Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R

Davis J. McCarthy [1,2,5]*, Kieran R. Campbell [2,4], Aaron T. L. Lun [6] and Quin F. Wills [2,3]

[1] European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI), Hinxton CB10 1SD, Cambridgeshire, UK;
[2] Wellcome Trust Centre for Human Genetics, University of Oxford, Oxford, Oxfordshire, UK;
[3] Weatherall Institute for Molecular Medicine, University of Oxford, Oxford, Oxfordshire, UK;
[4] Department of Anatomy, Physiology and Genetics, University of Oxford, Oxford, Oxfordshire, UK;
[5] St Vincent's Institute of Medical Research, 41 Victoria Parade Fitzroy Victoria 3065, Australia; and
[6] CRUK Cambridge Institute, Robinson Way, Cambridge CB2 0RE, Cambridgeshire, UK.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Single-cell RNA sequencing (scRNA-seq) is increasingly used to study gene expression at the level of individual cells. However, preparing raw sequence data for further analysis is not a straightforward process. Biases, artifacts, and other sources of unwanted variation are present in the data, requiring substantial time and effort to be spent on pre-processing, quality control (QC) and normalisation.
Results: We have developed the R/Bioconductor package *scater* to facilitate rigorous pre-processing, quality control, normalisation and visualisation of scRNA-seq data. The package provides a convenient, flexible workflow to go from raw reads through quality control to a tidy dataset ready for downstream analysis. *scater* provides a rich suite of plotting tools for single-cell data and a flexible data structure that is compatible with existing tools and can be used as infrastructure for future software development.
Availability: The open-source code, along with installation instructions, vignettes and case studies, is available through Bioconductor at http://bioconductor.org/packages/scater.
Contact: davis@ebi.ac.uk
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) describes a broad class of techniques which profile the transcriptome of individual cells. This provides insights into cellular processes at a resolution that cannot be matched by bulk RNA-seq experiments (Hebenstreit and Teichmann, 2011; Shalek *et al.*, 2013). With scRNA-seq data, the contributions of different cell types to the expression profile of a heterogeneous population can be explicitly determined. Rare cell types can be interrogated and new cell subpopulations can be discovered. Graduated processes such as development and differentiation can also be studied in greater detail. However, this improvement in resolution comes at the cost of increased technical noise and biases. This means that pre-processing, quality control and normalisation are critical to a rigorous analysis of scRNA-seq data. The increased complexity of the data across hundreds or thousands of cells also requires sophisticated visualisation tools to assist interpretation of the results.

Numerous statistical methods and software tools have been published for scRNA-seq data (Guo *et al.*, 2015; Kharchenko *et al.*, 2014; Finak *et al.*, 2015; Delmans and Hemberg, 2016; Angerer *et al.*, 2015; Kiselev *et al.*, 2016; Juliá *et al.*, 2015; Trapnell *et al.*, 2014). However, all of these assume that quality control and normalisation have already been applied. Fewer methods are available in the literature to perform these basic steps in scRNA-seq data processing (Ilicic *et al.*, 2016). The issue is exacerbated by the diversity of scRNA-seq data sets with respect to the experimental protocol and the biological context of the study, meaning that a single processing pipeline with fixed parameters is unlikely to be universally applicable. Rather, software tools are required that support an interactive approach to analysis, but nevertheless encourage good bioinformatic practices. This allows parameters to be fine-tuned for the study at hand in response to any issues diagnosed during data exploration, while supporting rigour and reproducibility.

One of the most popular frameworks for interactive analysis is the R programming language, extended for biological data analysis through the Bioconductor project (Huber *et al.*, 2015). While Bioconductor packages have been widely used for bulk RNA-seq data, the existing data structures (like the excellent ExpressionSet class) and methods are not sufficient for scRNA-seq data. An extension to ExpressionSet is desirable to add capabilities specifically useful for scRNA-seq analyses. Furthermore, single-cell datasets are now incorporating more data modalities than just transcriptomic data. FACS data, cell imaging, epigenetic and targeted genotyping assays and repeated measurements of cells using platforms such as the Polaris (www.fluidigm.com/products/polaris) are enabling richer datasets that become challenging to integrate. Successful integration requires an appropriate data structure to organise the expression data, other assay data and accompanying metadata. Current visualisation methods designed for exploratory data analysis of bulk transcriptomic experiments are unsuited to scRNA-seq data sets containing hundreds or thousands of cells. The large size of each dataset also favours methods such as *kallisto* (Bray *et al.*, 2016) and *Salmon* (Patro *et al.*, 2015) for rapidly quantifying gene expression, and there is a need for new computational infrastructure to process raw scRNA-seq sequence data into a high-quality expression dataset ready for downstream analysis.

Here we present *scater*, an open-source R/Bioconductor software package that implements a convenient data structure for representing scRNA-seq data and contains functions for pre-processing, quality control, normalisation and visualisation. The package provides wrapper functions for running *kallisto* and *Salmon* on raw read data and converting their output into gene-level expression values, methods for computing and visualising quality-control metrics for cells and genes, and methods for normalisation and correction of uninteresting covariates. This is done in a single software environment which enables seamless integration with a large number of existing tools for scRNA-seq data analysis in R. The *scater* package provides basic infrastructure upon which customized scRNA-seq analyses can be constructed, and we anticipate the package to be useful across the whole spectrum of users, from experimentalists and those less experienced to in bioinformatics to seasoned computational scientists.

2 Methods, Data and Implementation

2.1 Case study with scRNA-seq data

The results presented in the main paper and supplementary case study use an unpublished single-cell RNA-seq dataset consisting of 73 cells from two lymphoblast cell lines of two unrelated individuals. Cells were captured, lysed, and cDNA generated using the popular C1 platform from Fluidigm, Inc. (www.fluidigm.com/products/c1-system). The processing of the two cell lines was replicated across two machines, with the nuclei of the two cell lines stained with different dyes before mixing on each machine. Cells were imaged before lysis, with an example image provided together with these data (see Case Study in Supplementary Material). Further case studies using *scater* on published data, for example from 3000 mouse cortex cells (Zeisel *et al.*, 2015) and 1200 cells from early-development mouse embryos (Scialdone *et al.*, 2016) are available at dx.doi.org/10.5281/zenodo.59897.

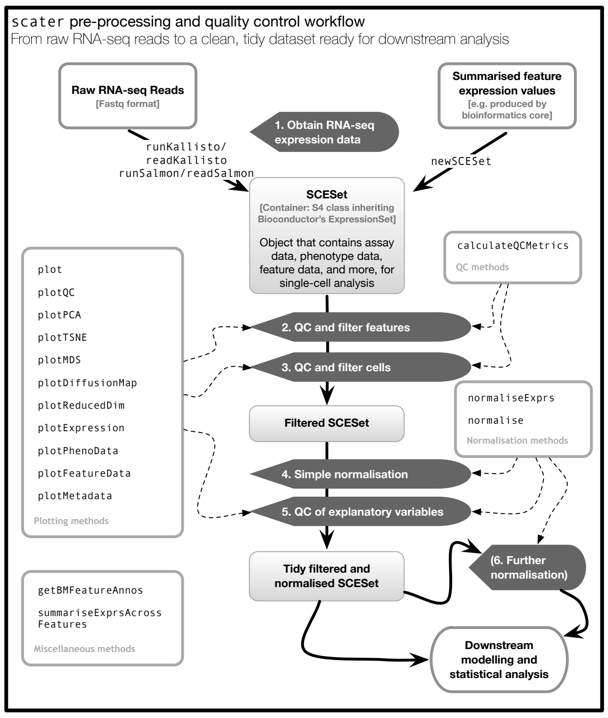

Fig. 1. An overview of the scater workflow, from raw sequenced reads to a tidy data set ready for higher-level downstream analysis. For step 5, explanatory variables include experimental covariates like batch, cell source and other recorded information, as well as QC metrics computed from the data. Step 6 describes an optional round of normalisation to remove effects of particular explanatory variables from the data. Automated computation of QC metrics and extensive plotting functionality support the workflow.

2.2 Implementation

The *scater* package is an open-source R package available through Bioconductor. Key aspects of the code are written in C++ to minimise computational time and memory use. The package builds on many other R packages: *Biobase* and *BiocGenerics* for core Bioconductor functionality (Huber *et al.*, 2015); *plyr* (Wickham, 2015), *reshape2* (Wickham, 2012), *dplyr* (Wickham and Francois, 2015), *data.table* (Dowle *et al.*, 2015) and *magrittr* (Bache and Wickham, 2014) for reading and tidying data; *ggplot2* (Wickham, 2016) for plotting; *biomaRt* (Durinck *et al.*, 2005) for feature annotation; *edgeR* (Robinson *et al.*, 2010) for computation of normalisation size factors and counts-per-million values; *limma* (Ritchie *et al.*, 2015) for efficient fitting of linear models to features; *rhdf5* (Fischer and Pau, 2016), *rjson* (Couture-Beil, 2014) and *tximport* (Soneson *et al.*, 2015) for reading in transcript-level expression values; *viridis* (Garnier, 2016) for perceptually-uniform colour maps for plotting; *parallel* for parallel computation; *matrixStats* (Bengtsson, 2016) for computation of summary statistics from matrices; *cowplot* (Wilke, 2016) for attractive plotting themes; *destiny* (Angerer *et al.*, 2015) for producing diffusion maps; *Rtsne* (Krijthe, 2015) for producing t-SNE plots; *mvoutlier* (Filzmoser and Gschwandtner, 2015) for multivariate outlier detection from PCA of QC metrics; *roxygen2* (Wickham *et al.*, 2015), *BiocStyle* (Huber *et al.*, 2015), *knitr* (Xie, 2013) and *rmarkdown* (Allaire *et al.*, 2016) for generating documentation; and *testthat* (Wickham, 2011) for unit testing. As well as functioning in the usual R environments, *scater* also has a GUI built using *shiny* (Chang *et al.*, 2016) and *shinydashboard* (Chang, 2015) for intuitive and interactive data visualisation. Calling the `scater_gui` function from within an R session opens up the GUI in a web browser.

3 Results

3.1 The *scater* package

The *scater* package offers a workflow to convert raw read sequences into a data set ready for higher-level analysis within the R programming environment (Figure 1). In addition, *scater* provides basic computational infrastructure to standardise and streamline scRNA-seq data analyses. Key features of *scater* include: (1) the "single-cell expression set" (SCESet) class, a data structure specialized for scRNA-seq data; (2) wrapper methods to run *kallisto* and *Salmon* and process their output into gene-level expression values; (3) automated calculation of quality control metrics, with QC visualisation and filtering methods to retain high-quality cells and informative features; (4) extensive visualisation capabilities for inspection of scRNA-seq data; and (5) methods to identify and remove uninteresting covariates affecting expression across cells. The package integrates many commonly used tools for scRNA-seq data analysis and provides a foundation on which future methods can be built. The methods in *scater* are agnostic to the form of the input data and are compatible with counts, transcripts-per-million, counts-per-million, FPKM or any other appropriate transformation of the expression values.

3.2 SCESet: a data structure for single-cell expression data

The *scater* package is built around the SCESet class (Supplementary Figure 1) which provides a sophisticated container for scRNA-seq data. This class inherits from the ExpressionSet class in Bioconductor's *Biobase* package (Huber *et al.*, 2015), which allows assay data (and multiple transformations thereof), gene or transcript metadata and sample metadata to be combined in a single object to empower robust analyses. ExpressionSet has proven a cornerstone of microarray and bulk RNA-seq analysis methods in Bioconductor, but extensions to it are desirable to add capabilities for scRNA-seq analyses. Specifically, the SCESet class adds slots for: a reduced-dimension representation of cells, cell-cell and gene-gene pairwise distance matrices, bootstrapped expression results (such as from *kallisto*), consensus clustering results, information about feature controls (such as ERCC spike-ins) and several more (Supplementary Figure 1). With these extra slots, SCESet objects can support analyses of scRNA-seq data in *scater* and in packages that build on *scater* that ExpressionSet could not.

An SCESet data object can be easily subsetted by row or column to remove unwanted genes or cells, respectively, from all data and metadata fields stored in the object. Furthermore, data and metadata in multiple SCESet objects can be easily combined e.g., to incorporate cells from different experimental batches. SCESet objects can also be converted to other R data structures, or saved to disk in structured, shareable formats. Further details on the class, including its motivation and execution, are available in the Supplementary Case Study and the package documentation. All methods available in *scater* are applicable to instances of the SCESet class and exploit the availability and richness of (meta)data stored in each SCESet object. Single-cell datasets are now incorporating more data modalities than just transcriptomic data; many of these (such as cell-level data such as FACS marker expression) can currently be captured in SCESet objects, and future development of the package will incorporate different types of assay data (such as targeted genotyping and epigenetic data) into the class.

3.3 Data pre-processing

Once raw read data have been obtained, the expression level of genomic features such as transcripts or genes must be quantified. Approaches to expression quantification from raw reads are, in principle, the same for scRNA-seq as they are for bulk RNA-seq (Kanitz *et al.*, 2015; Teng *et al.*, 2016). Read counts obtained from conventional quantification methods such as *HTSeq* (Anders *et al.*, 2015) and *featureCounts* (Liao *et al.*, 2014) can be readily used in a *scater* workflow (Figure 1). Another option is to use computationally-efficient pseudoalignment methods such as *kallisto* and *Salmon*. This is especially appealing for large scRNA-seq data sets containing hundreds to tens of thousands of cells. To this end, *scater* also provides wrapper functions for *kallisto* and *Salmon* so that fast quantification of transcript-level expression can be managed completely within an R programming environment. A common subsequent step for these methods is to collapse transcript-level expression to gene-level expression. Exploiting the *biomaRt* R/Bioconductor package, *scater* provides a convenient function for using Ensembl annotations to obtain gene-level expression values and gene or transcript annotations (Yates *et al.*, 2016).

3.4 Data quality control

The *scater* package provides methods to compute relevant QC metrics for an SCESet object. Given a set of control genes and/or cells, a variety of QC metrics will be computed and returned to the object in a single call to the calculateQCMetrics function (see package documentation). The only type of QC that *scater* cannot do is read-level QC, but this information can easily be incorporated into a *scater* object when such metrics are available from other alignment or quantification tools. The QC metrics computed in *scater* include, for each cell, the total count across all genes, the total number of expressed genes, and the percentage of counts allocated to control genes like spike-in transcripts or mitochondrial genes. These metrics (and others computed) are useful for building up a picture of the complexity of the transcriptome captured for the cell. For example, a high percentage of counts mapping to spike-ins typically indicates a small amount of RNA captured for the cell, suggesting protocol failure or death of the cell in processing and means it is unlikely to be suitable for downstream analyses. For each gene, QC metrics such as the average expression level and the proportion of cells in which the gene is expressed are computed. The metrics are used by *scater* to construct QC plots to explore the data and diagnose potential issues. This facilitates quality control which—despite attempts at automation (Ilicic *et al.*, 2016)—still requires manual intervention to account for aspects of the data specific to each study. The package documentation provides full details of the QC metrics produced.

In *scater*, the default plot method for an SCESet object produces a cumulative expression plot (Figure 2a). This is an underappreciated type of plot that describes how reads are distributed across genes, distinguishing between low-complexity libraries (where very few genes contain most of the counts) and their high-complexity counterparts (where counts are distributed more evenly across genes). There is substantial variability in library complexity among cells in this case study dataset. Some cells have profiles similar to the blank wells, suggesting that library preparation or sequencing failed for these cells and that the corresponding libraries should be removed prior to further analysis. Cell phenotype variables can be incorporated into these plots to highlight differences in expression distributions for different types of cells. For example, the curve for each cell is coloured by the type of well that produced the library (Figure 2a), while cells can also be split into separate facets by library type to show more metadata variables simultaneously (see Supplementary Case Study). Cumulative expression plots should be favoured over boxplots as the default method for visualising expression distributions across cells in a dataset, as the latter performs poorly at handling the long tail of low- and zero-expression observations in scRNA-seq data.

The plotPCA function implements an approach to automatic outlier detection using multivariate normal methods applied to the cell-level QC metrics (Ilicic *et al.*, 2016). Specifically, PCA is applied to the QC metrics for all cells and a plot is produced to automatically detect outliers in the higher-dimensional QC metric space (Figure 2b). These outliers

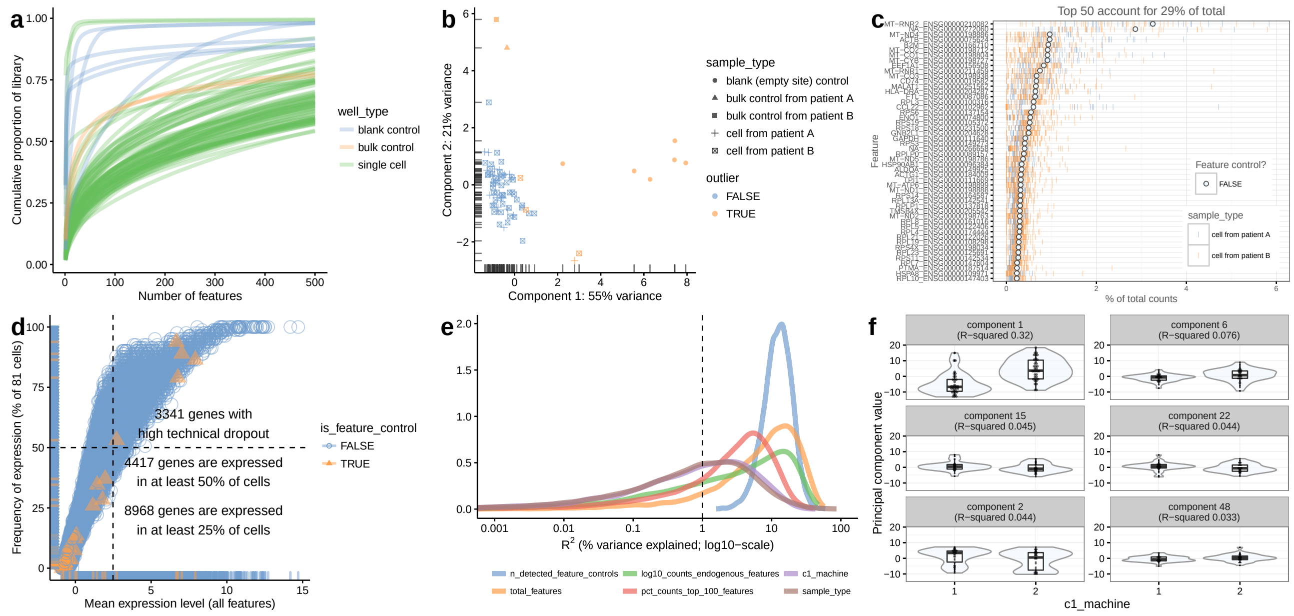

Fig. 2. Six types of QC plot made easily available in the scater package. (a) Cumulative expression plot showing the proportion of the library accounted for by the x most-expressed features (from 1 to 500). (b) PCA plot produced using a subset of the QC metrics computed with scater's calculateQCMetrics function. (c) Plot of the 50 most-expressed features (here, computed according to the highest total read counts) across all cells in the data set. For each feature, the circle represents the percentage of counts that the gene accounts for computed from counts pooled across all cells. The genes are ordered by this value. The bars for each cell show the percentage of counts accounted for by the gene for each individual cell, providing a visualisation of the distribution across cells. (d) Plot of frequency of expression (% of cells in which the feature is deemed expressed) against mean expression level across cells. (e) Density plot showing the percentage of variance explained by a set of explanatory variables across all genes. (f) Violin, scatter- and boxplots of principal component values against the C1 machine used for each cell for the six principal components most strongly correlated with C1 machine used. Each individual plot is produced by a single call with either the function plot (a), plotPCA (b) or plotQC (c-f).

correspond to low-quality cells with abnormal library characteristics (e.g., low total counts and few expressed genes) that should be removed prior to downstream analysis. This automated approach is generally less interpretable, and so complements simpler filtering approaches that apply thresholds to particular QC metrics.

The plotQC function generates many types of plots useful for quality control, such as a plot of visualising the most highly-expressed features in the dataset (Figure 2c). This provides a feature-centric overview of the dataset that simultaneously visualises the features with highest total expression across all cells, while also displaying the distribution of cell-level expression values for these features. It is common to see ERCC spike-ins (if used), mitochondrial and ribosomal genes among the highest expressed genes, while datasets consisting of healthy cells will also show high levels of constitutively expressed genes like *GAPDH* and *ACTB*. This plot allows the analyst to quickly check that the gene- or transcript-level quantification is behaving as expected, and to flag datasets where it is not.

A key feature of scRNA-seq is the high frequency of "dropout" events, that is, no observed expression (such as no read counts) in a particular cell for a gene that is actually expressed in that cell. Typically only a small set of genes are observed with detectable expression in every cell. The plotQC function can also be used to visualise the frequency of expression (inverse of the dropout percentage) of features against their average expression level (Figure 2d). Control features can be highlighted easily in the plot, and typical scRNA-seq datasets will show a broadly sigmoidal relationship between average expression level and frequency of expression across cells. This is consistent with expected behaviour where genes with greater average expression are more readily captured during library preparation and are detected at a greater frequency (Brennecke *et al.*, 2013; Kim *et al.*, 2015; Vallejos *et al.*, 2015).

Variation in expression data is driven by wanted and unwanted effects, and both need exploring. Thus, another important step in quality control is to identify variables that have explanatory power over both experimental metadata variables and computed QC metrics. The plotQC function provides a novel approach to identifying variables that have substantial explanatory power for many genes. For each variable in the phenoData slot of the SCESet object, we fit a linear model for each feature with just that variable as the explanatory variable. We then plot the distribution of the marginal R^2 values across all features for the variables with the most explanatory power for the dataset (Figure 2e). The variables are ranked by median R^2 across features in the plot, providing useful information for identifying variables that may need to be considered during normalisation or statistical modelling. The plotQC function can also assess the influence of variables of interest by plotting principal components of the expression matrix most strongly correlated with a variable of interest against that variable. For example, in the Case Study data, the first principal component is correlated with the C1 machine used to process the cell (Figure 2f).

We also introduce the plotPhenoData function for convenient plotting of cell phenotype information (including QC metrics), and the plotFeatureData function for plotting feature information (see examples in the Supplementary Case Study). These methods will work not only on the SCESet class defined in *scater*, but also on any ExpressionSet object, providing sophisticated plotting functionality for many other Bioconductor packages and contexts.

The *scater* graphical user interface (GUI) provides convenient access to *scater*'s QC and visualisation methods (Supplementary Figures 3–5). It should prove particularly useful for the less programmatically inclined, as it opens an interactive interface in a web browser that facilitates exploration of the data through QC plots and other visualisations. The GUI allows users to easily examine the effects of changing multiple parameters, so even for experienced programmers it can be helpful as a fast way of conducting exploratory data analysis to guide data processing scripts for reproducible research.

In summary, *scater* provides a variety of novel and convenient methods to visualise an scRNA-seq dataset for QC. Low-quality cells and uninteresting genes can then be easily removed by filtering and subsetting the SCESet data structure prior to further analysis.

3.5 Data visualisation

Dimensionality reduction techniques are necessary to convert high-dimensional expression data into low-dimensional representations for intuitive visualisation of the relationships, similarities and differences between cells. To this end, *scater* provides convenient functions to apply a variety of dimensionality reduction procedures to the cells in an SCESet object. Functions include plotPCA, to perform a principal components analysis; plotTSNE, to perform t-distributed stochastic neighbour embedding (Van der Maaten and Hinton, 2008; Maaten, 2009; Van der Maaten and Hinton, 2012), which has been widely used for scRNA-seq data (Amir *et al.*, 2013; Bendall *et al.*, 2014; Macosko *et al.*, 2015); plotDiffusionMap, to generate a diffusion map (Haghverdi *et al.*, 2015) for visualising differentiation processes; and plotMDS, to generate multi-dimensional scaling plots (Figure 3a–c). The plotReducedDim function can also be used to plot any reduced-dimension representation of cells (e.g., an independent component analysis produced by *monocle* (Trapnell *et al.*, 2013) or similar) that is stored in an SCESet object.

By default, the PCA and t-SNE plots are produced using the features with the most variable expression across all cells, though this can be changed with function arguments. We focus on the most variable genes to highlight any heterogeneity in the data that might be driving interesting differences between cells. Alternatively, we can apply *a priori* knowledge to define a set of genes that are associated with a biological process of interest, and construct plots using only these features. For example, Scialdone *et al.*, 2015 found that using prior knowledge to define feature sets is vital for exploring processes like the cell cycle, which can have substantial effects on single-cell expression measurements (Buettner *et al.*, 2015). The subsetting and filtering methods for SCESet objects make it easy to construct reduced-dimension plots for particular gene sets, in order to investigate certain effects in the data, such as those due to the cell cycle (Figure 3d–f).

The various types of reduced-dimension plots can be used to identify potentially problematic cells to filter out of the dataset and to interpret cell population structure. Cell-level variables stored in the SCESet object can be used to define the shape, colour and size of points plotted, allowing more information to be conveniently incorporated into each plot (e.g., cells are coloured by *CCND2* expression in Figure 3d–f). The plotExpression function is also provided for plotting expression levels of a particular gene against any of the cell phenotype variables or the expression level of another feature (Figure 3g). This allows the user to inspect the expression levels of a feature or set of features in full detail, rather than relying only on summary information and reduced-dimension plots where information is necessarily lost.

3.6 Data normalisation and batch correction

Normalisation using a size factor for each sample to scale RNA-seq library sizes to make samples more comparable has proven useful in the analysis of bulk RNA-seq data. The methods TMM (Robinson and Oshlack, 2010), relative log-expression (Anders and Huber, 2010) and upper-quartile (Bullard *et al.*, 2010) are frequently used. Size-factor normalisation is supported in *scater*, with these three methods available, as well as tight integration with the *scran* package that implements a method utilising cell

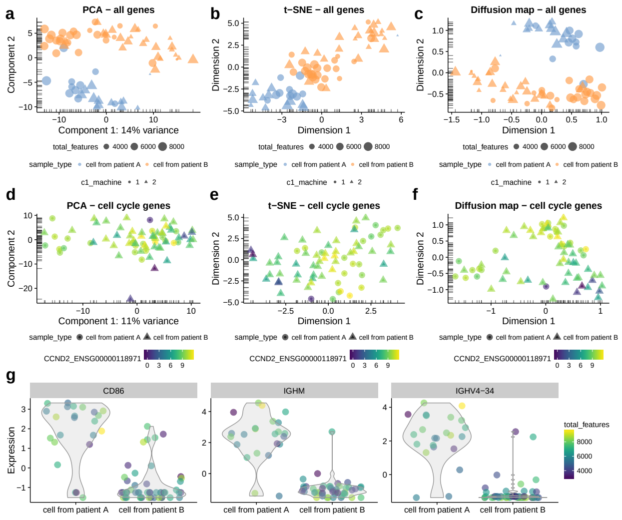

Fig. 3. Reduced dimension representations of cells and gene expression plots with scater. Plots are shown using all genes (a-c) and cell cycle genes only (d-f) using PCA (a,d), t-SNE (b,e) and diffusion maps (c,f). In the top row (a-c), points are coloured by patient of origin, sized by total features (number of genes with detectable expression) and the shape indicates the C1 machine used to process the cells. In the second row (d-f), points are coloured by the expression of CCND2 (ENSG00000118971), a gene associated with the G1/S phase transition of the cell cycle. With the plotExpression function, expression for sets of genes can be plotted against any cell metadata variables or the expression of a given gene (g). The function automatically detects whether the x-axis variable is categorical or continuous and plots the data accordingly, with x-axis values "jittered" to avoid excessive overplotting of points with the same x coordinate.

pooling and deconvolution to compute size factors better suited to scRNA-seq data (Lun *et al.*, 2016). Such normalisation is necessary, but further correction is typically required to ameliorate or remove batch effects. Here we present three possibilities, all easily implemented in a *scater* workflow. We emphasise that it is generally preferable to incorporate batch effects into statistical models used for inference. Where this is not possible, and for visualisations, approaches such as the following may be used.

In the case study dataset, cells from two patients were each processed on two C1 machines. Although C1 machine is not one of the most important explanatory variables on a per-gene level (Figure 2e), this factor is correlated with the first principal component of the log-expression data (Figure 2f). This effect cannot be removed by scaling normalisation methods, which target cell-specific biases and are not sufficient for removing large-scale batch effects that vary on a gene-by-gene basis (Figure 4a).

The C1 machine effect is known from the design of the experiment, so we can easily regress out this effect in *scater*. With the normaliseExprs function the user can supply a design matrix of variables to regress out of the expression values, and residuals from the linear model fit can be used as expression values for downstream analyses. For the dataset here, we fit a linear model to the *scran* normalised log-expression values with the C1 machine as an explanatory factor. (We also use the log-total counts from endogenous genes, percentage of counts from the top 100 most highly-expressed genes and percentage of counts from control genes as additional covariates.) We then use the residuals from the fitted model for further analyses (see Case Study in Supplementary Material). This approach successfully removes the C1 machine effect as a major source of variation between cells; the first principal component now separates the cells from the two patients, as expected (Figure 4b). This approach needs to be used carefully as single-cell data often deviate from normal

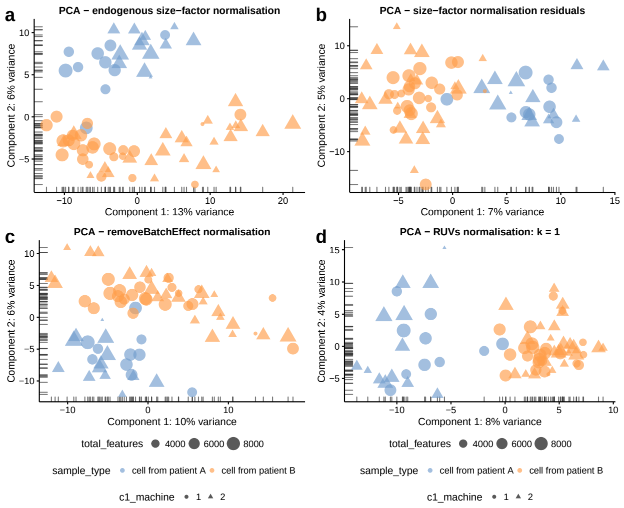

Fig. 4. Figure 5: Normalisation approaches made easy with scater. Principal component analysis plots showing cell structure in the first two PCA dimensions using various normalisation methods methods that can be easily applied in scater, including endogenous size-factor normalisation using methods from the scran package (a); expression residuals after applying size-factor normalisation and regressing out known, unwanted sources of variation (b); a "customised" expression conditional normalisation approach (c); and removal of one hidden factor identified using the RUVs method from the RUV package (d). In all plots, the colour of points is determined by the patient from which cells were obtained, shape is determined by the C1 machine used to process the cells and size reflects the total number of genes with detectable expression in the cell.

distributions, but in many cases, as here, it can successfully ameliorate large-scale known batch effects.

In addition to removing known batch effects, it can be important for large data sets to identify (potentially unknown) sources of unwanted variation (Leek *et al.*, 2010; Hicks *et al.*, 2015; Tung *et al.*, 2016; Bacher and Kendziorski, 2016; Grün and van Oudenaarden, 2015). *scater* is compatible with existing methods such as *svaseq* (Leek and Storey, 2007; Leek, 2014) and *RUVSeq* (Risso *et al.*, 2014) to identify and remove these unwanted sources of variation, and the removeBatchEffect method in the *limma* package (Ritchie *et al.*, 2015) to account for known batch effects. Applying removeBatchEffect from *limma* yields normalised data for which PCs are no longer correlated with C1 machine (see Case Study in Supplementary Material), but the patient effect is not the primary driver of variation amongst the PCs (Figure 4c). Here, just removing the first latent variable identified by the RUVs method from *RUVSeq* is sufficient to remove the machine effect, as the PCA plot now separates cells by patient rather than C1 machine (Figure 4d).

3.7 Software and data integration

As part of the R/Bioconductor ecosystem, *scater* can be easily integrated with other software for scRNA-seq data analysis (Supplementary Figure 2). Because the SCESet class builds on existing Bioconductor data structures, most Bioconductor packages for expression analyses are able to operate seamlessly with SCESet objects. Tools that can integrate easily with *scater* include many options for data normalisation (Lun *et al.*, 2016; Vallejos *et al.*, 2015; Ding *et al.*, 2015), differential expression analysis (Vallejos *et al.*, 2016; Trapnell *et al.*, 2014; Finak *et al.*, 2015; Vu *et al.*, 2016; Kharchenko *et al.*, 2014; Korthauer *et al.*, 2015; Andrews and Hemberg, 2016), heterogeneous gene expression analyses (Brennecke *et al.*, 2013; Kim *et al.*, 2015; Vallejos *et al.*, 2015), clustering (Kiselev *et al.*, 2016; Guo *et al.*, 2015; Fan *et al.*, 2016; Grün *et al.*, 2015), latent or hidden variable analysis (Leek and Storey, 2007; Leek, 2014; Risso *et al.*, 2014; Stegle *et al.*, 2012; Chikina *et al.*, 2015), cell cycle phase identification (Scialdone *et al.*, 2015) and pseudotime computation (Trapnell *et al.*, 2014; Angerer *et al.*, 2015; Juliá *et al.*, 2015; Campbell and Yau, 2016; Campbell *et al.*, 2015; Reid and Wernisch, 2015; Leng *et al.*, 2015; Haghverdi *et al.*, 2016). The *scater* package bridges the gap between raw reads and these downstream analysis tools by providing the pre-processing, QC, visualisation and normalisation methods and a data structure combining multiple data modalities and metadata necessary for convenient, robust and reproducible analyses of scRNA-seq data.

4 Discussion

Single-cell RNA sequencing is widely used for high-resolution gene expression studies investigating the behaviour of individual cells. While scRNA-seq data can provide substantial biological insights, the complexity and noise of the data is also much greater than that of conventional bulk RNA-seq. Thus, rigorous analysis of scRNA-seq data requires careful quality control to remove low-quality cells and genes, as well as normalisation to adjust for biases and batch effects in the expression data. Failure to carry out these procedures correctly is likely to compromise the validity of all downstream analyses (Leek *et al.*, 2010; Hicks *et al.*, 2015; Tung *et al.*, 2016; Bacher and Kendziorski, 2016; Grün and van Oudenaarden, 2015).

Here, we present an R/Bioconductor package, *scater*, that provides crucial infrastructure and methods for low-level scRNA-seq data analysis. The package introduces a data structure tailored to scRNA-seq data that is compatible with a vast number of existing tools in the Bioconductor project. The *scater* data structure combines multiple transformations of the expression data with cell and feature (gene or transcript) metadata and allows data sets to be easily standardised and shared. Wrapper functions for the popular RNA-seq quantification methods *kallisto* and *Salmon* facilitate the processing of raw read sequences to a SCESet object in R with expression data and accompanying metadata.

Quality control is a vital preliminary step for scRNA-seq and can be a time-consuming manual task. We present a tool for automated computation of QC metrics, novel plotting methods for QC and convenient subsetting and filtering methods to substantially simplify the process of filtering out unwanted or problematic cells and genes. The package provides a large array of sophisticated plotting functions so that cells can be visualised with a variety of popular dimensionality-reduction techniques in plots that incorporate cell metadata and expression values as plotting variables.

Normalisation remains a critical and continuously evolving aspect of pre-processing scRNA-seq data supported by *scater*. With judicious exploratory data analysis as part of the pre-processing, quality control and normalisation of scRNA-seq data using tools in *scater*, important known covariates can be identified. Simple size-factor normalisation methods, including the single-cell specific methods in the *scran* package, are seamlessly integrated into a *scater* workflow. Methods for correcting for batch effects and removing unwanted variation are supported both with internal methods and through harmonious integration with a multitude of tools available in the R/Bioconductor environment. Once identified, important covariates and latent variables can be flagged for inclusion in downstream statistical models or their effects regressed out of normalised expression values. The latter approach will often be necessary as many of the recently developed statistical methods for scRNA-seq data are not able to handle arbitrarily complex experimental designs.

The *scater* package is well supported and will continue to improve. Planned development includes further extensions to data structures that will enable tight integration of single-cell transcriptomic, genetic and epigenetic data, as well as further refinement of the methods available as the single-cell field matures. Although *scater* has been produced for scRNA-seq data, its capabilities are well suited for single-cell qPCR data and bulk RNA-seq data, and will prove useful for supporting analyses of these data types too.

5 Conclusion

The *scater* package eases the burden for a user tasked with producing a high-quality single-cell expression dataset for downstream analysis. The intuitive GUI implemented in *scater* provides an easier entry point into rigorous analysis of scRNA-seq data for users without a computational background, giving these users ability to go from raw reads to tidy data all in a single computing environment. Experienced users can take advantage of *scater*'s data structures, wide array of tools, suitability for scripted analyses and seamless integration with many other R/Bioconductor analysis tools. The data structures and methods in *scater* provide basic infrastructure upon which new scRNA-seq analysis tools can be developed. We anticipate that *scater* will be a useful resource for both analysts and software developers in the single-cell RNA sequencing field.

Acknowledgements

The authors would like to acknowledge Peter Donnelly and Oliver Stegle for support and helpful discussions.

ROME TEAM

Funding

DJM was supported by funding through an Early Career Fellowship from the National Health and Medical Research Council of Australia, and by core funding from EMBL. ATLL was supported by core funding from Cancer Research UK (award no. A17197). KRC is supported by a UK Medical Research Council studentship.

References

Allaire, J. J., Cheng, J., Xie, Y., McPherson, J., Chang, W., Allen, J., Wickham, H., Atkins, A., and Hyndman, R. (2016). rmarkdown: Dynamic Documents for R.

Amir, E.-A. D., Davis, K. L., Tadmor, M. D., Simonds, E. F., Levine, J. H., Bendall, S. C., Shenfeld, D. K., Krishnaswamy, S., Nolan, G. P., and Pe'er, D. (2013). viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology*, **31**(6), 545–552.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome biology*, **11**(10), R106.

Anders, S., Pyl, P. T., and Huber, W. (2015). HTSeq–a Python framework to work with high-throughput sequencing data. *Bioinformatics*, **31**(2), 166–169.

Andrews, T. S. and Hemberg, M. (2016). Modelling dropouts allows for unbiased identification of marker genes in scRNASeq experiments.

Angerer, P., Haghverdi, L., Büttner, M., Theis, F. J., Marr, C., and Buettner, F. (2015). destiny: diffusion maps for large-scale single-cell data in R. *Bioinformatics*.

Bache, S. M. and Wickham, H. (2014). Magrittr: A forward-pipe operator for R. *R package version*.

Bacher, R. and Kendziorski, C. (2016). Design and computational analysis of single-cell RNA-sequencing experiments. *Genome biology*, **17**(1), 1–14.

Bendall, S. C., Davis, K. L., Amir, E.-A. D., Tadmor, M. D., Simonds, E. F., Chen, T. J., Shenfeld, D. K., Nolan, G. P., and Pe'er, D. (2014). Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell*, **157**(3), 714–725.

Bengtsson, H. (2016). matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors).

Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*.

Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A. A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S. A., Marioni, J. C., and Heisler, M. G. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature methods*, **10**(11), 1093–1095.

Buettner, F., Natarajan, K. N., Paolo Casale, F., Proserpio, V., Scialdone, A., Theis, F. J., Teichmann, S. A., Marioni, J. C., and Stegle, O. (2015). Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, **33**(2), 155–160.

Bullard, J. H., Purdom, E. A., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC bioinformatics*, **Paper 247**, 1–62.

Campbell, K. and Yau, C. (2016). Ouija: Incorporating prior knowledge in single-cell trajectory learning using Bayesian nonlinear factor analysis.

Campbell, K., Ponting, C. P., and Webber, C. (2015). Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell RNA-seq profiles.

Chang, W. (2015). shinydashboard: Create Dashboards with 'Shiny'.

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., and McPherson, J. (2016). shiny: Web Application Framework for R.

Chikina, M., Zaslavsky, E., and Sealfon, S. C. (2015). CellCODE: a robust latent variable approach to differential expression analysis for heterogeneous cell populations. *Bioinformatics*, **31**(10), 1584–1591.

Couture-Beil, A. (2014). rjson: JSON for R.

Delmans, M. and Hemberg, M. (2016). Discrete distributional differential expression (D(3)E) - a tool for gene expression analysis of single-cell RNA-seq data. *BMC bioinformatics*, **17**(1), 110.

Ding, B., Zheng, L., Zhu, Y., Li, N., Jia, H., Ai, R., Wildberg, A., and Wang, W. (2015). Normalization and noise reduction for single cell RNA-seq experiments. *Bioinformatics*, **31**(13), 2225–2227.

Dowle, M., Srinivasan, A., Short, T., and Lianoglou, S. (2015). data.table: Extension of Data.frame.

Durinck, S., Moreau, Y., Kasprzyk, A., Davis, S., De Moor, B., Brazma, A., and Huber, W. (2005). BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, **21**(16), 3439–3440.

Fan, J., Salathia, N., Liu, R., Kaeser, G. E., Yung, Y. C., Herman, J. L., Kaper, F., Fan, J.-B., Zhang, K., Chun, J., and Kharchenko, P. V. (2016). Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nature methods*, **13**(3), 241–244.

Filzmoser, P. and Gschwandtner, M. (2015). mvoutlier: Multivariate outlier detection based on robust methods.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., Linsley, P. S., and Gottardo, R. (2015). MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome biology*, **16**, 278.

Fischer, B. and Pau, G. (2016). rhdf5: HDF5 interface to R.

Garnier, S. (2016). viridis: Default Color Maps from 'matplotlib'.

Grün, D. and van Oudenaarden, A. (2015). Design and Analysis of Single-Cell Sequencing Experiments. *Cell*, **163**(4), 799–810.

Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., and van Oudenaarden, A. (2015). Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, **525**(7568), 251–255.

Guo, M., Wang, H., Steven Potter, S., Whitsett, J. A., and Xu, Y. (2015). SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. *PLoS computational biology*, **11**(11), e1004575.

Haghverdi, L., Buettner, F., and Theis, F. J. (2015). Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*.

Haghverdi, L., Büttner, M., Alexander Wolf, F., Buettner, F., and Theis, F. J. (2016). Diffusion pseudotime robustly reconstructs lineage branching. *bioRxiv*.

Hebenstreit, D. and Teichmann, S. A. (2011). Analysis and simulation of gene expression profiles in pure and mixed cell populations. *Physical biology*, **8**(3), 035013.

Hicks, S. C., Teng, M., and Irizarry, R. A. (2015). On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data.

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Oleś, A. K., Pagès, H., Reyes, A., Shannon, P., Smyth, G. K., Tenenbaum, D., Waldron, L., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature methods*, **12**(2), 115–121.

Ilicic, T., Kim, J. K., Kolodziejczyk, A. A., Bagger, F. O., McCarthy, D. J., Marioni, J. C., and Teichmann, S. A. (2016). Classification of low quality cells from single-cell RNA-seq data. *Genome biology*, **17**(1), 29.

Juliá, M., Telenti, A., and Rausell, A. (2015). Sincell: an R/Bioconductor package for statistical assessment of cell-state hierarchies from single-cell RNA-seq. *Bioinformatics*, **31**(20), 3380–3382.

Kanitz, A., Gypas, F., Gruber, A. J., Gruber, A. R., Martin, G., and Zavolan, M. (2015). Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data. *Genome biology*, **16**(1), 150.

Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods*, **11**(7), 740–742.

Kim, J. K., Kolodziejczyk, A. A., Illicic, T., Teichmann, S. A., and Marioni, J. C. (2015). Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature communications*, **6**.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., and Hemberg, M. (2016). SC3 - consensus clustering of single-cell RNA-Seq data.

Korthauer, K. D., Chu, L.-F., Newton, M. A., Li, Y., Thomson, J., Stewart, R., and Kendziorski, C. (2015). scDD: A statistical approach for identifying differential distributions in single-cell RNA-seq experiments.

Krijthe, J. (2015). Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation. *0.10, URL http://CRAN. R-project. org/package= Rtsne*.

Leek, J. T. (2014). svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic acids research*, **42**(21), 0.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, **3**(9), 1724–1735.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Publishing Group*, **11**(10), 733–739.

Leng, N., Chu, L.-F., Barry, C., Li, Y., Choi, J., Li, X., Jiang, P., Stewart, R. M., Thomson, J. A., and Kendziorski, C. (2015). Oscope identifies oscillatory genes in unsynchronized single-cell RNA-seq experiments. *Nature methods*, **12**(10), 947–950.

Liao, Y., Smyth, G. K., and Shi, W. (2014). featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, **30**(7), 923–930.

Lun, A. T. L., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology*, **17**(1), 75.

Maaten, L. (2009). Learning a parametric embedding by preserving local structure. In *International Conference on Artificial Intelligence and Statistics*, pages 384–391. machinelearning.wustl.edu.

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., and McCarroll, S. A. (2015). Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, **161**(5), 1202–1214.

Patro, R., Duggal, G., and Kingsford, C. (2015). Salmon: Accurate, Versatile and Ultrafast Quantification from RNA-seq Data using Lightweight-Alignment. *bioRxiv*.

Reid, J. E. and Wernisch, L. (2015). Pseudotime estimation: deconfounding single cell time series. *bioRxiv*.

Risso, D., Ngai, J., Speed, T. P., and Dudoit, S. (2014). Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology*, **32**(9), 896–902.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research*, **43**(7), e47.

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome biology*, **11**(3), R25.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–140.

Scialdone, A., Natarajan, K. N., Saraiva, L. R., Proserpio, V., Teichmann, S. A., Stegle, O., Marioni, J. C., and Buettner, F. (2015). Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods*.

Scialdone, A., Tanaka, Y., Jawaid, W., Moignard, V., Wilson, N. K., Macaulay, I. C., Marioni, J. C., and Göttgens, B. (2016). Resolving early mesoderm diversification through single-cell expression profiling. *Nature*.

Shalek, A. K., Satija, R., Adiconis, X., Gertner, R. S., Gaublomme, J. T., Raychowdhury, R., Schwartz, S., Yosef, N., Malboeuf, C., Lu, D., Trombetta, J. J., Gennert, D., Gnirke, A., Goren, A., Hacohen, N., Levin, J. Z., Park, H., and Regev, A. (2013). Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, **498**(7453), 236–240.

Soneson, C., Love, M. I., and Robinson, M. D. (2015). Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, **4**, 1521.

Stegle, O., Parts, L., Piipari, M., Winn, J., and Durbin, R. (2012). Using probabilistic estimation of expression residuals (PEER) to obtain increased power and interpretability of gene expression analyses. *Nature protocols*, **7**(3), 500–507.

Teng, M., Love, M. I., Davis, C. A., Djebali, S., Dobin, A., Graveley, B. R., Li, S., Mason, C. E., Olson, S., Pervouchine, D., Sloan, C. A., Wei, X., Zhan, L., and Irizarry, R. A. (2016). A benchmark for RNA-seq quantification pipelines. *Genome biology*, **17**(1), 74.

Trapnell, C., Hendrickson, D. G., Sauvageau, M., Goff, L., Rinn, J. L., and Pachter, L. (2013). Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature biotechnology*, **31**(1), 46–53.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, **32**(4), 381–386.

Tung, P.-Y., Blischak, J. D., Hsiao, C., Knowles, D. A., Burnett, J. E., Pritchard, J. K., and Gilad, Y. (2016). Batch effects and the effective design of single-cell gene expression studies.

Vallejos, C. A., Marioni, J. C., and Richardson, S. (2015). BASiCS: Bayesian Analysis of Single-Cell Sequencing Data. *PLoS computational biology*, **11**(6), e1004333.

Vallejos, C. A., Richardson, S., and Marioni, J. C. (2016). Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome biology*, **17**(1), 70.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research: JMLR*, **9**(2579-2605), 85.

Van der Maaten, L. and Hinton, G. (2012). Visualizing non-metric similarities in multiple maps. *Machine learning*, **87**(1), 33–55.

Vu, T. N., Wills, Q. F., Kalari, K. R., Niu, N., Wang, L., Rantalainen, M., and Pawitan, Y. (2016). Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics*.

Wickham, H. (2011). testthat: Get started with testing. *The R journal*, **3**(1), 5–10.

Wickham, H. (2012). reshape2: Flexibly reshape data: a reboot of the reshape package. *R package version*.

Wickham, H. (2015). plyr: Tools for splitting, applying and combining data. R package version 1.8. 1. *R Found. Stat. Comput., Vienna*.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.

Wickham, H. and Francois, R. (2015). dplyr: A grammar of data manipulation. *R package version 0. 4*, **1**, 20.

Wickham, H., Danenberg, P., and Eugster, M. (2015). roxygen2: In-Source Documentation for R.

Wilke, C. O. (2016). cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'.

Xie, Y. (2013). *Dynamic Documents with R and knitr*, volume 29. CRC Press.

Yates, A., Akanni, W., Amode, M. R., Barrell, D., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Fitzgerald, S., Gil, L., Girón, C. G., Gordon, L., Hourlier, T., Hunt, S. E., Janacek, S. H., Johnson, N., Juettemann, T., Keenan, S., Lavidas, I., Martin, F. J., Maurel, T., McLaren, W., Murphy, D. N., Nag, R., Nuhn, M., Parker, A., Patricio, M., Pignatelli, M., Rahtz, M., Riat, H. S., Sheppard, D., Taylor, K., Thormann, A., Vullo, A., Wilder, S. P., Zadissa, A., Birney, E., Harrow, J., Muffato, M., Perry, E., Ruffier, M., Spudich, G., Trevanion, S. J., Cunningham, F., Aken, B. L., Zerbino, D. R., and Flicek, P. (2016). Ensembl 2016. *Nucleic acids research*, **44**(D1), D710–6.

Zeisel, A., Muñoz Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., Rolny, C., Castelo-Branco, G., Hjerling-Leffler, J., and Linnarsson, S. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**(6226), 1138–1142.