# 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

- Chapter order
- Publications
- Contributions

This preface provides a summary of the chapters in this thesis and described my contribution to them. This is a thesis *with* publication and where publications form part of a chapter that are listed here. Publications are included as they appear online and are designed to be read as stand-alone documents. Sections within these publications are not included in the table of contents and references are available at the end of each publication rather than the reference list for this thesis. These publications have authors other than myself and their contributions are explained below. I am the first author on these publications and contributed more than 50 percent of the work towards them including drafting, editing and revising the manuscripts. My co-authors have provided signed declarations acknowledging and supporting my contributions which have been submitted along with this thesis. Where publicly available datasets have been used these have been appropriately cited.

**Chapter 1** is an original work providing a background and overview relevant to understanding my work in this thesis including an introduction to RNA sequencing, single-cell RNA sequencing and kidney function and development.

**Chapter 2** is an original work describing a database of tools for analysing single-cell RNA sequencing data which has been published in *PLoS Computational Biology* as *"Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database"*. In addition to the publication I developed a website displaying the information in this database available at `https://scRNA-tools.org`. The database and code for building the website is available on GitHub at `https://github.com/Oshlack/scRNA-tools` under an MIT license.

Contributions to the work in this chapter:

- I complied and regularly updated the database of tools.
- I designed and built the public website used to display the database. Breon Schmidt provided assistance with implementing some of the website functionality and code for processing the database was based on a script written by Sean Davis.
- I performed the analysis of the database presented in the publication.
- I wrote the first draft of the manuscript and produced all the figures in the publication.

- Alicia Oshlack provided advice on planning the manuscript and edited draft versions.
- Belinda Phipson contributed to writing the manuscript.

**Chapter 3** is an original work describing a software package for simulating single-cell RNA sequencing data. This work was published in *Genome Biology* as *"Splatter: simulation of single-cell RNA sequencing data"*. The software package described in this publication is available through Bioconductor at `https://bioconductor.org/packages/splatter` and the code is shared on GitHub at `https://github.com/Oshlack/splatter` under a GPL-3.0 license.

Contributions to the work in this chapter:

- I designed and implemented the Splatter R package described in this chapter
- Belinda Phipson contributed to the design of the Splat simulation method described in the publication and provided statistical advice.
- I conducted the analysis presented in the publication and produced the figures shown.
- Belinda Phipson performed preprocessing for some of the public datasets used
- Alicia Oshlack helped to design and plan the analysis presented in the publication.
- I wrote the first draft of the manuscript and performed revisions.
- Alicia Oshlack assisted with planning the manuscript and edited drafts.
- Belinda Phipson helped write sections of the manuscript and edited drafts.
- Jovana Maksimovic proofread a section of the manuscript and provided comments.

**Chapter 4** is an original work describing a visualisation for showing clustering results across multiple resolutions and helping to choose a clustering resolution to use. This work has been published in *GigaScience* as *"Clustering trees: a visualization for evaluating clusterings at multiple resolutions"* and a software package implementing the algorithm described is available from CRAN at `https://cran.r-project.org/package=clustree`. The source code for this package can be found on GitHub at `https://github.com/lazappi/clustree` under a GPL-3.0 license.

Contributions to work in this chapter:

- I designed the clustering trees algorithm described in this chapter.
- I designed and wrote the clustree R package that implements this algorithm.
- I performed the analysis presented in the manuscript and designed and produced the figures shown.
- Alicia Oshlack provided advice on the design and planning of the analysis to present.
- I planned and wrote the first draft of the manscript.
- Alicia Oshlack provided advice on the structure of the manuscript and edited draft versions.
- I performed revisions and drafted responses to reviewers.
- Marek Cmero read and provided comments on a draft of the manuscript.

**Chapter 5** is an original work where I performed analysis of a single-cell RNA sequencing experiment from kidney organoids in order to identify and characterise the cell types present.

Contributions to work in this chapter:

- XXX performed the cell culture.
- XXX performed cell capture and library preparation of the samples.
- I performed preprocessing of the datasets
- I designed and performed the analysis with some input from Alicia Oshlack, Belinda Phipson, Melissa Little and Alex Combes.
- Alex Combes helped with interpreting gene lists describing cell types.
- I designed and created the figures shown in this chapter.

**Chapter 6** is an original work summarising the work in this thesis, placing it in the wider context of single-cell RNA sequencing analysis and outlining potential directions of the field.

**Other publications I have contributed to during my candidature but are not presented in this thesis**

- Organoid scRNA-seq paper
- Gene length paper
- Swirler organoid paper
- Reproducibility paper

# 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). Inspriation 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

# List of Tables

# List of Figures

# Chapter 1
# Introduction

- Central dogma
  - Flow of information in cell
  - DNA - long term storage
  - Transcription to RNA - working copy, amplification
    - Messenger RNA
  - Translation to protein - functional
  - Some RNA also functional

The central dogma of biology describes the flow of information within a cell, from DNA to RNA to protein. Deoxyribonucleic acid (DNA) is the long term data storage of the cell. This molecule has a well known double strand structure. Each strand of the helix consists of a series of nucleic acid molecules linked by phosphate groups. These nucleic acids come in four species, adenosine, cytosine, tyrosine and guanine and the two strands are bound together through hydrogen bonds between matching nucleic acids known as basepairs. Guanine forms three hydrogen bonds with cytosine and adenosine forms two with tyrosine. In computing terms DNA is similar to a hard drive that provides stable, consistent storage of important information. When the cell wants to use some of this information it produces a copy of it in the form of a ribonucleic acid (RNA) molecule through a process known as transcription, similar to a computer loading information it wants to use into it's random access memory. RNA is similar to a single strand of DNA except that the tyrosine base is replaced with another base called uracil. Because it is single-stranded RNA does not have a double helix structure but it can form complex shapes by binding to itself. There are several different types of RNA that serve different purposes. RNA molecules that are translated from genes are known as messenger RNA (mRNA). Other types of RNA include ribosomal RNA (rRNA) (that forms part of the ribosome), micro RNA which have a role in regulating gene expression and long-noncoding RNA. Genes are the sections of DNA that encode proteins and are made up of regions that code information (none as exons) with much larger non-coding regions between them (introns). When an mRNA molecule is transcribed it initially contains the intronic sequences but these are removed through a process known as splicing and a sequence of adenosine bases (a poly-A tail) is added to the end where transcription ends (the 3′ end) to mark a mature mRNA molecule. The process for producing a protein from an mRNA transcript occurs in a structure called the ribosome and is known as translation because

the information encoded by nucleic acids in RNA is converted to information stored as amino acids in the protein. Proteins complete most of the work required to keep a cell functioning and can be compared to the programs running on a computer. These functions include tasks such as sensing things in the external environment, transport nutrients into the cell, regulating the expression of genes, constructing new proteins, recycling molecules and metabolism. Understanding the molecules involved in the central dogma is central to our understanding of how a cell functions.

## 1.1 RNA sequencing

- Why RNA-seq?
  - What is happening?
  - High throughput
  - Easy
  - Unbiased
  - Complete
  - Different cell types

By looking at DNA we can see what genes are present in a cell but we cannot tell which of them are active and what processes they might be involved in. To do that we need to inspect the parts of the system that change dynamically. Ideally we might want to interrogate which proteins are present as they provide most of the functionality. However, while it is possible to do this using technologies such as mass spectrometry the readout they produce is difficult to interpret and the encoding is much more complex as there are 20 types of amino acids compared to only four nucleotides. In contrast RNA molecules are much easier to measure. High-throughput RNA sequencing (RNA-seq) provides a reliable method for high-quality measurement of RNA expression levels. RNA is isolated from a biological sample, converted to complementary DNA (cDNA) and provided as input to a sequencing machine. The output of an RNA-seq experiment is millions of short nucleotide sequences originating from the RNA transcripts present in the sample. In contrast to older techniques for measuring RNA, such as probe-based microarrays, RNA-seq requires no prior knowledge of existing sequences in order to measure a sample and is effective over a much greater range of expression levels.

### 1.1.1 Library preparation

- PolyA capture
- Ribosomal depletion

The first step in preparing a sample for RNA-seq is to chemically lyse the cells, disrupting the structure of the cell wall and releasing the molecules inside. RNA molecules can then be isolated, typically using a chemical process called phenol/chloroform extraction although this can also be done by physically separating different types of molecules by passing the sample through a silica column. The majority of the RNA in

a cell is ribosomal RNA, usually more than 80 percent {raz2011}. Most of the time this type of RNA is not of interest and sequencing it would reduce the ability to detect less abundant species. To select mRNA oligonucleotide probes that bind to the poly-A tail can be used but a downside of this approach is that it won't capture immature mRNA or other types of RNA molecules. An alternative method is to use a different kind of probe specific to each species that binds to the rRNA allowing it to be removed. The choice of selection method has been shown to introduce different biases into the resulting data {sultan2014}.

The Illumina sequencing typically used for RNA-seq experiments can only read short sequences of nucleotide of approximately 40-400 basepairs. Most mRNA molecules are longer than this so to read the full length they must be fragmented into smaller parts. Most sequencing machines also only work with DNA not directly with RNA so the sample must first be reverse-transcribed using a retroviral enzyme to produce a single strand of cDNA. Many protocols have been designed for this step with each requiring a specific primer sequence to be joined to the RNA molecules {roberts2011?}. The complementary strand of cDNA is produced using a second enzyme that is usually involved in copying DNA for cell division. For some protocols fragmentation is performed after conversion to cDNA rather than at the RNA stage.

Once the cDNA has been produced it is usually necessary to attach adaptor sequences that are used to bind the molecules and initiate sequencing. They may also contain barcodes for measuring multiple samples at once. It has become standard practice to perform paired-end sequencing where a section of sequence is read from one end of a molecule before it is flipped and the other end read and this process requires an additional set of adaptors. At each of the stages of library preparation there are quality control steps to be performed to make sure a high-quality cDNA sample is loaded on to the sequencing machine.

### 1.1.2 High-throughput sequencing

- Illumina sequencing
  - Sequence by synthesis
  - Paired end

Most RNA-seq experiments are sequenced on an Illumina maching using their Sequence by Synthesis technology. In this process the strands of cDNA fragments are separated and the adaptors bind to oligonucleotides coating a flow cell. The other end of the fragment can bind to a second oligonucleotide forming a structure where an enzyme synthesises a complementary DNA strand. This process of separation of strands and synthesis of new complementary strands is repeated until clusters of DNA fragments with the same sequence are formed. Once the clusters are significantly large the adaptor at one end of each fragment is cleaved leaving single stranded DNA attached to the flow cell and one end.

The sequencing process now begins. Nucleotides tagged with fluorescent markers are added and can bind to the next available position on a fragment if they are complementary. By adding all four nucleotides at once they compete for each position,

reducing the chance of an incorrect match. Any unbound nucleotides are washed away before a laser excites the florescent tags and an image is taken. Each nucleotide is tagged with a different and the order of colours produced by a cluster shows the sequence of nucleotides in a fragment. For paired-end sequencing the fragments can be flipped and the sequencing process repeated at the other end. The images from the sequencing machine are processed to produce millions of short nucleotide reads that are the starting point for computational analysis.

### 1.1.3 Analysis of RNA-seq data

- Experimental design
- Alignment
- Quantification
- Negative binomial
- Normalisation
- Differential expression testing
- Proportions

Many types of analyses can be performed using RNA-seq data such as identification of variants in the genetic sequence or detection of previously unannotated transcripts but the most common kind of analysis is to look for differences in the expression level of genes between groups. To do this reads are first aligned to a reference and the number of reads overlapping each genes is counted. In contrast to aligners designed for DNA sequencing RNA-seq aligners such as STAR, HiSAT2 and subread must taken into account the splicing of mRNA transcripts which causes parts of some reads to align in different locations. The alignment step is computational intensive and can take a significant amount of time. More recently tools such as kallisto and Salmon have been developed which attempt to directly quantify expression by estimating the probability that a read comes from a particular annotated transcript. These approaches are orders of magnitude faster than true alignment and potentially produce more accurate quantification at the cost of having an exact genomic position for each read.

At this stage the result is a matrix of counts known as an expression matrix where the rows are features (usually genes), the columns are samples and the values show the expression level of a particular feature in a sample. As these counts result from a sampling process that can be modeled using common statistical distributions. One option is the Poisson distribution, however this assumes that the mean and variance of each feature is equal. A better fit is the negative binomial (or Gamma-Poisson) distribution which includes an over-dispersion parameter allowing the variance to be larger than the mean. While each feature is quantified for each sample these values are not absolute measures of expression and are better understood as proportions of the total number of reads. Another complication of RNA-seq data is that the number of features (tens of thousands) is much larger than the number of samples (usually only a few per group).

The most successful methods for testing differential expression between groups of RNA-seq samples overcome this challenge by sharing information between genes.

Both the edgeR and DESeq (and later the DESeq2) packages model RNA-seq data using the negative binomial distribution while Before expression levels can be tested the differences between samples must be removed through normalisation. The edgeR packaged uses the Trimmed-Mean of M values (TMM) method where... DESeq has a similar method that... When an experiment has been conducted multiple batches and there are significant differences between them alternative normalisations such as Removel of Unwanted Variation (RUV) that ... may be required. The limma package uses an alternative approach where a method called voom transforms the data so that it is suitable for linear modelling methods originally designed for RNA microarray technology. Over the time the methods in these packages have been refined and new tests developed allowing for the routine analysis of RNA-seq experiments.

## 1.2 Single-cell RNA-sequencing

Traditional bulk RNA-seq experiments average the transcriptome across the many cells in a sample but recently it has become possible to perform single-cell RNA-sequencing (scRNA-seq) and investigate the transcriptome at the resolution of individual cell. There are many situations were it is important to understand how specific cell types react and where analyses may be affected by the unknown proportions of cell types in a sample. Studies into gene expression in specific cell types previously required a way to select and isolate the cells they were interested which removed them from the other cell types they are usually associated with and made it impossible to investigate how they interact. With scRNA-seq technologies it is now possible to look at the transcriptome of all the cell types in a tissue simultaneously which has lead to a better understanding of what makes cell types distinct and the discovery of previously unknown cell types.

One area that has particularly benefitted from the rise of scRNA-seq is developmental biology. Although the genes involved in the development of many organs are now well understood arriving at this knowledge has required many painstaking experiments. During development cells are participating in a continuous dynamic process involving the maturation from one cell type to another and the creation of new cell types. Single-cell RNA-seq captures a snapshot of this process allow the transcriptome of intermediate and mature cells to be studied. This has revealed that some of the genes thought to be markers of specific cell types are more widely expressed or involved in other processes.

## 1.3 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.

### 1.3.1 Droplet based cell capture

- Drop-seq
- Indrop
- 10x Chromium

An alternative to using microfludics to capture cells in wells is to capture them in nano-droplets. A dissociated cell mixture is fed into a microfluidic device while at another input beads coated in primers enter. The device is designed to form aqueous droplets within mineral and the inputs are arranged so that cells and beads can be simultaneously captured within a droplet. When this happens the reagents carried along with the bead lyse the cell and any PolyA tagged RNA molecules present can bind to the capture probes on the bead. Reverse transcription and PCR amplification then begins and an individual cDNA library is produced for each cell, tagged with the unique barcode sequence present on the bead. The main advantage of droplet-based capture technologies is the ability to capture many more cells at one time, up to tens of thousands. These approaches are also less selective about cell size and produce less doublets. As a result they are much cheaper per a cell, although as sequencing costs are fixed studies using droplet-based captures typically sequence individual cells at a much lower depth.

Droplet-based capture was popularised by the publication of the Drop-seq and InDrop platforms in 2015. This are both DIY systems and although they differ in how the beads are produced, when the droplets are broken and some aspects of the chemistry

they can both be constructed on a lab bench from syringes, automatic plungers, a micro scope and a small custom-made microfluidic chip. A similar commercially available platform is the 10x Genomics Chromium device which automates and streamlines much of the process. This device uses droplet-based technologies for a range of applications including capture of cells for scRNA-seq. More specialised captures, such as those aimed at profiling immune cell receptors are also possible and the company has recently announced kits for single-cell ATAC-seq capture.

### 1.3.2 Unique Molecular Identifiers

- Why?
- How they work

In contrast to plate-based capture methods, which often provide reads along the length of transcripts, droplet-based capture methods typically employ protocols which include short random nucleotide sequences known as Unique Molecular Identifiers (UMIs). Individual cells contain very small amounts of RNA and to obtain enough cDNA a PCR amplification step is necessary. Depending on their nucleotide sequence different transcripts may be amplified at different rates which can distort their relative proportions within a library. UMIs attempt to improve the quantification of gene expression by allowing the removal of PCR duplicates produced during amplification. The nucleotide probes used in droplet-based capture protocols include a PolyT sequence which binds to mature mRNA molecules, a barcode sequence which is the same for every probe on a bead and 8-10 bases of UMI sequence which is unique to each probe. The UMI sequences are long enough that the probability of capturing two copies of a transcript on two probes with the same UMI is extremely low. 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 and therefore should be PCR duplicates rather than truly expressed copies of a transcript. For this method to be effective each read must be associated with a UMI which means that only a small section at the 3' end of each transcript is sequenced. This has the side effect of reducing the amount of cDNA that needs to be sequenced and therefore increasing the number of cells that can be sequenced at a time. While the improvement in quantification of gene expression levels is useful for many downstream analyses it comes at the cost of coverage across the length of a gene which is required for applications such as variant detection and de-novo assembly. **READS ALONG GENE** Statistical methods designed for full-length data may also be affected by the difference properties of a UMI dataset. Datasets with UMIs also need extra processing steps which can be complicated by the possibility of sequencing errors in the UMI itself.

### 1.3.3 Recent advances

- New capture methods
- CITE-seq
- Cell hashing
- CRISPR

- Multiple measurements, same cell

Although droplet-based techniques are currently the most commonly used cell capture technologies other approaches have been proposed that promise to capture even more cells even more cheaply. These include approaches based around nanowells...

Extensions to the standard protocols have also been proposed that allow extra measurements from the same cell. One such protocol is CITE-seq which enables measurement of the levels of selected proteins at the same time as the whole transcriptome. Antibodies for the proteins of interest are labelled with short nucleotide sequences. These antibodies can then be applied to the dissociated cells and any that remain unbound washed away before cell capture. The antibody labels are then captured along with mRNA transcripts and a size selection step is applied to separate them before library preparation. Similar antibodies can be used to allow multiplexing of samples through a process known as cell hashing. In a typical scRNA-seq experiment each batch corresponds to a single sample. This complicated analysis as it is impossible to tell what is noise due to cells being processed in the same way and what is true biological signal. Cell hashing uses an antibody to a ubiquitously expressed protein but with a different nucleotide sequence for each sample. The samples can then be mixed, processed in batches and then the cells computationally separated based on which sequence they are associated with. An added benefit of this approach is the simple detection of doublets containing cells from different samples.

CRISPR-Cas9 gene editing has also been developed as an extension to scRNA-seq protocols. One possibility is to introduce a mutation at a known location that can then be used to demultiplex samples processed together. It is possible to do this with samples from different individuals or cell lines but the advantage of a gene editing based approach is that the genetic background remains similar between samples. It is also possible to investigate the effects of introducing a mutation. Protocols like Perturb-Seq introduce a range of guide RNA molecules to a cell culture, subject the cells to some stimulus then perform single-cell RNA sequencing. The introduced mutation can then be linked to the response of the cells to the stimulus and the associated broader changes in gene expression.

Other approaches that allow multiple measurements from individual cells include...

## 1.4 Analysing scRNA-seq data

- Low counts
  - Dropout
  - Bursting
  - Biology
- Ribosomal RNA

Cell capture technologies and scRNA-seq protocols have developed rapidly but there are still a number of challenges with the data they produce. Existing approaches

are inefficient, capturing around 10 percent of transcripts in a cell[Grun2014-zn]. When combined with the low sequencing depth per cell this results in a limited sensitivity and an inability to 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]. In order to capture cells they must first be dissociated into single-cell suspensions but this step can be non-trivial. Some tissues or cell types may be more difficult to separate than others and the treatments required to break them apart may effect the health of the cells and their transcriptional profiles. Other cell types may be too big or have other characteristics that prevent them being captured. In these cases related techniques that allow the sequencing of RNA from single nuclei may be more effective. Cells may be damaged during processing, multiple cells captured together or empty wells or droplets sequenced making quality control of datasets an important consideration.

As well as increasing technical noise the small amounts of starting material and low sequencing depth mean there are many occasions where zero counts are recorded, indicating no measured expression for a particular gene in a particular cell. These zero counts often represent true biological signal we are interested as we expect different cell types to express different genes. However they can also be the result of confounding biological factors such as stage in the cell cycle, transcriptional bursting and environmental interactions which cause genuine changes in expression but that might not be of interest to a particular study. On top of this there are effects that are purely technical factors in particular sampling effects which mean result in "dropout" events where a transcript is truly expressed in a sample but is not observed in the sequencing data. In bulk experiments these effects are limited by averaging across the cells in a sample but for single-cell experiments they can present a significant challenge for analysis as methods must account for the missing information and they may cause the assumptions of existing methods to be violated. One approach to tackling the problem of too many zeros is to use zero-inflated versions of common distributions but it is debatable whether scRNA-seq datasets are truly zero-inflated or the the additional zeros are better modeled with standard distributions with lower means. Another approach is to impute some of the zeros, replacing them with estimates of how expressed those genes truly are based on their expression in similar cells. However imputation comes with the risk of introducing false structure that is not really present in the data.

Bulk RNA-seq experiments usually involve predefined groups of samples, for example cancer cells and normal tissue, different tissue types or treatment and control groups. It is possible to design scRNA-seq experiments in the same way for example by sorting cells into known groups based on surface markers, sampling them at a series of time points or comparing treatment groups but often they are more exploratory. Many of the single-cell studies to date have sampled developing or mature tissues and attempted to profile the cell types that are present[Zeisel2015-rd; Patel2014-bl; Treutlein2014-wd; Usoskin2015-fz; Buettner2015-rq; Klein2015-iw; Trapnell2014-he]. This approach is best exemplified by the Human Cell Atlas project which is attempting to produce a reference of the transcriptional profiles of all the cell types in the human body. Similar projects exist for other species and specific tissues. As scRNA-seq datasets have become more widely available a standard workflow has developed

which can be applied to many experiments. This workflow can be divided into four phases: 1) Data acquisition, Pre-processing of samples to produce a cell by gene expression matrix, 2) Data cleaning, quality control to refine the dataset used for analysis, 3) Cell assignment, grouping or ordering of cells based on their transcriptional profile, and 4) Gene identification to find genes that represent particular groups and can be used to interpret them. Within each phase a range processes may be used and there are now many tools available for completing each of them, with over XXX tools currently available. An introduction to the phases of scRNA-seq analysis is provided here but the analysis tools landscape is more fully explored in Chapter X.

### 1.4.1 Pre-processing and quality control

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

The result of a sequencing experiment is typically a set of image files from the sequencer or a FASTQ file containing nucleotide reads but for most analyses we use an expression matrix. To produce this matrix there is a series of pre-processing steps, typically beginning will some quality control of the raw reads. Reads are then aligned to a reference genome and the number of reads overlapping annotated features (genes or transcripts) is counted. In recent years probabilistic quantification methods such as kallisto[Bray2016-tm] or Salmon[Patro2015-kl] that estimate transcript expression directly without requiring complete alignment have become popular as they dramatically reduce processing time and potentially produce more accurate quantification. These can be applied to full-length scRNA-seq datasets but have required adaptations such as the Alevin method for UMI-based datasets. When using conventional alignment UMI samples need extra processing with tools like UMI-tools[Smith2016-bt] or umis[Svensson2016-eg] in order to assign cell barcodes and deduplicate UMIs. For datasets produced using the Chromium platform the Cell Ranger software is a complete preprocessing pipeline that also includes an automated downstream analysis. Other packages such as scPipe also aim to streamline this process with some such as XXX designed to work on scalable cloud based infrastructure which may be required as bigger datasets continue to be produced.

Quality control of individual cells is important as experiments will contain low-quality cells that can be uninformative or lead to misleading results. Quality control can be performed on various levels, from the quality scores of the reads themselves, how or where reads align to features of the expression matrix. Particular types of cells that are commonly removed include damaged cells, doublets where multiple

cells have been captured together and empty droplets or wells that have been sequenced but do not contain a cell. The Cellity package attempts to automate this process by inspecting a series of biological and technical features and using machine learning methods to distinguish between high and low-quality cells[Ilicic2016-wy]. However the authors found that many of the features were cell type specific and more work needs to be done to make this approach more generally applicable. The scater package[McCarthy2016-cw] emphasises a more exploratory approach to quality control at the expression matrix level but providing a series of functions for visualising various features of a dataset. These plots can then be used for selecting thresholds for removing cells. Plate-based capture platforms can produce additional biases based on the location of individual wells, a problem which is addressed by the OEFinder package which attempts to identify and visualise these "ordering effects"[Leng2016-it].

Filtering and selection of features also deserves attention. Genes or transcripts that are lowly expressed are typically removed from datasets in order to reduce computational time and multiple-testing correction but it is unclear how many counts indicate that a gene is truly "expressed". Many downstream analysis operate on a selected set of genes which can have a dramatic effect on their results. These features are often selected based on how variable they are across the dataset but this may be a result of noise rather than biological importance. Alternative selection methods have been proposed such as M3Drop which...

### 1.4.2 Normalisation and integration

- Why?
- Seurat CCA
- New methods
- Tung?
- Different data types

Technical variation is a known problem in high-throughput genomics studies, for example it has been estimated 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 and similarly this is true for single-cell experiments. Some full-length 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 the total number of reads per cell and gene length. For UMI data the gene length correction is not required as reads only come from the ends of transcripts. Normalisation methods designed for detecting differential expression between bulk samples such as Trimmed Mean of M-Values (TMM)[Robinson2010-ll] or the DESeq method[Anders2010-pq] can be applied, but is is unclear how suitable they are for the single-cell context. Most of the early normalisation methods developed specifically for scRNA-seq data made use of spike-ins, synthetic RNA sequences added to cells in known quantities such as the ERCC.... 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]. Using spike-ins for normalisation assumes that they properly capture the dynamics of the underlying dataset and even if this is the case it is restricted to protocols where they can be added which does not include droplet-based capture techniques. The scrna package implements a method that doesn't rely on 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 size factors that are deconvolved back to the original cells[Lun2016-mq]. The BASiCS method has also been adapted to experiments without spike-ins by..., but only for designed experiments where groups are known in advance.

Early scRNA-seq studies often made use of only a single sample but as technologies have become cheaper and more widely available it is common to see studies with multiple batches or making use of publicly available data produced by other groups. While this expands the potential insights to be gained it presents a problem as to how to integrate these datasets and a range of computational approaches for doing this have been developed. The alignment approach in the Seurat package uses Canonical Correlation Analysis (CCA) to identify a multi-dimensional subspace that is consistent between datasets. Dynamic Time Warping (DTW) is then used to stretch and align these dimensions so that the datasets are similarly spread along them. Clustering can then be performed using these aligned dimensions but as the original expression matrix is unchanged the integration is not used for other tasks such as differential expression testing. The authors of scran using a Mutual Nearest Neighbours (MNN) approach that... A recent update to the Seurat method combines these approaches by identifying "anchors" that... Alternative integration methods such as...

### 1.4.3 Grouping cells

- Clustering
- Seurat
- Other approaches
- Comparison
- Classification

Grouping similar cells is a key step in analysing scRNA-seq datasets that is not usually required for bulk experiment and as such it has been a key focus of methods development with over XXX tools released for clustering cells. Some of these methods include SINgle CEll RNA-seq profiling Analysis (SINCERA)[Guo2015-mf], Single-Cell Consensus Clustering (SC3)[Kiselev2016-fa], single-cell latent variable model (scLVM)[Buettner2015-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]. All of these tools attempt to cluster similar cells together based on their expression profiles, forming groups of cells of the same type. One clustering method that has become popular is that included in the Seurat package. This method begins by selecting a set of highly variable genes then performing PCA on them.**NEW GENE SELECTION** A set of dimensions is then selected that contains most of the variation in the dataset. Alterna-

tively if Seurat's alignment method has been used to integrate datasets the aligned CCA dimensions are used instead. Next an MNN graph is constructed by considering the distance between cells in this multidimensional space. In order to separate cells into clusters a community detection algorithm such as Louvain optimisation is run on the graph with a resolution parameter that controls the number of clusters that are produced. Seurat's clustering method has been shown too....

For tissue types that are well understood or where comprehensive references are available an alternative is to directly classify cells. This can be done using a gating approach based on the expression of known marker genes similar to that commonly used for flow cytometry experiments. Alternatively machine learning algorithms can be used to perform classification based on the overall expression profile. Methods such as ... take this approach. For example... Classification has the advantage of making use of existing knowledge and avoids manual annotation and interpretation of clusters which can often be difficult and time consuming. However it is biased by what is present in the reference datasets used typically can not reveal previously unknown cell types or states. As projects like the Human Cell Atlas produce well-annotated references based on scRNA-seq data the viability of classification and other reference-based methods will improve.

### 1.4.4 Ordering cells

- Pseudotime
- Monocle
- Other approaches
- Comparison

In some studies, for example in development where stem cells are differentiating into mature cell types, it may make sense to order cells along a continuous trajectory from one cell type to another instead of assigning them to distinct groups. Trajectory analysis was pioneered by the Monocle package 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 other developed including TSCAN[Ji2016-ws], SLICER[Welch2016-cw], CellTree[DuVerle2016-ni], Sincell[Julia2015-zc] and Mpath[Chen2016-kx]. In their comprehensive review and comparison of trajectory inference methods Cannoodt, Saelens and Saeys break the process into two steps. In the first step dimensionality reduction techniques such as PCA or t-SNE[Maaten2008-ne] are used to project cells into lower**[?]** dimensions where the cells are clustered or a graph constructed between them. The trajectory is then created by finding a path through the cells and ordering the cells along it. This review compares the performance on a range of datasets... They found that...

An alternative continuous approach is the cell velocity method in the velocyto package. RNA-seq studies typically focus on the expression of complete mature mRNA molecules but a sample will also contain immature mRNA that are yet to be spliced. Examining these reads assigned to introns can indicate newly transcribed mRNA molecules and therefore which genes are currently active. Instead of assigning cells

to discrete groups or along a continuous path velocyto uses reads from unspliced regions to place them in a space and create a vector indicating the direction in which the transcriptional profile is heading. This vector can show the a cell is differentiation in a particular way or that a specific transcriptional program has been activated.

Deciding on which assignment approach to use depends on the source of the data, the goals of the study and the questions that are being asked. Both grouping and ordering can be informative and it is often useful to attempt both on a dataset and see how they compare.

### 1.4.5 Gene detection and interpretation

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

Once cells are assigned by clustering or ordering the problem is to interpret what these groups represent. For clustered datasets this is usually done by identifying genes that are differentially expressed across the groups or marker genes that are expressed in a single cluster. Many methods have been suggested for testing differential expression some of which take in to account the unique features of scRNA-seq data. For example…The large number of cells in scRNA-seq datasets mean that some of the problems that made standard statistical tests unsuitable for bulk RNA-seq experiments do not apply and simple methods like the unpaired Wilcoxon rank-sum test (or Mann-Whitney U test) may give reasonable results in this setting. Methods originally developed for bulk experiments have have also been applied to scRNA-seq datasets. Some of these methods have well understood statistical frameworks and have been shown to perform well in multiple comparisons. However the assumptions they make may not be appropriate for single-cell data and methods such as ZiNB-WaVe may be required to transform the data that is appropriate for their use.

Often the goal is not to find all the genes that are differentially expressed between groups but to identify genes which uniquely mark particular clusters. This goal is open to alternative approaches such as the Gini coefficient which measures unequal distribution across a population. Another approach is to construct machine learning classifiers for each genes to distinguish between one group and all other cells. Genes that give good classification performance should be good indicators of what is specific to that cluster.

When cells have been ordered along a continuous trajectory the task is slightly different. Instead of testing for a difference in means between two groups the goal is to find genes that have a relationship between expression and pseudotime. This can be accomplished by fitting splines and testing the coefficients. For more complex trajectories it can also be useful to find genes that are differently expressed along each side of a branch points. Monocle's BEAM method does this by… Genes that are associated

with a trajectory are important in their own right as they describe the biology along a path but they can also be used to identify cell types at end points.

Interpreting the meaning of detected markers genes is a difficult task as is likely to remain so. Gene set testing to identify related categories such as Gene Ontology terms can help but often it is necessary to rely the results of previous functional studies. This can only be reliably done by working closely with experts who have significant domain knowledge in the cell types being studied. An additional concern for unsupervised scRNA-seq studies is that the same genes are used for clustering or ordering and determining what those clusters or trajectories mean. This is a problem addressed by XXX who suggest a differential expression test using a long-tailed distribution for testing genes following clustering.

### 1.4.6 Alternative analyses

- Variant detection
- Cancer
- Immune cells

Some uses of scRNA-seq data fall outside the most common workflow and methods have been developed for a range of other purposes. For example methods have been designed for assigning haplotypes to cells, detecting allele-specific expression, identifying alternative splicing or calling single nucleotide or complex genomic variants. Other methods have been designed for specific cell types or tissues such as XXX which can assign immune cell receptors and XXX which interrogate the development of cancer samples. Most future studies can be expected to continue to follow common practice but it also expected that researchers will continue to push the boundaries of what it is possible to study using scRNA-seq technologies.

## 1.5 Kidney development

### 1.5.1 Structure and function

- Kidney structure
- Nephron structure
- Important cell types

In mammals the kidney is an organ responsible for filtering the blood in order to remove waste products. Kidneys grow as a pair with each being around the size of an adult fist and weighing about 150 g. with each being functional. Blood flows into the kidney via the renal artery and the blood vessels form a tree-like branching with ever smaller capillaries. At the end of these branches are nephrons, the functional filtration unit of the kidney. Humans can have around 1 million nephrons that are formed during development and just after birth, however they cannot be regenerated after around ... of age. A capillary loop is formed inside a structure at the end of the nephron called a glomerulous and surrounded by Bowman's capsule. Here specialised cells called podocytes create a structure called the slit diaphragm that allows water, metal ions

and small molecules to be filtered while keeping blood cells and larger species such as proteins trapped within the bloodstream. The rest of the nephron is divided into segments that are responsible for balancing the concentration of these species in the filtrate. The lumen of the nephron is surrounded by capillaries which allows content to be transferred between the filtrate and blood as required. The first segment of the nephron is the proximal tubule. Here common biomolecules such as glucose, amino acids and bicarbonate are reabsorbed into the bloodstream, as is most of the water. Other molecules including urea and ammonium ions are secreted from the blood into the filtrate at his stage. This proximal tubule is followed by the Loop of Henle and the distal tubule where ions are reabsorbed including potassium, chlorine, magnesium and calcium. The final segment is the collecting duct that balances salt concentrations by exchanging sodium in the filtrate for potassium in the bloodstream using a process controlled by the hormone aldosterone. The remaining filtrate is then passed to the ureter where it is carried to the bladder and collected as urine while the blood leaves via the renal vein. In order to preform this complex series of reabsorption and and secretion each segment of the nephron is made up specialised cell types with there own set of signaling and transporter proteins. The filtration process is repeated about 12 times every hour with around 200 litres of blood being filtered every day. Aside from removing waste and maintaining the balance of species in the bloodstream the kidneys also play a role in the activation of vitamin D and synthesises the hormones erythropoietin, which stimulates red blood cell production, and renin which is part of the pathway that controls fluid volume and the constriction of arteries to regulate blood pressure.

Chronic kidney disease is a major health problem in Australia with XXX percent of the population to experience it during their lifetime. Early stages of the disease can be managed but once it becomes severe the only treatment options are dialysis, which is expensive, time consuming and unpleasant, or a kidney transplant. There are also a range of developmental kidney disorders that have limited treatment options and can profoundly affect quality of life. Understanding how the kidney grows and develops is key to developing new treatments that may improve kidney function or repair damage.

### 1.5.2 Stages of development

- Lineage
- Important genes

The kidney develops from a region of the early embryo called the intermediate mesoderm and occurs in three phases with a specific spatial and temporal order. The first phase results in the pronephros which consists of 6-10 pairs of tubules that forms the mature kidney in most primitive vertebrates such as hagfish. By about the fourth week of human embryonic development this structure dies off and is replaced by the mesonephros which is the form of kidney present in most fish and amphibians. The mesonephros is functional during weeks 4-8 of human embryonic development before degenerating although parts of it's duct system go on for form part

of the male reproductive system. The final phase of human kidney development results in the metanephros which begins developing at around five weeks to become the permanent and functional kidney. Individual nephrons grow in a similar series of stages. Cells from the duct that will become the ureter begin to invade the surrounding metanephric mesenchyme forming a ureteric bud. Interactions between these cell types, including WNT signaling, cause mesenchymal cells to condense around the ureteric bud forming a stem cell population known as the cap mesenchyme that expresses genes such as Six2 and Cited1. Cells from the cap mesenchyme first form a renal vesicle, a primitive structure with a lumen, which extends to form an s-shaped body. By this stage the lumen has joined with the ureteric bud to form a continuous tubule. The s-shaped body continues to with podocytes beginning to develop and form a glomerulus at one end and other specialised cells arising along the length of the tubule to form the various nephron segments. Several signaling pathways and cell-cell interactions are involved in this process including Notch signaling.

Most of our understanding of kidney development comes from studies using mouse models and other model species. While these have greatly added to our knowledge they do not completely replicate human kidney development and there are known to be significant differences in the developmental timeline, signaling pathways and gene expression between species. To better understand human kidney development we need models that reproduce the human version of this process.

### 1.5.3 Growing kidney organoids

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

One alternative model of human kidney development is to grow miniature organs if a lab. Known as organoids these tissues are grown from stem cells provided with the right sequence of conditions and growth factors. Naturally occurring embryonic stem cells can be used but a more feasible approach is to reprogram mature cell types (typically fibroblasts from skin samples) using a method discovered by .... Under this protocol cells are supplied with ... followed by .... The resulting cells have the ability to differentiate into any cell type and are known as induced pluripotent stem cells (iPSCs). By culturing iPSCs under the right conditions the course of differentiation can be directed and protocols for growing eye, brain and ... tissues have been developed. The first protocol first protocol for growing kidney organoids was published in 2015 by Takasato et al.

Using this protocol iPSCs are first grown on a plate where Wnt signaling is induced by the presence of CHIR, an inhibitor of glycogen kinase synthase 3. After several days of growth the growth factor FGF9 is added which is required to form the

intermediate mesoderm. Following several more days of growth the cells are removed from the plate and formed into three dimensional pellets. A short pulse of CHIR is added to again induce Wnt signaling and the pellets continue to be cultured in the presence of FGF9. Growth factors are removed after about five days of 3D culture and the organoids continue to grow for a further two weeks at which point tubular structures have formed. These kidney organoids have been extensively characterised using both immunofluorescence imaging and transcriptional profiling by RNA-seq. Imaging showed that the tubules are segmented and express markers of podocytes, proximal tubule, distal tubule and collecting duct, however individual tubules are not connected in the same way they would be in a real kidney. By comparing RNA-seq profiles with those from a range of developing tissues the organoids from this protocol were found to be most similar to trimester one and two fetal kidney. While the bulk transcriptional profiles may be similar this analysis does not confirm that individual cells types the same lab-grown kidney organoids and the true developing kidney. Further studies using this protocol have shown that it is reproducible with organoids grown at the same time being having very similar transcriptional profiles however organoids from different batches can be significantly different, potentially due to differences in the rate at which they develop.

While they are not a perfect model of a developing human kidney organoids have several advantages over other models. In particular they have great potential for uses in the modeling of developmental kidney diseases. Cells from a patient with a particular mutation can be reprogrammed and used to grow organoids that can then be used for functional studies or drug screening. Alternatively gene editing techniques can be used to insert the mutation into an existing cell line or correct the mutation in the patient line allowing comparisons on the same genetic background. Modified versions of the protocol that can produce much larger numbers of organoids, for example by growing them in swirler cultures, could potentially be used to produce cells in sufficient numbers for cellular therapies. Extensive work is been done to improve the protocol in other ways as well such as improving the maturation of the organoids or directing them more towards particular segments. Overall kidney organoids open up many possibilities for studies to better help use understand kidney development and potentially help develop new treatments for kidney disease.

# Chapter 2
# The scRNA-seq tools landscape

## 2.1 Introduction

When I began my PhD in early 2016 single-cell RNA-sequencing technologies were just beginning to become widely available. Since then there has been a rapid uptake and there are now many studies using this approach. Along with the growth in the adoption of scRNA-seq technologies there has been an explosion in the number of software tools for analysing these datasets. This chapter charts the growth in the scRNA-seq analysis landscape over time.

In 2016 there were relatively few analysis methods available and to answer questions like how many tools perform a particular task or which areas were developers focusing on or was there a tool for doing this I began to record details about them. Inspired by similar projects such as Sean Davis' Awesome Single Cell page I decided to make this collection public. This turned out to be useful to other researchers and over time a simple spreadsheet became the scRNA-tools database and website (`https://scRNA-tools.org`). A paper published in *PLoS Computational Biology* describing this resource forms the main part of this chapter.

By having access to details about existing analysis tools we were able to explore how the field has developed. We found that computational researchers had focused their efforts on analysis tasks specific to scRNA-seq data such as clustering and ordering of cells or handling the larger numbers of zeros. We also saw that many of the tools performed tasks common to several stages of analysis including dimensionality reduction of various kinds and visualisation. Developers of scRNA-seq analysis tools tend to embrace a open-source and open-science approach. Most tools are developed on GitHub were others can ask questions and submit improvements. The majority are also available under open-source licenses allowing their code to be reused for other purposes, although there is also a significant proportion that do not have any associated license. Tools are commonly made public by releasing a preprint publication, making them available to the community much more quickly and giving early adopters a chance to contribute to their development.

A section at the end of this chapter presents an updated version of some of this analysis based on the most recent version of the database.

# Chapter 3

# Simulating scRNA-seq data

## 3.1 Introduction

To be accepted and used any computational method for data analysis needs to demonstrate that it is effective at the task it aims to complete. Ideally this can be done by evaluating performance on a real dataset where the results are already known. Unfortunately in many case such gold standard datasets are not available. This is particular true for genomic data where it is difficult to know what the truth is or it is limited to only small sections of the genome. It is possible to create some genomic datasets where the truth is known, for example through carefully performed mixing experiments, but these often do not capture the true biological complexity. In many cases the most effective way to evaluate an analysis method is by testing it on a simulated datasets. Simulations have the additional advantage of relatively cheap and easy to produce allowing exploration of a wide range of possible parameters. This is the approach taken by many early methods for scRNA-seq analysis but often the simulations they used where not well explained, code for reproducing them was not available and perhaps most importantly they didn't show that the synthetic datasets were similar to real scRNA-seq data.

This chapter presents Splatter, a software package for simulating scRNA-seq datasets presented in a publication in *Genome Biology*. Splatter is designed to provide a consistent, easy-to-use interface for multiple scRNA-seq simulation models previously used to develop analysis tools. We do this by providing two functions for each model, one which estimates parameters from a real dataset and a second that generates a synthetic dataset using those parameters. Each model has different assumptions and reproduces different aspects of scRNA-seq data and we explain these differences in the paper. We also present Splat, our own simulation model based on the Gamma-Poisson distribution. This model includes several aspects of scRNA-seq data including highly expressed outliers genes, differences in library sizes between cells a relationship between the mean and the variance of each gene and the ability to add a dropout effect linked to gene expression. When designing the Splat simulation our goal was to reproduce scRNA-seq data as well as possible rather than test a specific method with the result being that the model is highly flexible and able to generate a range of scenarios including datasets with multiple groups of cells, batch effects and continuous trajectories.

In the paper we compare how well each simulations reproduces a range of scRNA-seq datasets including UMI and full-length protocol, different capture methods and homogenous and complex tissues. We found that the Splat simulation was a good match for some of these simulations across a range of methods, however it was also clear that some models more faithfully reproduced different aspects of the data, particularly for datasets from different sources. The Splatter R package is available for download from Bioconductor (`https://bioconductor.org/packages/splatter`).

## 3.2 Splatter publication

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

Genome Biology

METHOD Open Access

# Splatter: simulation of single-cell RNA sequencing data

Luke Zappia[1,2], Belinda Phipson[1] 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 journals or as preprints [4]. The focus of these tools is often different from those designed for the analysis of a 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 relatively few genes in each cell. The observed zero counts have both biological (different cell types express different genes) and technical (an expressed RNA molecule might not be captured) causes, with technical zeros often referred to as "dropout". Some analysis methods (ZIFA [11], MAST [12], ZINB-WaVE [13]) incorporate dropout into their models while others (MAGIC [14], SAVER [15], scImpute [16]) attempt to infer what the true expression levels should be.

* Correspondence: alicia.oshlack@mcri.edu.au
[1]Murdoch Childrens Research Institute, Royal Children's Hospital, 50 Flemington Rd, Parkville, VIC 3052, Australia
[2]School of Biosciences, The University of Melbourne, Parkville, VIC 3010, Australia





## Results

### The Splatter framework

Currently, Splatter implements six different simulation

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 $i$, $i = 1, ..., N$, is simulated from a gamma distribution and the count for each cell $j$, $j = 1, ..., M$, 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 $\alpha$ and rate $\beta$. While the gamma distribution is a good fit for gene means it does not always capture extreme expression levels. To counter this a probability ($\pi^O$) 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 $\mu^O$ and scale $\sigma^O$.

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 $\mu^L$ and scale $\sigma^L$) 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 ($\lambda_{i,j}$) 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 $\lambda_{i,j}$. 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 $Y_{i,j}$ 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

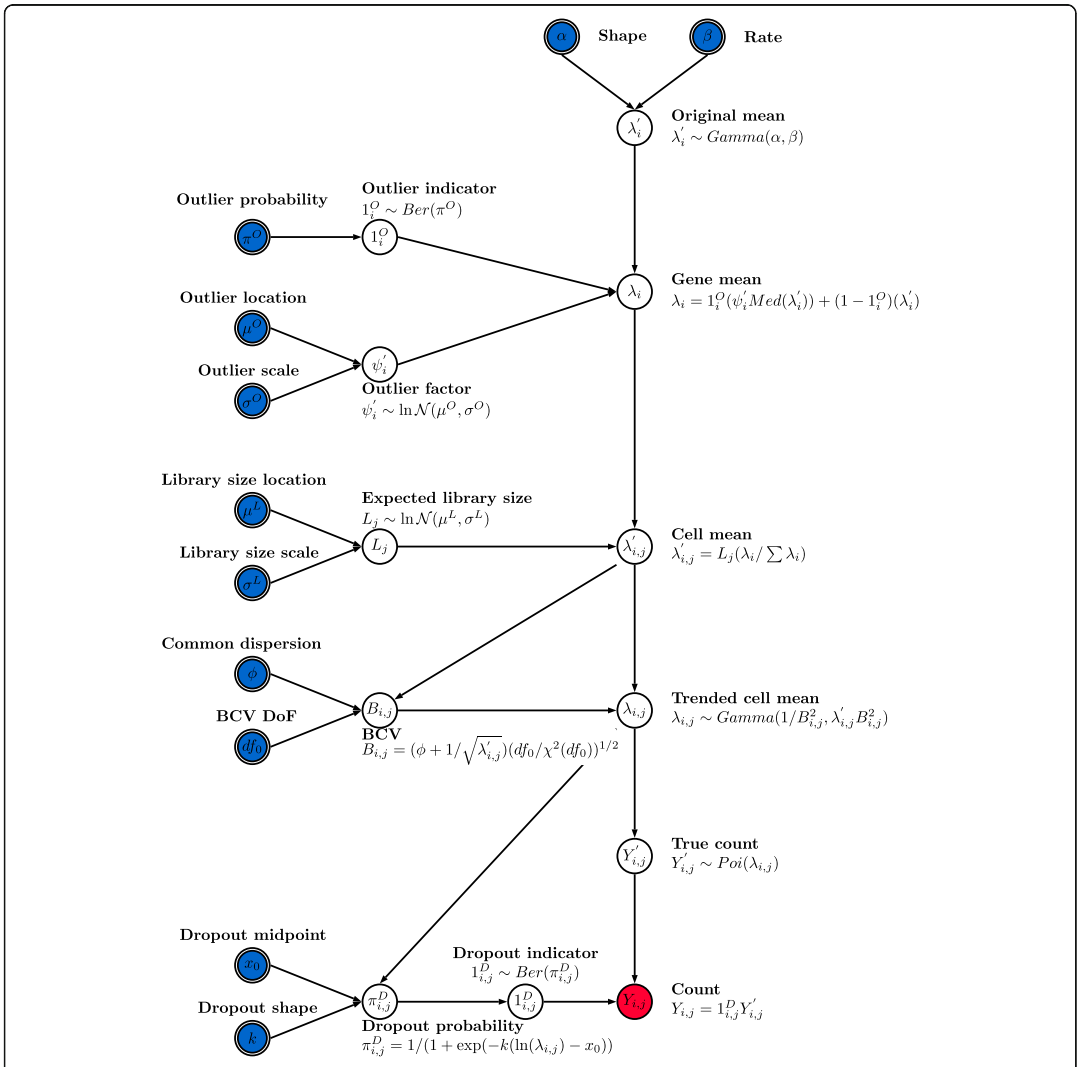


**Fig. 1** The Splat simulation model. Input parameters are indicated with *double borders* and those that can be estimated from real data are shaded *blue*. *Red shading* indicates the final output. The simulation begins by generating means from a gamma distribution. Outlier expression genes are added 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 | $\alpha$ | Shape parameter for the mean gene expression gamma distribution |
| Mean rate | $\beta$ | Rate parameter for the mean gene expression gamma distribution |
| Library size location | $\mu^L$ | Location parameter for the library size log-normal distribution |
| Library size scale | $\sigma^L$ | Scale parameter for the library size log-normal distribution |
| Outlier probability | $\pi^O$ | Probability that a gene is an expression outlier |
| Outlier location | $\mu^O$ | Location parameter for the expression outlier factor log-normal distribution |
| Outlier scale | $\sigma^O$ | Scale parameter for the expression outlier factor log-normal distribution |
| Common BCV | $\varphi$ | 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

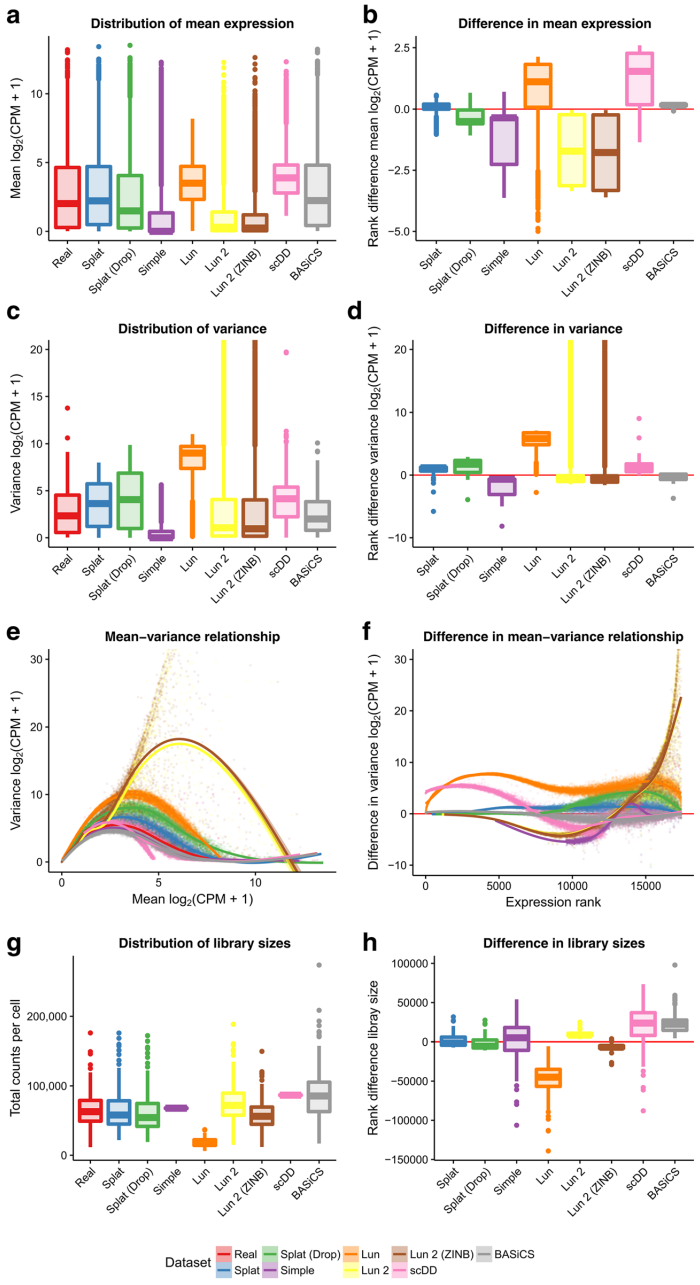


**Fig. 2** Comparison of simulations based on the Tung dataset. The left column panels show the distribution of mean expression (**a**), variance (**c**) and library size (**g**) across the real dataset and the simulations as boxplots, along with a scatter plot of the mean–variance relationship (**e**). The right column shows boxplots of the ranked differences between the real data and simulations for the same statistics: mean (**b**), variance (**d**), mean–variance relationship (**f**), and library size (**h**). Note that the *y-axis* for plots of the variance has been limited in order to show more detail. Variances for the Lun and Lun 2 simulations extend beyond what has been shown

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

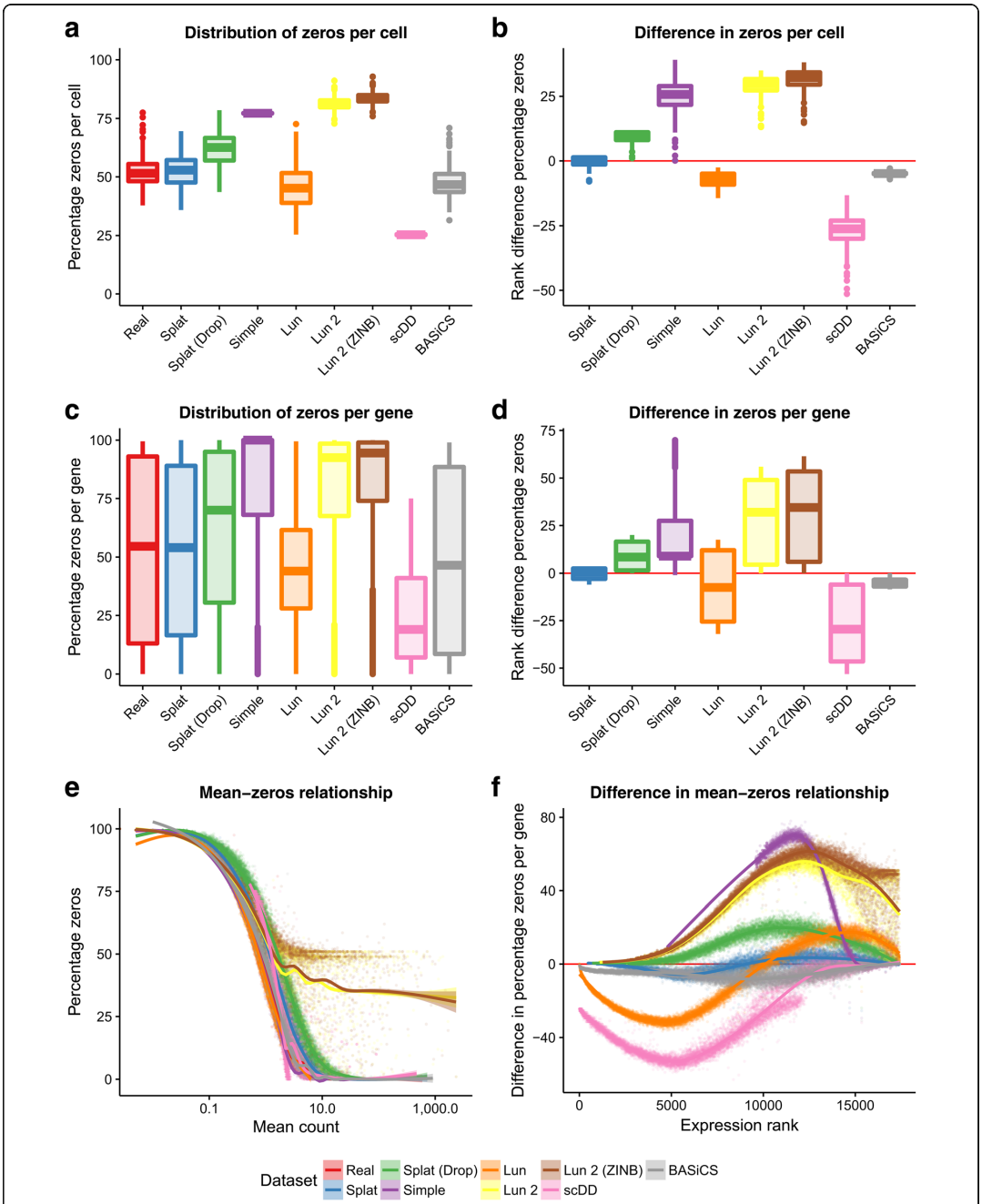


**Fig. 3** Comparison of zeros in simulations based on the Tung dataset. The *top row* shows boxplots of the distribution of zeros per cell (**a**) and the difference from the real data (**b**). The distribution (**c**) and difference (**d**) in zeros per gene are shown in the *middle row*. The *bottom row* shows scatter plots of the relationship between the mean expression of a gene (including cells with zero counts) and the percentage of zeros as both the raw observations (**e**) and ranked differences from the real data (**f**)

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

**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 |

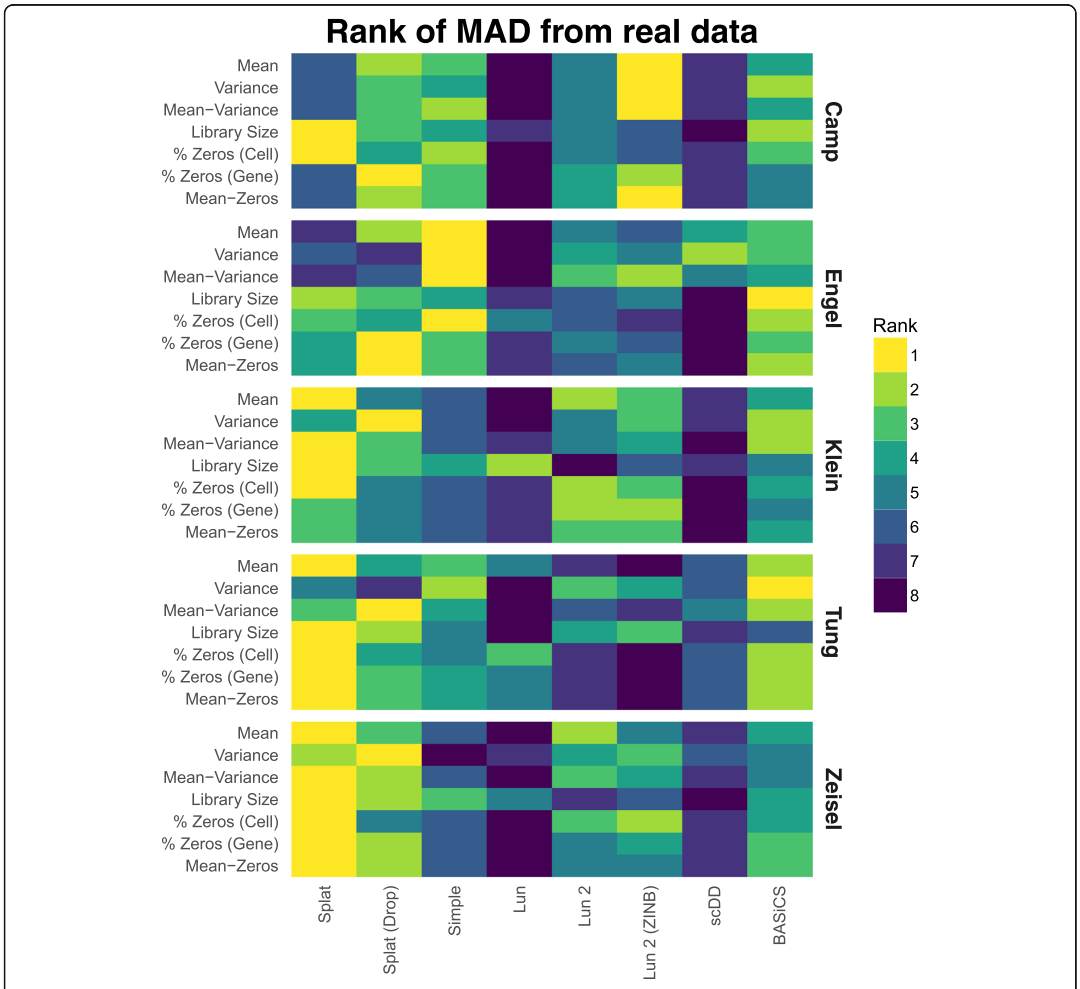


**Fig. 4** Comparison of simulation models based on various datasets. For each dataset parameters were estimated and synthetic datasets generated using various simulation methods. The median absolute deviation (*MAD*) between each simulation and the real data was calculated for a range of metrics and 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

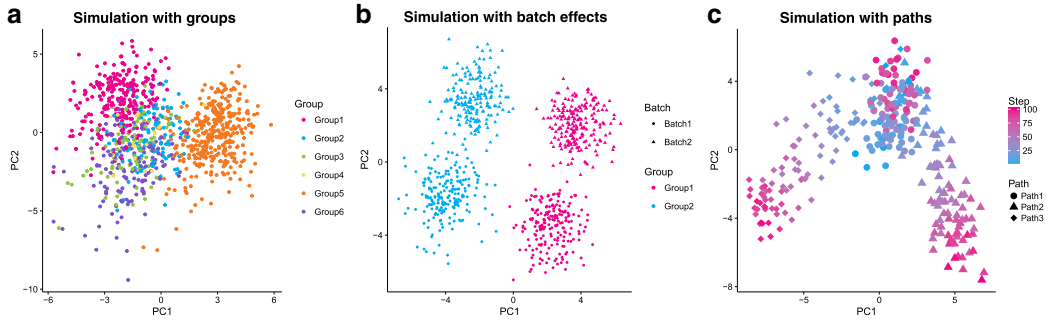


**Fig. 5** Examples of complex Splat simulations. **a** A principle components analysis (PCA) plot of a simulation with six groups with varying numbers of cells and levels of differential expression. **b** A PCA plot of a simulation with two groups (*pink* and *blue*) and two batches (*circle* and *triangle*). *PC1* separates groups (wanted biological variation) while *PC2* separates batches (unwanted technical variation). **c** A PCA plot of a simulation with differentiation paths; the colored gradient indicates how far along a path each cell is from *blue* to *pink*. A progenitor cell type (*blue circles*) differentiates into an intermediate cell type (*pink circles/blue triangles* or *diamonds*), which becomes one of two (*pink triangle* or *diamond*) mature cell types

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.

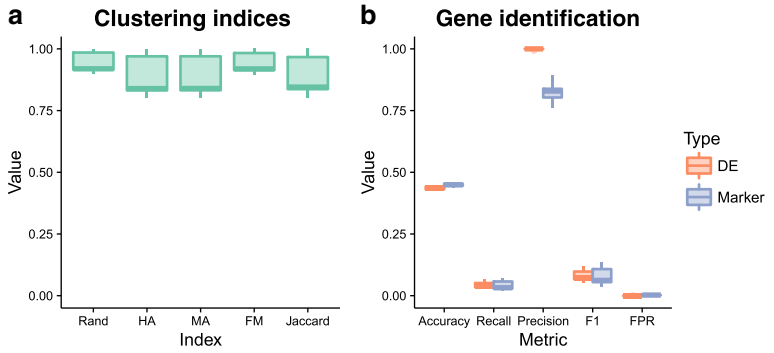


**Fig. 6** Evaluation of SC3 results. Metrics for the evaluation of clustering (**a**) include the Rand index, Hubert and Arabie's adjusted Rand index (*HA*), Morey and Agresti's adjusted Rand index (*MA*), Fowlkes and Mallows index (*FM*), and the Jaccard index. Detection of differentially expressed 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 ($\alpha$ and $\beta$) 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 ($\mu^L$ and $\sigma^L$) 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 $\pi^O$ is then calculated as the proportion of genes that are outliers. Parameters for the outlier factors ($\mu^O$ and $\sigma^O$) 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**


1. Goodwin S, McPherson JD, Richard McCombie W. Coming of age: ten years of next-generation sequencing technologies. Nat Rev Genet. 2016;17:333–51.
2. Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. Nat Rev Genet. 2011;12:87–98.
3. 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.
4. scRNA-tools. http://www.scrna-tools.org/.
5. 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.
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.
7. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33:495–502.
8. 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.
9. 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.
10. 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.
11. Pierson E, Yau C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. 2015;16:241.
12. 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.
13. 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.
14. van Dijk D, Nainys 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.
15. 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.
16. Li WV, Li JJ. scImpute: accurate and robust imputation for single cell RNA-Seq data. 2017. http://biorxiv.org/content/early/2017/05/24/141598.
17. 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.
18. 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.
19. Lun ATL, Marioni JC. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. Biostatistics. 2017;18:451–64.
20. 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.
21. Vallejos CA, Marioni JC, Richardson S. BASiCS: Bayesian analysis of single-cell sequencing data. PLoS Comput Biol. 2015;11:e1004333.
22. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26:139–40.
23. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol. 2010;11:R106.
24. Korthauer K. scDD vignette. 2017. https://bioconductor.org/packages/release/bioc/vignettes/scDD/inst/doc/scDD.pdf.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. Akima H. A new method of interpolation and smooth curve fitting based on local procedures. JACM. 1970;17:589–602.
32. Akima H, Gebhardt A. akima: interpolation of irregularly and regularly spaced data. 2016. https://CRAN.R-project.org/package=akima.
33. Huber W, Carey JV, Gentleman R, et al. Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods. 2015;12:115–21.
34. Phipson B, Zappia L, Oshlack A. Gene length and detection bias in single cell RNA sequencing protocols. F1000Res. 2017;6:595.
35. Delignette-Muller M, Dutang C. fitdistrplus: an R package for fitting distributions. J Stat Softw. 2015;64:1–34.
36. Sadedin SP, Pope B, Oshlack A. Bpipe: a tool for running and managing bioinformatics pipelines. Bioinformatics. 2012;28:1525–6.
37. 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.
38. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics. 2014;30:923–30.
39. 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.
40. Soneson C, Love MI, Robinson MD. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Res. 2015;4:1521.
41. R Core Team. R: a language and environment for statistical computing. Vienna: R Foundation for Statistical Computing; 2016. https://www.R-project.org/.
42. 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.
43. Wickham H. ggplot2: elegant graphics for data analysis. New York: Springer; 2010.
44. 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.
45. Engel I, Seumois 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.
46. 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.
47. Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnerberg 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.

# Chapter 4

# Visualising clustering across resolutions

## 4.1 Introduction

Clustering of cells to form groups is a common task when analysing scRNA-seq data that is not required for bulk RNA-seq experiments and one that has received a lot of attention from analysis methods developers. The need to group samples is not unique to genomic data and clustering techniques are used in many other fields for a wide variety of purposes. Whatever kind of data you are interested in and whatever clustering method is being used a question that commonly comes up is how many clusters do we want to have? This can be controlled by setting an exact value, changing a parameter that indirectly controls the clustering resolution or affected by the values of other parameters and the number of clusters that are used can often have a profound affect on how the results are interpreted. Existing measures of clustering typically only consider a single clustering resolution at a time or require multiple rounds or permutations and clustering which can be infeasible for large datasets. In this chapter I propose an alternative visualisation-based aid for deciding which clustering resolution to use.

Clusterings of the same dataset at different resolutions are often related and it is common for new clusters formed at higher resolutions to be formed by splitting existing clusters. However when comparing clusterings it is not always clear what those relationships are and how significant they might be. The method I describe here was published in *GigaScience* and suggests clustering datasets at multiple resolutions then considering the overlap in samples at neighbouring resolutions. By doing this we can build a graph structure we call a "clustering tree". Visualising this tree allows us to see where new clusters form, how they are related and the stability of particular clustering resolutions. In the paper we demonstrate this approach using simulated datasets, a simple dataset commonly used as an example for machine learning techniques and a complex scRNA-seq dataset from blood.

While the structure of clustering trees can help choose a clustering resolutions to use they are more generally a compact, information dense visualisation that can show information across clustering resolutions. This is something that is not possible with traditional visualisations used for clustering results such as t-SNE projections and is

achieved by trading individual information about each sample for summarised information about clusters and adding a resolution dimension. Overlaying important domain knowledge (such as the expression of known marker genes) onto these visualisations can be particularly informative and we also demonstrate this in our paper.

Clustering trees can be produced using the clustree R package which is built on the tidygraph and ggraph packages and is available from CRAN (`https://cran.r-project.org/package=clustree`).

# Chapter 5

# Analysis of kidney organoid scRNA-seq data

# Chapter 6
# Conclusion

# References