

Tools and techniques for single-cell RNA-seq data

Luke Zappia
ORCID ID: 0000-0001-7744-8565

Doctor of Philosophy

November 2018

School of Biosciences
The University of Melbourne

Submitted in Total Fulfillment of the Requirements of the Degree of Doctor of
Philosophy

Abstract

The preface pretty much says it all.
Second paragraph of abstract starts here.

Declaration

This is to certify that:

- i. the thesis comprises only their original work towards the [name of the award] except where indicated in the preface;
- ii. due acknowledgement has been made in the text to all other material used; and
- iii. the thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices or that the thesis is [number of words] as approved by the Research Higher Degrees Committee.

Luke Zappia, November 2018

Preface

This is an example of a thesis setup to use the reed thesis document class (for LaTeX) and the R bookdown package, in general.

Acknowledgements

This template is based on `thesisdown` (<https://github.com/ismayc/thesisdown>) and makes use of `RMarkdown` (<https://rmarkdown.rstudio.com/>) and `bookdown` (<https://bookdown.org/yihui/bookdown/>). The LaTeX template is based on John Papandriopoulos' University of Melbourne thesis template (<https://github.com/jpap/phd-thesis-template>). Inspiration also comes from similar projects including `beaverdown` (<https://github.com/zkamvar/beaverdown>), `aggidown` (<https://github.com/ryanpeek/aggiedown>), `huskydown` (<https://github.com/benmarwick/huskydown>) and `jayhawkdown` (<https://github.com/wjakethompson/jayhawkdown>).

Contents

1	Introduction	1
1.1	RNA sequencing	1
1.1.1	Capture and reverse transcription	1
1.1.2	High-throughput sequencing	1
1.1.3	Analysis of RNA-seq data	1
1.2	Single-cell capture technologies	1
1.2.1	Droplet based cell capture	2
1.2.2	Unique Molecular Identifiers	3
1.2.3	Recent advances	3
1.3	Features of single-cell RNA-seq data	3
1.4	Analysing scRNA-seq data	3
1.4.1	Pre-processing and quality control	5
1.4.2	Integrating multiple datasets	6
1.4.3	Grouping cells	6
1.4.4	Ordering cells	7
1.4.5	Gene detection and interpretation	8
1.5	Kidney development	8
1.5.1	Structure and function	8
1.5.2	Stages of development	8
1.5.3	Growing kidney organoids	8
2	The scRNA-seq tools landscape	9
3	Simulating scRNA-seq data	11
3.1	Introduction	12
3.2	Splatter publication	12
4	Visualising clustering across resolutions	27
5	Analysis of kidney organoid scRNA-seq data	29
6	Conclusion	31
References		33

List of Tables

List of Figures

Chapter 1

Introduction

1.1 RNA sequencing

- Central dogma
- Why RNA-seq?

1.1.1 Capture and reverse transcription

- PolyA capture
- Ribosomal depletion

1.1.2 High-throughput sequencing

- Illumina sequencing
 - Sequence by synthesis

1.1.3 Analysis of RNA-seq data

- Experimental design
- Negative binomial
- Normalisation
- Differential expression testing

1.2 Single-cell capture technologies

- First protocol
- Fluidigm

The first scRNA-seq protocol was published in 2009, just a year after the first bulk RNA-seq publication. While this approach allowed measurements of the transcriptome in individual cells it required manual manipulation and was restricted to inspecting a few precious cells. Further studies quickly showed that cell types could be identified without sorting cells and approaches were developed to allow unbiased capture of the whole transcriptome. Since then many scRNA-seq protocols have been

developed including The first commercially available cell capture platform was the Fluidigm C1. This system uses microfluidics to passively separate cells into individual wells on a plate where they are lysed, reverse-transcribed and the collected cDNA is PCR amplified. After this stage the product is extracted from the plate and libraries prepared for Illumina sequencing. Most C1 data has been produced using a 96 well plate but more recently an 800 well plate has become available, greatly increasing the number of cells that can be captured at a time. One of the disadvantages of microfluidic cell capture technologies is that the chips have a fixed size window, meaning that only cells of a particular sizes can be captured in a single run. However as cells are captured in individual wells they can be imaged before lysis, potentially identifying damaged or broken cells, empty wells or wells containing more than one cell. Capturing multiple cells is a known issue with Macosko et al. finding that when preparing a mixture of mouse and human cells 30 percent of the resulting libraries contained transcripts from both species but only about a third of these doublets were visible in microscopy images[Macosko2015-rl]. The newer Polaris system from Fluidigm also uses microfluidics to capture cells but can select particular cells based on staining or fluorescent reporter expression and then hold them for up to 24 hours while introducing various stimuli. The cells can be imaged during this time before being lysed and prepared for RNA sequencing. This platform provides opportunities for a range of experiments that aren't possible using other capture technologies.

1.2.1 Droplet based cell capture

- Drop-seq
- Indrop
- 10x Chromium

The alternative approach is to capture cells in microfluidic droplets. Cells are introduced into a microfluidic device while at another input beads coated in primers enter. At the output of the device aqueous droplets are formed in a mineral oil. The device is designed such that cells and beads may arrive and be captured simultaneously. When this occurs the cells are lysed and reverse transcription begins. The primary advantage of droplet-based approaches is the ability to capture many more cells in a single run, up to tens of thousands. These approaches are also less selective about cell size and much cheaper per cell, although as the sequencing costs are constant studies that use droplet-based captures typically sequence individual cells at a lower depth. There are two DIY droplet-based platforms, Drop-seq[@Macosko2015-rl] and InDrops[@Klein2015-iw; @Zillionis2017-gu]. They differ in how the beads are made when the droplets are broken and some aspects of the chemistry but they can both be constructed on a lab bench from syringes, plungers, a microscope and a small custom-made microfluidic chip (Figure \\\ref{fig:indrops}). More recently 10x Genomics have commercialised the droplet capture method as their Chromium device, which automates much of the process[@Zheng2016-sj].

1.2.2 Unique Molecular Identifiers

- Why?
- How they work

There are various ways that protocols for these platforms can be modified, as evidenced by the number of published protocols including Smart-seq2[@Picelli2013-or; @Picelli2014-pc], CEL-seq[@Hashimshony2012-dh], CEL-seq2[@Hashimshony2016-yi], Quartz-seq[@Sasagawa2013-uc] and SCRIB-seq[@Soumillion2014-nk]. Two decisions researchers need to make when deciding how to process their samples is whether to use a UMI or full-length protocol and whether to include external spike-in controls. UMIs attempt to improve the quantification of gene expression by allowing the removal of PCR duplicates produced during amplification. Individual RNA molecules are randomly tagged with short nucleotide sequences as part of the capture process. After reverse-transcription, amplification, sequencing and alignment de-duplication can be performed by identifying reads with the same UMI that align to the same position (Figure \\\ref{fig:UMIs}). These reads should be PCR duplicates rather than reads that truly originate from expression of that gene. While the use of UMIs should improve the quality of the final data used for downstream analyses it also introduces some difficulties. Extra processing is required to match reads with aligned reads and remove duplicates, this is further complicated by the potential for sequencing errors in the UMI itself[@Smith2016-bt]. Analyses that require full coverage of a gene such as variant detection or de-novo assembly are not possible and statistical analyses that were designed using full-length data may not reliably transfer to a UMI dataset.

1.2.3 Recent advances

- New capture methods
- CITE-seq
- Cell hashing

1.3 Features of single-cell RNA-seq data

- Why use single-cell?
- Low counts
 - Dropout
 - Bursting
 - Biology

1.4 Analysing scRNA-seq data

Single-cell RNA-seq technologies have developed rapidly but there are still many challenges. Existing protocols are inefficient, capturing only around

10 percent of transcripts in a cell[@Grun2014-zn]. Combined with the relatively low sequencing depth that is commonly used, this results in a limited sensitivity and an inability to reliably detect lowly expressed transcripts. The small amount of starting material also contributes to high levels of technical noise, complicating downstream analysis and making it difficult to detect biological differences[@Liu2016-wq]. All high-throughput scRNA-seq protocols and platforms require tissues to be dissociated into single-cell suspensions before capture. This step can be non-trivial. Some tissues or cell types may not readily separate and the treatments used to break them down may effect the health of the cells and therefore their transcriptional profiles. Additionally some cell types may be too big or have other physical characteristics that prevent them being captured using current methods. Cells may also be damaged during processing, or missing or multiple cells may have been sequenced, making quality control an important consideration.

As well as introducing technical noise, the small amounts of starting material and low sequencing depth mean there are many occasions where there is no measured expression for a particular gene in a particular cell. Some of these zero counts are due to the biology we wish to study, for example we expect different cell types to express different genes, but there are additional biological factors such as the cell cycle, transcriptional bursting and environmental interactions which cause genuine differences in expression between cells performing the same function. On top of this are the technical effects that have already been discussed including that existing protocols may not reliably capture all the RNA present, resulting in "dropout" events where a gene is expressed in a sample but not observed in the sequencing data. These zeros can make analysis difficult as methods must account for the missing information and they may violate the assumptions of existing approaches. For example the DEseq2 package[@Love2014-tw] has proven successful for detecting differential expression in bulk RNA-seq but it relies on the presence of genes without any zero counts, and therefore typically fails on scRNA-seq data. Examples of scRNA-seq analysis packages that tackle this problem are Zero Inflated Factor Analysis (ZIFA) [@Pierson2015-qp], which explicitly models dropout as it affects dimensionality reduction, and Clustering through Imputation and Dimensionality Reduction (CIDR) [@Lin2016-yu], which implicitly imputes zeros as it clusters cells.

Bulk RNA-seq experiments are usually conducted on predefined groups of samples for example cancer cells and normal tissue, different tissue types or treated and control cells. Some scRNA-seq experiments are done in a similar way where cells are sorted into known types based on surface markers or selected at a series of time points, but often they are more exploratory. Many of the current studies have taken samples of developing or mature tissues and attempted to identify what cell types are present[@Zeisel2015-rd; @Patel2014-bl; @Treutlein2014-wd; @Usoskin2015-fz; @Buettnner2015-rq; @Klein2015-iw;

@Trapnell2014-he]. This requires a new set of analysis techniques to be developed which attempt to identify cell types. Success at this task is crucial to the reliability of results from more developed analysis methods such as cell type assignment. Additionally it is impossible to truly replicate an individual cell, therefore thought must be given as to what constitutes a replicate for the purposes of statistical analysis.

1.4.1 Pre-processing and quality control

- Alignment
- Droplet selection
- UMIs
- Doublet detection
- Bad cells
- Gene filtering
- Cell ranger
- scater

As scRNA-seq data has become available there has been a rapid development of bioinformatics tools attempting to unlock its potential. Currently there are at least 80 available software packages that have been designed specifically for the analysis of scRNA-seq data, the majority of which have been published in peer-reviewed journals or as preprints. A table of scRNA-seq software is available at [<https://goo.gl/4wcVwn>()]. Prior to analysis the sequencing reads from an scRNA-seq experiment are processed in much the same way as a bulk experiment. Typically there is some quality control of the raw reads, reads aligned to a reference genome and the number of reads overlapping annotated features (genes or transcripts) is counted. Alternatively, for full-length reads, probabilistic quantification methods such as kallisto[@Bray2016-tm] or Salmon[@Patro2015-kl] can be used. These approaches can greatly improve processing time which is important when there may be tens of thousands of samples but they are currently incompatible with UMI protocols. When using conventional alignment UMI samples need extra processing with tools such as UMI-tools[@Smith2016-bt] or umis[@Svensson2016-eg]. The resulting gene by cell matrix of expression values is the starting point for most analysis.

Quality control of individual cells is important as most experiments will contain some low-quality cells that could be uninformative or misleading. Quality control can be performed on various levels: on the quality scores of the reads themselves, how or where the reads align or features of the expression matrix such as the total expression, expression of spike-ins or expression of particular genes. The Cellity package attempts to do this by inspecting a set of biological and technical features and using principal component analysis or machine learning methods to distinguish between high and low-quality cells[@Ilicic2016-wy]. However the authors found that many of the features were not cell type specific and more work needs to be done to make this approach more accurate.

generally applicable. Jiang, Thomson and Stewart take a different approach, assuming that expression outliers are associated with poor sequencing quality[@Jiang2016-ys]. The scater package[@McCarthy2016-cw] emphasises a more exploratory approach to quality control. While it cannot automatically detect low-quality cells, scater provides a convenient object for storing scRNA-seq data with functions for plotting associated features, making it easy for the user to define their own filtering thresholds. Plate-based platforms such as the Fluidigm C1 can have additional biases based on the location of individual wells. The OEFinder package attempts to identify and visualise these "ordering effects"[@Leng2016-it].

1.4.2 Integrating multiple datasets

- Why?
- Seurat CCA
- New methods

Technical variation is a known problem in high-throughput studies and Kim et al predict that only 17.8 percent of allele-specific expression is due to biological variation with the rest being technical noise[@Kim2015-mo]. Effective normalisation has been shown to be a crucial aspect of analysis for bulk RNA-seq datasets, but how normalisation should be applied to single-cell datasets is yet to be clearly established. Some studies use simple transformations like Reads (or Fragments) Per Kilobase per Million (RPKM/FPKM) [@Mortazavi2008-vu] or Transcripts Per Million (TPM) [@Wagner2012-qf] which correct for library size and gene length. Alternatively, normalisation methods designed for detecting differential expression in bulk samples such as the Trimmed Mean of M-Values (TMM) [@Robinson2010-ll] or the DESeq method[@Anders2010-pq] can be applied, but it is unclear how suitable they are for the single-cell context. Most of the methods that have been developed specifically for estimating technical variance in scRNA-seq data make use of spike-ins. Brennecke et al.[@Brennecke2013-pt], Ding et al.[@Ding2015-ht] and Grn, Kester and van Oudenaarden[@Grun2014-zn] all propose methods for estimating technical variance using spike-ins, as does Bayesian Analysis of Single-Cell Sequencing data (BASiCS) [@Vallejos2015-ef]. However, using spike-ins for normalisation relies on the assumption that they properly capture the dynamics of the underlying datasets, and even if this is the case is it restrictive as they are not compatible with all current sequencing protocols. Lun, Bach and Marioni don't make use of spike-ins, instead using a pooling approach to compensate for the large number of zero counts, where expression levels are summed across similar cells before calculating normalisation factors that are deconvolved back to the individual cell level[@Lun2016-mq].

1.4.3 Grouping cells

- Clustering

- Seurat
- Other approaches

Once a set of high-quality cells has been established the true analysis can begin. Many of the current packages focus on the task of assigning cells to groups before applying more traditional differential expression testing. This approach makes sense for a sample with a defined set of mature cell types as taken by tools such as SINGle CELl RNA-seq profiling Analysis (SINCERA) [@Guo2015-mf], Single-Cell Consensus Clustering (SC3) [@Kiselev2016-Seurat] [@Satija2015-or], single-cell latent variable model (scLVM) [@Buettnner2015-rq] and Spanning-tree Progression Analysis of Density-normalised Events (SPADE) [@Anchang2016-vo], as well as BackSPIN which was used to identify nine cell types and 47 distinct subclasses in the mouse cortex and hippocampus [@Zeisel2015-rd]. These tools attempt to cluster similar cells together based on their expression profiles, forming groups of cells of the same type. Often a dimensionality reduction step is included which can help to remove some of the noise present in scRNA-seq data. Once groups of cells are identified many of these packages can test genes for changes in expression, identifying genes that are differentially expressed across the groups, or maybe genes that are expressed in a single group. These genes can be used to identify which cell types each group represents, or alternatively known marker genes can be inspected.

1.4.4 Ordering cells

- Pseudotime
- Monocle
- Other approaches

In other situations, for example where stem cells are differentiating into mature cell types, it may be more appropriate to order cells along a continuous trajectory from one cell type to another. Trajectory analysis was pioneered by Monocle which used dimensionality reduction and computation of a minimum spanning tree to explore a model of skeletal muscle differentiation[@Trapnell2014-he]. Since then the Monocle algorithm has been updated and a range of others developed, including TSCAN[@Ji2016-ws], SLICER[@Welch2016-cw], CellTree[@DuVerle2016-ni], Sincell[@Julia2015-zc] and Mpath[@Chen2016-kx]. In their review of methods for trajectory inference, Cannoodt, Saelens and Saeys break the process into two parts, dimensionality reduction and then trajectory modelling (Figure \\\ref{fig:trajectory-inference})[@Cannoodt2016-iv]. Dimensionality reduction consists of calculating similarities between cells, projecting onto lower dimensions using manifold learning techniques such as PCA or t-SNE[@Maaten2008-ne] then clustering cells or constructing a graph between them. The trajectory is then formed by finding a path between cells and ordering the cells along it. Once a path has been inferred important genes can be

identified by looking for those that change expression over the course of the path. These genes can be important in their own right as they describe the biology of the path, but that can also be used to identify cell types at the points of the path or where the path branches. Deciding which assignment approach is most appropriate depends on the source of that data and the questions you intend to ask. There are currently no studies comprehensively comparing the performance of different methods for each approach.

1.4.5 Gene detection and interpretation

- DE
- Marker genes
 - Alternatives - Gini, classifiers
- Reviews
- Classification

1.5 Kidney development

1.5.1 Structure and function

- Kidney structure
- Nephron structure
- Important cell types

1.5.2 Stages of development

- Lineage
- Important genes

1.5.3 Growing kidney organoids

- Why?
 - Disease modelling
- Protocol
- Growth factors
- Characterisation
- Reproducibility

Chapter 2

The scRNA-seq tools landscape

Chapter 3

Simulating scRNA-seq data

3.1 Introduction

3.2 Splatter publication

Zappia et al. *Genome Biology* (2017) 18:174
DOI 10.1186/s13059-017-1305-0

Genome Biology

METHOD Open Access

 CrossMark

Splatter: simulation of single-cell RNA sequencing data

Luke Zappia^{1,2} , Belinda Phipson¹  and Alicia Oshlack^{1,2*} 

Abstract

As single-cell RNA sequencing (scRNA-seq) technologies have rapidly developed, so have analysis methods. Many methods have been tested, developed, and validated using simulated datasets. Unfortunately, current simulations are often poorly documented, their similarity to real data is not demonstrated, or reproducible code is not available. Here, we present the Splatter Bioconductor package for simple, reproducible, and well-documented simulation of scRNA-seq data. Splatter provides an interface to multiple simulation methods including Splat, our own simulation, based on a gamma-Poisson distribution. Splat can simulate single populations of cells, populations with multiple cell types, or differentiation paths.

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

Background

The first decade of next-generation sequencing has seen an explosion in our understanding of the genome [1]. In particular, the development of RNA sequencing (RNA-seq) has enabled unprecedented insight into the dynamics of gene expression [2]. Researchers now routinely conduct experiments designed to test how gene expression is affected by various stimuli. One limitation of bulk RNA-seq experiments is that they measure the average expression level of genes across the many cells in a sample. However, recent technological developments have enabled the extraction and amplification of minute quantities of RNA, allowing sequencing to be conducted on the level of single cells [3]. The increased resolution of single-cell RNA-seq (scRNA-seq) data has made a range of new analyses possible.

As scRNA-seq data have become available there has been a rapid development of new bioinformatics tools attempting to unlock its potential. Currently there are at least 120 software packages that have been designed specifically for the analysis of scRNA-seq data, the majority of which have been published in peer-reviewed

bulk RNA-seq experiment. In a bulk experiment, the groups of samples are known and a common task is to test for genes that are differentially expressed (DE) between two or more groups. In contrast, the groups in a single-cell experiment are usually unknown and the analysis is often more exploratory.

Much of the existing software focuses on assigning cells to groups based on their expression profiles (clustering) before applying more traditional DE testing. This approach is taken by tools such as SC3 [5], CIDR [6], and Seurat [7] and is appropriate for a sample of mature cells where it is reasonable to expect cells to have a particular type. In a developmental setting, for example, where stem cells are differentiating into mature cells, it may be more appropriate to order cells along a continuous trajectory from one cell type to another. Tools such as Monocle [8], CellTree [9], and Sincell [10] take this approach, ordering cells along a path, then identifying patterns in the changes of gene expression along that path.

Another defining characteristic of scRNA-seq data is its sparsity; typically expression is only observed for

or more SCESet objects, combines them (keeping any cell or gene-level information that is present in all of them) and produces a series of diagnostic plots comparing aspects of scRNA-seq data. The combined datasets are also returned, making it easy to produce additional comparison plots or statistics. Alternatively, one SCESet can be designated as a reference, such as the real data used to estimate parameters, and the difference between the reference and the other datasets can be assessed. This approach is particularly useful for comparing how well simulations recapitulate real datasets. Examples of these comparison plots are shown in the following sections.

Simulation models

Splatter provides implementations of our own simulation model, Splat, as well as several previously published simulations. The previous simulations have either been published as R code associated with a paper or as functions in existing packages. By including them in Splatter, we have made them available in a single place in a more accessible way. If only a script was originally published, such as the Lun [18] and Lun 2 [19] simulations, the simulations have been re-implemented in Splatter. If the simulation is available in an existing R package, for example, scDD [20] and BASiCS [21], we have simply written wrappers that provide consistent input and output but use the package implementation. We have endeavored to keep the simulations and estimation procedures as close as possible to what was originally published while providing a consistent interface within Splatter. The six different simulations currently available in Splatter are described below.

Simple

The negative binomial is the most common distribution used to model RNA-seq count data, as in the edgeR [22] and DESeq [23] packages. The Simple simulation is a basic implementation of this approach. A mean expression level for each gene is simulated using a gamma distribution and the negative binomial distribution is used to generate a count for each cell based on these means, with a fixed dispersion parameter (default = 0.1; Additional file 1: Figure S1). This simulation is primarily included as a baseline reference and is not intended to accurately reproduce many of the features of scRNA-seq data.

Lun

Published in “Pooling across cells to normalize single-cell RNA sequencing data with many zero counts” [18], the Lun simulation builds on the Simple simulation by adding a scaling factor for each cell (Additional file 1:

Figure S2). The cell factors are randomly sampled from a normal distribution with mean 1 and variance 0.5. The inverse- \log_2 transformed factors are used to adjust the gene means, resulting in a matrix in which each cell has a different mean. This represents the kinds of technical effects that scaling normalization aims to remove. The matrix of means is then used to sample counts from a negative binomial distribution, with a fixed dispersion parameter. This simulation can also model differential expression between multiple groups with fixed fold changes.

Lun 2

In “Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data” [19] Lun and Marioni extended the negative binomial model from the Lun simulation. This simulation samples input parameters from real data, with very little random sampling from statistical distributions. In the Lun 2 simulation the cell factors are replaced with a library size factor and an additional level of variation is added by including a batch effects factor. While the library size factor acts on individual cells the batch effects are applied to groups of cells from the same batch. This simulation is thus highly specific to the scenario when there are known batch effects present in the data, for example, Fluidigm C1 plate effects. Differential expression can be added between two sets of batches and the user can choose to use a zero-inflated negative binomial (ZINB) model. Counts are simulated from a negative binomial using the library size and plate factor adjusted gene means and gene-wise dispersion estimates obtained from the real data. If the ZINB model is chosen, zero inflated estimates of gene means and dispersions are used instead. An additional step then randomly sets some counts to zero, based on the gene-wise proportions of zeros observed in the data. Additional file 1: Figure S3 shows the model assumptions and parameters for this simulation.

scDD

The scDD package aims to test for differential expression between two groups of cells but also more complex changes such as differential distributions or differential proportions [20]. This is reflected in the scDD simulation, which can contain a mixture of genes simulated to have different distributions, or differing proportions where the expression of the gene is multi-modal. This simulation also samples information from a real dataset. As the scDD simulation is designed to reproduce a high quality, filtered dataset, it only samples from genes with less than 75% zeros. As a result, it only simulates relatively highly expressed genes. The Splatter package

simply provides wrapper functions to the simulation function in the scDD package, while capturing the necessary inputs and outputs needed to compare to other simulations. The full details of the scDD simulation are described in the scDD package vignette [24].

BASiCS

The BASiCS package introduced a model for separating variation in scRNA-seq data into biological and technical components based on the expression of external spike-in controls [21]. This model also enables cell-specific normalization and was extended to detect differential expression between groups of cells [25]. Similar to the scDD simulation, Splatter provides a wrapper for the BASiCS simulation function, which is able to produce datasets with both endogenous and spike-in genes as well as multiple batches of cells. As the BASiCS simulation contains both biological and technical variation, it can be used to test the ability of methods to distinguish between the two.

Splat

We have developed the Splat simulation to capture many features observed in real scRNA-Seq data, including high expression outlier genes, differing sequencing depths (library sizes) between cells, trended gene-wise dispersion, and zero-inflation. Our model uses parametric distributions with hyper-parameters estimated from real data (Fig. 1). The core of the Splat simulation is the gamma-Poisson hierarchical model where the mean expression level for each gene i , $i = 1, \dots, N$, is simulated from a gamma distribution and the count for each cell j , $j = 1, \dots, M$, is subsequently sampled from a Poisson distribution, with modifications to include expression outliers and to enforce a mean-variance trend.

More specifically, the Splat simulation initially samples gene means from a Gamma distribution with shape α and rate β . While the gamma distribution is a good fit for gene means it does not always capture extreme expression levels. To counter this a probability (π^O) that a gene is a high expression outlier can be specified. We then add these outliers to the simulation by replacing the previously simulated mean with the median of the simulated gene means multiplied by an inflation factor. The inflation factor is sampled from a log-normal distribution with location μ^O and scale σ^O .

The library size (total number of counts) varies within an scRNA-seq experiment and can be very different between experiments depending on the sequencing depth. We model library size using a log-normal distribution (with location μ^L and scale σ^L) and use the simulated library sizes (L_j) to proportionally adjust the gene means for each cell. This allows us to alter the

number of counts per cell independently of the underlying gene expression levels.

It is known that there is a strong mean-variance trend in RNA-Seq data, where lowly expressed genes are more variable and highly expressed genes are more consistent [26]. In the Splat simulation we enforce this trend by simulating the biological coefficient of variation (BCV) for each gene from a scaled inverse chi-squared distribution, where the scaling factor is a function of the gene mean. After simulating the BCV values we generate a new set of means (λ_{ij}) from a gamma distribution with shape and rate parameters dependent on the simulated BCVs and previous gene means. We then generate a matrix of counts by sampling from a Poisson distribution, with lambda equal to λ_{ij} . This process is similar to the simulation of bulk RNA-seq data used by Law et al. [27].

The high proportion of zeros is another key feature of scRNA-seq data [11], one cause of which is technical dropout. We use the relationship between the mean expression of a gene and the proportion of zero counts in that gene to model this process and use a logistic function to produce a probability that a count should be zero. The logistic function is defined by a midpoint parameter (x_0), the expression level at which 50% of cells are zero, and a shape parameter (k) that controls how quickly the probabilities change from that point. The probability of a zero for each gene is then used to randomly replace some of the simulated counts with zeros using a Bernoulli distribution.

Each of the different steps in the Splat simulation outlined above are easily controlled by setting the appropriate parameters and can be turned off when they are not desirable or appropriate. The final result is a matrix of observed counts Y_{ij} where the rows are genes and the columns are cells. The full set of input parameters is shown in Table 1.

Comparison of simulations

To compare the simulation models available in Splatter we estimated parameters from several real datasets and then generated synthetic datasets using those parameters. Both the standard and zero-inflated versions of the Splat and Lun 2 simulations were included, giving a total of eight simulations. We began with the Tung dataset which contains induced pluripotent stem cells from three HapMap individuals [28].

To reduce the computational time we randomly sampled 200 cells to use for the estimation step and each simulation consisted of 200 cells. Benchmarking showed a roughly linear relationship between the number of genes or cells and the processing time required (Additional file 1: Figures S4 and S5). The estimation procedures for the Lun 2 and BASiCS simulations are particularly time consuming; however, the Lun 2 estimation can be run using

Zappia et al. *Genome Biology* (2017) 18:174

Page 5 of 15

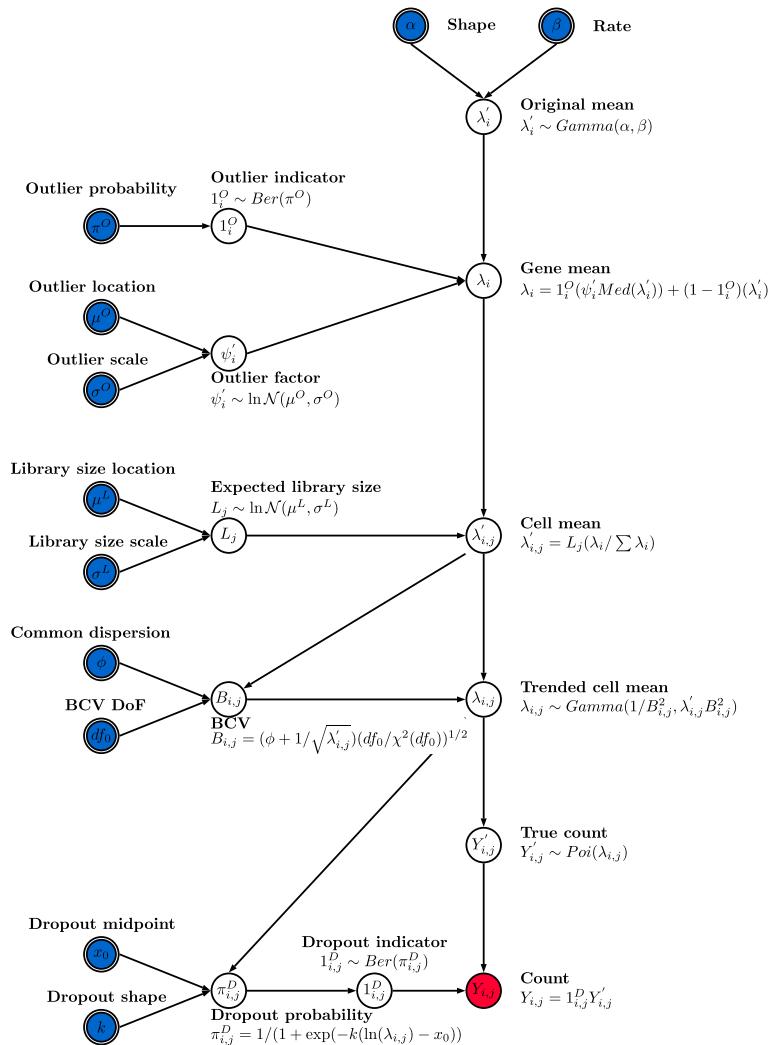


Fig. 1 The Splat simulation model. Input parameters are indicated with *double borders* and those that can be estimated from real data are shaded *blue*. *Red shading* indicates the final output. The simulation begins by generating means from a gamma distribution. Outlier expression genes are added by multiplying by a log-normal factor and the means are proportionally adjusted for each cell's library size. Adjusting the means using a simulated Biological Coefficient of Variation (*BCV*) enforces a mean-variance trend. These final means are used to generate counts from a Poisson distribution. In the final step dropout is (optionally) simulated by randomly setting some counts to zero, based on each gene's mean expression. *DoF* degrees of freedom

multiple cores unlike the BASiCS estimation procedure. We did not perform any quality control of cells and only removed genes that were zero in all of the selected cells. We believe this presents the most challenging situation to simulate, as there are more likely to be violations of the underlying model. This scenario is also possibly the most useful as it allows any analysis method to be evaluated, from low-level filtering to complex downstream analysis.

Figure 2 shows some of the plots produced by Splatter to compare simulations based on the Tung dataset.

We compared the gene means, variances, library sizes, and the mean-variance relationship. From these diagnostic plots, we can evaluate how well each simulation reproduces the real dataset and how it differs. One way to compare across the simulations is to look at the overall distributions (Fig. 2, left column).

Table 1 Input parameters for the Splat simulation model

Name	Symbol	Description
Mean shape	α	Shape parameter for the mean gene expression gamma distribution
Mean rate	β	Rate parameter for the mean gene expression gamma distribution
Library size location	μ^L	Location parameter for the library size log-normal distribution
Library size scale	σ^L	Scale parameter for the library size log-normal distribution
Outlier probability	π^O	Probability that a gene is an expression outlier
Outlier location	μ^O	Location parameter for the expression outlier factor log-normal distribution
Outlier scale	σ^O	Scale parameter for the expression outlier factor log-normal distribution
Common BCV	ϕ	Common BCV dispersion across all genes
BCV degrees of freedom	df	Degrees of freedom for the BCV inverse chi-squared distribution
Dropout midpoint	x_0	Midpoint for the dropout logistic function
Dropout shape	k	Shape of the dropout logistic function

Alternatively, we can choose a reference (in this case the real data) and look at departures from that data (Fig. 2, right column). Examining the mean expression levels across genes, we see that the scDD simulation is missing lowly expressed genes, as expected, as is the Lun simulation. In contrast, the Simple and Lun 2 simulations are skewed towards lower expression levels (Fig. 2a, b). The BASiCS simulation is a good match to the real data as is the Splat simulation. Both versions of the Lun 2 simulation produce some extremely highly variable genes, an effect which is also seen to a lesser extent in the Lun simulation. The difference in variance is reflected in the mean–variance relationship where genes from the Lun 2 simulation are much too variable at high expression levels for this dataset. Library size is another aspect in which the simulations differ from the real data. The simulations that do not contain a library size component (Simple, Lun, scDD) have different median library sizes and much smaller spreads. In this example, the BASiCS simulation produces too many large library sizes, as does the Lun 2 simulation to a lesser degree.

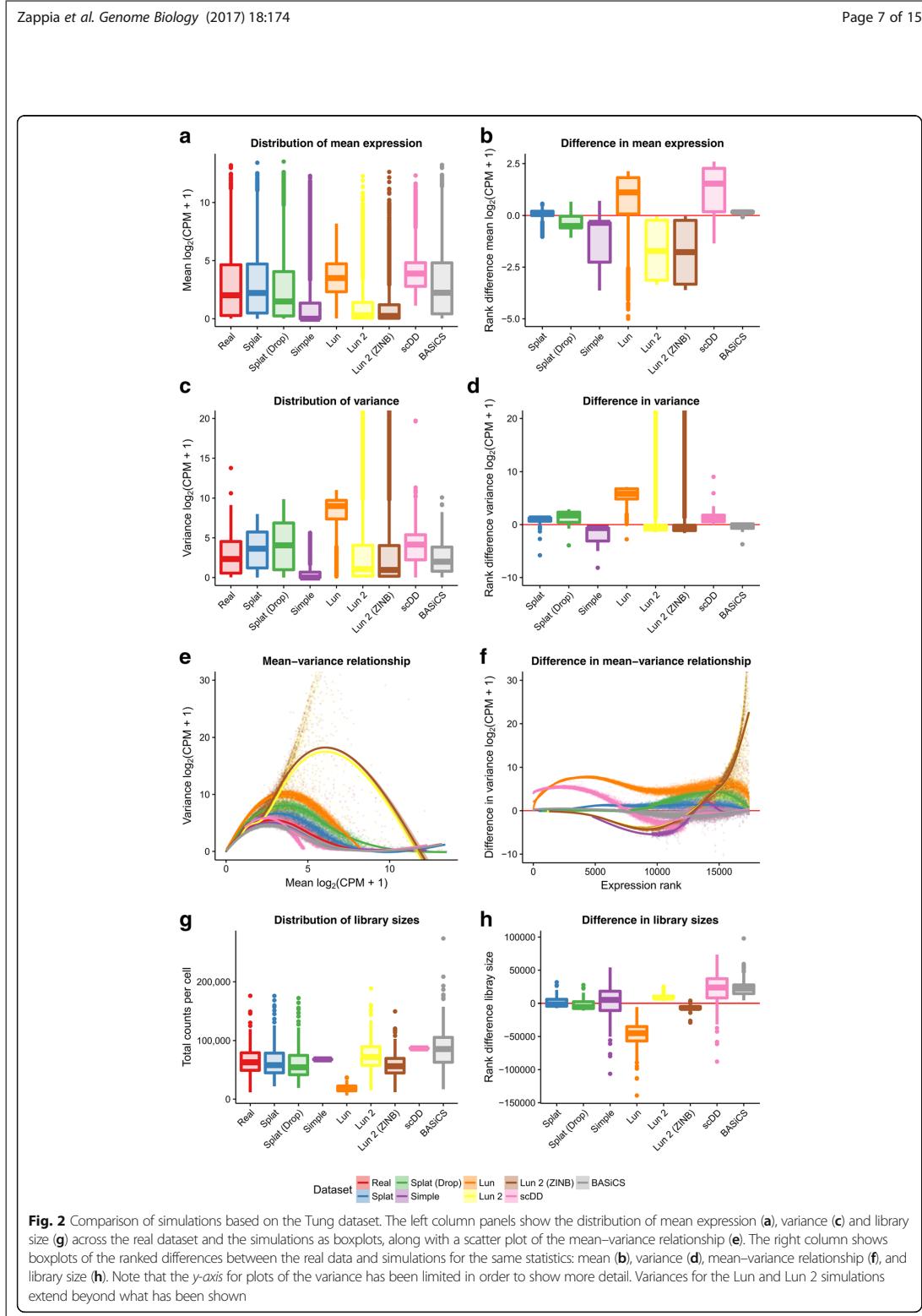
A key aspect of scRNA-seq data is the number of observed zeros. To properly recreate an scRNA-seq dataset a simulation must produce the correct number of zeros but also have them appropriately distributed across both genes and cells. In addition, there is a clear relationship between the expression level of a gene and the number of observed zeros [29] and this should be reproduced in simulations. Figure 3 shows the distribution of zeros for the simulations based on the Tung dataset.

For this dataset the Simple and Lun 2 simulations produce too many zeros across both genes and cells while the Lun and scDD simulations produce too few. Interestingly, the Splat simulation produces a better fit to this dataset when dropout is not included, suggesting that additional dropout is not present in the Tung dataset. However, this is not the case for all data and sometimes simulating additional dropout produces a better fit to the data (for example, the Camp dataset presented below). We can also consider the relationship between the expression level of a gene, calculated including cells with zero counts, and the percentage of zero counts in that gene. The Lun and scDD simulations produce too few zeros at low expression levels, while the Simple and Lun 2 simulations produce too many zeros at high expression levels. It is important to note that as the scDD simulation removes genes with more than 75% zeros prior to simulation this model can never produce genes with high numbers of zeros as shown in Fig. 3c. Both the Splat and BASiCS models are successful at distributing zeros across genes and cells as well as maintaining the mean–zeros relationship.

Although the analysis presented in Figs. 2 and 3 allows us to visually inspect how simulations compare with a single dataset, we also wished to compare simulations across a variety of datasets. To address this we performed simulations based on five different datasets (outlined in Table 2) that varied in terms of library preparation protocol, cell capture platform, species, and tissue complexity. Three of the datasets used Unique Molecular Identifiers (UMIs) [30] and two used full-length protocols. Complete comparison panels for all the datasets are provided in Additional file 1: Figures S5–S10 and processing times for all datasets are shown in Additional file 1: Figure S11.

For each dataset, we estimated parameters and produced a synthetic dataset as described previously. We then compared simulations across metrics and datasets by calculating a median absolute deviation (MAD) for each metric. For example, to get a MAD for the gene expression means, the mean expression values for both the real data and the simulations were sorted and the real values were subtracted from the simulated values. The median of these absolute differences was taken as the final statistic. To compare between simulations, we ranked the MADs for each metric with a rank of one being most similar to the real data. Figure 4 summarizes the ranked results for the five datasets as a heatmap. A heatmap of the MADs is presented in Additional file 1: Figure S12 and the values themselves in Additional file 2.

Looking across the metrics and datasets we see that the Splat simulations are consistently highly ranked. In



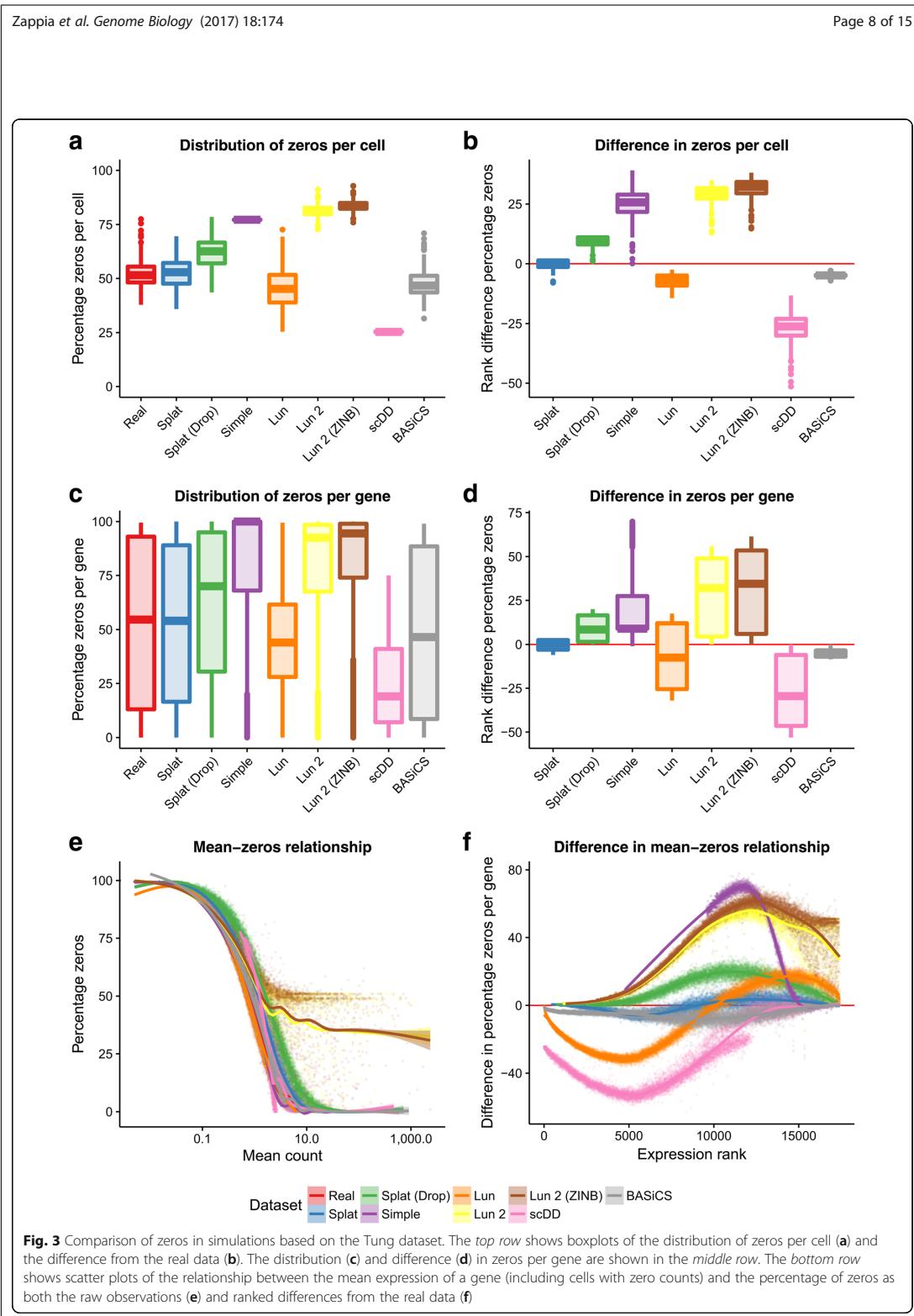


Table 2 Details of real datasets

Dataset	Species	Cell type	Platform	Protocol	UMI	Number of cells
Camp [44]	Human	Whole brain organoids	Fluidigm C1	SMARTer	No	597
Engel [45]	Mouse	Natural killer T cells	Flow cytometry	Modified Smart-seq2	No	203
Klein [46]	Human	K562 cells	InDrop	CEL-Seq	Yes	213
Tung [28]	Human	Induced pluripotent stem cells	Fluidigm C1	Modified SMARTer	Yes	564
Zeisel [47]	Mouse	Cortex and hippocampus cells	Fluidigm C1	STRT-Seq	Yes	3005

Rank of MAD from real data

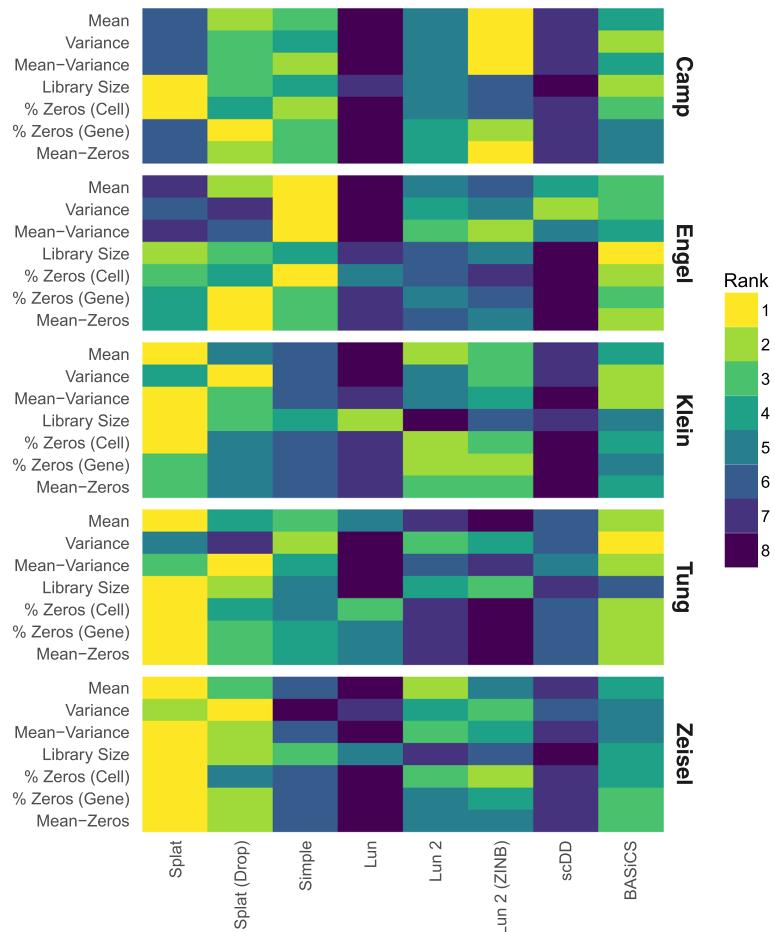


Fig. 4 Comparison of simulation models based on various datasets. For each dataset parameters were estimated and synthetic datasets generated using various simulation methods. The median absolute deviation (MAD) between each simulation and the real data was calculated for a range of metrics and the simulations ranked. A heatmap of the ranks across the metrics and datasets is presented here. We see that the Splat simulation (with and without dropout) performs consistently well, with the BASICS simulation and the two versions of the Lun 2 simulation also performing well.

general, it seems that the datasets are not zero-inflated and thus the zero-inflated simulations do not perform as well as their regular counterparts. The Splat simulations were least successful on the Camp cerebral organoid and Engel T-cell datasets. The complex nature of the Camp data (many cell types) and the full-length protocols used by both may have contributed to Splat's poorer performance. In this situation the semi-parametric, sampling-based models may have an advantage and the Lun 2 simulation was the best performer on most aspects of the Camp data. Interestingly, the Simple simulation was the best performer on the Engel dataset. This result suggests that the additional features of the more complex simulations may be unnecessary in this case or that other models may be more appropriate. The Lun simulation is consistently among the worst performing. However, given that this model is largely similar to the others, it is likely due to the lack of an estimation procedure for most parameters rather than significant problems with the model itself. The scDD simulation also often differed significantly from the real data, which is unsurprising as this simulation is designed to produce a filtered dataset, not the raw datasets used here. A comparison based on a filtered version of the Tung dataset, showing scDD to be a better match, is provided in Additional file 1: Figure S13.

Most importantly we see that simulations perform differently on different datasets. This emphasizes the importance of evaluating different models and demonstrating their similarity to real datasets. Other comparisons may also be of interest for evaluation, such as testing each simulated gene to see if it matches known distributions, an example of which is shown in Additional file 1 Figure S14. The Splatter framework makes these comparisons between simulation models straightforward, making it easier for researchers to choose simulations that best reflect the data they are trying to model.

Complex simulations with Splat

The simulation models described above are sufficient for simulating a single, homogeneous population but not to reproduce the more complex situations seen in some real biological samples. For example, we might wish to simulate a population of cells from a complex tissue containing multiple mature cell types or a developmental scenario where cells are transitioning between cell types. In this section, we present how the Splat simulation can be extended to reproduce these complex sample types (Fig. 5).

Simulating groups

Splat can model samples with multiple cell types by creating distinct groups of cells where several genes are differentially expressed between the different groups. Previously published simulations can reproduce this situation to

some degree but are often limited to fixed fold changes between only two groups. In the Splat simulation, however, differential expression is modeled using a process similar to that for creating expression outliers and can be used to simulate complex cell mixtures. Specifically a multiplicative differential expression factor is assigned to each gene and applied to the underlying mean. For DE genes, these factors are generated from a log-normal distribution while for other genes they are equal to one. Setting the number of groups and the probability that a cell comes from each group allows flexibility in how different groups are defined. Additionally, parameters controlling the probability that genes are differentially expressed as well as the magnitude and direction of DE factors can be set individually for each group. The resulting SCESet object contains information about which group each cell comes from as well as the factors applied to each gene in each group (Fig. 5a).

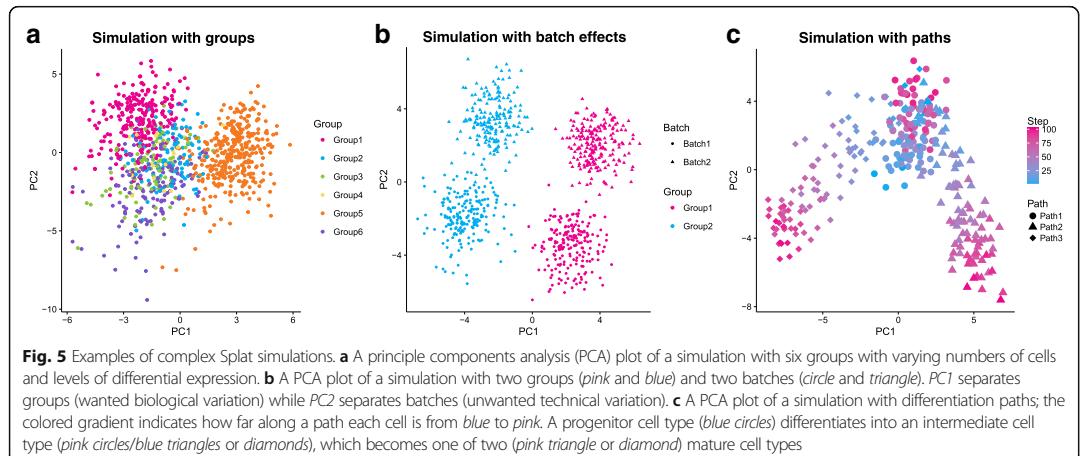
Simulating batches

A common technical problem in all sequencing experiments is batch effects, where technical variation is created during sample collection and preparation. The Splat simulation can model these effects using multiplicative factors that are applied to all genes for groups of cells. Adding this extra layer of variation allows researchers to evaluate how methods perform in the presence of unwanted variation (Fig. 5b).

Simulating paths

A common use of scRNA-seq is to study cellular development and differentiation. Instead of having groups of mature cells, individual cells are somewhere on a continuous differentiation path or lineage from one cell type to another. To model this, the Splat simulation uses the differential expression process described above to define the expression levels of a start and end cell for each path. A series of steps is then defined between the two cells types and the simulated cells are randomly assigned to one of these steps, receiving the mean expression levels at that point. Therefore, the simulation of lineages using Splat is defined by the differential expression parameters used to create the differences between the start and end of each path. It also incorporates the parameters that define the path itself, such as the length (number of steps) and skew (whether cells are more likely to come from the start or end of the path).

In real data it has been observed that expression of genes can change in more complex, non-linear ways across a differentiation trajectory. For example, a gene may be lowly expressed at the beginning of a process, highly expressed in the middle and lowly expressed at the end. Splat models these kinds of changes by



generating a Brownian bridge (a random walk with fixed end points) between the two end cells of a path, which is then smoothed and interpolated using an Akima spline [31, 32]. This random element allows many possible patterns of expression changes over the course of a path (Additional file 1: Figure S15). While non-linear changes are possible they are not the norm. Splat defines parameters that control the proportion of genes that are non-linear and how variable those genes can be.

Further complexity in simulating differentiation paths can be achieved by modeling lineages with multiple steps or branches. For example, a stem cell that differentiates into an intermediate cell type that then changes into one of two mature cell types. These possibilities are enabled by allowing the user to set a starting point for each path (Fig. 5c).

Example: using Splatter simulations to evaluate a clustering method

To demonstrate how the simulations available in Splatter could be used to evaluate an analysis method we present an example of evaluating a clustering method. SC3 [5] is a consensus k -means-based approach available from Bioconductor [33]. As well as assigning cells to groups, SC3 is able to detect genes that are differentially expressed between groups and marker genes that uniquely identify each group. To test SC3 we estimated Splat simulation parameters from the Tung dataset and simulated 400 cells from three groups with probabilities of 0.6, 0.25, and 0.15. The probability of a gene being differentially expressed in a group was 0.1, resulting in approximately 1700 DE genes per group. We then ran SC3 with three clusters ($k = 3$) and compared the results to the true groupings (Fig. 6a). We also assessed the detection of DE and marker genes. True DE genes were taken as

genes with simulated DE in any group and true marker genes as the subset of DE genes that were DE in only a single group (Fig. 6b). This procedure was repeated 20 times with different random seeds to get some idea of the variability and robustness of the method.

Figure 6 shows the evaluation of SC3's clustering and gene identification on the simulated data. Five measures were used to evaluate the clustering: the Rand index (Rand), Hubert and Arabie's (HA) adjusted Rand index and Morey and Agresti's (MA) adjusted Rand index (both of which adjust for chance groupings), Fowlkes and Mallows index (FM) and the Jaccard index (Jaccard). All of these indices attempt to measure the similarity between two clusterings, in this case the clustering returned by SC3 and the true groups in the simulation. SC3 appears to identify clusters well for the majority of simulations, in some cases producing a near-perfect clustering. It may be interesting to examine individual cases further in order to identify when SC3 is able to perform better. Both the DE genes and marker genes identified by SC3 show a similar pattern across our classification metrics of accuracy, precision, recall, and F1 score. On average approximately 2700 of the truly DE genes and 2500 of the true marker genes passed SC3's automatic filtering (with additional non-DE genes). SC3 then detected around 100 DE genes per simulation, along with 99 marker genes (median values). Precision (the proportion of identified genes that are true positives) is very high while recall (the proportion of true positives that were identified, or true positive rate) is very low. This tells us that in this scenario SC3 is producing many false negatives, but that the genes that it finds to be markers or DE are correct. This result is often desirable, particularly for marker genes, and is reflected in the very low false positive rate.

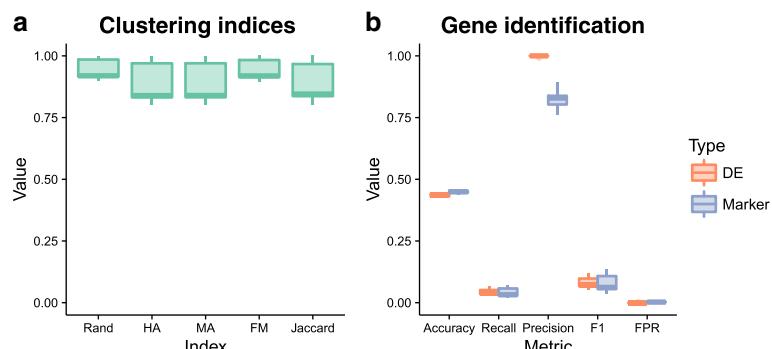


Fig. 6 Evaluation of SC3 results. Metrics for the evaluation of clustering (a) include the Rand index, Hubert and Arabie's adjusted Rand index (HA), Morey and Agresti's adjusted Rand index (MA), Fowlkes and Mallows index (FM), and the Jaccard index. Detection of differentially expressed and marker genes were evaluated (b) using accuracy, recall (true positive rate), precision, F1 score (harmonic mean of precision and recall), and false positive rate (FPR). All of the metrics are presented here as boxplots across the 20 simulations

While it is beyond the scope of this paper, clearly this evaluation could be extended, for example, by including more clustering methods, more variations in simulation parameters, and investigating why particular results are seen. However, these data, and the code used to produce them, are an example of how such an evaluation could be conducted using the simulations available in Splatter.

Discussion and conclusions

The recent development of single-cell RNA sequencing has spawned a plethora of analysis methods, and simulations can be a powerful tool for developing and evaluating them. Unfortunately, many current simulations of scRNA-seq data are poorly documented, not reproducible, or fail to demonstrate similarity to real datasets. In addition, simulations created to evaluate a specific method can sometimes fall into the trap of having the same underlying assumptions as the method that they are trying to test. An independent, reproducible, and flexible simulation framework is required in order for the scientific community to evaluate and develop sophisticated analysis methodologies.

Here we have developed Splatter, an independent framework for the reproducible simulation of scRNA-seq data. Splatter is available as an R package from Bioconductor, under a GPL-3 license, and implements a series of simulation models. Splatter can easily estimate parameters for each model from real data, generate synthetic datasets and quickly create a series of diagnostic plots comparing different simulations and datasets.

As part of Splatter we introduce our own simulation called Splat. Splat builds on the gamma-Poisson (or negative binomial) distribution commonly used to represent RNA-seq data, and adds high-expression outlier

genes, library size distributions, a mean-variance trend, and the option of expression-based dropout. Extensions to Splat include the simulation of more complex scenarios, such as multiple groups of cells with differing sizes and levels of differential expression, experiments with several batches, or differentiation trajectories with multiple paths and branches, with genes that change in non-linear ways.

We performed an evaluation of the six simulation models currently available in Splatter by comparing synthetic data generated using estimated parameters to five published datasets. Overall Splat performed well, ranking highly on most metrics. However, other simulations performed better for some metrics or better reproduced specific datasets. We found the Camp cerebral organoid dataset the most challenging to simulate, perhaps because of the complex nature of this sample, which is comprised of many different cell types. In addition, this dataset (along with the Engel data) used a full-length protocol, which may contain additional noise compared to the UMI datasets [34].

One of the key features of scRNA-seq data is the high number of zero counts where no expression is observed for a particular gene in a particular cell. This can be especially challenging to simulate as not only must there be the correct number of zeros but they must be correctly distributed across genes and cells. We found that introducing dropout (in Splat) or zero-inflation (in Lun 2) often failed to improve the match to real datasets, suggesting that they are not truly zero-inflated. Together, the results demonstrate that no simulation can accurately reproduce all scRNA-seq datasets. They also emphasize the variability in scRNA-seq data, which arises from a complex set of biological (for example, species, tissue type, cell

type, treatment, and cell cycle) and technical (for example, platform, protocol, or processing) factors. Non-parametric simulations that permute real data could potentially produce more realistic synthetic datasets but at the cost of flexibility in what can be simulated and knowledge of the underlying parameters.

Finally, we demonstrated how Splatter could be used for the development and evaluation of analysis methods, using the SC3 clustering method as an example. Splatter's flexible framework allowed us to quickly generate multiple test datasets, based on parameters from real data. The information returned about the simulations gave us a truth to test against when evaluating the method. We found that SC3 accurately clustered cells and was precise in identifying DE and marker genes.

The simulations available in Splatter are well documented, reproducible, and independent of any particular analysis method. Splatter's comparison functions also make it easy to demonstrate how similar simulations are to real datasets. Splatter provides a framework for simulation models, makes existing scRNA-seq simulations accessible to researchers and introduces Splat, a new scRNA-seq simulation model. As more simulation models become available, such as those replicating newer technologies including k-cell sequencing, they can be adapted to Splatter's framework. The Splat model will continue to be developed and may, in the future, include additional modules such as the ability to add gene lengths to differentiate between UMI and full-length data. We hope that Splatter empowers researchers to rapidly and rigorously develop new scRNA-seq analysis methods, ultimately leading to new discoveries in cell biology.

Methods

Splat parameter estimation

To easily generate a simulation that is similar to a given dataset, Splatter includes functions to estimate the parameters for each simulation from real datasets. Just as with the simulation models themselves, the estimation procedures are based on what has been published and there is variation in how many parameters can be estimated for each model. We have given significant attention to estimating the parameters for the Splat simulation. The parameters that control the mean expression of each gene (α and β) are estimated by fitting a gamma distribution to the winsorized means of the library size normalized counts using the *fitdistrplus* package [35]. The library size normalization is a basic normalization where the counts in the original dataset are adjusted so that each cell has the same number of total counts (in this case the median across all cells) and any genes that are all zero are removed. We found that genes with extreme means affect the fit of the gamma distribution

and that this effect was mitigated by winsorizing the top and bottom 10% of values to the 10th and 90th percentiles, respectively. Parameters for the library size distribution (μ^L and σ^L) are estimated in a similar way by fitting a log-normal distribution to the unnormalised library sizes.

The procedure for estimating expression outlier parameters is more complex. Taking the library size normalized counts, outliers are defined as genes where the mean expression is more than two MADs greater than the median of the gene expression means. The outlier probability π^O is then calculated as the proportion of genes that are outliers. Parameters for the outlier factors (μ^O and σ^O) are estimated by fitting a log-normal distribution to the ratio of the means of the outlier genes to the median of the gene expression means.

BCV parameters are estimated using the *estimateDisp* function in the *edgeR* package [22]. When testing the estimation procedure on simulated datasets we observed that the *edgeR* estimate of common dispersion was inflated (Additional file 1: Figure S16); therefore, we apply a linear correction to this value ($\hat{\phi} = 0.1 + 0.25\hat{\phi}_{\text{edgeR}}$).

The midpoint (x_0) and shape (k) parameters for the dropout function are estimated by fitting a logistic function to the relationship between the log means of the normalized counts and the proportion of samples that are zero for each gene (Additional file 1: Figure S17).

While we note that our estimation procedures are somewhat ad hoc, we found that these procedures are robust, efficient, and guaranteed to produce parameter estimates on all datasets we tested.

Datasets

Each of the real datasets used in the comparison of simulations is publicly available. Raw FASTQ files for the Camp dataset were downloaded from SRA (accession SRP066834) and processed using a Bpipe (v0.9.9.3) [36] pipeline that examined the quality of reads using FastQC (v0.11.4), aligned the reads to the hg38 reference genome using STAR (v2.5.2a) [37], and counted reads overlapping genes in the Gencode V22 annotation using featureCounts (v1.5.0-p3) [38]. Matrices of gene by cell expression values for the Klein (accession GSM1599500) and Zeisel (accession GSE60361) datasets were downloaded from GEO. For the Tung dataset the matrix of molecules (UMIs) aligned to each gene available from <https://github.com/jdblischak/singleCellSeq> was used. These data are also available from GEO (accession GSE77288). The Salmon [39] quantification files for the Engel dataset were download from the Conquer database (<http://imlspenticton.uzh.ch:3838/conquer/>) and converted to a gene by cell matrix using the *tximport* [40] package.

Simulation comparison

For each dataset the data file was read into R (v3.4.0) [41] and converted to a gene by cell matrix. We randomly selected 200 cells without replacement and filtered out any genes that had zero expression in all cells or any missing values. The parameters for each simulation were estimated from the selected cells and a synthetic dataset generated with 200 cells and the same number of genes as the real data. Simulations were limited to 200 cells (the size of the smallest dataset) to reduce the computational time required. When estimating parameters for the Lun 2, scDD, and BASiCS simulations cells were randomly assigned to two groups. For the Splat and Lun 2 simulations both the regular and zero-inflated variants were used to simulate data. The resulting eight simulations were then compared to the real data using Splatter's comparison functions and plots showing the overall comparison produced. To compare simulations across the datasets summary statistics were calculated. For each of the basic metrics (mean, variance, library size, zeros per gene, and zeros per cell) the genes were sorted individually for each simulation and the difference from the sorted values and the real data calculated. When looking at the relationship between mean expression level and other metrics (variance, zeros per gene) genes in both the real and simulated data were sorted by mean expression and the difference between the metric of interest (e.g., variance) calculated. The median absolute deviation for each metric was then calculated and ranked for each dataset to give the rankings shown in Fig. 4.

Clustering evaluation

Parameters for Splat simulations used in the example evaluation of SC3 were estimated from the Tung dataset. Twenty synthetic datasets were generated using these parameters with different random seeds. Each simulation had three groups of different cells, with probabilities of 0.6, 0.25 and 0.1, and a probability of a gene being differentially expressed of 0.1. Factors for differentially expressed genes were generated from a log-normal distribution with location parameter equal to -0.1 and scale parameter equal to 0.3. For each simulation the SC3 package was used to cluster cells with $k = 3$ and asked to detect DE and marker genes, taking those with adjusted p values less than 0.05. True DE genes were defined as genes where the simulated DE factor was not equal to 1 in one or more groups. Marker genes were defined as genes where the DE factor was not equal to 1 in a single group (and 1 in all others). Clustering metrics were calculated using the clues R package [42]. To evaluate the DE and marker gene detection we calculated the numbers of true negatives (TN), true positives (TP), false negatives (FN),

and false positives (FP). We then used these values to calculate the metrics shown in Fig. 6: accuracy ($Acc = (TP + TN) / Total\ number\ of\ genes$), recall ($Rec = TP / (TP + FN)$), precision ($Pre = TP / (TP + FP)$), F1 score ($F1 = 2 * ((Pre * Rec) / (Pre + Rec))$), and false positive rate ($FPR = FP / (FP + TN)$). Metrics were aggregated across the 20 simulations and boxplots produced using the ggplot2 package [43].

Session information describing the packages used in all analysis steps is included as Additional file 3. The code and dataset files are available at <https://github.com/Oshlack/splatter-paper> under an MIT license.

Additional files

Additional file 1: Figures S1–S17 Diagrams of other simulation models, Splatter comparison output for all datasets, example non-linear gene dispersion estimate correction, mean-zeros fit, benchmarking, and processing times (PDF 17991 kb)

Additional file 2: Table of the median absolute deviations used to produce Fig. 4 in CSV format. (CSV 37 kb)

Additional file 3: Session information. Details of the R environment and packages used for analysis. (PDF 118 kb)

Acknowledgements

We would like to thank the authors of the BASiCS and scDD packages for their responses to our questions about how to include their simulations in Splatter as well as Mark Robinson and Charlotte Soneson for discussions regarding the simulation of scRNA-seq data. Our thanks also to Jovana Maksimovic and Sarah Blood for their comments on the manuscript.

Funding

Luke Zappia is supported by an Australian Government Research Training Program (RTP) Scholarship. Alicia Oshlack is supported through a National Health and Medical Research Council Career Development Fellowship APP1126157. MCRI is supported by the Victorian Government's Operational Infrastructure Support Program.

Availability of data and materials

The datasets analyzed during the current study are available from the repositories specified in the methods. The code used to analyze them is available under an MIT license from the repository for this paper <https://github.com/Oshlack/splatter-paper> (doi: 10.5281/zenodo.833571). Copies of the datasets are also provided in this repository. The Splatter package is available from Bioconductor (<http://bioconductor.org/packages/splatter/>) and is being developed on Github (<https://github.com/Oshlack/splatter>) under a GPL-3 license. The specific version of Splatter used in this paper, which includes the BASiCS simulation, is available at <https://github.com/Oshlack/splatter/releases/tag/v1.1.3-basics> (doi: 10.5281/zenodo.833574).

Authors' contributions

LZ developed the software and performed the analysis. BP contributed to the statistics and supervision. AO oversaw all aspects of the project. All authors contributed to drafting the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 3 May 2017 Accepted: 22 August 2017
 Published online: 12 September 2017

References

- Goodwin S, McPherson JD, Richard McCombie W. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet.* 2016;17:333–51.
- Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. *Nat Rev Genet.* 2011;12:87–98.
- 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;6:377–82.
- scRNA-tools. <http://www.scRNA-tools.org/>.
- 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;14:483–6.
- Lin P, Troup M, Ho JWK. CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* 2017;18:59.
- Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol.* 2015;33:495–502.
- Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol.* 2014;32:381–6.
- DuVerle DA, Yotsukura S, Nomura S, Aburatani H, Tsuda K. Cell Tree: an R bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics.* 2016;17:363.
- Juliá M, Telenti A, Rausell A, Sincell: an R/Bioconductor package for statistical assessment of cell-state hierarchies from single-cell RNA-seq. *Bioinformatics.* 2015;31:3380–2.
- Pierson E, Yau C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* 2015;16:241.
- Finak G, McDavid A, Yajima M, Deng J, Gersuk V, Shalek AK, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* 2015;16:278.
- Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert J-P. ZINB-WaVE: a general and flexible method for signal extraction from single-cell RNA-seq data. 2017. <http://www.biorxiv.org/content/early/2017/04/06/125112>.
- van Dijk D, Nairys J, Sharma R, Kathail P, Carr AJ, Moon KR, et al. MAGIC: a diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. 2017. <http://biorxiv.org/content/early/2017/02/25/111591>.
- Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, et al. Gene expression recovery for single cell RNA sequencing. 2017. <http://biorxiv.org/content/early/2017/05/17/138677>.
- Li WV, Li JJ. sclImpute: accurate and robust imputation for single cell RNA-Seq data. 2017. <http://biorxiv.org/content/early/2017/05/24/141598>.
- 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;33:1179–86.
- Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* 2016;17:1–14.
- Lun ATL, Marioni JC. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics.* 2017;18:451–64.
- Korthauer KD, Chu L-F, 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;17:222.
- Vallejos CA, Marioni JC, Richardson S. BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS Comput Biol.* 2015;11:e1004333.
- Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics.* 2010;26:139–40.
- Anders S, Huber W. Differential expression analysis for sequence count data. *Genome Biol.* 2010;11:R106.
- Korthauer K. scDD vignette. 2017. <https://bioconductor.org/packages/release/bioc/vignettes/scDD/inst/doc/scDD.pdf>.
- Vallejos CA, Richardson S, Marioni JC. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biol.* 2016;17:70.
- McCarthy DJ, Chen Y, Smyth GK. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res.* 2012;40:4288–97.
- Law CW, Chen Y, Shi W, Smyth GK. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* 2014;15:R29.
- Tung P-Y, 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.
- Andrews TS, Hemberg M. Modelling dropouts allows for unbiased identification of marker genes in scRNASeq experiments. 2016. <http://biorxiv.org/content/early/2016/07/21/065094>.
- Kivioja T, Vähärautio A, Karlsson K, Bonke M, Enge M, Linnarsson S, et al. Counting absolute numbers of molecules using unique molecular identifiers. *Nat Methods.* 2012;9:72–4.
- Akima H. A new method of interpolation and smooth curve fitting based on local procedures. *JACM.* 1970;17:589–602.
- Akima H, Gebhardt A. akima: interpolation of irregularly and regularly spaced data. 2016. <https://CRAN.R-project.org/package=akima>.
- Huber W, Carey JV, Gentleman R, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods.* 2015;12:115–21.
- Phipson B, Zappia L, Oshlack A. Gene length and detection bias in single cell RNA sequencing protocols. *F1000Res.* 2017;6:595.
- Delignette-Muller M, Dutang C. fitdistrplus: an R package for fitting distributions. *J Stat Softw.* 2015;64:1–34.
- Sadedin SP, Pope B, Oshlack A. Bpipe: a tool for running and managing bioinformatics pipelines. *Bioinformatics.* 2012;28:1525–6.
- Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics.* 2013;29:15–21.
- Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics.* 2014;30:923–30.
- Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods.* 2017;14:417–9.
- Soneson C, Love MI, Robinson MD. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Res.* 2015;4:1521.
- R Core Team. R: a language and environment for statistical computing. Vienna: R Foundation for Statistical Computing; 2016. <https://www.R-project.org/>.
- Chang F, Qiu W, Zamar R, Lazarus R, Wang X. clues: an R package for nonparametric clustering based on local shrinking. *J Stat Softw.* 2010;33:1–16.
- Wickham H. ggplot2: elegant graphics for data analysis. New York: Springer; 2010.
- Camp JG, Badsha F, Florio M, Kanton S, Gerber T, Wilsch-Bräuninger M, et al. Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proc Natl Acad Sci U S A.* 2015;112:15672–7.
- Engel I, Seumos G, Chavez L, Samaniego-Castruita D, White B, Chawla A, et al. Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. *Nat Immunol.* 2016;17:728–39.
- Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell.* 2015;161:1187–201.
- Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnérberg P, La Manno G, Juréus A, et al. Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science.* 2015;347:1138–42.

Submit your next manuscript to BioMed Central and we will help you at every step:

- We accept pre-submission inquiries
- Our selector tool helps you to find the most relevant journal
- We provide round the clock customer support
- Convenient online submission
- Thorough peer review
- Inclusion in PubMed and all major indexing services
- Maximum visibility for your research

Submit your manuscript at
www.biomedcentral.com/submit



Chapter 4

Visualising clustering across resolutions

Chapter 5

Analysis of kidney organoid scRNA-seq data

Chapter 6

Conclusion

References