Orchestrating Single-Cell Analysis with Bioconductor

Robert A. Amezquita[1], Vince J. Carey[*2], Lindsay N. Carpp[*1], Ludwig Geistlinger[*3,4], Aaron T. L. Lun[*5], Federico Marini[*6,7], Kevin Rue-Albrecht[*8], Davide Risso[*9,10], Charlotte Soneson[*11,12], Levi Waldron[*3,4], Hervé Pagès[1], Mike Smith[13], Wolfgang Huber[13], Martin Morgan[14], Raphael Gottardo[†1], and Stephanie C. Hicks[†15]

[1]Fred Hutchinson Cancer Research Center, Seattle, WA, USA
[2]Channing Division of Network Medicine, Brigham And Women's Hospital, MA, USA
[3]Graduate School of Public Health and Health Policy, City University of New York, NY, USA
[4]Institute for Implementation Science in Population Health, City University of New York, NY, USA
[5]Cancer Research UK Cambridge Institute, University of Cambridge, Cambridge CB2 0RE, UK
[6]Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), Mainz, Germany
[7]Center for Thrombosis and Hemostasis, Mainz, Germany
[8]Kennedy Institute of Rheumatology, University of Oxford, Oxford, OX3 7FY, UK
[9]Department of Statistical Sciences, University of Padua, Italy
[10]Division of Biostatistics and Epidemiology, Department of Healthcare Policy and Research, Weill Cornell Medicine, New York, NY, USA
[11]Friedrich Miescher Institute for Biomedical Research, Basel, Switzerland
[12]SIB Swiss Institute of Bioinformatics, Basel, Switzerland
[13]European Molecular Biology Laboratory, Genome Biology Unit, Heidelberg, Germany
[14]Biostatistics and Bioinformatics, Roswell Park Comprehensive Cancer Center, Buffalo, NY, USA
[15]Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, MD, USA

March 26, 2019

Abstract

Recent developments in experimental technologies such as single-cell RNA sequencing have enabled the profiling a high-dimensional number of genome-wide features in individual cells, inspiring the formation of large-scale data generation projects quantifying unprecedented levels of biological variation at the single-cell level. The data generated in such projects exhibits unique characteristics, including increased sparsity and scale, in terms of both the number of features and the number of samples. Due to these unique characteristics, specialized statistical methods are required along with fast and efficient software implementations in order to successfully derive biological insights. Bioconductor - an open-source, open-development software project based on the R programming language - has pioneered the analysis of such high-throughput, high-dimensional biological data, leveraging a rich history of software and methods development that has spanned the era of sequencing. Featuring state-of-the-art computational methods, standardized data infrastructure, and interactive data visualization tools that are all easily accessible as software packages, Bioconductor has made it possible for a diverse audience to analyze data derived from cutting-edge single-cell assays. Here, we present an overview of single-cell RNA sequencing analysis for prospective users and contributors, highlighting the contributions towards this effort made by Bioconductor.
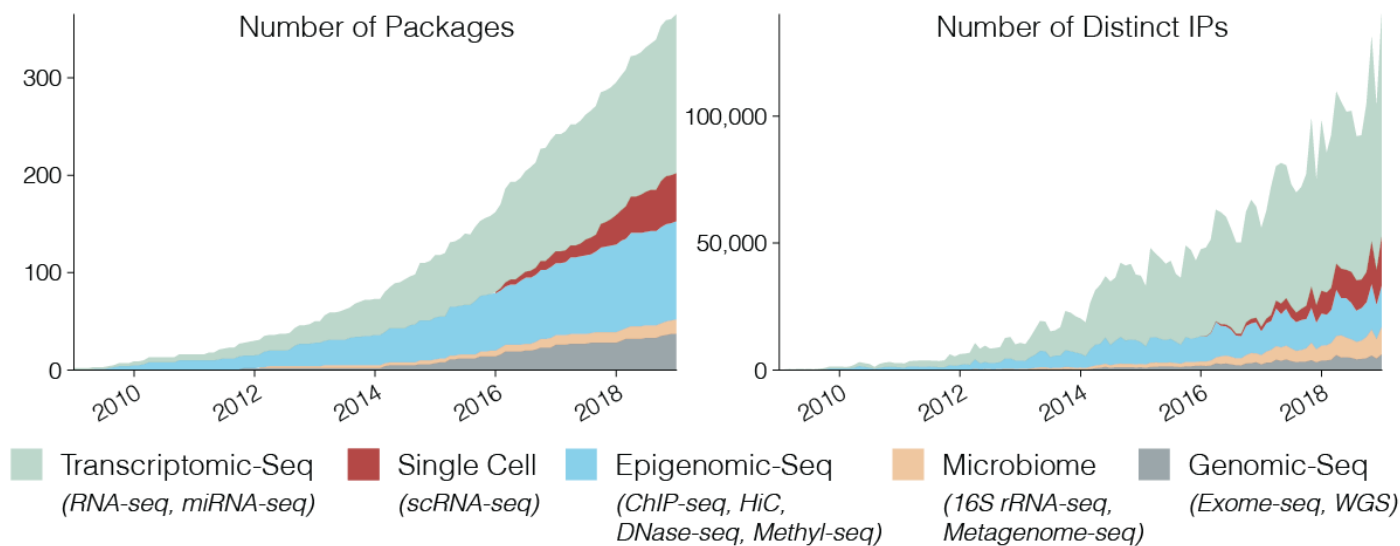
Figure 1: **10 years of Bioconductor in the high-throughput sequencing era.** Bioconductor software packages associated with the analysis of sequencing technology were tracked by the total number of packages (left) and the number of distinct IPs (data recorded monthly) visiting their online documentation (right) over the course of ten years. Software packages were uniquely defined by their primary sequencing technology association, with examples of specific terms used for annotation below in parentheses.

Introduction

Bioconductor [1] is a central repository for open-source software packages based in the R programming language [2] for the analysis and comprehension of high-throughput biological data. Since 2001, it has drawn together a rich community of developers and users from diverse scientific fields including genomics, proteomics, metabolomics, flow cytometry, quantitative imaging, chemoinformatics, and other high-throughput data.

Bioconductor supports the analysis of traditional bulk DNA, RNA, and epigenomic profiling assays [3–10] (**Figure 1**). Such bulk profiling technologies have yielded a wealth of important scientific insights (reviewed in e.g. [11–13]). However, a number of critical biological questions can be only answered at the single-cell level. Characterizing the extent of genetic heterogeneity within a tumor, identifying and characterizing rare cell populations with differential features, and defining the mechanisms of cell lineage differentiation, are all examples of biological questions that are intractable without single-cell approaches. Furthermore, revisiting questions that were previously tackled with bulk approaches can potentially provide new perspectives. Biotechnology is an ever-evolving field, with an ever-changing vocabulary (see **Box 1** for our definitions used throughout), and new single-cell experimental protocols and technologies such as single-cell RNA-sequencing (scRNA-seq) have emerged that can help tackle these unresolved questions.

In addition, Bioconductor has been a pioneer for analyzing high-throughput data from single-cell cytometry assays that can be used to obtain samples from hundreds of thousands to millions of cells [14, 15], but are comparatively low-dimensional, capturing around 20 to 50 features per cell. In contrast, the more recently developed single-cell genomics technologies are both high-throughput and high-dimensional, capturing thousands of traits within a single cell in an unbiased manner. To successfully derive biological insights from such work, two unique characteristics of the resulting data sets must be successfully dealt with. One is the increased scale on which samples, or cells, are measured, i.e. thousands to millions of cells within each data set, for example in the compendiums from the Human Cell Atlas [16]. In contrast, data derived from bulk assays, such as RNA-sequencing (RNA-seq) and chromatin immunoprecipitation sequencing (ChIP-seq), typically have samples on the order of tens to hundreds. A second distinctive feature of single-cell assay data is their increased

---

*Co-second authors. These authors (VJC, LNC, LG, ATLL, FM, KR, DR, CS, LW) contributed equally and are listed alphabetically.

†Co-senior authors. These authors (RG, SCH) contributed equally.

sparsity, due to biological fluctuations in the measured traits or limited sensitivity in quantifying small numbers of molecules [17–19]. In addition, data derived from single-cell assays have revealed more heterogeneity than previously seen [20–27]. This has led to the rapid development of statistical methods to address the increased sparsity and heterogeneity seen in this data [28–31]. The profound increase in the complexity of data measured at the single-cell level, along with the continued increases in the number of samples measured, have precipitated the need for fundamental changes in data access, management, and infrastructure to make data analyses scalable to empower scientific progress. Specifically, specialized statistical methods along with fast and memory-efficient software implementation are required to reap the full scientific potential of high-throughput, high-dimensional data from single-cell assays.

Bioconductor has developed state-of-the-art and widely used software packages (**Table S1**) for the analysis of high-dimensional bulk assays, such as RNA-sequencing (RNA-seq) and high-throughput, low-dimensional single-cell assays, such as flow cytometry and mass cytometry (CyTOF) data. Therefore, Bioconductor is a natural home for software development for data derived from high-throughput, high-dimensional single-cell technologies. In particular, Bioconductor has developed state-of-the-art software packages (**Table S1**) and workflows (**Table 1**) to analyze data from such technologies (**Figure 2**). To help users get started with Bioconductor, we describe some of these packages and present a series of workflows (**Box 2**) to demonstrate how to leverage the robust and scalable Bioconductor framework to analyze data derived from single-cell assays. We primarily focus on the analysis of scRNA-seq data, although many of the packages mentioned herein are also generalizable to other types of single-cell assays.

Broadly, these new Bioconductor contributions provide profound changes in how users access, store, and analyze data, including: (1) memory-efficient data import and representation, (2) common data containers for storing data from single-cell assays for interoperability between packages, (3) fast and robust methods for transforming raw single-cell data into processed data ready for downstream analyses, (4) interactive data visualization, and (5) downstream analyses, annotation and biological interpretation. As a companion to this manuscript, we also provide an online book that provides extensive resources for running R and Bioconductor, and furthermore demonstrates select workflows corresponding to topics covered in this manuscript. The book can be found at: `https://osca.bioconductor.org`

Box 2. Key Definitions

Sample: a single biological unit that is assayed.

Feature: a trait of a sample that is measured. Examples include mRNAs in RNA-seq experiments, genomic loci for ChIP-seq experiments, and cell markers in flow/CyTOF experiments.

Experiment: a procedure where a set of features are measured for each sample; in this usage, typically involves multiple samples, possibly with varying conditions (e.g. treatments, time points).

High-throughput assay: an assay that captures and measures features from many samples. Examples include flow cytometry, CyTOF, and certain scRNA-seq platforms, which can quantify tens or hundreds of thousands to millions of cells. For this reason, in our review, most bulk assays are not considered high-throughput as they profile a limited number of samples.

High-dimensional assay: an assay that captures thousands or tens of thousands of features per single sample unit. In our review, high throughput assays such as flow cytometry are not considered high-dimensional as they profile a limited number of proteins.

Bulk assay: an assay that measures pools of cells to produce a set of measured features as a single observation unit per pool.

Single-cell assay: a technology where a single sample corresponds to a single cell; includes flow cytometry, CyTOF, and single-cell RNA-seq (scRNA-seq) across various platform technologies (plate-based, droplet, etc.).

3

Box 2. Getting Started

To accompany this manuscript, we have written a book that is freely accessible online and discusses in detail how to get started with using R and Bioconductor. The book covers installation, learning how to get help, and workflows covering specific case studies to illustrate the usage of R and Bioconductor based workflows. See the book at: `https://osca.bioconductor.org`

Preprocessing sequencing data

The analysis of sequencing-based assays often begins with quantification of measured traits from raw sequencing reads. For high-throughput scRNA-seq assays, sequenced reads are typically aligned to the transcriptome and quantified to create a matrix of expression values for each cell across the features of interest (i.e. genes or transcripts) for further analysis. While much of the specific choices defining the workflow that generates processed data from raw data are often technology- or platform-dependent, we will briefly touch on the topic of alignment. A pipeline to process scRNA-seq data based on the R/Bioconductor software project is now available through the *scPipe* package [32], which uses the *Rsubread* Bioconductor package [33, 34] to provide alignments within R. For droplet-based scRNA-seq technologies, such as 10X Genomics [35], the *DropletUtils* Bioconductor package can read in a matrix of UMI counts, which were produced from, for example, the *Cell Ranger* [35] 10X Genomics pipeline, where each row of the matrix corresponds to a gene, and each column corresponds to a cell barcode.

Outside of R, recent developments in alignment methods have produced a class of pseudoaligners capable of running on personal computers, such as the Salmon [36] and Kallisto [37] utilities. The *tximport* [38] Bioconductor package imports the results from these pseudoaligners as matrices into an R session. In combination with the *tximeta* [39] package, an instance of a well-supported Bioconductor class can be created. Common Bioconductor methods and classes are the foundation of the single-cell R/Bioconductor analysis software discussed herein. The specific infrastructure used to represent single-cell data is discussed in more detail in the next section.

Data Infrastructure

A key focus and advantage of embracing Bioconductor workflows is the use of common data infrastructures. First and foremost, the implementation of the data containers is done with the aim of enabling modularity between packages and interoperability across packages. Hence, the containers are designed to support diverse workflows, while still being accessible to end-users. To this end, Bioconductor makes use of the S4 object-oriented programming style that allows for the creation of classes that standardize how data is stored and accessed. Furthermore, a single object or data container can contain a rich set of metadata annotation, as well as various forms of primary data essential for description, analysis, and portability. Bioconductor has recently focused its efforts on the creation of the *SingleCellExperiment* class to support single-cell data platforms, which is described in depth below.

The *SingleCellExperiment* container

The *SingleCellExperiment* class is a lightweight data container for storing data from single-cell assays or experiments [40] that is sufficiently flexible to work with a diverse set of packages. Because the *SingleCellExperiment* class was developed as an extension of the *SummarizedExperiment* [41] class, the *SingleCellExperiment* class contains all the advantages, structure, and specially engineered features to accommodate large-scale data, with a number of additions that provide convenient methods and structures that are common to many single-cell analyses. Specifically, the inheritance of *SingleCellExperiment* class from the *SummarizedExperiment* class [41] has enabled the immediate use of previously developed methods, many of which are discussed by Huber *et al.* 2015 in a previous review of Bioconductor tools [1]).
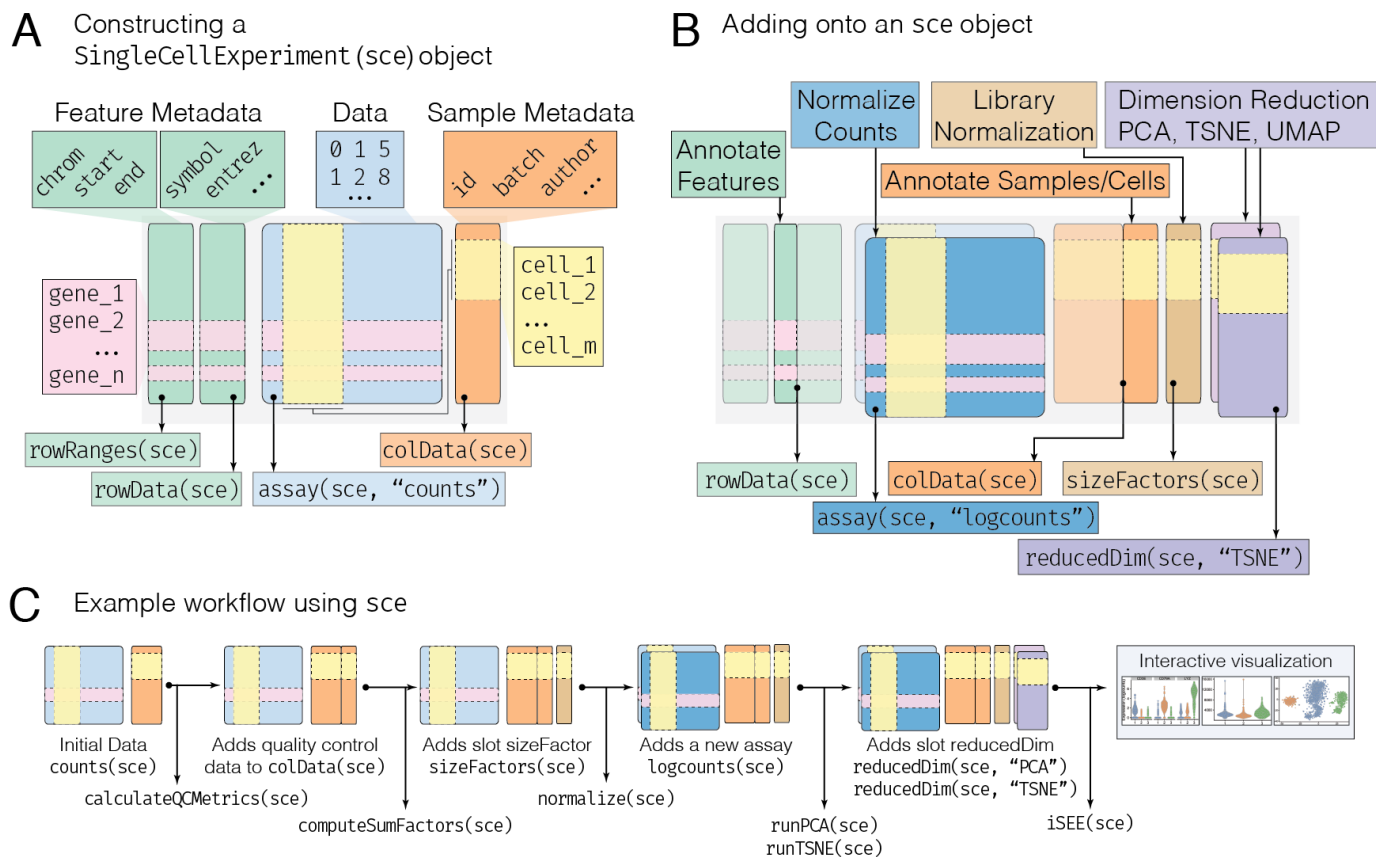
Figure 2: **Overview of the *SingleCellExperiment* class and workflow**. The *SingleCellExperiment* (`sce`) class and data container from the *SingleCellExperiment* package [40] stores multifaceted large-scale data and enables interoperability across Bioconductor packages. References to R functions and objects are shown in `monospace` (code) font. **(A)** A minimal `sce` object is constructed by supplying data such as a matrix of counts per cell as an assay (blue box), consisting of features, such as genes (rows), and cells (columns). Metadata describing the cells may also be supplied, wherein the cells are represented as rows and known characteristics of the cells are columns (orange box). Similarly, metadata describing the features may be added as well (green box). Each of these different types of data are stored in distinct parts of the `sce` object, which are referred to as slots. The data within each slot may be accessed programmatically via accessors named after their respective slot (arrows), such that `rowRanges` refers to feature metadata, `colData` refers to sample metadata, and `assay` refers to data. (**B**) Analysis using *SingleCellExperiment* (`sce`) compatible workflows appends data to the initial `sce` object. For example, calculating library normalization factors per cell creates a new slot (pink box). These can then used to derive a normalized count matrix, which is stored in the same `assay` slot alongside the initial counts data (dark blue box). The `assay` slot is thus capable of storing any number of data transformations. Cell quality metrics, which describe cell characteristics, are appended to the sample metadata slot `colData`. Finally, in a similar fashion to the `assay` slot, any number of dimensionally reduced representations of the data can be stored, residing in their own slot, `reducedDim` (purple box). **(C)** The `sce` object evolves throughout the course of a typical analysis, storing various metrics and representations derived from the initial data. For more information on the *SingleCellExperiment* class, see the *SingleCellExperiment* vignette (`https://bioconductor.org/packages/SingleCellExperiment`).

Made up of multiple compartments of information called *slots*, the *SingleCellExperiment* object holds various data representations (**Figure 3**). Primary data, such as count matrices representing sequencing read or unique molecular identifier (UMI) counts, are stored in the `assays` slot as one or more matrices (including sparse or dense Matrix [42] objects), where rows represent features (e.g. genes, transcripts, genomic regions) and columns represent samples, or in the case of single-cell experiments, cells. Furthermore, each row and column can be
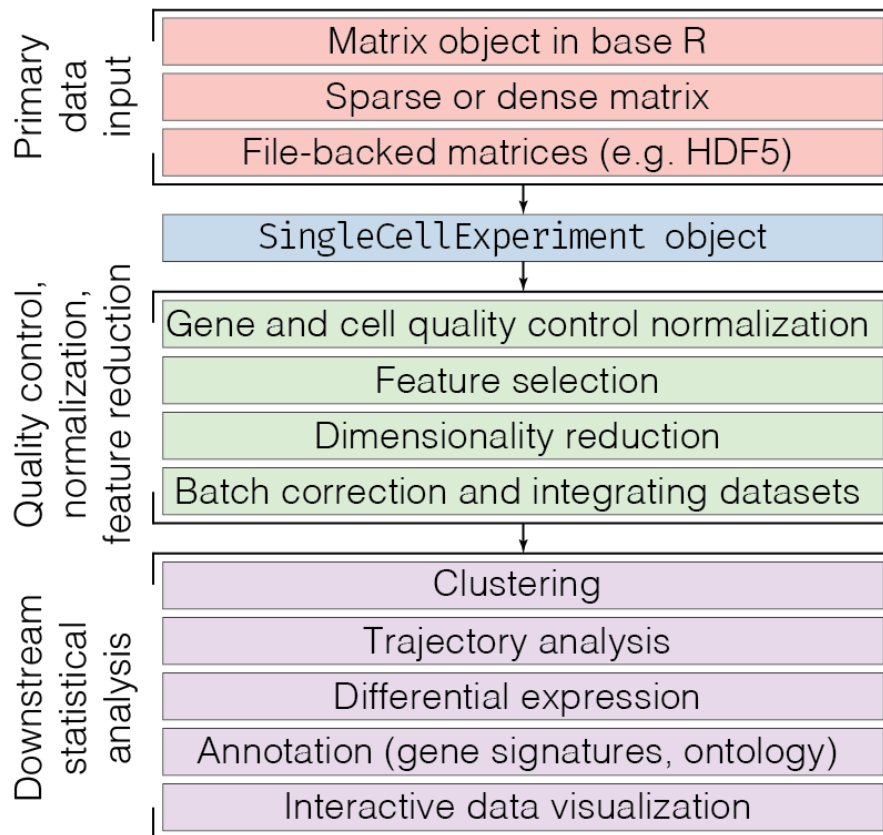
Figure 3: **Bioconductor workflow for analyzing single-cell data**. Primary data is collected and coalesced into a matrix representation, which can exist in various forms (top, red box). This data is then used to create a *SingleCellExperiment* object, which contains both primary data and metadata essential to understanding the experimental setup (blue box). The *SingleCellExperiment* object is then transformed through preprocessing workflows that ultimately produce a cleaned expression matrix (green box). This result can then be used in various downstream statistical analyses (purple box, bottom). This paper is structured to roughly follow this analytical workflow from top to bottom.

annotated with a rich set of metadata. For example, row metadata is stored in the `rowData` slot and could include Entrez Gene IDs [43] and GC content. Further, for rows corresponding to genomic features, a special `rowRanges` slot can be created to hold genome coordinates. Column metadata is stored in the `colData` slot and can contain information about sample-level characteristics. Additional column metadata can be added to this slot as well - for example, summary quality control statistics such as the total number of reads per cell. This column metadata is particularly useful for subsetting the data, such as removing cells with a low number of reads. Separately, the `sizeFactors` slot also refers to columns (cells) and contains information necessary for normalization. A recent innovation that is specifically designed for single-cell data is the *reducedDims* slot that contains low-dimensional representations of data, such as principal components analysis (PCA), *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) [44], or Uniform Manifold Approximation and Projection (UMAP) [45].

Lastly, throughout the various slots of the *SingleCellExperiment* class, disk-backed representations of the data (e.g. HDF5) are supported to enable analyses that would otherwise be impossible to perform due to memory constraints. Recent software innovations - both in data handling and processing - are required to make full use of this capability, and are demonstrated in our online supplement (**Table 1**).

Altogether, these different types of primary data and metadata slots reside in a single data container, allowing for a portable, full representation and annotation of the data alongside continual validity checks to prevent

malformed data input. This has made the *SingleCellExperiment* data container the foundation of many packages oriented toward single-cell analysis available today in Bioconductor, providing seamless interoperability and facilitating the development and usage of cutting-edge computational methods, from initialization of data containers to the end stages of analysis.

In addition, recent advances in technology and protocols allow the simultaneous measurement of genetic, epigenetic, and transcriptomic information from the same cells [46–52]. The *MultiAssayExperiment* [53] package integrates heterogeneous data types that may be individually represented by *SingleCellExperiment*, *DelayedArray*, or other standard R/Bioconductor data structures.

Quality Control and Normalization

After quantifying a measured trait from a single-cell assay and creating a *SingleCellExperiment* object, one of the first steps in analyzing the data is to identify, remove, and correct for low quality features or samples. In the analysis of scRNA-seq data, this typically translates to transforming a raw count matrix to a processed expression matrix ready for downstream analyses, such as clustering or differential expression. This process includes filtering out low-quality cells, selecting for informative features, applying cell and gene-specific normalization methods to remove cell and gene-specific biases, and adjusting for known covariates as well as latent sources of variation. In this section, we discuss these steps in greater detail, as well as methods for integrating data from multiple single-cell experiments.

Cell and Gene Quality Control

For droplet-based scRNA-seq technologies such as 10X Genomics [35], the *DropletUtils* package can be used to perform key quality control tasks such as distinguishing empty droplets from cells [54] and reducing the effects of barcode swapping [55]. The *scater* [56] package automates the calculation of a number of key quality control metrics. Amongst these, the library size (the total number of read or UMI counts across all genes in a given cell), the proportion of counts assigned to spike-ins [57] or mitocondrial genes, and the number of genes detected are commonly used to remove low quality cells. Furthermore, genes can be annotated with their abundance measurements, such as average expression, or frequency of detection, enabling the removal of uninformative genes [58]. The *simpleSingleCell* [59] Bioconductor workflow demonstrates how to use these packages to apply such quality control metrics for both read counts and UMI counts scRNA-seq data.

Gene Expression Normalization

Though many normalization procedures exist for data derived from traditional bulk assays [3, 7, 9, 60–62], using data derived from single-cell assays, such as scRNA-seq, introduces new challenges that require cell and gene-specific biases to be eliminated prior to downstream analyses that depend on explicit gene expression value comparisons, such as differential expression analyses [19]. These biases can be reduced through the use of cell- (and possibly gene-) specific scaling factors, also known as size factors, which are used to make cells with different properties comparable. The family of approaches discussed herein explicitly address technical, experimental, and biological factors to make cells within a single scRNA-seq experiment comparable through the calculation of a corrected or "clean" gene expression matrix.

The *SCnorm* [63], *scran* [29] and *scater* [56] packages provide standalone normalization methods whose results can then be used in any downstream analysis. The *SCnorm* package estimates and removes gene-specific variability due to sequencing read depth, along with accounting for other feature-level biases such as GC content and gene-length effects. The *scran* package calculates cell-specific size factors used for normalization in a manner that accounts for population heterogeneity by first clustering groups of cells with similar expression profiles. Finally, *scater* can calculate size factors from the library sizes. However, these approaches can only account for intrinsic factors that can be deduced directly from the data, and thus cannot correct for experimental factors such as batch effects.

Alternatively, there are approaches that propose statistical models to address not only normalization, but also other analyses such as dimensionality redution or differential expression. The Bioconductor packages

BASiCS [64, 65], *zinbwave* [31], and *MAST* [28] are not specific for normalization, but each provide unique statistical frameworks tailored for scRNA-seq data. Such frameworks have the capacity to adjust not only for intrinsic technical factors, but also for known artifacts such as time, treatment, and batch effects. Furthermore, these technical or experimental factors are typically unwanted variation, however at times, they may be potentially confounded with interesting biological factors or variation [19].

To compare the effects of different normalization strategies, the *scone* package [66] can be used to explore the results of various strategies and parameterizations. To learn more about scRNA-seq normalization methods, including a thorough discussion on the use of synthetic control genes such as ERCC spike-ins [57], we refer the interested reader to reviews by McDavid et al. [67] and Vallejos et al. [68].

Adjusting for Cell-Cycle Heterogeneity

Unless a cell cycle synchronization step is performed, the individual cells in a population will be at different stages of the cell cycle when the traits of interest are analyzed. Thus, a special type of normalization that is unique to single-cell data is adjusting for the biological effect of cell cycle phase on gene expression profiles [69]. The biological heterogeneity induced from differences in cell-cycle phases across cells is often considered a source of unwanted, technical variation, as they can obscure primary biological effects or conversely be highly correlated with biological effects [70]. For this reason, computational methods may be needed to deconvolute cell-cycle heterogeneity. For instance, latent variable models have been proposed to remove cell-cycle variability [70]. Within the Bioconductor framework, the *scran* [59] package implements a trained cell-cycle classifier using known cell cycle genes in humans and in mice to assign G1, G2/M, and S scores to cells. These scores can then be used as covariates to regress out the effect of cell cycle. In addition, the *Oscope* [71] package can be used to identify oscillators, or oscillating genes, when a single cell's mRNA expression is oscillating through cell states.

Integrating Datasets

As scRNA-seq approaches continue to gain in popularity and decrease in cost, large-scale projects that combine and integrate independently generated datasets will become standard [72]. However, overcoming the inherent batch effects [73] of such approaches presents a unique challenge. While linear or generalized linear modeling frameworks can be used to integrate disparate datasets, the performance of these frameworks in the scRNA-seq context may be sub-optimal, due to their underlying assumption that the composition of cell populations is either known or identical across groups of cells [74]. While this assumption can be addressed through explicit consideration of similar clusters of cells, new approaches to integrate datasets from distinct single-cell experiments that are largely independent of linear models, and which do not even rely on producing a corrected gene expression matrix, have been developed to address this issue.

One approach involves the identification of the most biologically similar cells between batches using mutual nearest neighbors (MNNs) [74], which is implemented in the *scran* [59] and *batchelor* [75] packages. The MNN-corrected data can be directly used in downstream analyses, similar to other dimensionality reduction approaches, such as clustering or trajectory analyses. The *scMerge* package draws upon a similar approach, but instead uses mutual nearest clusters to merge datasets [76]. However, *scMerge* requires a user to supply the number of clusters *a priori* or have annotated cell types to run in unsupervised or semi-supervised mode, respectively. Similar to MNN in *scran* and *scMerge*, the *scmap* package [77] performs an approximate nearest neighbor search to project cells from a query dataset or experiment onto the cell types (clusters) or individual cells in a different dataset or experiment. Alternative implementations based on unsupervised deep learning methods such as the *scAlign* package [78] have also recently been proposed. Finally, several data integration methods developed for bulk assays are available in the *mixOmics* [79] package and have shown good performance on single-cell data [80].

Overall, the approaches that have been tailored for scRNA-seq and explicitly address dataset integration have performed well with respect to ameliorating batch effects in recent benchmarks comparing against traditional statistical modeling frameworks [80]. However, care should be taken in utilizing corrected gene expression matrices, and further validation should be applied to ensure the result is sufficiently free from artifacts. The integration result is especially useful for clustering and visualization. Statistical frameworks can thus then

be applied on the (raw) counts data in service of downstream methods such as differential expression, which become more reliable thanks to the integration approach separating different cell types.

Beyond simply addressing batch effects, one particularly exciting application of these integration approaches is in facilitating comparisons of *de novo* scRNA-seq data with published reference compendiums such as the Human Cell Atlas [16] or, for the model organism Mus musculus, *Tabula Muris* [81, 82]. Such approaches may not only facilitate cross-comparisons, but furthermore enable the annotation of *de novo* datasets on the basis of previously annotated references (see the *Annotation* section for a broader discussion).

In our online supplement, we demonstrate an example of combining datasets derived from different sources using the integration approaches described above (**Table 1**).

Feature Selection

In most experiments, only a subset of genes drive heterogeneity across the population of cells profiled within a single-cell experiment. Feature selection is the process of identifying the most informative features or genes that drive the biological variation, such as genes with high variance from biological effects, rather than technical effects. In the analysis of scRNA-seq data, feature selection is an important step because it reduces the computational burden and removes noise in downstream analyses.

In cases with known groups of cells, correlation-based approaches [83] or the identification of differentially expressed genes [84] across groups can be used to select key features. However, in practice, these groups of similar cells are not often known *a priori* unless the experimental design contains sorted populations or other markers to designate known cell populations, such as engineered gene constructs. More commonly in scRNA-seq data, feature selection is either: expression-based, selecting genes with a high overall mean expression across cells; variance-based, selecting highly variable genes relative to overall mean expression [59, 64, 65]; dropout-based, selecting high dropout genes relative to overall mean expression [84]; deviance-based, selecting genes based on how well each gene fits a null model of constant expression across cells [85]; or a mixture of these strategies. For recent reviews comparing feature selection methods for scRNA-seq data, including the concordance across different approaches, see Yip *et al.* [86] and Andrews *et al.* [84].

Dimensionality Reduction

Cutting-edge single-cell assays can potentially measure thousands of features genome-wide, in hundreds of thousands to millions of cells. With data on this scale, many types of analyses quickly become computationally intractable. While feature selection can mitigate this high-dimensional problem to a certain extent, it is often insufficient in reducing the complexity of single-cell data. Dimension reduction approaches elegantly resolve this dilemma by creating low dimensional representations that nonetheless preserve meaningful structure. The end result can then be subsequently used for data visualization as well as downstream analyses such as clustering and trajectory analysis.

The *SingleCellExperiment* container has a dedicated slot, `reducedDims`, for holding such reduced dimension representations of single-cell data. This slot is used by single-cell Bioconductor packages to provide uniform storage and access of linear and non-linear reduced representations of the data. For example, the *scater* [56] package uses this slot to store and visualize reduced dimension representations of single-cell data after applying dimensionality reduction methods. These include the top principal components (PCs) after performing principal components analysis (PCA), the *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) components [44] using the *Rtsne* [87] R package, the uniform manifold approximation and projection (UMAP) components [45] using the *umap* [88] R package, and diffusion maps [89] using the *destiny* Bioconductor package, respectively. In addition, the *BiocSingular* [90] package provides access to both exact and approximate singular value decomposition (SVD) methods for developers of Bioconductor packages to implement various forms of SVD within their own package. To improve the speed of computations in SVD methods, *BiocSingular* uses the *BiocParallel* [91] Bioconductor framework to parallelize operations.

The *zinbwave* [31] Bioconductor package takes an alternative approach, calculating a model-based dimensionality reduction based on the ZINB-WaVE model [31] tailored for zero-inflated count data that allows for adjustment for confounding factors, as described above. In a similar fashion to *scater*, the *zinbwave* [31] package

9

works seamlessly with counts from *SingleCellExperiment* objects, and can store its results in the `reducedDims` slot for downstream analysis.

Downstream Statistical Analysis

The choice of statistical analyses and workflows can differ greatly depending on the specific goals of the investigation. Following preprocessing of the data as described above, here we illustrate how the Bioconductor framework can be used to answer a variety of biological questions from single-cell data, using tools that are interoperable with the *SingleCellExperiment* class and scale with the number of cells. Also, we provide an online workbook (**Box 2**) that provides workflows and case studies for users on how to perform many of these downstream analyses with single-cell data.

Clustering

Data derived from single-cell assays have enabled researchers to unravel tissue heterogeneity at unprecedented levels of detail, enabling the identification of novel cell types, as well as rare cell populations that were previously unidentifiable using bulk assays [92–94]. Unsupervised clustering – the process of grouping cells based on a similarity metric without a known reference – is a fundamental step in deconvoluting heterogeneous single-cell data into clusters that relate to biological concepts, such as discrete cell types or cell states. Clustering is also essential for other analyses, such as differential expression, in order to identify distinct cellular sub-populations. While, the degree of the separation between the detected clusters is relevant to the robustness and confidence in the clusters, it is more important to think about using clustering methods to yield some hypothesis for experimental validation.

While many algorithms and software packages have previously been used to cluster data from bulk assays and single-cell flow cytometry [95], the complexities of scRNA-seq data pose unique challenges for clustering tasks, specifically a richer feature space, large numbers of cells, and data sparsity [96]. To address these challenges, Bioconductor has developed software packages that incorporate recent advance in nearest-neighbors and clustering algorithms that improve computational efficiency through approaches such as using approximate methods instead of exact methods, thereby trading an acceptable amount of accuracy for vastly improved runtimes. For example, the *BiocNeighbors* package [97–100] can be used to search for nearest neighbors and then a shared nearest neighbor graph using cells as nodes can be built using the *scran* [59]. Further, approximate methods have the advantage of smoothing over noise and sparsity, and thus potentially providing a better fit to the data [101]. Other approaches for improving computational speed include implementing specially designed versions of classical algorithms that support parallelization, enabling multicore and/or multinode processing [91].

Two implementations of unsupervised clustering frameworks come from two Bioconductor packages, *SC3* [102] and *clusterExperiment* [103], which calculate consensus clusters derived from multiple parametrizations. The *SIMLR* package [104] uses kernels to learn a distance metric between cells tailored to improve sensitivity in noisy single-cell data. For large data that require a file-backed representation such as a HDF5 file, the mini-batch k-means algorithm [105] has been implemented in the *mbkmeans* package [106], efficiently replicating the results of classical k-means clustering. For experiments with control genes present in the form of spike-in controls, the *BEARscc* [107] package estimates cluster variability due to technical noise [57]. For evaluating clustering performance across methods and parameter spaces, the results of these various methods can be assessed quantitatively and visually using the *SC3*, *clusterExperiment*, and *clustree* [108] packages. Together, these packages directly support the *SingleCellExperiment* class, allowing for seamless interoperability across a host of clustering methods.

Amongst these methods, the *SC3* package was found to be a top performer in two separate benchmarking studies [109, 110]. However, it should be noted that for the methods described above, only *SC3* was tested in the benchmarking studies. Thus, care should be taken in the choice of clustering method, as each method may excel in different contexts, and results should always be evaluated over various instantiations. For a demonstration of implementing clustering approaches, we refer the interested reader to our online supplement (**Table 1**).

Differential Expression

Given identifiable groups of cells, differential expression analysis can be used to identify features that uniquely distinguish each of the biological groups (clusters) of cells. The results from this analysis can then be used to identify the cell populations present. Another application involves comparing cells within a given population across various conditions, such as time or treatment. In both cases, it is important to correct for confounders such as systematic batch effects [19, 111]; furthermore, it is important to consider how these effects may drive clustering, and thus confound differential expression analyses from the start.

The distinct challenges described in the previous section for scRNA-seq data, namely data size and sparsity, have also spurred many methodological and computational developments in Bioconductor for identifying differentially expressed features between biological groups. Moreover, multiple recent benchmark papers have highlighted that many of the top performing scRNA-seq differential expression tools are Bioconductor packages (e.g. MAST and edgeR, mentioned below) [112–114].

Across these differential expression methods, two general approaches stand out. The first approach retrofits frameworks initially designed for bulk RNA-seq analysis, namely the *edgeR* [3, 62], *DESeq2* [7], or *limma* [115] frameworks. In this approach, the *zinbwave* [31] package can be used to model the single-cell data as arising from the zero-inflated negative binomial distribution (ZiNB) to account for the sparsity inherent to scRNA-seq data [116]. Namely, the *zinbwave* package downweights the excess zeros observed in scRNA-seq data in the dispersion estimation and model fitting steps, thereby enabling improved differential expression analysis. While this approach is slightly more complex at the onset, the *edgeR*, *DESeq2*, and *limma* packages have the benefit of being well-supported, robust, and with ample documentation.

The second class of approaches is uniquely tailored for single-cell data because the statistical methods proposed directly model the zero-inflation component, frequently observed in scRNA-seq data. These methods explicitly separate gene expression into two components: the discrete component, which describes the frequency of a binary component (zero versus non-zero expression), and the continuous component, where the level of gene expression is quantified. While all the methods mentioned herein can test for differences in the continuous component, only this second class of approaches can explicitly model the discrete component, and thus test for differences in the frequency of expression. To do this, the *MAST* [28] package uses a hurdle model framework, whereas the *scDD* [117], *BASiCS* [64, 65], and *SCDE* [18, 118] packages use Bayesian mixture and hierarchical models. Together, these methods are able to provide a broader suite of testing functionality and can be directly used on the scRNA-seq data contained within a *SingleCellExperiment* object.

For a guide in performing differential expression analyses, we refer the interested reader to our online supplement (**Table 1**).

Trajectory Analysis

In contrast to comparing distinct groups of cells within an experiment as described above, heterogeneity in some cases may be better explained as a continuous spectrum, arising due to processes such as cell differentiation. A specialized application of dimension reduction, trajectory analysis - also known as pseudotime inference - uses a broad array of phylogenetic methods to order cells (e.g. by beginning state, intermediate state, end state) along a trajectory of interest, such as a developmental process occurring over time (see [119, 120] for further discussion). From this inferred trajectory, it is possible to identify, for example, new subsets of cells, a differentiation process, or events responsible for bifurcations, such as branch-points, in a dynamic cellular process [121, 122]. As trajectory inference is intractable with bulk data, this type of analysis presents an exciting new avenue for methods development in single-cell applications.

New developments in trajectory inference methods have greatly expanded their capabilities. Whereas early versions required more guidance from the user - such as an expected topology - modern approaches have largely minimized the need for extensive parametrization. Furthermore, modern methods have led the development of novel statistical applications that can test for significant gene expression changes along a continuum and at branch points [123, 124].

In a recent review by Saelens et al. [125], they created a wrapper package - *dynverse* - to perform an extensive benchmarking of trajectory inference methods using both real and simulated datasets. From their evaluation, which included both performance aspects and qualitative features such as documentation, four of

the five top performers were the Bioconductor packages *slingshot* [126], *TSCAN* [30], *monocle* [123, 124, 127], and *cellTree* [128]; the fifth was the *SCORPIUS* [129] package currently hosted on CRAN.

However, each of the methods above can produce drastically different results. This is largely due to the inherent complexity of the task and choices made by the authors of the packages in setting sensible default parameters, which may favor a specific type of topology. Therefore, it is essential to extensively test a suite of methods and parametrizations to assess the robustness of results from trajectory analysis methods. To facilitate such testing, the *monocle*, *slingshot*, and *CellTrails* (a newer method that as of this writing has not been tested by Saelens et al.) packages provide explicit support for *SingleCellExperiment* objects. For a more in-depth discussion on methods and benchmarking of trajectory analysis methods, we refer the interested reader to the previously mentioned paper [125] and a recent review by Tanay et al. [130].

Annotation

One of the challenges of working with high-throughput data, such as that typically encountered in scRNA-seq, is the characterization of the data into familiar terms. This has led to the development of approaches that characterize expression changes of individual genes in the context of gene sets, often termed "gene signatures". The basic structure of all gene signatures is that they provide a list of genes with a shared biological context. This context can be derived from myriad sources - from experimentally derived differential expression analyses to manually curated compilations of genes involved in metabolic and molecular pathways. More complex versions of gene signatures can also possess additional characteristics, such as quantitative or qualitative metrics, or even relationships between the elements of a given signature that produces a network representation. Quantifying the enrichment of these signatures for significant changes in expression has been a hallmark of functional annotation. In addition to this classical approach, novel methodologies developed in the era of scRNA-seq now bypass the usage of gene signatures, applying a data-centric approach to the annotation of *de novo* data using a reference dataset.

Here, we cover how such publicly available gene signatures can be applied to quantify the enrichment of gene signatures in scRNA-seq, methodologies that rely on reference data for annotation, and finally discuss a specialized application of these two approaches in assigning cell labels to single-cells or clusters of cells within an scRNA-seq experiment.

Accessing Public Gene Signatures

The characterization of gene signatures has made great strides through the coordinated efforts of many groups, which have pioneered strategies for standardizing cell type representations and developing statistical methods for the identification of necessary and sufficient markers for cell types and functional signatures [131, 132]. The field has generated countless public knowledge-bases from which such signatures can be used for downstream enrichment analysis. These databases vary in their approach to the definition of signatures - some are based on experimental approaches, such as differential expression studies between conditions in the immunological module of MSigDB [133], whereas others rely on curated knowledge of well known molecular functions and biological processes, such as KEGG [134], Reactome [135], and Gene Ontology (GO) [136]. While covering the various available knowledge bases is outside of the scope of this manuscript, we encourage the use of programmatically accessible databases (e.g. those with application programming interfaces such as REST) and data packages which provide ready to use annotation data in order to facilitate reproducible analyses.

Gene Signature Enrichment

In this section, we focus on methods which rely on explicit gene signatures, either from manually curated or experimentally derived sources akin to those described above. Similar to identifying differentially expressed genes in single-cell data, there are two general approaches to test for an enrichment of genes in functional gene signatures. The first approach adapts existing gene set analysis methods originally developed for the analysis of microarray and bulk RNA-seq, such as *GSEA* [133] (or a fast implementation of pre-ranked GSEA in the *fgsea* package [137]), *goseq* [138], and *PADOG* [139], using observational weights to account for the excess zero

observed in scRNA-seq data [116]. For a comprehensive set of available gene set analysis methods available on Bioconductor, see the *EnrichmentBrowser* package [140], which facilitates the usage of 10 different gene set based methods, with further options to combine resulting gene set rankings across methods. The second approach is a set of enrichment methods specifically tailored for scRNA-seq data. The *MAST* [28] package implements a competitive gene set test that accounts for inter-gene correlation using the hurdle model whereas the *AUCell* [141] package scores the activity level of gene sets using a rank-based scoring method and computes a gene set activation scores for each cell. Finally, the *slalom* [142] package uses a factorial single-cell latent variable model to explain variation in an scRNA-seq data set as a function pre-annotated gene sets.

Data-centric Enrichment Methods

A complementary approach to using published gene signatures relies on learning such signatures *de novo* from reference data. Such approaches, while still nascent, have the potential benefit of being able to characterize biological processes through the use of more comprehensive, quantitative signature definitions, as they go beyond just a list of genes. For example, the *scmap* [77] package projects cells from an scRNA-seq experiment onto the cell-types or individual cells identified in a different experiment. Similarly, the *scCoGAPS* [143, 144] package generates gene expression signatures and then maps the learned signatures onto new datasets to learn shared biological characteristics.

Labeling of Cells

A specialized application of annotation methods in the analysis of scRNA-seq data is in automating the classification of unknown cells to known cell types. To accomplish this, a source of prior knowledge is first required. This can be in the form of a marker panel or a well-annotated reference scRNA-seq dataset, reflecting the division between gene signature and data-centric enrichment methods described above. Secondly, the choice of method will depend on the desired resolution of the annotation, as the labels are assigned either at the single-cell level or on (predefined) clusters of cells.

Methods that rely on reference scRNA-seq datasets that have been annotated *a priori* at the level of clusters, and thus apply data-centric enrichment approaches, possess some key advantages. Chiefly, by defining the labels at the cluster level from the reference dataset, e.g. from pools of cells, issues with sparsity can be overcome, and the level of uncertainty (variability) in the characteristic signature quantified. However, the definition of clusters, especially when it comes to defining biologically meaningful clusters, is an inherently empirical process, and thus any results relying on clusters - either from the reference dataset or the *de novo* dataset - will be subject to this initial bias. Ameliorating this, the definition of the characteristic signature pertaining to each reference cluster is much more quantitative than traditional manually curated gene signatures. Methods that adopt a cluster-centric labeling approach include the *celaref* [145] and *scmap* [77] packages. Interestingly, the *scmap* package also possesses the capability of annotating a *de novo* scRNA-seq experiment at the single-cell level (while still using the reference scRNA-seq clusters), thus removing one layer of potential bias arising from clustering.

Conversely, approaches that rely on manually curated panels have the benefit of being well-defined based on prior knowledge. Furthermore, they are easily adaptable, as the user can readily tweak the marker panel - extending or shortening signatures, as well as adding or removing cell type definitions. Naturally, such manual definitions also possess some disadvantages compared to data-centric definitions - for example, marker panels often rely on protein level characterization that may not be applicable to scRNA-seq, and these panels are usually much more limited in scope. Thus, it is important to apply such panels in a formalized manner, and one such package that takes this approach comes from *cellassign* [146], which labels individual cells from a *de novo* scRNA-seq experiment.

Accessible and Reproducible Analysis

The richness of data from single-cell assays has immensely increased the space of possible data exploration. Oftentimes however, bespoke visualization frameworks to communicate results are limited in scope or may lack
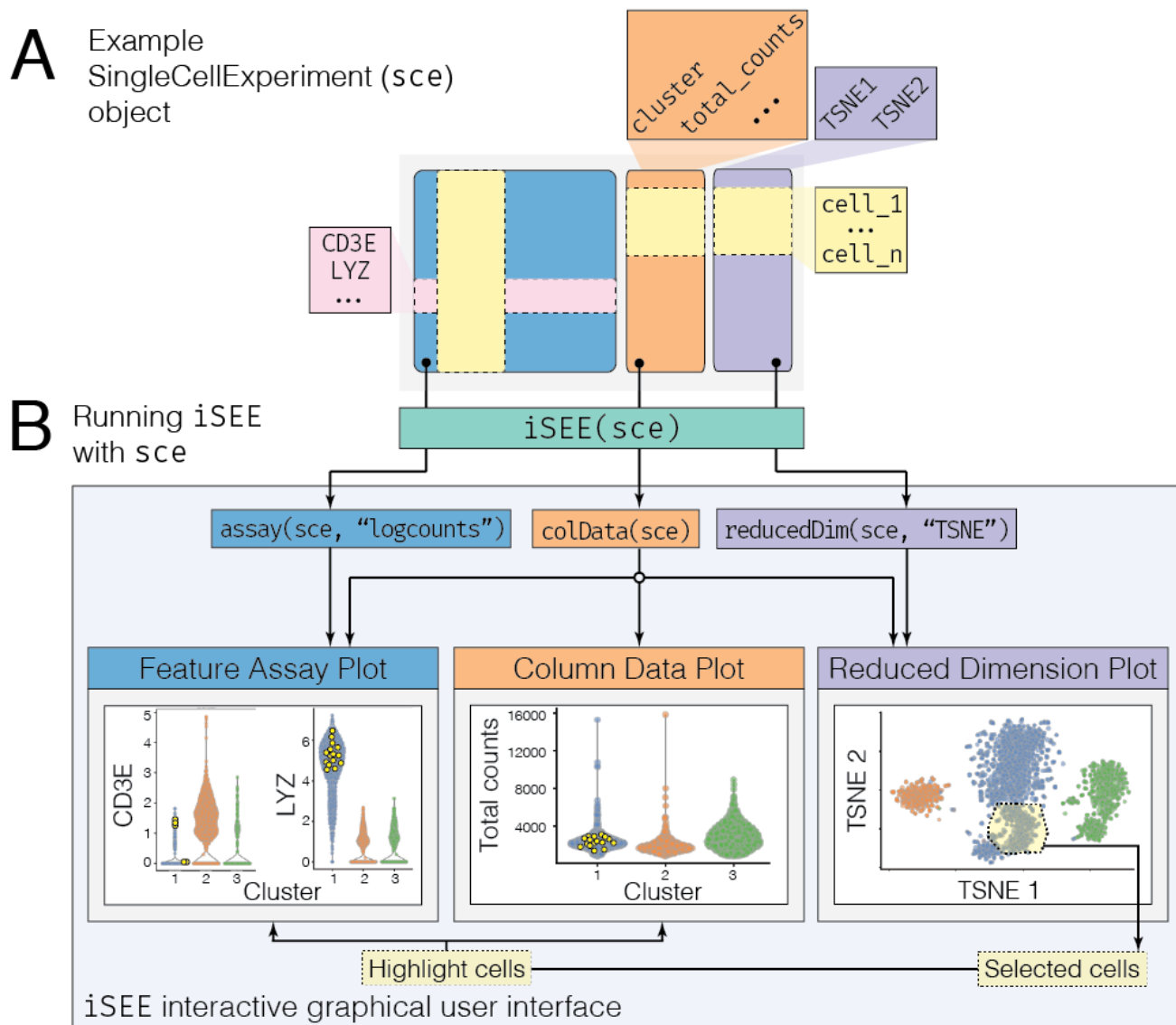
Figure 4: **Interactive Data Exploration in Bioconductor**. **A** An example *SingleCellExperiment* object, sce, is shown with a normalized expression assay that has been log-transformed (logcounts), sample metadata with a column representing the cluster label for each cell amongst other quality control metrics such as total_counts, and a reduced dimension TSNE representation. **B** The sce object is passed to the *iSEE* package using the iSEE() function (green box). This function produces an interactive data visualization accessible via a web browser, encapsulating various types of plots into a single application (large light gray box). The various components of the sce data container are used differently by different plots (arrows pointing from slots to plots). The feature assay plot (left) derives its data from the assays slot (blue box), whereas the reduced dimension plot (right) derives its coordinates from the reducedDims slot (purple box). Clustering information from the colData slot is used for each plot to separate and color the data by clusters (middle), as well as to visualize cell metadata (orange box). Cells from any individual plot can be selected via a shiny brush or lasso selection tool, and then transmitted and visualized in any other plot (yellow selection in reduced dimension plot, purple dots on feature assay and column data plot).

essential infrastructure that ensures reproducibility over time. To address these challenges, the Bioconductor community has embraced both software solutions and best practices, as described below.

Interactive Data Visualization

The maturation of web technologies has opened new avenues for the interactive exploration of data. These web technologies have been embraced by the broader R community through the use of *shiny* [147], an R package that facilitates the development of complex interfaces with relative ease. Given the complexity of scRNA-seq data, it is crucial to facilitate the exploration of data across teams of researchers directly working with the data and also to facilitate the communication of results to the broader community. Furthermore, such communication must accommodate audiences with varying levels of programming expertise. To this end, the *iSEE* [148] package provides a full-featured application for the interactive visualization of scRNA-seq data sets through an internet browser, eliminating the need for any programming experience if the instance is hosted on the web (**Figure 4**). The *iSEE* package directly interfaces with the *SingleCellExperiment* data container to enable the simultaneous exploration of single-cell data and metadata. For example, the *iSEE* package includes functionality to visualize dimensionality reduction results such as *t*-SNE or UMAP while coloring cells, or alternatively, the expression level of any feature by their assigned cluster or cell type. Lastly, the *iSEE* package lets the user export the code required to reproduce the interactive graphics as well, facilitating long-term reproducibility of exploratory analyses.

Report Generation

Ensuring that results are well-documented, shareable, and reproducible is of utmost importance for publication and external validation of results. To address this, Bioconductor promotes the usage of R packages like *rmarkdown* [149] and *bookdown* [150] to produce analytical reports, tools which were all used to generate our companion online book (see **Box 2**). Furthermore, Bioconductor has published the *BiocStyle* package [151] to provide standardized formats for package associated vignettes that illustrate software functionality.

Code and reports inherent to a project, especially those associated with documentation or publication, are encouraged to be shared as part of a package as an included vignette as well as in public code repositories such as GitHub. For example, Bioconductor publishes data packages with vignettes that demonstrate how to reproduce an associated manuscript's relevant figures [152]. In addition, packages such as *iSEE* [148] report all the code used to generate a visualization that was manually specified via the interface described prior.

Beyond bespoke reports or packages, Bioconductor has also added packages that automate the creation of shareable, standalone reports. In particular, one area of focus is the automation of quality control documentation. Namely, the *countsimQC* [153] and *batchQC* [154] packages automate the visualization of various sample characteristics, such as library sizes, the number of genes quantified, and additionally illustrate what batch correction procedures may need to be applied prior to downstream analysis.

Published and Simulated Data

Access to data – both simulated and published – is essential to validate and benchmark previously established and new methods. Given the rapid ascent of high-throughput single-cell assays, Bioconductor has actively encouraged the community to publish data packages, as well as use standardized data simulation frameworks to foster further development in methods and synthesis of analytic results.

Single-cell Data Packages

As new single-cell assays, statistical methods and corresponding software continue to be developed, it becomes increasingly important to facilitate the publication of datasets, both to reproduce existing analysis as well as to enable comparisons across new and existing tools. Bioconductor encourages the publication of data packages, which are primarily focused on providing accessible, well annotated, clean versions of data that are ready to be analyzed by end users.

To standardize the querying of published data packages on Bioconductor, the *ExperimentHub* [155] Bioconductor package was created to enable programmatic access of published datasets using a standardized interface. Amongst available scRNA-seq datasets, the *TENxPBMCData*, and *TENxBrainData* data packages from the 10X Genomics platforms provide streamlined access to the *ExperimentHub* resource for their respective data

sets. In addition, the *HCAData* and *HCABrowser* packages provide access to the data portal from the Human Cell Atlas [16] consortium.

It is important to note that with *ExperimentHub* [155], data is processed and submitted by the authors of the package, and thus data packages from multiple sources may not be readily comparable to each other. To address this, single-cell data compendiums, such as *conquer* [112], provide users an interactive online resource of many scRNA-seq datasets where each has been processed in a consistent manner and with additional quality control steps applied. Such an approach helps standardize various pipelines used to process scRNA-seq data in order to make them more readily comparable [112].

Benchmarking Datasets

Of note within single-cell data packages, benchmarking datasets are labeled with some known "ground truth" designed for the evaluation of new methods. Amongst these, the *CellBench* [80] data set is one such example specific to scRNA-seq applications, comprised of data from various platforms and cell lines. As such, it can be used to validate batch correction, differential expression, and clustering methods. Another benchmarking dataset for scRNA-seq data is from Tung *et al.* (2017) [156], which can be used for benchmarking methods using data derived from plate-based protocols. Finally, the *DuoClustering2018* package [109] contains data from a variety of sources which were used to test the performance of various clustering methods, with the data primarily chosen to represent different degrees of difficulty in the clustering task.

Simulating Data

Simulated data is logical prerequisite for method development where it is necessary to fully know the generative model underlying the data, and thus, accurately benchmark the performance of a method. For scRNA-seq data, the *splatter* package [157] can simulate the presence of multiple cell types, batch effects, varying levels of dropout events, differential gene expression, as well as trajectories, providing a rich canvas for methods testing and development. Furthermore, the *splatter* package uses both its own simulation framework and wraps other simulation frameworks with differing generative models such as *scDD* and *BASiCS* to provide a comprehensive resource for single-cell data simulation.

Discussion

The open-source and open-development Bioconductor community has developed state-of-the-art computational methods, standardized data infrastructure, and interactive data visualization tools available as software packages for the analysis of data derived from cutting-edge single-cell assays. The rapid development of high-dimensional single-cell assays producing data sets of increasing size, complexity and sparsity, has the Bioconductor community to implement profound changes in how users access, store, and analyze data, including: (1) memory-efficient data import and representation, (2) common data containers for storing data from single-cell assays for interoperability between packages, (3) fast and robust methods for transforming raw single-cell data into processed data ready for downstream analyses, (4) interactive data visualization, and (5) downstream analyses, annotation and biological interpretation. In addition, emerging single-cell technologies in epigenomics, T-cell and B-cell repertoires, and multiparameter assays from single cells (such as joint/simultaneous protein and transcriptional profiling) promise to continue to push forward advances in computational biology. Based on the unique strengths of Bioconductor, in particular its strong connection between users and developers, the high degree of responsiveness of developers to user needs that arise, and the reach of the Bioconductor community, we are optimistic that it will be the hub of development for these technologies as well.

Being a part of the broader R community presents unique advantages for Bioconductor, including accessibility to statisticians and data scientists. The Bioconductor software project has established best practices for coordinated package versioning and code review. Alongside community-contributed packages, a team of core developers (https://www.bioconductor.org/about/core-team/) implements and maintains the infrastructure needed by the global project, as well as reviews contributed packages to ensure they satisfy Bioconductor package guidelines. Taken together, these practices result in high-quality and consistently maintained packages.

Table 1: **Bioconductor workflows for single-cell analyses.**

Name	Description	URL
Integrating Datasets	Using two datasets from the *TENxPBMCData* data packages – one with 3000 cells and one with 4000 cells – the workflow demonstrates how to preprocess this scRNA-seq data and integrate the two datasets using mutual nearest neighbors. We visually inspect the results using the first two principal components.	`http://osca.bioconductor.org/workflow-integrating-datasets.html`
Clustering	Using the *CellBench* [80] benchmarking dataset (specifically the `sc_10x_5cl` dataset, which contains 5 sorted cell lines that were sequencing on the 10X Genomics platform), the workflow demonstrates how to preprocess the scRNA-seq data and highlights several clustering packages and methods, which are suitable for large datasets.	`http://osca.bioconductor.org/workflow-clustering.html`
Differential Expression	Using the *Tabula muris* data compendium of mouse organ systems, this workflow illustrates performing differential expression analysis across different cell populations.	`http://osca.bioconductor.org/differential-expression.html`
Large-scale Analyses	Using the *HCAData* [16] package, the workflow demonstrates current practices for dealing with large-scale datasets, illustrated the use of *DelayedArray* operations and methods that support disk-backed (HDF5) data representations.	`http://osca.bioconductor.org/large-scale-data.html`

In addition, Bioconductor provides standardized data containers that enable interoperability between Bioconductor packages, between Bioconductor and CRAN [2] packages, and between R and other programming languages. For instance, it is simple to convert between a *SingleCellExperiment* object and the format used by the popular single-cell CRAN package *Seurat* [158] and Python [159] package *scanpy* [160] and vice versa. Indeed, R has a long history of interoperability with other programming languages. Two notable examples are the *Rcpp* [161–163] package for integrating C++ compiled code into R and the *reticulate* [164] package for interfacing with Python. This interoperability enables common machine learning frameworks such as TensorFlow/Keras to be used directly in R.

To the newcomer, the wealth of single-cell analyses possible in Bioconductor can be daunting. To address the rapid growth of contributed packages within the single-cell analysis space, we have summarized and highlighted state-of-the-art methods and software packages and organized the packages into the broad sections of a typical single-cell analysis workflow (**Figure 2**) alongside companion code-based workflows showcasing their use (`https://osca.bioconductor.org`). Finally, Bioconductor software packages are organized into `BiocViews`, an ontology of topics that classify packages by task or technology. This effort increases discoverability and interoperability. In the future, additional meta packages may be developed to wrap similar methods together.

Author Contributions

Acknowledgements

Competing Financial Interests

The authors declare no competing financial interests.

References

[1] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, Raphael Gottardo, Florian Hahne, Kasper D Hansen, Rafael A Irizarry, Michael Lawrence, Michael I Love, James MacDonald, Valerie Obenchain, Andrzej K Oleś, Hervé Pagès, Alejandro Reyes, Paul Shannon, Gordon K Smyth, Dan Tenenbaum, Levi Waldron, and Martin Morgan. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods*, 12(2):115–21, 02 2015. doi:10.1038/nmeth.3252.

[2] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

[3] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–40, 2010. doi:10.1093/bioinformatics/btp616. URL `https://bioconductor.org/packages/edgeR`.

[4] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biol*, 11(10):R106, 2010. doi:10.1186/gb-2010-11-10-r106. URL `https://bioconductor.org/packages/DESeq`.

[5] Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T Morgan, and Vincent J Carey. Software for computing and annotating genomic ranges. *PLoS Comput Biol*, 9(8):e1003118, 2013. doi:10.1371/journal.pcbi.1003118. URL `https://bioconductor.org/packages/IRanges`.

[6] Martin J Aryee, Andrew E Jaffe, Hector Corrada-Bravo, Christine Ladd-Acosta, Andrew P Feinberg, Kasper D Hansen, and Rafael A Irizarry. Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics*, 30(10):1363–9, 2014. doi:10.1093/bioinformatics/btu049. URL `https://bioconductor.org/packages/minfi`.

[7] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*, 15(12):550, 2014. doi:10.1186/s13059-014-0550-8. URL `https://bioconductor.org/packages/DESeq2`.

[8] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res*, 43(7):e47, 2015. doi:10.1093/nar/gkv007. URL `https://bioconductor.org/packages/limma`.

[9] Davide Risso, John Ngai, Terence P Speed, and Sandrine Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nat Biotechnol*, 32(9):896–902, 2014. doi:10.1038/nbt.2931. URL `https://bioconductor.org/packages/RUVSeq`.

[10] Martin Morgan, Hervé Pagès, Valerie Obenchain, and Nathaniel Hayden. *Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import*, 2018. URL `http://bioconductor.org/packages/Rsamtools`. R package version 1.34.0.

[11] Simona Serratì, Simona De Summa, Brunella Pilato, Daniela Petriella, Rosanna Lacalamita, Stefania Tommasi, and Rosamaria Pinto. Next-generation sequencing: advances and applications in cancer diagnosis. *Onco Targets Ther*, 9:7355–7365, 2016. doi:10.2147/OTT.S99807.

[12] Ryuichiro Nakato and Katsuhiko Shirahige. Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation. *Brief Bioinform*, 18(2):279–290, 2017. doi:10.1093/bib/bbw023.

[13] Kimberly R Kukurba and Stephen B Montgomery. RNA sequencing and analysis. *Cold Spring Harb Protoc*, 2015(11):951–69, 2015. doi:10.1101/pdb.top084970.

[14] Greg Finak, Jacob Frelinger, Wenxin Jiang, Evan W Newell, John Ramey, Mark M Davis, Spyros A Kalams, Stephen C De Rosa, and Raphael Gottardo. OpenCyto: an open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis. *PLoS Comput Biol*, 10(8):e1003806, 2014. doi:10.1371/journal.pcbi.1003806. URL `https://bioconductor.org/packages/openCyto`.

[15] Greg Finak, Wenxin Jiang, and Raphael Gottardo. CytoML for cross-platform cytometry data sharing. *Cytometry A*, 93(12):1189–1196, 2018. doi:10.1002/cyto.a.23663. URL `https://bioconductor.org/packages/CytoML`.

[16] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N Schumacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W Shin, Oliver Stegle, Michael Stratton, Michael J T Stubbington, Fabian J Theis, Matthias Uhlen, Alexander van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, Nir Yosef, and Human Cell Atlas Meeting Participants. The Human Cell Atlas. *Elife*, 6, 2017. doi:10.7554/eLife.27041.

[17] Andrew McDavid, Greg Finak, Pratip K Chattopadyay, Maria Dominguez, Laurie Lamoreaux, Steven S Ma, Mario Roederer, and Raphael Gottardo. Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments. *Bioinformatics*, 29(4):461–7, 2013. doi:10.1093/bioinformatics/bts714.

[18] Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nat Methods*, 11(7):740–2, 2014. doi:10.1038/nmeth.2967. URL `https://bioconductor.org/packages/scde`.

[19] Stephanie C Hicks, F William Townes, Mingxiang Teng, and Rafael A Irizarry. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 19(4):562–578, 2018. doi:10.1093/biostatistics/kxx053.

[20] Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza, David N Louis, Orit Rozenblatt-Rosen, Mario L Suvà, Aviv Regev, and Bradley E Bernstein. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–401, 2014. doi:10.1126/science.1254257.

[21] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–6, 2014. doi:10.1126/science.1245316.

[22] Christelle Borel, Pedro G Ferreira, Federico Santoni, Olivier Delaneau, Alexandre Fort, Konstantin Y Popadin, Marco Garieri, Emilie Falconnet, Pascale Ribaux, Michel Guipponi, Ismael Padioleau, Piero Carninci, Emmanouil T Dermitzakis, and Stylianos E Antonarakis. Biased allelic expression in human primary fibroblast single cells. *Am J Hum Genet*, 96(1):70–80, 2015. doi:10.1016/j.ajhg.2014.12.001.

[23] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol*, 33(5):495–502, 2015. doi:10.1038/nbt.3192.

[24] Kaia Achim, Jean-Baptiste Pettit, Luis R Saraiva, Daria Gavriouchkina, Tomas Larsson, Detlev Arendt, and John C Marioni. High-throughput spatial mapping of single-cell RNA-seq data to tissue of origin. *Nat Biotechnol*, 33(5):503–9, 2015. doi:10.1038/nbt.3209.

[25] Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, 2nd, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, Mohammad Fallahi-Sichani, Ken Dutton-Regester, Jia-Ren Lin, Ofir Cohen, Parin Shah, Diana Lu, Alex S Genshaft, Travis K Hughes, Carly G K Ziegler, Samuel W Kazer, Aleth Gaillard, Kellie E Kolb, Alexandra-Chloé Villani, Cory M Johannessen, Aleksandr Y Andreev, Eliezer M Van Allen, Monica Bertagnolli, Peter K Sorger, Ryan J Sullivan, Keith T Flaherty, Dennie T Frederick, Judit Jané-Valbuena, Charles H Yoon, Orit Rozenblatt-Rosen, Alex K Shalek, Aviv Regev, and Levi A Garraway. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science*, 352(6282):189–96, 2016. doi:10.1126/science.aad0501.

[26] Mihriban Karaayvaz, Simona Cristea, Shawn M Gillespie, Anoop P Patel, Ravindra Mylvaganam, Christina C Luo, Michelle C Specht, Bradley E Bernstein, Franziska Michor, and Leif W Ellisen. Unravelling subclonal heterogeneity and aggressive disease states in TNBC through single-cell RNA-seq. *Nat Commun*, 9(1):3588, 2018. doi:10.1038/s41467-018-06052-0.

[27] Hanna Mendes Levitin, Jinzhou Yuan, and Peter A Sims. Single-Cell Transcriptomic Analysis of Tumor Heterogeneity. *Trends Cancer*, 4(4):264–268, 2018. doi:10.1016/j.trecan.2018.02.003.

[28] G Finak, A McDavid, M Yajima, and others. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol*, 16:278, 2015. doi:s13059-015-0844-5. URL https://bioconductor.org/packages/MAST.

[29] Aaron T L Lun, Karsten Bach, and John C Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol*, 17:75, 2016. doi:10.1186/s13059-016-0947-7. URL https://bioconductor.org/packages/scran.

[30] Zhicheng Ji and Hongkai Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res*, 44(13):e117, 2016. doi:10.1093/nar/gkw430. URL https://bioconductor.org/packages/TSCAN.

[31] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat Commun*, 9(1):284, 2018. doi:10.1038/s41467-017-02554-5. URL https://bioconductor.org/packages/zinbwave.

[32] Luyi Tian, Shian Su, Xueyi Dong, Daniela Amann-Zalcenstein, Christine Biben, Azadeh Seidi, Douglas J Hilton, Shalin H Naik, and Matthew E Ritchie. scPipe: A flexible R/Bioconductor pre-processing pipeline for single-cell RNA-sequencing data. *PLoS Comput Biol*, 14(8):e1006361, 2018. doi:10.1371/journal.pcbi.1006361. URL https://bioconductor.org/packages/scPipe.

[33] Yang Liao, Gordon K Smyth, and Wei Shi. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Res*, 41(10):e108, 2013. doi:10.1093/nar/gkt214.

[34] Yang Liao, Gordon K Smyth, and Wei Shi. The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. *Nucleic Acids Res*, 2019. doi:10.1093/nar/gkz114. URL https://bioconductor.org/packages/Rsubread.

[35] Grace X Y Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, Mark T Gregory, Joe Shuga, Luz Montesclaros, Jason G Underwood, Donald A Masquelier, Stefanie Y Nishimura, Michael Schnall-Levin, Paul W Wyatt, Christopher M Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D Ness, Lan W Beppu, H Joachim Deeg, Christopher McFarland, Keith R Loeb, William J Valente, Nolan G Ericson, Emily A Stevens, Jerald P Radich, Tarjei S Mikkelsen, Benjamin J Hindson, and Jason H Bielas. Massively parallel digital transcriptional profiling of single cells. *Nat Commun*, 8:14049, 2017. doi:10.1038/ncomms14049.

[36] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*, 14(4):417–419, 2017. doi:10.1038/nmeth.4197.

[37] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic rna-seq quantification. *Nat Biotechnol*, 34(5):525–7, 2016. doi:10.1038/nbt.3519.

[38] Charlotte Soneson, Michael I Love, and Mark D Robinson. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, 4:1521–21, 2015. doi:10.12688/f1000research.7563.2. URL `https://bioconductor.org/packages/tximport`.

[39] Michael Love, Rob Patro, Peter Hickey, and Charlotte Soneson. *tximeta: Transcript Quantification Import with Automatic Metadata*, 2019. URL `https://bioconductor.org/packages/tximeta`. R package version 1.0.3.

[40] Aaron T L Lun, Davide Risso, and K Korthauer. *SingleCellExperiment: S4 Classes for Single Cell Data*, 2019. URL `https://bioconductor.org/packages/SingleCellExperiment`. R package version 1.4.1.

[41] Martin Morgan, Valerie Obenchain, Jim Hester, and Hervé Pagès. *SummarizedExperiment: Summarized-Experiment container*, 2018. URL `https://bioconductor.org/packages/SummarizedExperiment`. R package version 1.12.0.

[42] Douglas Bates and Martin Maechler. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2018. URL `https://CRAN.R-project.org/package=Matrix`. R package version 1.2-15.

[43] Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res*, 39:D52–7, 2011. doi:10.1093/nar/gkq1237.

[44] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.

[45] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *arXiv*, 2018. URL `https://arxiv.org/abs/1802.03426`.

[46] Siddharth S Dey, Lennart Kester, Bastiaan Spanjaard, Magda Bienko, and Alexander van Oudenaarden. Integrated genome and transcriptome sequencing of the same cell. *Nat Biotechnol*, 33(3):285–289, 2015. doi:10.1038/nbt.3129.

[47] Iain C Macaulay, Mabel J Teng, Wilfried Haerty, Parveen Kumar, Chris P Ponting, and Thierry Voet. Separation and parallel sequencing of the genomes and transcriptomes of single cells using GT-seq. *Nat Protoc*, 11(11):2081–103, 2016. doi:10.1038/nprot.2016.138.

[48] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nat Methods*, 14(9):865–868, 2017. doi:10.1038/nmeth.4380.

[49] Payam Shahi, Samuel C Kim, John R Haliburton, Zev J Gartner, and Adam R Abate. Abseq: Ultrahigh-throughput single cell protein profiling with droplet microfluidic barcoding. *Sci Rep*, 7:44447, 2017. doi:10.1038/srep44447.

[50] Christof Angermueller, Stephen J Clark, Heather J Lee, Iain C Macaulay, Mabel J Teng, Tim Xiaoming Hu, Felix Krueger, Sebastien Smallwood, Chris P Ponting, Thierry Voet, Gavin Kelsey, Oliver Stegle, and Wolf Reik. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nat Methods*, 13(3):229–232, 2016. doi:10.1038/nmeth.3728.

[51] Junyue Cao, Darren A Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A Pliner, Andrew J Hill, Riza M Daza, Jose L McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, Frank J Steemers, Andrew C Adey, Cole Trapnell, and Jay Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, 2018. doi:10.1126/science.aau0730.

[52] Stephen J Clark, Ricard Argelaguet, Chantriolnt-Andreas Kapourani, Thomas M Stubbs, Heather J Lee, Celia Alda-Catalinas, Felix Krueger, Guido Sanguinetti, Gavin Kelsey, John C Marioni, Oliver Stegle, and Wolf Reik. scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. *Nat Commun*, 9(1):781, 2018. doi:10.1038/s41467-018-03149-4.

[53] Marcel Ramos, Lucas Schiffer, Angela Re, Rimsha Azhar, Azfar Basunia, Carmen Rodriguez, Tiffany Chan, Phil Chapman, Sean R Davis, David Gomez-Cabrero, Aedin C Culhane, Benjamin Haibe-Kains, Kasper D Hansen, Hanish Kodali, Marie S Louis, Arvind S Mer, Markus Riester, Martin Morgan, Vince Carey, and Levi Waldron. Software for the integration of multiomics experiments in Bioconductor. *Cancer Res*, 77(21):e39–e42, 2017. doi:10.1158/0008-5472.CAN-17-0344. URL https://bioconductor. org/packages/MultiAssayExperiment.

[54] Aaron T. L. Lun, Samantha Riesenfeld, Tomas Gomes Tallulah Andrewsand The Phuong Dao, participants in the 1st Human Cell Atlas Jamboree, and John C. Marioni. Emptydrops: distinguishing cells from empty droplets in droplet-based single-cell rna sequencing data. *Genome Biology*, 20:63, 2019. doi:10.1186/s13059-019-1662-y. URL https://bioconductor.org/packages/DropletUtils.

[55] Jonathan A Griffiths, Arianne C Richard, Karsten Bach, Aaron T L Lun, and John C Marioni. Detection and removal of barcode swapping in single-cell RNA-seq data. *Nat Commun*, 9(1):2667, 2018. doi:10.1038/s41467-018-05083-x. URL https://bioconductor.org/packages/DropletUtils.

[56] Davis J McCarthy, Kieran R Campbell, Aaron T L Lun, and Quin F Wills. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*, 33(8):1179–1186, 2017. doi:10.1093/bioinformatics/btw777. URL https://bioconductor.org/packages/scater.

[57] Lichun Jiang, Felix Schlesinger, Carrie A Davis, Yu Zhang, Renhua Li, Marc Salit, Thomas R Gingeras, and Brian Oliver. Synthetic spike-in standards for RNA-seq experiments. *Genome Res*, 21(9):1543–51, 2011. doi:10.1101/gr.121095.111.

[58] Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. *Proc Natl Acad Sci USA*, 107(21):9546–51, 2010. doi:10.1073/pnas.0914005107.

[59] Aaron T. L. Lun, Davis J. McCarthy, and John C. Marioni. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Res*, 5: 2122, 2016. doi:10.12688/f1000research.9501.2. URL https://www.bioconductor.org/packages/ simpleSingleCell.

[60] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–27, 2007. doi:10.1093/biostatistics/kxj037. URL https://bioconductor.org/packages/sva.

[61] Jeffrey T Leek. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res*, 42(21), 2014. doi:10.1093/nar/gku864. URL https://bioconductor.org/packages/sva.

[62] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. Differential expression analysis of multifactor RNA-seq experiments with respect to biological variation. *Nucleic Acids Res*, 40(10):4288–97, 2012. doi:10.1093/nar/gks042. URL https://bioconductor.org/packages/edgeR.

[63] Rhonda Bacher, Li-Fang Chu, Ning Leng, Audrey P Gasch, James A Thomson, Ron M Stewart, Michael Newton, and Christina Kendziorski. SCnorm: robust normalization of single-cell RNA-seq data. *Nat Methods*, 14(6):584–586, 2017. doi:10.1038/nmeth.4263. URL https://bioconductor.org/packages/ SCnorm.

[64] Catalina A Vallejos, John C Marioni, and Sylvia Richardson. BASiCS: Bayesian Analysis of Single-Cell Sequencing Data. *PLoS Comput Biol*, 11(6):e1004333, 2015. doi:10.1371/journal.pcbi.1004333. URL https://bioconductor.org/packages/BASiCS.

[65] Catalina A Vallejos, Sylvia Richardson, and John C Marioni. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biol*, 17:70, 2016. doi:10.1186/s13059-016-0930-3. URL `https://bioconductor.org/packages/BASiCS`.

[66] Michael B Cole, Davide Risso, Allon Wagner, David DeTomaso, John Ngai, Elizabeth Purdom, Sandrine Dudoit, and Nir Yosef. Performance assessment and selection of normalization procedures for Single-Cell RNA-seq. *bioRxiv*, 2018. doi:10.1101/235382. URL `https://bioconductor.org/packages/scone`.

[67] Andrew McDavid, Greg Finak, and Raphael Gottardo. The contribution of cell cycle to heterogeneity in single-cell RNA-seq data. *Nat Biotechnol*, 34(6):591–3, 2016. doi:10.1038/nbt.3498.

[68] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell rna sequencing data: challenges and opportunities. *Nat Methods*, 14(6):565–571, 2017. doi:10.1038/nmeth.4292.

[69] Dominic Grün and Alexander van Oudenaarden. Design and analysis of single-cell sequencing experiments. *Cell*, 163(4):799–810, 2015. doi:10.1016/j.cell.2015.10.039.

[70] Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat Biotechnol*, 33(2):155–60, 2015. doi:10.1038/nbt.3102.

[71] Ning Leng, Li-Fang Chu, Chris Barry, Yuan Li, Jeea Choi, Xiaomao Li, Peng Jiang, Ron M Stewart, James A Thomson, and Christina Kendziorski. Oscope identifies oscillatory genes in unsynchronized single-cell RNA-seq experiments. *Nat Methods*, 12(10):947–950, 2015. doi:10.1038/nmeth.3549. URL `https://bioconductor.org/packages/Oscope`.

[72] Blanca Pijuan-Sala, Jonathan A Griffiths, Carolina Guibentif, Tom W Hiscock, Wajid Jawaid, Fernando J Calero-Nieto, Carla Mulas, Ximena Ibarra-Soria, Richard C V Tyser, Debbie Lee Lian Ho, Wolf Reik, Shankar Srinivas, Benjamin D Simons, Jennifer Nichols, John C Marioni, and Berthold Göttgens. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019. doi:10.1038/s41586-019-0933-9.

[73] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet*, 11(10):733–9, 2010. doi:10.1038/nrg2825.

[74] Laleh Haghverdi, Aaron T L Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol*, 36(5): 421–427, 2018. doi:10.1038/nbt.4091.

[75] Aaron T L Lun. *Bioconductor package containing a collection of single-cell batch correction methods*, 2019. URL `https://github.com/LTLA/batchelor`.

[76] Yingxin Lin, Shila Ghazanfar, Kevin Wang, Johann A Gagnon-Bartsch, Kitty K Lo, Xianbin Su, Ze-Guang Han, John T Ormerod, Terence P Speed, Pengyi Yang, and Jean Y H Yang. scMerge: Integration of multiple single-cell transcriptomics datasets leveraging stable expression and pseudo-replication. *bioRxiv*, pages 1–13, 2018. doi:10.1101/393280.

[77] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell RNA-seq data across data sets. *Nat Methods*, 15(5):359–362, 2018. doi:10.1038/nmeth.4644. URL `https://bioconductor.org/packages/scmap`.

[78] Nelson Johansen and Gerald Quon. scAlign: a tool for alignment, integration and rare cell identification from scRNA-seq data. *bioRxiv*, 2019. doi:10.1101/504944. URL `https://github.com/quon-titative-biology/scAlign`.

[79] Florian Rohart, Benoit Gautier, Amrit Singh, and Kim-Anh Lê Cao. mixomics: An r package for 'omics feature selection and multiple data integration. *PLoS Comput Biol*, 13(11):e1005752, 2017. doi:10.1371/journal.pcbi.1005752. URL https://bioconductor.org/packages/mixOmics.

[80] Luyi Tian, Xueyi Dong, Saskia Freytag, Kim-Anh Le Cao, Shian Su, Daniela Amann-Zalcenstein, Tom S Weber, Azadeh Seidi, Shalin H Naik, and Matthew E Ritchie. scRNA-seq mixology: towards better benchmarking of single cell RNA-seq protocols and analysis methods. *bioRxiv*, 2018. doi:10.1101/433102. URL https://github.com/LuyiTian/CellBench_data.

[81] Charlotte Soneson. *TabulaMurisData: 10x And SmartSeq2 Data From The Tabula Muris Consortium*, 2018. URL https://bioconductor.org/packages/TabulaMurisData. R package version 1.0.0.

[82] Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562(7727):367–372, 2018. doi:10.1038/s41586-018-0590-4.

[83] Tiago Lubiana and Helder Nakaya. *FCBF: Fast Correlation Based Filter for Feature Selection*, 2019. URL https://bioconductor.org/packages/FCBF. R package version 1.0.1.

[84] Tallulah Andrews and Martin Hemberg. M3Drop: Dropout-based feature selection for scRNASeq. *Bioinformatics*, 2018. doi:10.1093/bioinformatics/bty1044. URL https://bioconductor.org/packages/M3Drop.

[85] F William Townes, Stephanie C Hicks, Martin J Aryee, and Rafael A Irizarry. Feature Selection and Dimension Reduction for Single Cell RNA-Seq based on a Multinomial Model. *bioRxiv*, 2019. doi:10.1101/574574.

[86] Shun H Yip, Pak Chung Sham, and Junwen Wang. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief Bioinform*, 2018. doi:10.1093/bib/bby011.

[87] Jesse H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. URL https://CRAN.R-project.org/package=Rtsne. R package version 0.15.

[88] Tomasz Konopka. *umap: Uniform Manifold Approximation and Projection*, 2018. URL https://CRAN.R-project.org/package=umap. R package version 0.2.0.0.

[89] Philipp Angerer, Laleh Haghverdi, Maren Büttner, Fabian J Theis, Carsten Marr, and Florian Buettner. destiny: diffusion maps for large-scale single-cell data in R. *Bioinformatics*, 32(8):1241–3, 2016. doi:10.1093/bioinformatics/btv715. URL https://bioconductor.org/packages/destiny.

[90] Aaron Lun. *BiocSingular: Singular Value Decomposition for Bioconductor Packages*, 2019. URL https://bioconductor.org/packages/BiocSingular. R package version 0.99.13.

[91] Martin Morgan, Valerie Obenchain, Michel Lang, Ryan Thompson, and Nitesh Turaga. *BiocParallel: Bioconductor facilities for parallel evaluation*, 2019. URL https://bioconductor.org/packages/BiocParallel. R package version 1.16.6.

[92] Akira Nguyen, Weng Hua Khoo, Imogen Moran, Peter I Croucher, and Tri Giang Phan. Single Cell RNA Sequencing of Rare Immune Cell Populations. *Front Immunol*, 9:1553, 2018. doi:10.3389/fimmu.2018.01553.

[93] Andrew J Shih, Andrew Menzin, Jill Whyte, John Lovecchio, Anthony Liew, Houman Khalili, Tawfiqul Bhuiya, Peter K Gregersen, and Annette T Lee. Identification of grade and origin specific cell populations in serous epithelial ovarian cancer by single cell RNA-seq. *PLoS One*, 13(11):e0206785, 2018. doi:10.1371/journal.pone.0206785.

[94] Aashi Jindal, Prashant Gupta, Jayadeva, and Debarka Sengupta. Discovery of rare cells from voluminous single cell expression data. *Nat Commun*, 9(1):4719, 2018. doi:10.1038/s41467-018-07234-6.

[95] Nima Aghaeepour, Greg Finak, FlowCAP Consortium, DREAM Consortium, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nat Methods*, 10(3):228–38, 2013. doi:10.1038/nmeth.2365.

[96] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 2019. doi:10.1038/s41576-018-0088-9.

[97] Dirk Eddelbuettel. *RcppAnnoy: 'Rcpp' Bindings for 'Annoy', a Library for Approximate Nearest Neighbors*, 2018. URL https://CRAN.R-project.org/package=RcppAnnoy. R package version 0.0.11.

[98] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Wenjie Zhang, and Xuemin Lin. Approximate Nearest Neighbor Search on High Dimensional Data — Experiments, Analyses, and Improvement. *arXiv*, 2016. URL https://arxiv.org/abs/1610.02455.

[99] Yu A Malkov and D A Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *arXiv*, March 2016. URL https://arxiv.org/abs/1603.09320.

[100] Aaron Lun. *BiocNeighbors: Nearest Neighbor Detection for Bioconductor Packages*, 2018. URL https://bioconductor.org/packages/BiocNeighbors. R package version 1.0.0.

[101] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nat Methods*, 16(3): 243–245, 2019. doi:10.1038/s41592-018-0308-4.

[102] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, and Martin Hemberg. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods*, 14(5):483–486, 2017. doi:10.1038/nmeth.4236. URL https://bioconductor.org/packages/SC3.

[103] Davide Risso, Liam Purvis, Russell B Fletcher, Diya Das, John Ngai, Sandrine Dudoit, and Elizabeth Purdom. clusterExperiment and RSEC: A Bioconductor package and framework for clustering of single-cell and other large gene expression datasets. *PLoS Computational Biology*, 14(9):e1006378–16, 09 2018. URL https://bioconductor.org/packages/clusterExperiment.

[104] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods*, 14(4):414–416, 2017. doi:10.1038/nmeth.4207. URL https://bioconductor.org/packages/SIMLR.

[105] D Sculley. *Web-Scale K-means Clustering*, 2010. URL https://www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf.

[106] Davide Risso, Stephanie Hicks, and Elizabeth Purdom. *Mini-batch k-means*, 2019. URL https://github.com/drisso/mbkmeans.

[107] D T Severson, R P Owen, M J White, X Lu, and B Schuster-Böckler. BEARscc determines robustness of single-cell clusters using simulated technical replicates. *Nat Commun*, 9(1):1187, 2018. doi:10.1038/s41467-018-03608-y. URL https://bioconductor.org/packages/BEARscc.

[108] Luke Zappia and Alicia Oshlack. Clustering trees a visualisation for evaluating clusterings at multiple resolutions. *bioRxiv*, 2018. doi:10.1101/274035.

[109] Angelo Duò, Mark D Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7:1141, 2018. doi:10.12688/f1000research.15666.2. URL https://bioconductor.org/packages/DuoClustering2018.

[110] S Freytag, L Tian, I Lönnstedt, M Ng, and M Bahlo. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. *F1000Research*, 7(1297), 2018. doi:10.12688/f1000research.15809.2.

[111] Aaron T L Lun and John C Marioni. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics*, 18(3):451–464, 2017. doi:10.1093/biostatistics/kxw055.

[112] Charlotte Soneson and Mark D Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nat Methods*, 15(4):255–261, 2018. doi:10.1038/nmeth.4612.

[113] Alessandra Dal Molin, Giacomo Baruzzo, and Barbara Di Camillo. Single-Cell RNA-Sequencing: Assessment of Differential Expression Analysis Methods. *Frontiers in Genetics*, 8:r106–11, 05 2017.

[114] Maria K Jaakkola, Fatemeh Seyednasrollah, Arfa Mehmood, and Laura L Elo. Comparison of methods to detect differentially expressed genes between single-cell populations. *Brief Bioinform*, 18(5):735–743, 2017. doi:10.1093/bib/bbw057.

[115] Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*, 15(2):R29, 2014. doi:10.1186/gb-2014-15-2-r29. URL https://bioconductor.org/packages/limma.

[116] Koen Van den Berge, Fanny Perraudeau, Charlotte Soneson, Michael I Love, Davide Risso, Jean-Philippe Vert, Mark D Robinson, Sandrine Dudoit, and Lieven Clement. Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol*, 19(1):24, 2018. doi:10.1186/s13059-018-1406-4.

[117] Keegan D Korthauer, Li-Fang Chu, Michael A Newton, Yuan Li, James Thomson, Ron Stewart, and Christina Kendziorski. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol*, 17(1):222, 2016. doi:10.1186/s13059-016-1077-y. URL https://bioconductor.org/packages/scDD.

[118] Jean Fan, Hae-Ock Lee, Soohyun Lee, Da-Eun Ryu, Semin Lee, Catherine Xue, Seok Jin Kim, Kihyun Kim, Nikolaos Barkas, Peter J Park, Woong-Yang Park, and Peter V Kharchenko. Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell RNA-seq data. *Genome Res*, 28(8): 1217–1227, 2018. doi:10.1101/gr.228080.117.

[119] Jonathan A Griffiths, Antonio Scialdone, and John C Marioni. Using single-cell genomics to understand developmental processes and cell fate decisions. *Mol Syst Biol*, 14(4):e8046, 2018. doi:10.15252/msb.20178046.

[120] Robrecht Cannoodt, Wouter Saelens, and Yvan Saeys. Computational methods for trajectory inference from single-cell transcriptomics. *Eur J Immunol*, 46(11):2496–2506, 2016. doi:10.1002/eji.201646347.

[121] Tallulah S Andrews and Martin Hemberg. Identifying cell populations with scRNASeq. *Mol Aspects Med*, 59:114–122, 02 2018. doi:10.1016/j.mam.2017.07.002.

[122] Charles A Herring, Amrita Banerjee, Eliot T McKinley, Alan J Simmons, Jie Ping, Joseph T Roland, Jeffrey L Franklin, Qi Liu, Michael J Gerdes, Robert J Coffey, and Ken S Lau. Unsupervised Trajectory Analysis of Single-Cell RNA-Seq and Imaging Data Reveals Alternative Tuft Cell Origins in the Gut. *Cell Syst*, 6(1):37–51.e9, 2018. doi:10.1016/j.cels.2017.10.012.

[123] Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nat Methods*, 14(10):979–982, 2017. doi:10.1038/nmeth.4402. URL https://bioconductor.org/packages/monocle.

[124] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mRNA quantification and differential analysis with Census. *Nat Methods*, 14(3):309–315, 03 2017. doi:10.1038/nmeth.4150. URL https://bioconductor.org/packages/monocle.

[125] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods: towards more accurate and robust tools. *bioRxiv*, 2018. doi:10.1101/276907.

[126] Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(1):477, 2018. doi:10.1186/s12864-018-4772-0. URL `https://bioconductor.org/packages/slingshot`.

[127] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol*, 32(4):381–386, 2014. doi:10.1038/nbt.2859. URL `https://bioconductor.org/packages/monocle`.

[128] David A duVerle, Sohiya Yotsukura, Seitaro Nomura, Hiroyuki Aburatani, and Koji Tsuda. Cell-Tree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics*, 17(1):363, 2016. doi:10.1186/s12859-016-1175-6. URL `http://bioconductor.org/packages/cellTree`.

[129] Robrecht Cannoodt, Wouter Saelens, Dorine Sichien, Simon Tavernier, Sophie Janssens, Martin Guilliams, Bart N Lambrecht, Katleen De Preter, and Yvan Saeys. SCORPIUS improves trajectory inference and identifies novel modules in dendritic cell development. *bioRxiv*, 2016. doi:10.1101/079509.

[130] Amos Tanay and Aviv Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, 2017. doi:10.1038/nature21350.

[131] Trygve Bakken, Lindsay Cowell, Brian D. Aevermann, Mark Novotny, Rebecca Hodge, Jeremy A. Miller, Alexandra Lee, Ivan Chang, Jamison McCorrison, Bali Pulendran, Yu Qian, Nicholas J. Schork, Roger S. Lasken, Ed S. Lein, and Richard H. Scheuermann. Cell type discovery and representation in the era of high-content single cell phenotyping. *BMC Bioinformatics*, 18(Suppl 17), 2017. doi:10.1186/s12859-017-1977-1.

[132] Brian D. Aevermann, Mark Novotny, Trygve Bakken, Jeremy A. Miller, Alexander D. Diehl, David Osumi-Sutherland, Roger S. Lasken, Ed S. Lein, and Richard H. Scheuermann. Cell type discovery using single-cell transcriptomics: Implications for ontological representation. *Human Molecular Genetics*, 27 (R1):R40–R47, 2018. doi:10.1093/hmg/ddy100.

[133] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102(43):15545–50, 2005. doi:10.1073/pnas.0506580102.

[134] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res*, 45(D1):D353–D361, 2017. doi:10.1093/nar/gkw1092.

[135] Antonio Fabregat, Florian Korninger, Kerstin Hausmann, Konstantinos Sidiropoulos, Mark Williams, Phani Garapati, Steven Jupe, Henning Hermjakob, Bijay Jassal, Bruce May, Guanming Wu, Joel Weiser, Karen Rothfels, Marija Milacic, Marissa Webber, Robin Haw, Sheldon McKay, Marc Gillespie, Lincoln Stein, Lisa Matthews, Veronica Shamovsky, and Peter D'Eustachio. The Reactome pathway Knowledge-base. *Nucleic Acids Research*, 44(D1):D481–D487, 12 2015. doi:10.1093/nar/gkv1351.

[136] M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–9, 2000. doi:10.1038/75556.

[137] A Sergushichev. An algorithm for fast preranked gene set enrichment analysis using cumulative statistic calculation. *bioRxiv*, 2016. doi:10.1101/060012. URL `https://bioconductor.org/packages/fgsea`.

[138] Matthew D Young, Matthew J Wakefield, Gordon K Smyth, and Alicia Oshlack. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biol*, 11(2):R14, 2010. doi:10.1186/gb-2010-11-2-r14. URL `https://bioconductor.org/packages/goseq`.

[139] A L Tarca, S Draghici, G Bhatti, and R Romero. Down-weighting overlapping genes improves gene set analysis. *BMC Bioinformatics*, 13:136, 2012. doi:10.1186/1471-2105-13-136. URL `https://bioconductor.org/packages/PADOG`.

[140] L Geistlinger, G Csaba, and R Zimmer. Bioconductor's EnrichmentBrowser: seamless navigation through combined results of set and network-based enrichment analysis. *BMC Bioinformatics*, 17:45, 2016. doi:10.1186/s12859-016-0884-1. URL `https://bioconductor.org/packages/EnrichmentBrowser`.

[141] Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, Joost van den Oord, Zeynep Kalender Atak, Jasper Wouters, and Stein Aerts. SCENIC: single-cell regulatory network inference and clustering. *Nat Methods*, 14(11):1083–1086, 2017. doi:10.1038/nmeth.4463. URL `https://bioconductor.org/packages/AUCell`.

[142] Florian Buettner, Naruemon Pratanwanich, Davis J McCarthy, John C Marioni, and Oliver Stegle. f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol*, 18(1): 212, 2017. doi:10.1186/s13059-017-1334-8. URL `https://bioconductor.org/packages/release/bioc/html/slalom.html`.

[143] Elana J Fertig, Jie Ding, Alexander V Favorov, Giovanni Parmigiani, and Michael F Ochs. CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data. *Bioinformatics*, 26(21):2792–3, 2010. doi:10.1093/bioinformatics/btq503. URL `https://bioconductor.org/packages/CoGAPS`.

[144] Genevieve L Stein-O'Brien, Brian S Clark, Thomas Sherman, Christina Zibetti, Qiwen Hu, Rachel Sealfon, Sheng Liu, Jiang Qian, Carlo Colantuoni, Seth Blackshaw, Loyal A Goff, and Elana J Fertig. Decomposing cell identity for transfer learning across cellular measurements, platforms, tissues, and species. *bioRxiv*, 2018. doi:10.1101/395004. URL `https://bioconductor.org/packages/CoGAPS`.

[145] Sarah Williams. *celaref: Single-cell RNAseq cell cluster labelling by reference*, 2019. URL `https://bioconductor.org/packages/celaref`. R package version 1.1.1.

[146] Allen W Zhang, Ciara O'Flanagan, Elizabeth Chavez, Jamie LP Lim, Andrew McPherson, Matt Wiens, Pascale Walters, Tim Chan, Brittany Hewitson, Daniel Lai, Anja Mottok, Clementine Sarkozy, Lauren Chong, Tomohiro Aoki, Xuehai Wang, Andrew P Weng, Jessica N McAlpine, Samuel Aparicio, Christian Steidl, Kieran R Campbell, and Sohrab P Shah. Probabilistic cell type assignment of single-cell transcriptomic data reveals spatiotemporal microenvironment dynamics in human cancers: Supplementary tables. *bioRxiv*, 2019. doi:10.1101/521914. URL `https://github.com/Irrationone/cellassign`.

[147] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2018. URL `https://CRAN.R-project.org/package=shiny`. R package version 1.2.0.

[148] K Rue-Albrecht, F Marini, C Soneson, and Aaron T L Lun. iSEE: Interactive SummarizedExperiment Explorer. *F1000Research*, 7:741, 2018. doi:10.12688/f1000research.14966.1. URL `https://bioconductor.org/packages/iSEE`.

[149] Y. Xie, J.J. Allaire, and G Grolemund. *R Markdown - The Definitive Guide*. Chapman and Hall/CRC, 2018. URL `https://bookdown.org/yihui/rmarkdown`.

[150] Y Xie. *Bookdown: Authoring Books and Technical Documents with R Markdown.* Chapman and Hall/CRC, 2016. URL `https://bookdown.org/yihui/bookdown`.

[151] Andrzej Oleś, Martin Morgan, and Wolfgang Huber. *BiocStyle: Standard styles for vignettes and other Bioconductor documents*, 2018. URL `https://bioconductor.org/packages/BiocStyle`. R package version 2.11.0.

[152] Bioconductor Core Team. *Experiment Data Packages*, 2019. URL `https://www.bioconductor.org/packages/release/data/experiment`.

[153] Charlotte Soneson and Mark D Robinson. Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics*, 34(4):691–692, 2018. doi:10.1093/bioinformatics/btx631. URL `https://bioconductor.org/packages/release/bioc/html/countsimQC.html`.

[154] Solaiappan Manimaran, Heather Marie Selby, Kwame Okrah, Claire Ruberman, Jeffrey T Leek, John Quackenbush, Benjamin Haibe-Kains, Hector Corrada Bravo, and W Evan Johnson. BatchQC: interactive software for evaluating sample and batch effects in genomic data. *Bioinformatics*, 32(24):3836–3838, 2016. doi:10.1093/bioinformatics/btw538. URL `http://bioconductor.org/packages/BatchQC`.

[155] Bioconductor Package Maintainer. *ExperimentHub: Client to access ExperimentHub resources*, 2018. URL `https://bioconductor.org/packages/ExperimentHub`. R package version 1.8.0.

[156] Po-Yuan Tung, John D Blischak, Chiaowen Joyce Hsiao, David A Knowles, Jonathan E Burnett, Jonathan K Pritchard, and Yoav Gilad. Batch effects and the effective design of single-cell gene expression studies. *Sci Rep*, 7:39921, 2017. doi:10.1038/srep39921.

[157] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol*, 18(1):174, 2017. doi:10.1186/s13059-017-1305-0. URL `https://bioconductor.org/packages/splatter`.

[158] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol*, 36(5):411–420, 2018. doi:10.1038/nbt.4096. URL `https://CRAN.R-project.org/package=Seurat`.

[159] Python Core Team. *Python: A dynamic, open source programming language.* Python Software Foundation, 2019. URL `https://www.python.org/`.

[160] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol*, 19(1):15, 2018. doi:10.1186/s13059-017-1382-0. URL `https://github.com/theislab/scanpy`.

[161] Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi:10.18637/jss.v040.i08. URL `https://CRAN.R-project.org/package=Rcpp`.

[162] Dirk Eddelbuettel. *Seamless R and C++ Integration with Rcpp.* Springer, New York, 2013. doi:10.1007/978-1-4614-6868-4. URL `https://CRAN.R-project.org/package=Rcpp`.

[163] Dirk Eddelbuettel and James Joseph Balamuta. Extending extitR with extitC++: A Brief Introduction to extitRcpp. *PeerJ Preprints*, 5:e3188v1, 08 2017. doi:10.7287/peerj.preprints.3188v1. URL `https://CRAN.R-project.org/package=Rcpp`.

[164] JJ Allaire, Kevin Ushey, Yuan Tang, and Dirk Eddelbuettel. *reticulate: Interface to 'Python'*, 2018. URL `https://CRAN.R-project.org/package=reticulate`. R package version 1.10.

[165] Aaron T L Lun, Hervé Pagès, and Mike L Smith. beachmat: A Bioconductor C++ API for accessing high-throughput biological data from a variety of R matrix types. *PLoS Comput Biol*, 14(5):e1006135, 2018. doi:10.1371/journal.pcbi.1006135. URL `https://bioconductor.org/packages/beachmat`.

[166] The HDF Group. *Hierarchical Data Format, version 5*, 1997. URL `http://www.hdfgroup.org/HDF5/`.

[167] B Fischer, G Pau, and M Smith. *rhdf5: HDF5 interface to R. R package version 2.26.2e*, 2019. URL `https://github.com/grimbough/rhdf5`.

[168] Hervé Pagès. *HDF5Array: HDF5 backend for DelayedArray objects*, 2018. URL `https://bioconductor.org/packages/HDF5Array`. R package version 1.10.1.

[169] Hervé Pagès, with contributions from Peter Hickey, and Aaron Lun. *DelayedArray: Delayed operations on array-like objects*, 2019. URL `https://bioconductor.org/packages/DelayedArray`. R package version 0.8.0.

[170] Peter Hickey. *DelayedMatrixStats: Functions that Apply to Rows and Columns of 'DelayedMatrix' Objects*, 2018. URL `https://bioconductor.org/packages/DelayedMatrixStats`. R package version 1.4.0.

Supplementary Tables

Table S1: **Bioconductor software package for single-cell analyses.**

Analysis	Package name	Description
Data Input	*scPipe* [32]	Uses the *Rsubread* [33, 34] package to align sequencing reads
	tximport [38]	Wrappers for psuedoalignment algorithms such as Salmon [36] and Kallisto [37]
	DropletUtils [54]	Reading in a matrix of UMI counts from droplet-based protocols, such as 10X Genomics [35]
	beachmat [165]	A C++ interface for accessing single-cell data with sparse, dense and file-backed matrices, such as the HDF5 file format [166]
	rhdf5 [167]	Access and store data in HDF5 file formats [166]
Data Infrastructure	*HDF5Array* [168], *DelayedArray* [169], *DelayedMatrixStats* [170]	Efficiently work with data in HDF5 file formats [166] using array-like containers and delayed operations
	SingleCellExperiment [40]	Package for the *SingleCellExperiment* class that stores and accesses data from single-cell assays or experiments
Quality control, normalization, feature reduction	*DropletUtils* [54], *scater* [56], *simpleSingleCell* [59]	Perform cell and gene quality control for both read and UMI count data
	SCnorm [63], *scran* [29], *BASiCS* [64, 65], *zinbwave* [31], and *MAST* [28]	Normalizations methods to remove gene and/or cell-specific biases in scRNA-seq data
	scone [66]	Compare normalization strategies and parameterizations
	scran [59], *Oscope* [71]	Adjust for variation from differences in cell-cycle phases in data from humans and mice
	batchelor [75], *scran* [59, 74], *scMerge* [76], *scmap* [77], *scAlign* [78]	Methods for integrating two or more scRNA-seq datasets
	scran, [59], *BASiCs* [64, 65], *M3Drop* [84]	Methods for feature selection
	scater [56]	Performs and visualizes results from dimensionality reduction methods including PCA, *t*-SNE components [44], UMAP components [45, 88], and diffusion maps [89]
	BiocSingular [90]	Exact and approximate methods for singular value decomposition that uses the *BiocParallel* [91] framework to parallelize operations
	zinbwave [31]	Performs model-based dimensionality reduction based on the ZINB-WaVE model [31]
Downstream statistical analyses	*BiocNeighbors* [97–100]	Exact and approximate methods for nearest neighbor detection that uses the *BiocParallel* [91] framework to parallelize operations
	SC3 [102], *clusterExperiment* [103], *SIMLR* [104], *mbkmeans* [106], *BEARscc* [107], *clustree* [108]	Unsupervised clustering frameworks for single-cell data
	edgeR [3, 62], *DESeq2* [7], *limma* [115]	Methods developed for bulk RNA-seq differential expression that can be used in combination with methods such as *zinbwave* [31, 116] to account for the zero-inflation
	MAST [28], *scDD* [117], *BASiCS* [64, 65], *SCDE* [118]	Methods to identify differentially expressed features using statistical models that directly model zero-inflation
	slingshot [126], *TSCAN* [30], *monocle* [123, 124, 127], *cellTree* [128]	Methods for trajectory analysis or pseudotime inference
	MAST [28], *AUCell* [141], *scmap* [77], *PADOG* [139], *fgsea* [137], *goseq* [138], *slalom* [142], *scCoGAPS* [143, 144], *EnrichmentBrowser* [140]	Methods for gene set / signature enrichment analysis
	iSEE [148]	Interactive data exploration and visualization
	countsimQC [153], *batchQC* [154], *iSEE* [148]	Making analyses accessible and reproducible for producing analytical reports
	ExperimentHub [152], *DuoClustering2018* [109], *CellBench* [80]	Single-cell published data packages such as *TENxPBMCData*, *TENxBrainData*, *HCAData*, *HCABrowser* or benchmarking datasets
	splatter [157]	Packages to simulate single-cell data useful for benchmarking statistical methods