth is challenging or infeasible to establish. For example, which genes and cell subpopulations are affected by batch effects cannot be controlled, independent of whether control samples might be used to quantify these effects. Similarly, intra- and inter-cluster effects are unclear, even if cluster annotations might be obtained through cell-sorting or manual annotation by an expert. And, effects on gene expression remain unknown, despite controlled perturbation or time-series studies through which discrete labels might be given. Taken together, although some level of ground truth may be experimentally attainable, simulations remain indispensable owing to (i) their feasibility and (ii) the information they provide (e.g., which genes and cell subpopulations are affected).

The most truthful model for real data is real data. Artificial data alterations (e.g., applying fold changes to a specified subset of gene expression means in certain subsets of cells) are unlikely to mimic biological differences. Even if founded on a thorough investigation of *realistic* changes, non-reference based simulations are difficult to evaluate, and conclusions drawn from *de novo* simulations in terms of method evaluations should be treated with caution.

While tools to evaluate the quality of simulated data exist, they are seldomly taken advantage of. For example, *scater* [55] offers a range of gene- and cell-level quality control summaries; *countsimQC* [56] can generate a comprehensive report comparing an input set of count matrices (e.g., real against synthetic data), and many dataset summaries are easy to compute and compare manually. Having such reference-simulation

comparisons available every time that a simulator is proposed or used, as well as in every (non-neutral) benchmark would add credibility to the results.

In addition to evaluating the faithfulness of simulated data, we investigated whether and to what extent benchmark results are affected by the simulator used. Our results suggest that method performances for integration and clustering of scRNA-seq data deviate from those obtained from real data; in addition, simulators that better mimic reference datasets do not necessarily yield more similar method comparison results. For example, *muscat* was among the highest ranked simulators in our study, but integration and clustering method ranking obtained from *muscat* simulations were rather inconsistent with those from real data. On the other hand, *SPsimSeq* ranked mediocre in terms of mimicking real datasets, but gave the most faithful integration method ranking. In the context of clustering, there was a consistent over-optimistic performance of methods, independent of the simulator used.

This discrepancy between the faithfulness of simulated data and benchmark results brings to question which set of summaries is sufficient to capture relevant data structure. Here, simulators were ranked by their average performance across summaries. However, many of these may be redundant (see below) or differ in their suitability to capture group-related structures (e.g., batch-/cluster-effects). Thus, simulators that are performant “overall” are not guaranteed to be suitable for evaluating methods for a specific task (e.g., integration/clustering), where global structure should take priority over gene-/cell-specific summaries. An open question that needs to be answered is what summaries are important for a given task.

Besides the capabilities each method has to offer and its performance, i.e., how realistic its simulations are, there are other criteria we did not explore thoroughly. For example, *splatter* offers a well-documented, easy-to-use framework that is both flexible and interpretable. While other methods might outperform *splatter*, they return parameters that are less applicable to benchmarking computational tools. For example, artificially introducing DE genes provides a binary ground truth (e.g., whether a gene is DE), whereas estimating and mimicking cluster effects might not (i.e., the user defines which genes are DE based on gene-wise parameters returned by the simulator).

Here, we have focused on methods that generate a single group or multiple groups of cells; in particular, we distinguished between “singular” (type n), multi-batch (type b), and multi-cluster (type k) datasets. However, there are various methods that are aimed at simulating data where gene expression profiles evolve along a discrete or continuous trajectory or time-course (e.g., *dyngen* [18], *PROSSTT* [19], *SERGIO* [20]). These have been applied in, for example, benchmarking methods for trajectory inference [12].

The combination of scRNA-seq with CRISPR/Cas9 genome editing has enabled the joint readout of gene expression and cell lineage barcodes [57]. Salvador-Martínez et al. [58] have proposed a simulator for lineage barcode data that, however, does not generate gene expression data. A recent method, *TedSim* [59], is capable of outputting combined readouts and can be used to study tools for either or both data types, including more genuine investigation of trajectory inference methods.

Most of these methods employ fairly sophisticated and well-designed models for data generation, but require a complex set of inputs that is specific to each method and difficult to justify. Meanwhile, very few trajectory simulators support the estimation of

simulation parameters from a reference dataset, making it challenging to evaluate them and opening the question of how faithful performance assessments based on them are. Overall, validating the faithfulness of synthetically generated trajectories in single-cell data remains challenging.

Conclusions

Taken together, while a set of performant methods to generate synthetic scRNA-seq data exist, current methods are (i) limited in the level of complexity they are able to accommodate; (ii) often reliant—in full or in part—on inputs by the user to introduce (artificial) expression differences; and (iii) more or less suitable to evaluate other tools, depending on the data characteristics they can capture faithfully. Secondly, simulation-based benchmark studies are affected by the simulator used, and more performant simulators do not necessarily yield more reliable readouts of, e.g., integration and clustering methods. And thirdly, the chosen quality control summaries and their prioritization have an impact on the assessment of simulations and, consequently, the conclusions drawn from them. Thus, identifying the nature, number, and significance of summaries to faithfully capture scRNA-seq data structure warrants future work in order to improve method evaluations.

Methods

Reference datasets

Each reference dataset was retrieved from a publicly available source, including public GitHub repositories, Bioconductor's *ExperimentHub* [60], and databases such as the Gene Expression Omnibus (GEO). Raw data were formatted into objects of class *SingleCellExperiment* [61, 62] and, with few exceptions, left *as is* otherwise. Datasets cover various organisms, tissue types, technologies, and levels of complexity (i.e., number of genes and cells, clusters, and/or batches and/or experimental conditions). A summary of each dataset's characteristics and source is given in Additional file 1: Table S1.

References underwent minimal filtering in order to remove groups (clusters, batches) with an insufficient number of cells, as well as genes and cells of low quality (e.g., low detection rate, few counts overall). Secondly, we drew various subsets from each reference to retain a reduced number of observations (genes and cells), as well as a known number of batches, clusters, or neither (see Additional file 1: Table S2 and Additional file 1: Sec. 6.1).

Simulation methods

With few exceptions, methods were run using default parameters and, if available, following recommendations given by the authors in the corresponding software documentation. All packages were available from a public GitHub repository, through CRAN, or Bioconductor [63]. A brief overview of each method's model framework and support for parallelization is given in Table 1. For the explicit arguments used for parameter estimation and data simulation, we refer to the method wrappers available at <https://github.com/HelenaLC/simulation-comparison> [64] (snapshot on Zenodo [65]).

Quality control summaries

We computed a set of five summaries at the gene-level: average and variance of logCPM, coefficient of variation, detection frequency (i.e., proportion of cells with non-zero count), and gene-to-gene correlation of logCPM. Here, logCPM correspond to log1p-transformed counts per million computed with *scater's calculateCPM* function [55]. We also computed six summaries at the cell-level: library size (i.e., total counts), detection frequency (i.e., fraction of detected genes), cell-to-cell correlation (of logCPM), cell-to-cell distance (Euclidean, in PCA space), the number of times a cell occurs as a k-nearest neighbor (KNN), and local density factors [52] (LDF) that represent a relative measure of a cell's local density compared to those in its neighborhood (in PCA space), and aim to quantify group (batch/cluster for type *b/k*) structure. For PC-based summaries, we ran *scran's* [66] *modelGeneVar* (on logCPM) and *getTopHVGs* to select the $n = 500$ most highly variable features, and *scater's calculatePCA* to compute their first $n_{components} = 50$ PCs. For datasets other than type *n*, each summary was computed for each of three cell groupings: globally (i.e., across all cells), at the batch-, and at the cluster-level. Three additional summaries—the percent variance explained (PVE) [67] at the gene-, and the cell-specific mixing score (CMS) [52], and silhouette width [51] at the cell-level—were computed globally. Here, the PVE corresponds to the fraction of expression variance accounted for by a cell's group assignment (batch/cluster for type *b/k*). Summaries are described in more detail in Additional file 1: Table S3.

Evaluation statistics

For each reference-simulation pair of summaries, we computed the Kolmogorov-Smirnov (KS) test statistic using the *ks.test* function of the *stats* R package, and the Wasserstein metric using the *wasserstein_metric* function of the *waddR* R package [68]. In addition, we computed the two-dimensional KS statistic [69] (using *MASS' kde2d* function [70]) and earth mover's distance (EMD) [71] (using *emdist's emd2d* function [72]) between relevant pairs of summaries, i.e., between unique combinations of gene- and cell-level summaries, respectively, excluding global summaries (PVE, CMS, and silhouette width) as well as gene-to-gene and cell-to-cell correlations. One- and two-dimensional evaluations are detailed under Additional file 1: Sec. 4.

Runtime evaluation

To quantify simulator runtimes, we selected one reference per type and drew five random subsets of 400–4000 genes (fixing the number of cells) and 100–2600 cells (fixing the number of genes). For each method and subset (eight in total), we separately measured the time required for parameter estimation and data simulation. For each step, we set a time limit of 10^6 s after which computations were interrupted.

Integration evaluation

Integration methods were implemented as in Chazarra-Gil et al. [73] (see Additional file 1: Sec. 5.1), including *ComBat* [74], *Harmony* [75], *fastMNN* and *mnnCorrect* [76], *limma* [77], and *Seurat* [78]. To evaluate method performances, cell-specific mixing scores (CMS) and the difference in local density factors (Δ LDF) were computed using the *cms* and *ldfDiff* function, respectively, of the *CellMixS* package [52]. To make metrics

more interpretable and comparable across datasets, we (i) subtracted 0.5 to center CMS at 0 (denoted CMS*) and (ii) centered (at 0) and scaled (to range 1) Δ LDF (denoted Δ LDF*). Overall integration scores correspond to the unweighted average of CMS* and Δ LDF*. Thus, for all three metrics, a value of 0 indicates “good” mixing for a given cell. When aggregating results (e.g., for heatmap visualizations), metrics were first averaged across cells within each batch and, secondly, across batches.

Clustering evaluation

Clustering methods were implemented as in Duó et al. [31] (see Additional file 1: Sec. 5.2), including *CIDR* [79], hierarchical clustering (HC), and k-means (KM) [80] on PCA, *pcaReduce* [81], *SC3* [82], *Seurat* [78], *TSCAN* [83], and KM on t-SNE [84]. If applicable, the number of clusters was set to match the number of true (annotated respective simulated) clusters. To evaluate the performance of each method, we matched true and predicted cluster labels using the Hungarian algorithm [54] and computed cluster-level precision, recall, and F1 score (the harmonic mean of precision and recall).

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s13059-023-02904-1>.

Additional file 1: Supplementary Text. **Tables S1-2** (reference datasets), **Table S3** (quality control summaries), and **Table S4** (method parameters); description of evaluation statistics (for 1/2D comparisons, clustering and integration), computational *Snakemake* workflow, and supplementary R data objects.

Additional file 2: Figures S1-3 (t-SNEs of reference datasets), **Figs. S4-10** (comparison of 1/2D test statistics), **Figs. S11-13** (runtime and memory usage), **Figs. S14-16** (integration results), **Figs. S17-24** (t-SNEs of integrated datasets), **Figs. S25-28** (meta-analysis of summaries).

Additional file 3. Review History.

Acknowledgements

The authors thank members of the Robinson Lab at the University of Zurich for valuable feedback on methodology, benchmarking, and exposition.

Peer review information

Andrew Cosgrove was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history

The review history is available as Additional file 3.

Authors' contributions

HLC implemented method comparisons with significant contributions from SM. CS assisted in several conceptual aspects of the benchmark. HLC, SM, and MDR drafted the manuscript with feedback from CS. All authors read and approved the final paper.

Funding

This work was supported by the Swiss National Science Foundation (grant Nos. 310030_175841 and CRSII5_177208) and the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation (grant No. 2018-182828). MDR acknowledges support from the University Research Priority Program Evolution in Action at the University of Zurich.

Availability of data and materials

Raw data are available from 10x Genomics, ArrayExpress, Broad Institute's Single Cell Portal (SCP), and the Gene Expression Omnibus (GEO) database. Specifically, data from the following studies and accession numbers were used: Tian et al. [85] (GSE118767 [86]), Gierahn et al. [87] (GSE92495 [88]), Ding et al. [89] (SCP425 [90]), Han et al. [91] (GSE108097 [92]), Kang et al. [93] (GSE96583 [94]), Koh et al. [95] (GSE85066 [96]), Mereu et al. [97] (GSE133549 [98]), Oetjen et al. [99] (GSE120221 [100]), Tabula Muris Consortium et al. [101] (GSE109774 [102]), Tung et al. [103] (GSE77288 [104]), Zheng et al. [105] (10x Genomics [106]), and human pancreatic islet cell datasets across 5 technologies (GSE81076 [107], GSE85241 [108], GSE86469 [109], E-MTAB-5061 [110]).

R objects (.rds files) to reproduce key results of this study are available from Zenodo [111]. These include global and gene- and cell-level quality control summaries of reference and simulated data for different cell groupings, one- and two-dimensional test statistics across all datasets and methods, clustering and integration results for reference and simulated data, and runtimes for 5 replicates per gene- and cell-subsets for one dataset per type; see Additional file 1: Sec. 7 for a comprehensive description.

All analyses were run in R v4.1.0 [112], with Bioconductor v3.13 [63]. The computational workflow was implemented using *Snakemake* v5.5.0 [113], with Python v3.6.8. Package versions used throughout this study are captured in the *session_info.txt* file on Zenodo [111]. All code to reproduce the results presented herein is accessible on GitHub [64] (snapshot on Zenodo [65]). Workflow structure, code organization, and script naming schemes are described in more detail on the repository's landing page.

Both the supplementary data [111] and code [65] described above are under a Creative Commons Attribution 4.0 International license.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 23 November 2021 Accepted: 20 March 2023

Published: 29 March 2023

References

1. Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods*. 2009;6(5):377–82.
2. Svensson V, da Veiga Beltrame E, Pachter L. A curated database reveals trends in single-cell transcriptomics. *Database*. 2020;2020:baaa073.
3. Zappia L, Phipson B, Oshlack A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Comput Biol*. 2018;14(6): e1006245.
4. Zappia L, Theis FJ. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol*. 2021;22(1):301.
5. Mangul S, Martin LS, Hill BL, Lam AKM, Distler MG, Zelikovsky A, et al. Systematic benchmarking of omics computational tools. *Nat Commun*. 2019;10(1):1393.
6. Weber LM, Saelens W, Cannoodt R, Soneson C, Hapfelmeier A, Gardner PP, et al. Essential guidelines for computational method benchmarking. *Genome Biol*. 2019;20(1):125.
7. Buchka S, Hapfelmeier A, Gardner PP, Wilson R, Boulesteix AL. On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biol*. 2021;22:152.
8. Boulesteix AL, Groenwold RH, Abrahamowicz M, Binder H, Briel M, Hornung R, et al. Introduction to statistical simulations in health research. *BMJ Open*. 2020;10(12):e039921.
9. Dal Molin A, Baruzzo G, Di Camillo B. Single-cell RNA-sequencing: assessment of differential expression analysis methods. *Front Genet*. 2017;8:62.
10. Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. *Nat Methods*. 2018;15(4):255–61.
11. Wang T, Li B, Nelson CE, Nabavi S. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinformatics*. 2019;20(1):40.
12. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. *Nat Biotechnol*. 2019;37(5):547–54.
13. Tran HTN, Ang KS, Chevrier M, Zhang X, Lee NYS, Goh M, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol*. 2020;21(1):12.
14. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nat Methods*. 2022;19(1):41–50.
15. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol*. 2017;18(1):174.
16. Assefa AT, Vandesompele J, Thas O. SPsimSeq: semi-parametric simulation of bulk and single cell RNA sequencing data. *Bioinformatics*. 2020.
17. Cao Y, Yang P, Yang JYH. A benchmark study of simulation methods for single-cell RNA sequencing data. *bioRxiv*. 2021. p. 2021.06.01.446157.
18. Cannoodt R, Saelens W, Deconinck L, Saeys Y. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat Commun*. 2021;12(1):3942.
19. Papadopoulos N, Gonzalo PR, Söding J. PROSST: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes. *Bioinformatics*. 2019;35(18):3517–9.
20. Dibaieina P, Sinha S. SERGIO: a single-cell expression simulator guided by gene regulatory networks. *Cell Syst*. 2020;11(3):252–271.e11.
21. Germain PL, Sonrel A, Robinson MD. pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biol*. 2020;21(1):227.
22. Xi NM, Li JJ. Benchmarking computational doublet-detection methods for single-cell RNA sequencing data. *Cell Syst*. 2021;12(2):176–194.e6.
23. Vieth B, Parekh S, Ziegenhain C, Enard W, Hellmann I. A systematic evaluation of single cell RNA-seq analysis pipelines. *Nat Commun*. 2019;10(1):4667.
24. Yip SH, Sham PC, Wang J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief Bioinform*. 2019;20(4):1583–9.

25. Andrews TS, Hemberg M. False signals induced by single-cell imputation. *F1000Research*. 2018;7:1740.
26. Cole MB, Risso D, Wagner A, DeTomaso D, Ngai J, Purdom E, et al. Performance assessment and selection of normalization procedures for single-cell RNA-seq. *Cell Syst*. 2019;8(4):315–328.e8.
27. Gilbert AC, Vargo A. Comparison of marker selection methods for high throughput scRNA-seq data. *bioRxiv*. 2019. p. 679761.
28. Krzak M, Raykov Y, Boukouvalas A, Cutillo L, Angelini C. Benchmark and parameter sensitivity analysis of single-cell RNA sequencing clustering methods. *Front Genet*. 2019;10:1253.
29. Sun S, Zhu J, Ma Y, Zhou X. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome Biol*. 2019;20(1):269.
30. Chen W, Zhang S, Williams J, Ju B, Shaner B, Easton J, et al. A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing. *Comput Struct Biotechnol J*. 2020;18:861–73.
31. Duò A, Robinson MD, Sonesson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*. 2020;7:1141 [v3].
32. Heiser CN, Lau KS. A quantitative framework for evaluating single-cell data structure preservation by dimensionality reduction techniques. *Cell Rep*. 2020;31(5):107576.
33. Huang Q, Liu Y, Du Y, Garmire LX. Evaluation of cell type annotation R packages on single-cell RNA-seq data. *Genomics Proteomics Bioinforma*. 2020;19(2):267–81.
34. Zhang L, Zhang S. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM Trans Comput Biol Bioinforma*. 2020;17(2):376–89.
35. Li R, Guan J, Zhou S. Single-cell RNA-seq data clustering: a survey with performance comparison study. *J Bioinforma Comput Biol*. 2020;18(4):2040005.
36. Westoby J, Herrera MS, Ferguson-Smith AC, Hemberg M. Simulation-based benchmarking of isoform quantification in single-cell RNA-seq. *Genome Biol*. 2018;19(1):191.
37. Vallejos CA, Marioni JC, Richardson S. BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS Comput Biol*. 2015;11(6):e1004333.
38. Tian J, Wang J, Roeder K. ESCO: single cell expression simulation incorporating gene co-expression. *Bioinformatics*. 2021;37(16):2374–81.
39. Zimmerman KD, Langefeld CD. Hierarchicell: an R-package for estimating power for tests of differential expression with single-cell data. *BMC Genomics*. 2021;22(1):1–8.
40. Crowell HL, Sonesson C, Germain PL, Calini D, Collin L, Raposo C, et al. muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nat Commun*. 2020;11(1):6077.
41. Su K, Wu Z, Wu H. Simulation, power evaluation and sample size recommendation for single-cell RNA-seq. *Bioinformatics*. 2020;36(19):4860–8.
42. Vieth B, Ziegenhain C, Parekh S, Enard W, Hellmann I. powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics*. 2017;33(21):3486–8.
43. Korthauer KD, Chu LF, Newton MA, Li Y, Thomson J, Stewart R, et al. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol*. 2016;17(1):222.
44. Li WW, Li JJ. A statistical simulator scDesign for rational scRNA-seq experimental design. *Bioinformatics*. 2019;35(14):i41–50.
45. Sun T, Song D, Li WW, Li JJ. scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome Biol*. 2021;22(1):163.
46. Qin F, Luo X, Xiao F, Cai G. SCRIP: an accurate simulator for single-cell RNA sequencing data. *Bioinformatics*. 2021.
47. Baruzzo G, Patuzzi I, Di Camillo B. SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics*. 2020;36(5):1468–75.
48. Zhang X, Xu C, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. *Nat Commun*. 2019;10(1):2611.
49. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert JP. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat Commun*. 2018;9(1):284.
50. Van den Berge K, Sonesson C, Love MI, Robinson MD, Clement L. zinger: unlocking RNA-seq tools for zero-inflation and single cell applications. *bioRxiv*. 2017:157982.
51. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*. 1987;20:53–65.
52. Lütge A, Zyrpych-Walczak J, Brykczynska Kunzmann U, Crowell HL, Calini D, Malhotra D, et al. Cell MixS: quantifying and visualizing batch effects in single-cell RNA-seq data. *Life Sci Alliance*. 2021;4(6):e202001004.
53. Massey FJ. The Kolmogorov-Smirnov test for goodness of fit. *J Am Stat Assoc*. 1951;46(253):68–78.
54. Kuhn HW. The Hungarian method for the assignment problem. *Nav Res Logist*. 2005;52(1):7–21.
55. McCarthy DJ, Campbell KR, Lun ATL, Wills QF. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*. 2017;33(8):1179–86.
56. Sonesson C, Robinson MD. Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics*. 2018;34(4):691–2.
57. Raj B, Wagner DE, McKenna A, Pandey S, Klein AM, Shendure J, et al. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nat Biotechnol*. 2018;36(5):442–50.
58. Salvador-Martínez I, Grillo M, Averof M, Telford MJ. Is it possible to reconstruct an accurate cell lineage using CRISPR recorders? *elife*. 2019;8.
59. Pan X, Li H, Zhang X. TedSim: temporal dynamics simulation of single cell RNA-sequencing data and cell division history. *bioRxiv*. 2021.
60. Morgan M, Shepherd L. ExperimentHub: Client to access ExperimentHub resources. R package. 2016.
61. Lun A, Risso D, Korthauer K. SingleCellExperiment: S4 classes for single cell data. R package version. 2021;1.14.1.
62. Amezcua RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, et al. Orchestrating single-cell analysis with Bioconductor. *Nat Methods*. 2019;17:137–45.

63. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods*. 2015;12(2):115–21.
64. Crowell HL, Leonardo SM, Soneson C, Robinson MD. Snakemake workflow to benchmark scRNA-seq data simulators. GitHub. 2022. <https://github.com/HelenaLC/simulation-comparison>.
65. Crowell HL, Leonardo SM, Soneson C, Robinson MD. Supplementary Code for "The shaky foundations of simulating single-cell RNA sequencing data". Zenodo. 2022:10.5281/zenodo.6979699.
66. Lun ATL, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research*. 2016;5:2122 [v2].
67. Hoffman GE, Schadt EE. variancePartition: interpreting drivers of variation in complex gene expression studies. *BMC Bioinformatics*. 2016;17(1):483.
68. Scheffzik R, Flesch J, Goncalves A. Fast identification of differential distributions in single-cell RNA-sequencing data with waddR. *Bioinformatics*. 2021;37(19):3204–11.
69. Peacock JA. Two-dimensional goodness-of-fit testing in astronomy. *Mon Not R Astron Soc*. 1983.
70. Venables WN, Ripley BD. *Modern Applied Statistics with S*. 4th ed. New York: Springer; 2002.
71. Rubner Y. The earth mover's distance as a metric for image retrieval. *Int J Comput Vis*. 2000;40(2):99–121.
72. Urbanek S, Rubner Y. emdlist: Earth Mover's Distance. R package version. 2012;0.3-1.
73. Chazarra-Gil R, van Dongen S, Kiselev VY, Hemberg M. Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. *Nucleic Acids Res*. 2021;49(7):e42.
74. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*. 2007;8(1):118–27.
75. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods*. 2019;16(12):1289–96.
76. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol*. 2018;36(5):421–7.
77. Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res*. 2015;43(7):e47.
78. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol*. 2015;33(5):495–502.
79. Lin P, Troup M, Ho JWK. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol*. 2017;18(1):59.
80. Wong MA, Hartigan JA. Algorithm as 136: A k-means clustering algorithm. *J R Stat Soc: Ser C: Appl Stat*. 1979;28(1):100–8.
81. Žurauskienė J, Yau C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*. 2016;17:140.
82. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods*. 2017;14(5):483–6.
83. Ji Z, Ji H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res*. 2016;44(13):e117.
84. Maaten Lvd, Hinton G. Visualizing Data using t-SNE. *J Mach Learn Res*. 2008;9(Nov):2579–2605.
85. Tian L, Dong X, Freytag S, Lê Cao KA, Su S, JalalAbadi A, et al. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat Methods*. 2019;16(6):479–87.
86. Tian L, Dong X, Freytag S, Lê Cao KA, Su S, JalalAbadi A, et al. Data from 'Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments'. *Gene Expression Omnibus (GEO)*. 2018. p. GSE118767.
87. Gierahn TM, Wadsworth MH 2nd, Hughes TK, Bryson BD, Butler A, Satija R, et al. Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat Methods*. 2017;14(4):395–8.
88. Gierahn TM, Wadsworth MH 2nd, Hughes TK, Bryson BD, Butler A, Satija R, et al. Data from 'Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput'. *Gene Expression Omnibus (GEO)*. 2017. p. GSE92495.
89. Ding J, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, Marjanovic ND, et al. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nat Biotechnol*. 2020;38:737–46.
90. Ding J, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, Marjanovic ND, et al. Data from 'Systematic comparison of single-cell and single-nucleus RNA-sequencing methods'. *Single Cell Portal (Broad Institute)*; 2020. p. SCP425.
91. Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, et al. Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell*. 2018;172(5):1091–1107.e17.
92. Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, et al. Data from 'Mapping the Mouse Cell Atlas by Microwell-Seq'. *Gene Expression Omnibus (GEO)*. 2018. p. GSE108097.
93. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat Biotechnol*. 2018;36(1):89–94.
94. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, et al. Data from 'Multiplexed droplet single-cell RNA-sequencing using natural genetic variation'. *Gene Expression Omnibus (GEO)*. 2017. p. GSE96583.
95. Koh PW, Sinha R, Barkal AA, Morganti RM, Chen A, Weissman IL, et al. An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Sci Data*. 2016;3:160109.
96. Koh PW, Sinha R, Barkal AA, Morganti RM, Chen A, Weissman IL, et al. Data from 'An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development'. *Gene Expression Omnibus (GEO)*. 2016. p. GSE85066.
97. Mereu E, Lafzi A, Moutinho C, Ziegenhain C, McCarthy DJ, Álvarez-Varela A, et al. Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. *Nat Biotechnol*. 2020;38:747–55.
98. Mereu E, Lafzi A, Moutinho C, Ziegenhain C, McCarthy DJ, Álvarez-Varela A, et al. Data from 'Benchmarking single-cell RNA-sequencing protocols for cell atlas projects'. *Gene Expression Omnibus (GEO)*. 2019. p. GSE133549.

99. Oetjen KA, Lindblad KE, Goswami M, Gui G, Dagur PK, Lai C, et al. Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry. *JCI Insight*. 2018;3(23).
100. Oetjen KA, Lindblad KE, Goswami M, Gui G, Dagur PK, Lai C, et al. Data from 'Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry'. Gene Expression Omnibus (GEO). 2018. p. GSE120221.
101. Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*. 2018;562(7727):367–372.
102. Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, et al. Data from 'Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris'. Gene Expression Omnibus (GEO). 2018. p. GSE109774.
103. Tung PY, Blischak JD, Hsiao CJ, Knowles DA, Burnett JE, Pritchard JK, et al. Batch effects and the effective design of single-cell gene expression studies. *Sci Rep*. 2017;7:39921.
104. Tung PY, Blischak JD, Hsiao CJ, Knowles DA, Burnett JE, Pritchard JK, et al. Data from 'Batch effects and the effective design of single-cell gene expression studies'. Gene Expression Omnibus (GEO). 2016. p. GSE77288.
105. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. *Nat Commun*. 2017;8:14049.
106. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Data from 'Massively parallel digital transcriptional profiling of single cells'. 2017. <https://www.10xgenomics.com/resources/datasets>.
107. Grün D, Muraro MJ, Boisset JC, Wiebrands K, Lyubimova A, Dharmadhikari G, et al. Data from 'De Novo Prediction of Stem Cell Identity using Single-Cell Transcriptome Data'. Gene Expression Omnibus (GEO). 2016. p. GSE81076.
108. Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, et al. Data from 'A Single-Cell Transcriptome Atlas of the Human Pancreas'. Gene Expression Omnibus (GEO). 2016. p. GSE85241.
109. Lawlor N, George J, Bolisetty M, Kursawe R, Sun L, Sivakamasundari V, et al. Data from 'Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes'. Gene Expression Omnibus (GEO). 2016. p. GSE86469.
110. Segerstolpe Å, Palasantza A, Eliasson P, Andersson EM, Andréasson AC, Sun X, et al. Data from 'Single-Cell Transcriptome Profiling of Human Pancreatic Islets in Health and Type 2 Diabetes'. ArrayExpress (BioStudies). 2016. p. E-MTAB-5061.
111. Crowell HL, Leonardo SM, Sonesson C, Robinson MD. Supplementary Data for "The shaky foundations of simulating single-cell RNA sequencing data". Zenodo. 2022. p. 10.5281/zenodo.6980272.
112. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing; 2019.
113. Köster J, Rahmann S. Snakemake - a scalable bioinformatics workflow engine. *Bioinformatics*. 2012;28(19):2520–2.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.