nature methods

PERSPECTIVE

https://doi.org/10.1038/s41592-019-0654-x

Corrected: Publisher Correction

Orchestrating single-cell analysis with Bioconductor

Robert A. Amezquita[1], Aaron T. L. Lun[2,16], Etienne Becht[1], Vince J. Carey[3], Lindsay N. Carpp[1], Ludwig Geistlinger[4,5], Federico Marini[6,7], Kevin Rue-Albrecht[8], Davide Risso[9,10], Charlotte Soneson[11,12], Levi Waldron[4,5], Hervé Pagès[1], Mike L. Smith[13], Wolfgang Huber[13], Martin Morgan[14], Raphael Gottardo[1]* and Stephanie C. Hicks[15]*

Recent technological advancements have enabled the profiling of a large number of genome-wide features in individual cells. However, single-cell data present unique challenges that require the development of specialized methods and software infrastructure to successfully derive biological insights. The Bioconductor project has rapidly grown to meet these demands, hosting community-developed open-source software distributed as R packages. Featuring state-of-the-art computational methods, standardized data infrastructure and interactive data visualization tools, we present an overview and online book (https://osca.bioconductor.org) of single-cell methods for prospective users.

Since 2001, the Bioconductor project[1] has attracted a rich community of developers and users from diverse scientific fields, driving the development of open-source software packages using the R language for the analysis of high-throughput biological data[2–6]. While bulk profiling technologies have yielded important scientific insights and methods[7–9], recent advancements in sequencing technologies to profile samples at single-cell resolution have emerged that can answer previously inaccessible scientific questions[10–20]. Bioconductor has been home to a wide range of software packages used in analyzing bulk profiling data, and more recently it has expanded significantly into the realm of single-cell data analysis with a rapidly growing list of community-contributed software packages (Fig. 1).

Current single-cell assays can be both high-throughput, measuring thousands to millions of cells, and high dimensional, measuring thousands of features within each individual cell. Compared to bulk assays, there are two defining characteristics of single-cell data that must be specially handled to achieve biological insight: (1) the increased scale of the number of observations (that is, cells) that are assayed in large compendiums such as those from the Human Cell Atlas[21,22] and the Mouse Cell Atlas[23]; and (2) the increased sparsity of the data due to biological fluctuations in the measured traits or limited sensitivity for quantifying small numbers of molecules[13,24–26]. These unique characteristics have motivated the development of statistical methods tailored for single-cell data analysis[27–30]. Furthermore, as single-cell technologies mature, the increasing complexity and volume of data require fundamental changes in data access, management and infrastructure alongside specialized methods to facilitate scalable analyses.

To address these challenges, software packages developed for the analysis of single-cell data have become an integral part of the Bioconductor project. Herein, we primarily focus on the analysis of single-cell RNA-seq (scRNA-seq) data, much of the concepts mentioned are also generalizable to other types of single-cell assays. We cover data import, common data containers for storing single-cell assay data, fast and robust methods for transforming raw single-cell data into processed data suitable for downstream analyses, interactive data visualization, and downstream analyses. To help users leverage this robust and scalable framework, we describe selected packages and present an online book (https://osca.bioconductor.org) covering installation, sources of help, specialized topics pertaining to specific aspects of scRNA-seq analysis and complete workflows analyzing various scRNA-seq datasets. The references for all packages are available at http://bioconductor.org/packages/.

Data infrastructure

One of Bioconductor's strongest advantages is the availability of common representations and infrastructure for complex, highly interdependent data sets[1]. Bioconductor uses standardized data containers to enable modularity and interoperability of diverse packages while maintaining robust end-user accessibility. To this end, Bioconductor employs a flexible object-oriented paradigm called S4 (ref. [31]) that enables encapsulation of multiple object components into a single instance with a rich and user-friendly interface. Such an approach is especially important for biological analysis, as there are often many links between primary data and metadata that need to be preserved throughout an analysis.

The SingleCellExperiment container. Bioconductor uses the SingleCellExperiment class for storing single-cell assay data and metadata (Fig. 2). Primary data, such as count matrices, are stored in the assays component as one or more matrices, where rows represent features (for example, genes and transcripts) and columns represent cells. In addition, low-dimensional representations of

[1]Fred Hutchinson Cancer Research Center, Seattle, WA, USA. [2]Cancer Research UK Cambridge Institute, University of Cambridge, Cambridge, UK. [3]Channing Division of Network Medicine, Brigham And Women's Hospital, Boston, MA, USA. [4]Graduate School of Public Health and Health Policy, City University of New York, New York, NY, USA. [5]Institute for Implementation Science in Population Health, City University of New York, New York, NY, USA. [6]Center for Thrombosis and Hemostasis, Mainz, Germany. [7]Institute of Medical Biostatistics, Epidemiology and Informatics, Mainz, Germany. [8]Kennedy Institute of Rheumatology, University of Oxford, Oxford, UK. [9]Department of Statistical Sciences, University of Padua, 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, Baltimore, MD, USA. [16]Present address: Bioinformatics and Computational Biology, Genentech Inc., San Francisco, CA, USA. *e-mail: rgottard@fredhutch.org; shicks19@jhu.edu

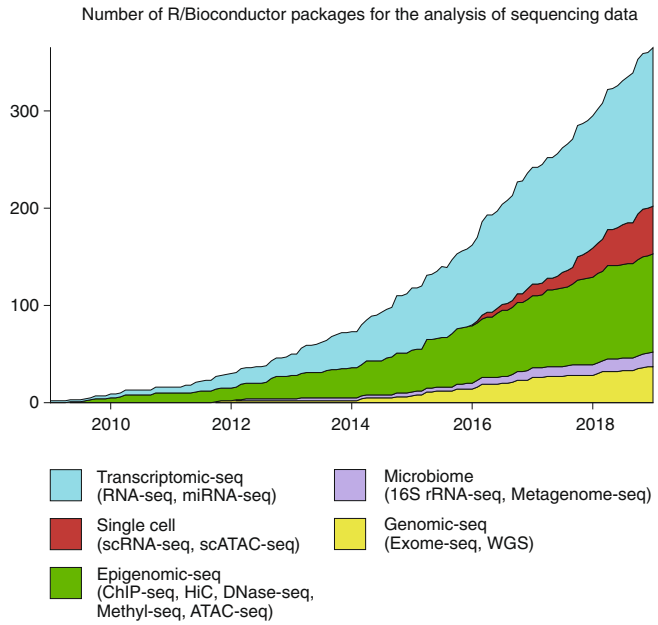

Fig. 1 | Number of Bioconductor packages for the analysis of high-throughput sequencing data over ten years. Bioconductor software packages associated with the analysis of sequencing data were tracked by date of submission over the course of ten years. Software packages were uniquely defined by their primary sequencing technology association, with examples of specific terms used for annotation in parentheses.

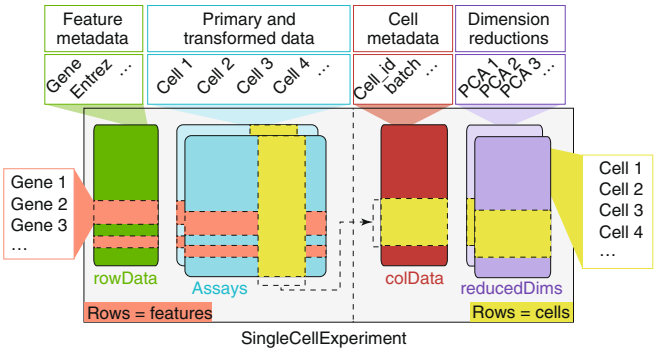

Fig. 2 | Overview of the SingleCellExperiment class. The SingleCellExperiment class instantiates an object (SingleCellExperiment, abbreviated as sce) capable of storing various datatypes generated from single-cell assays. An sce object is organized into components (for example, rowData, assays, colData, reducedDims). In the assays component, the rows represent features such as genes (horizontal pink bands), and the columns represent cells (vertical yellow band). The rowData and colData components can hold information (such as metadata) about those features and cells, respectively. Note that in the colData and reducedDims components, cells are represented as rows (horizontal yellow bands) and the number of columns in the assays component must match the number of rows in the colData and reducedDims components.

the primary data, and metadata describing cell or feature characteristics, can also be stored in the SingleCellExperiment object. Through the SingleCellExperiment class, all pertinent data and results relevant to a scRNA-seq experiment can be stored in a single instance. By standardizing the storage of single cell data and results, Bioconductor fosters interoperability between single-cell analysis packages and facilitates the development and usage of complex analysis workflows.

Data processing

The aim of this section is to describe the precursor steps that are common to most scRNA-seq analyses. These preliminary steps follow a general workflow (Fig. 3): (1) preprocessing raw sequencing data to produce a per-gene (or transcript) per-cell expression count matrix, followed by creating a SingleCellExperiment object; (2) applying quality control metrics and subsequent removal of low quality cells that would otherwise interfere with downstream analyses; (3) converting counts into normalized expression values to eliminate cell- and gene-specific biases; (4) performing feature selection to pick a subset of biologically relevant genes for downstream analyses; (5) applying dimensionality reduction methods to compact the data and reduce noise; and (6), if applicable, integrating multiples batches of scRNA-seq data.

Preprocessing. For scRNA-seq data, preprocessing involves the alignment of sequencing reads to a reference transcriptome and quantification into a per-cell and per-gene count matrix of expression values. While various preprocessing methods are available as command line software, Bioconductor packages such as scPipe[32] and scruff[33] provide a preprocessing workflow that is entirely written in R. For preprocessing workflows utilizing command line software, the DropletUtils[34] and tximeta Bioconductor packages can import the results from various tools, including Cell Ranger[35] (10X Genomics), Kallisto-Bustools[36] and Alevin[37]. Notably, pseudo-alignment methods such as Alevin and Kallisto significantly reduce compute time and memory usage.

In all the above workflows, the end result is the import of a count matrix into R and creation of a SingleCellExperiment object. For specific file formats, we can use dedicated methods from the DropletUtils (for 10X data) or tximeta (for pseudo-alignment methods) packages.

Quality control. Low-quality libraries in scRNA-seq data can arise from a variety of sources such as cell damage during dissociation or failure in library preparation (for example, inefficient reverse transcription or PCR amplification). These usually manifest as 'cells' with low total counts, few expressed genes and high mitochondrial read proportions. These low-quality libraries are problematic as they can contribute to misleading results in downstream analyses.

For droplet-based protocols, it is common to exclude data from droplets that did not contain exactly one cell. The DropletUtils[34] package distinguishes between empty—ambient RNA-containing—and cell-containing droplets, based on the frequency of each droplet barcode observed and a comparison of their respective expression profile with that of the ambient solution. It can also remove artificial cells generated by barcode swapping in droplet-based experiments[38]. Similarly, droplets that likely contain more than one cell (doublets) can be identified using the scran[28] or scds[39] packages, which compare the droplets in question against the expression profile of simulated doublets.

After excluding empty droplets and identifying potential doublets, droplets containing potentially damaged cells or exhibiting poor read coverage are filtered out. The library size—defined as the total sum of counts across all relevant features for each cell—is an oft-used metric for filtering. Cells with small library sizes are more likely to be of low quality, as the RNA has been lost at some point during library preparation, either due to cell lysis or inefficient cDNA capture and amplification. Another metric is the number of expressed features in each cell, defined as the number of endogenous genes with non-zero counts for that cell. Cells with very few expressed genes are likely to be of poor quality as the diverse transcript population has not been successfully captured. The proportion of reads mapped to genes in the mitochondrial genome

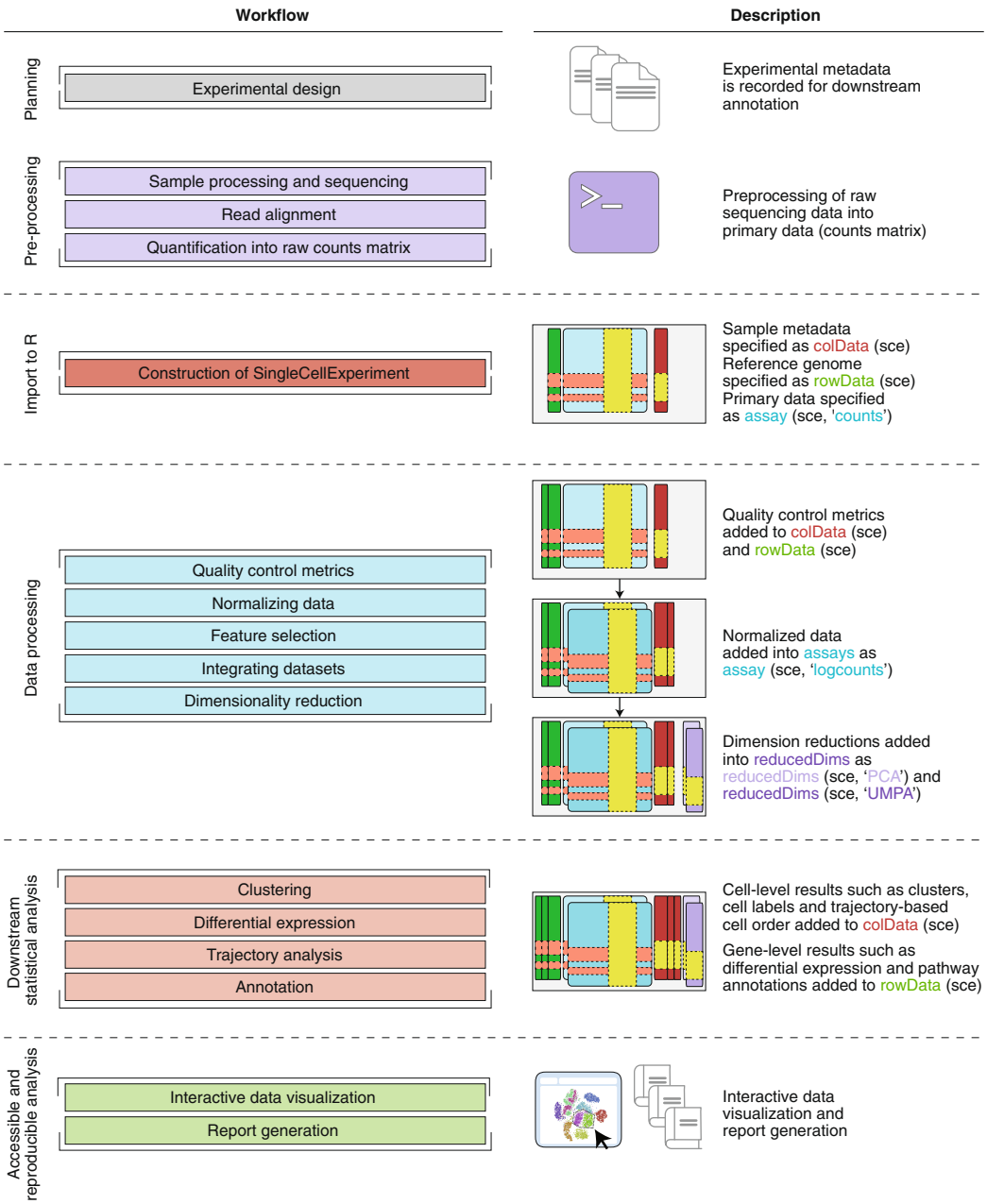

Fig. 3 | Bioconductor workflow for analyzing single-cell data. A typical analytical workflow using Bioconductor leads to the creation and evolution of a SingleCellExperiment (sce) object during data processing and downstream statistical analysis (left column). An example of an sce object evolving throughout the course of a workflow is shown, including visualization, analysis and annotation (right column).

can also be used, as high proportions indicate the possible loss of cytoplasmic RNA due to cell damage, wherein the mitochondria—being larger than individual transcript molecules—are less likely to escape through holes in the cell membrane[40]. The scater[41] package simplifies the calculation of these various metrics.

Normalization. Systematic differences in coverage between libraries are often observed in scRNA-seq data, such as differences due to sequencing depth[25,28,42]. This typically arises from differences in cDNA capture or PCR amplification efficiency across cells, attributable to the difficulty of achieving consistent library preparation with minimal starting material. Normalization aims to remove these systematic differences such that they do not interfere with comparisons of the expression profiles between cells, for example during clustering or differential expression analyses.

Here, we consider methods that moderate systematic differences within a single scRNA-seq experiment that bias all genes in a similar manner. This includes, for example, a change in sequencing depth that scales the expected coverage of all genes by a certain factor. Library size normalization is the simplest strategy for performing scaling normalization, as implemented in scater[41]. While this approach makes the assumption that there is no imbalance in the differentially expressed genes (DEGs) between any pair of cells, normalization accuracy is usually not a major consideration for exploratory scRNA-seq analysis, as there are minimal effects on cluster separation.

Accurate normalization, however, is important for procedures that involve estimation and interpretation of per-gene statistics, as in DEGs. Composition biases that systematically shift log-fold changes are most often observed when multiple cell types are

present in a given scRNA-seq dataset. Normalization by deconvolution overcomes this by pooling counts from many cells to increase the size of the counts for accurate size factor estimation, followed by deconvolution into cell-based factors for normalization per-cell, as implemented in scran[28].

Alternatively, BASiCS[43], zinbwave[30] and MAST[27] provide model-based approaches to normalization that can not only handle such library size or composition biases, but also can adjust for known covariates or other intrinsic technical factors that could conceal biologically meaningful variation[25]. These methods enable more complex scaling strategies such as non-linear transformations of the data. For reviews on this topic, see ref. [42].

Imputation. Imputation methods have been proposed to address the challenge of data sparsity in single-cell assays[44,45]. As scRNA-seq experiments frequently fail to measure expression for some genes, leading to an overabundance of zero-values[46], zero-inflated models have been developed. However, there are differences in the degree of zero-inflation depending on the type of assay or protocol[46–48], suggesting that the optimal method is assay-dependent. Furthermore, imputation methods for scRNA-seq data have been shown to generate false-positive results and decrease the reproducibility of cell-type specific markers[49].

Feature selection. Exploratory analyses of scRNA-seq data is often directed to characterize heterogeneity across cells. Procedures such as clustering and dimensionality reduction, compare cells based on their gene expression profiles. However, the choice of genes to use in these calculations has a major impact on the behavior and performance of such downstream methods. Feature selection methods aim to identify genes that contain useful information about the biology of the system while removing genes that contain random noise. By limiting analyses to such genes, interesting biological structure is preserved without the variance that obscures that structure. Furthermore, focusing on such a subset of the transcriptome can significantly reduce the size of the dataset, improving the computational efficiency of downstream analyses. See refs. [50,51] for reviews in feature selection methods.

The simplest approach to feature selection is to select the most variable genes based on their expression across the population. This assumes that genuine biological differences will manifest as increased variation in the affected genes, compared to other genes that are only affected by technical noise or a baseline level of uninteresting biological variation (for example, from transcriptional bursting). However, the log-transformation does not achieve perfect variance stabilization. This means that the variance of a gene is more affected by its abundance than the underlying biological heterogeneity. Thus, calculation of the per-gene variance for feature selection requires modelling of the mean-variance relationship. Packages such as scran[52], BASiCS[43] and scFeatureFilter adopt this approach.

Alternate metrics to variance have also been proposed, such as selecting genes based on their deviance, a metric that quantifies how well each gene fits a null model of constant expression across cells[48]. Unlike variance-based feature selection approaches, calculating the deviance is done on raw unique molecular identifier (UMI) counts, thus making the approach less sensitive to errors brought on by normalization. The deviance can be calculated using the glmpca package.

Dimensionality reduction. Dimensionality reduction aims to reduce the number of separate dimensions in the data. This is possible because different genes are correlated if they are affected by the same biological process. Thus, we do not need to store separate information for individual genes, but can instead compress multiple features into a single dimension. Dimensionality reduction approaches thus create low-dimensional representations that aim to preserve the most meaningful structures in the dataset. This has the additional benefit of reducing noise by averaging across multiple genes to obtain a more precise representation of patterns in the data (for example, related to a specific pathway). Computational work in downstream analyses is also reduced, as calculations only need to be performed for a few dimensions rather than thousands of genes. More aggressive dimensionality reduction schemes yield two- or three-dimensional representations that can be directly visualized to assist in the interpretation of the results.

A common first step to dimensionality reduction of scRNA-seq data is principal components analysis (PCA). PCA discovers axes (principal components, PCs) in high-dimensional space that capture the largest amount of variation. The top PCs capture the dominant factors of heterogeneity in the data set, and thus can be used to efficiently perform dimensionality reduction. This takes advantage of the well-studied theoretical properties of the PCA—namely, that a low-rank approximation formed from the top PCs is the optimal approximation of the original data for a given matrix rank. Given this property, calculations performed using the top PCs (or any similar low-rank approximation) takes advantage of data compression and denoising, which includes downstream analyses such as clustering.

No matter the approach, dimensionality reduction for visualization necessarily involves discarding information and distorting the distances between cells. Thus, it is ill-advised to directly analyze the low-dimensional coordinates used for plotting. Rather, these plots should only be used to interpret or communicate the results of quantitative analyses based on a more accurate, higher-rank representation of the data. This ensures that analyses make use of the information that was lost during compression into two dimensions. For example, given a discrepancy between the visible clusters on a 2-dimensional plot and those identified by clustering using the top PCs, one would be inclined to favor the latter.

The SingleCellExperiment class has a dedicated component, reducedDims, for storing lower dimensional representations of the assay data (Fig. 2). The scater[41] package provides convenience wrapper functions for dimensionality reduction algorithms, including Principal Components Analysis (PCA), *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE)[53], and Uniform Manifold Approximation and Projection (UMAP)[54]. Diffusion map methods are available via the destiny[55] package. The zinbwave[30] and glmpca[48] packages use a zero-inflated negative binomial model and a multinomial model, respectively, for model-based dimensionality reduction approaches that can account for confounding factors.

Integrating datasets. Large scRNA-seq projects usually need to generate data across multiple batches due to logistical constraints. However, the processing of different batches is often subject to uncontrollable differences, for example, changes in operator or differences in reagent quality. This results in systematic differences in the observed expression in cells from different batches. Furthermore, as the prevalence of scRNA-seq data expands and reference datasets become available, encountering such confounding variables will become inevitable in meta-analysis contexts. Such batch effects are problematic as they can be major drivers of heterogeneity in the data, masking relevant biological differences and complicating the interpretation of results.

While generalized linear modeling frameworks can be used to integrate disparate data sets[6], these frameworks may be sub-optimal in the scRNA-seq context. This is often due to the underlying assumption that the composition of cell populations is either known or identical across batches of cells. To overcome these limitations, bespoke methods have been developed for batch correction of single-cell data[56,57] that do not require a priori knowledge about the composition of the population. This enables exploratory analyses of scRNA-seq data where such knowledge is usually unavailable.

Before batch correction, it is important to examine the presence of a batch effect. This can be examined by performing PCA

on the log-expression values of select genes, followed by graph-based clustering to obtain a summary of the population structure. Ideally, clusters should consist of cells from replicate scRNA-seq datasets. However, if instead clusters are comprised of cells from a single batch, this indicates that cells of the same type are artificially separated due to technical differences. Approaches such as *t*-SNE and UMAP will also typically show a strong separation between cells from different batches that are consistent with such clustering results. Notably, such a diagnostic that relies on the degree of intermingling may not be effective when the batches involved may indeed contain unique subpopulations, but is nonetheless a useful first approximation.

Supervised integration via the labeling of cells a priori (see the section 'Annotation') can be used via packages, such as scMerge[57] and scmap[58], to guide the application of any batch correction on the gene-expression values or to adjust lower dimensional representations. On the other hand, unsupervised approaches, such as mutual nearest neighbours (MNN), identify pairs of cells from different batches that belong in each other's set of nearest neighbours. Thus, the difference between cells in MNN pairs can be used as an estimate of the batch effect, the subtraction of which yields batch-corrected values[56]. Vitally, by altering the number of *k*-nearest neighbors that are considered, the aggressiveness of the batch correction can be tuned, wherein a higher *k*-value results in more generous matching of subpopulations across batches. This MNN-based approach is implemented in the batchelor package.

The success of the batch correction is contingent on the preservation of biological heterogeneity, as one could envision a correction method of simply aggregating all cells together, which would achieve perfect mixing but also discard the biology of interest. To this end, the CellMixS package can be used to evaluate the degree of cell mixing across batches. Another useful heuristic is to compare clusters identified in the merged data against those identified per batch. Ideally, we should see a many-to-one mapping, where the across-batch clustering is nested inside the within-batch clustering, indicating that any within-batch structure was preserved post-correction. A summary statistic such as the Rand index can then be calculated, where larger Rand indices are more desirable.

Downstream statistical analysis

The choice of methods and workflows can differ greatly depending on the specific goals of the investigation and the experimental protocol used. Following data processing, Bioconductor can be used to generate new biological insights from single-cell data, using tools that are interoperable with the SingleCellExperiment class and that scale with cell number. Our online book (https://osca.bioconductor.org) provides prospective users with workflows and case studies for downstream analyses and visualizations (Fig. 4).

Clustering. Clustering is used in scRNA-seq data analysis to empirically define groups of cells with similar expression profiles. This allows us to describe population heterogeneity in terms of discrete labels that can be more easily understood, rather than attempting to comprehend the high-dimensional manifold on which the cells truly reside. After annotation based on differentially expressed marker genes, the clusters can be treated as proxies for more abstract biological concepts, such as cell types or states.

It is worth highlighting the distinction between clusters and cell types. The former is an empirical construct while the latter is a biological truth (albeit a vaguely defined one). Thus, it is helpful to realize that clustering, like a microscope, is simply a tool to explore the data. One can zoom in and out by changing the resolution of the clustering parameters, and experiment with different clustering algorithms to obtain alternative perspectives of the data.

Graph-based clustering is a flexible and scalable technique for clustering large scRNA-seq datasets. A graph is constructed where each node is a cell that is connected to its nearest neighbours (NN) in the high-dimensional space. Edges are weighted based on the similarity between the cells involved, with higher weight given to cells that are more closely related. Algorithms such as louvain and leiden[59] can then be used to identify clusters of cells.

BiocNeighbors provides an engine for both exact and approximate nearest-neighbor detection, with scran building the actual graph. Notably, for large scRNA-seq datasets, approximate NN methods trade an acceptable loss in accuracy for vastly improved run times, with the added advantage of smoothing over noise and sparsity. Alternative approaches include the SIMLR package[60], which uses multiple kernels to learn a distance metric between cells that best fits the data, and can then be used for clustering and dimension reduction. For large data, the mbkmeans package implements a scalable version of the *k*-means algorithm. Finally, the SC3[61] and clusterExperiment[62] packages calculate consensus clusters derived from multiple parameterizations.

Many of these packages allow quantitative and visual evaluation of the clustering results, alongside external packages designed solely for data visualization and evaluation (for example, clustree). Clusters can also be evaluated independently by assessing metrics such as cluster modularity or the silhouette coefficient.

Differential expression. Differential gene expression (DGE) analysis can be used to identify marker genes that drive the separation between clusters. These marker genes allow us to assign biological meaning to each cluster based on their functional annotation. In the most obvious case, the marker genes for each cluster are a priori associated with particular cell types, allowing for clustering to serve as a proxy for cell-type identity. The same principle can be applied to detect more subtle differences, such as activation status or differentiation state. An alternative to DGE analysis for cell-type annotation is gene-set enrichment analysis, which groups genes into pre-specified gene modules or biological pathways to facilitate biological interpretation. We discuss this topic in the section 'Annotation'.

DGE can also be used to compare individual cells within a given population across conditions, such as time or treatment, while adjusting for covariates (for example, patient identification or batch effects).

Across differential expression methods, two general approaches stand out. The first approach retrofits well-supported and long-standing DE analysis frameworks initially designed for bulk RNA-sequencing (edgeR (ref. [2]), DESeq2 (ref. [5]) and limma-voom (ref. [6])) that have made the transition to scRNA-seq through various approaches, such as by creating pseudo-bulk RNA-seq profiles. Alternatively, approaches such as zinbwave[30] can be used to down-weight excess zeros observed in scRNA-seq data during the dispersion estimation and model fitting steps prior to assessing differential expression (DE), and consequently further enabling the adaptation of bulk RNA-seq-based DE methods for use with scRNA-seq data[63].

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 discrete 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[27] package utilizes a hurdle model framework, whereas the scDD[64], BASiCS[43] and SCDE[14] use Bayesian mixture and hierarchical models, respectively. Together, these methods are able to provide a broader suite of testing functionality and

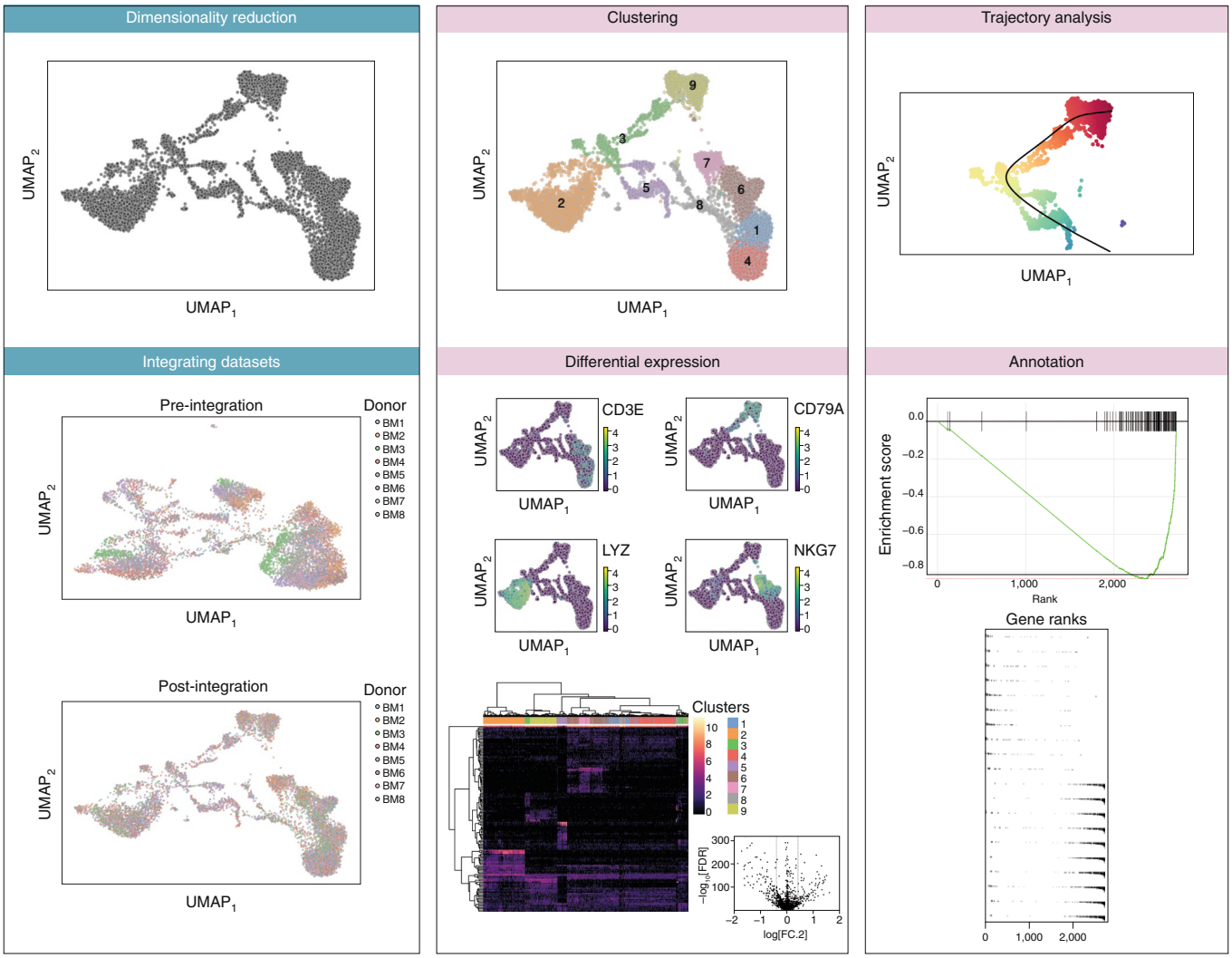

Fig. 4 | Select visualizations derived from various Bioconductor workflows. Various visualizations associated with pre-processing (blue boxes) and downstream statistical analyses (pink boxes). The example data set used throughout was generated as part of the Human Cell Atlas[21]. Details on the generation of these figures are described in our online companion book (https://osca.bioconductor.org).

can be directly utilized on scRNA-seq data contained within the SingleCellExperiment class.

For more details regarding DE analysis and the benchmarking of the various packages mentioned above, see refs. [65–67].

Trajectory analysis. Heterogeneity may also be modeled as a continuous spectrum arising from biological processes, such as cell differentiation. A specialized application of dimension-reduction specific to single-cell analysis—trajectory analysis or pseudotime inference—uses phylogenetic methods to order cells along an (often time-continuous) trajectory, such as development over time. Inferred trajectories can identify transition between cell states, a differentiation process, or events responsible for bifurcations in a dynamic cellular process[68].

Modern approaches for trajectory inference have minimized the need for extensive parameterization and can test for differential gene expression across various topologies (for example, Monocle[69], LineagePulse and switchde[70]). Moreover, several Bioconductor packages for trajectory inference (for example, slingshot[71], TSCAN[29], Monocle[69], cellTree[72] and MFA[73]) were recently demonstrated to have excellent performance[74]. As different methods can produce drastically different results for the same dataset, a suite of methods and parameterizations must be tested to assess robustness. Bioconductor facilitates such testing by providing standardized data representation, such as the SingleCellExperiment class objects. See ref. [74] for further discussion.

Annotation

The most challenging task in scRNA-seq data analysis is arguably the interpretation of the results. Obtaining clusters of cells is fairly straightforward, but it is more difficult to determine what biological state is represented by each of those clusters. Doing so requires bridging the gap between the current dataset and prior biological knowledge, and the latter is not always available in a consistent and quantitative manner. As such, interpretation of scRNA-seq data is often manual and is a common bottleneck in the analysis workflow.

To expedite this step, various computational approaches can be applied that exploit prior information to assign meaning to an uncharacterized scRNA-seq dataset. The most obvious sources of prior information are curated gene sets associated with particular biological processes (for example, from the Gene Ontology (GO) or the Kyoto Encyclopedia of Genes and Genomes (KEGG) collections).

An alternative approach involves directly comparing expression profiles to published reference datasets where each sample or cell has already been annotated with its putative biological state by domain experts.

Gene-set enrichment. Classical gene-set enrichment (GSE) approaches have the advantage of not requiring reference expression values. This is particularly useful when dealing with gene sets derived from the literature or other qualitative forms of biological knowledge. In the context of cell annotation, GSE is typically performed on a group of cells (or cluster) to identify the gene set (or pathway) that is enriched in these cells. The enriched pathway can then be used to deduce a cell type (or state).

Bioconductor provides dedicated packages to programmatically access predefined gene signatures from databases such as MSigDB[75], KEGG[76], Reactome[77] and Gene Ontology (GO)[78]. EnrichmentBrowser[79] simplifies the compilation of gene-set collections from such repositories. This prior knowledge is used to test for the enrichment of specific gene modules in scRNA-seq data, often adapting existing gene-set analysis methods originally developed for bulk data. The EnrichmentBrowser[79], EGSEA[80] and fgsea packages each provide some version of classical GSE analysis. Alternative approaches to testing for GSE are implemented in MAST[27], AUCell[81] and slalom[82].

Automated classification of cells. A conceptually straightforward annotation approach is to compare the single-cell expression profiles with previously annotated reference datasets. Labels can then be assigned to each cell in an uncharacterized dataset based on the most similar reference sample(s) or on some other similarity metric. This is a common classification challenge that can be tackled by standard machine-learning techniques, such as random forests and support vector machines. Any published and labelled RNA-seq dataset (bulk or single-cell) can be used as a reference, though its reliability depends greatly on the domain expertise of the original authors who assigned the labels in the first place.

The SingleR method[83] provides one such automated system for cell type annotation assignment. SingleR labels cells based on the reference samples with the highest Spearman rank correlations, and thus can be considered a rank-based variant of *k*-nearest-neighbor classification. To reduce noise, SingleR identifies marker genes between pairs of labels and computes the correlation using only those markers. A number of built-in reference datasets are included with the package that are derived from a variety of sources and tissues, including Immunological Genome project (ImmGen), ENCODE and the Database for Immune Cell Expression (DICE).

Accessible analysis

With the increased interest in data from single-cell assays, Bioconductor has developed not only the methods and software to analyze the data, but also has prioritized making the data itself and the data analysis tools more easily accessible to both users and developers. Specifically, the community has contributed data packages, containing both publicly available published data and simulated data, and interactive data visualization tools. Making single-cell data and data analysis tools more accessible allows researchers to leverage these resources in their own work and democratizes data analysis.

Benchmarking. As new single-cell assays, statistical methods and corresponding software are developed, it is increasingly important to facilitate the publication of data sets, to reproduce existing analyses as well as to enable comparisons across new and existing tools. Bioconductor houses a collection of data packages focused on providing accessible and well-annotated versions of data ready for analysis, alongside vignettes that can be used to reproduce manuscript figures and showcase data characteristics.

To facilitate querying of published data packages on Bioconductor, the ExperimentHub package enables programmatic access of published data sets using a standardized interface. Of note, the scRNAseq package provides direct access to a curated selection of high-quality scRNA-seq data from various contexts. In addition, simulated data are useful for benchmarking methods.

Alternately, the splatter package[84] can simulate scRNA-seq data that contains multiple cell types, batch effects, varying levels of dropout events, differential gene expression and trajectories. The splatter package uses both its own simulation framework and wraps around other simulation frameworks with differing generative models to provide a comprehensive resource for single-cell data simulation.

To promote the reproducibility of benchmark comparisons assessing the performance of single-cell methods, software packages have been developed that provide infrastructure to compute and store the results of applying different methods to a data set. The SummarizedBenchmark[85] and CellBench[86] packages provide interfaces for which to store metadata (method parameters and package versions) and evaluation metrics.

Interactive data visualization. The maturation of web technologies has opened new avenues for interactive data exploration, aided by shiny, an R package facilitating development of rich graphical user interfaces. The iSEE[87] and singleCellTK packages provide full-featured applications for interactive visualization of scRNA-seq datasets through an internet browser, eliminating the need for programming experience if the instance is hosted on the web. Both packages directly interface with the SingleCellExperiment data container to enable scRNA-seq analysis results.

Outlook

Since the early days of genomics, the Bioconductor project has embraced the development of open-source and open-development software through the R statistical programming language. Bioconductor has established best practices for coordinated package versioning and code review. Alongside community-contributed packages, a core developer team (https://www.bioconductor.org/about/core-team) implements and maintains the essential infrastructure, and reviews contributed packages to ensure they satisfy a set of guidelines to guarantee interoperability across packages. These packages are organized into BiocViews, an ontology of topics that classify packages by task or technology. For example, topics in single-cell analysis are labeled under the view SingleCell. Most importantly, the broader Bioconductor community—accessible through various means, including forums, Slack or mailing lists—is a model of altruism in code sharing and technical help. Together, these practices produce high-quality, well maintained packages, contributing to a unified and stable environment for biological research.

Most recently, the Bioconductor community has developed state-of-the-art computational methods, infrastructure and interactive data visualization tools available as software packages for the analysis of data derived from single-cell experiments. Emerging single-cell technologies in epigenomics, T cell and B cell repertoires, spatial profiling, and sequencing-based protein profiling[88–95], promise to continue driving advances in computational biology. In particular, technologies enabling multimodal profiling are rapidly developing, and Bioconductor has laid the groundwork necessary to support statistical methodologies that fully leverage such approaches.

In addition, Bioconductor's standardized data containers enable interoperability within and between Bioconductor packages as well as other software. Analysis stored in a SingleCellExperiment can be converted to formats usable with Seurat[96], Monocle[69] and Python's scanpy[97], enabling the use of tools that best serve the objective at hand. Indeed, R has a long history of interoperability with other programming languages. Four examples are the Rcpp[98] package for integrating C++ compiled code into R, the rJava package to call Java code from within R, the.Fortran() function in base R to call Fortran code, and the reticulate CRAN 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 data infrastructure (Fig. 2), methods and software, and organized the packages along a typical workflow (Fig. 3) for the most common single-cell analyses (Fig. 4). Finally, we have developed an online companion book that provides more details on focused topics as well as complete coding workflows (https://osca.bioconductor.org). This effort will be continuously updated and maintained with new packages as they emerge, which increases discoverability of Bioconductor resources.

Received: 26 March 2019; Accepted: 14 October 2019;
Published online: 02 December 2019

References

1. Huber, W. et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods* **12**, 115–121 (2015).
2. Robinson, M. D. et al. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).
3. Lawrence, M. et al. Software for computing and annotating genomic ranges. *PLoS Comput. Biol.* **9**, e1003118 (2013).
4. Aryee, M. J. et al. Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics* **30**, 1363–1369 (2014).
5. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
6. Ritchie, M. E. et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **43**, e47 (2015).
7. Serratì, S. et al. Next-generation sequencing: advances and applications in cancer diagnosis. *Onco. Targets Ther.* **9**, 7355–7365 (2016).
8. Nakato, R. & Shirahige, K. Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation. *Brief. Bioinform.* **18**, 279–290 (2017).
9. Kukurba, K. R. & Montgomery, S. B. RNA sequencing and analysis. *Cold Spring Harb. Protoc.* **2015**, 951–969 (2015).
10. Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C. & Teichmann, S. A. The technology and biology of single-cell RNA sequencing. *Mol. Cell* **58**, 610–620 (2015).
11. Patel, A. P. et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344**, 1396–401 (2014).
12. Tirosh., I. et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
13. Karaayvaz, M. et al. Unravelling subclonal heterogeneity and aggressive disease states in TNBC through single-cell RNA-seq. *Nat. Commun.* **9**, 3588 (2018).
14. Jean Fan. et al. Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell RNA-seq data. *Genome Res.* **28**, 1217–1227 (2018).
15. Levitin, H. M., Yuan, J. & Sims, P. A. Single-cell transcriptomic analysis of tumor heterogeneity. *Trends Cancer* **4**, 264–268 (2018).
16. Paulson, K. G. et al. Acquired cancer resistance to combination immunotherapy from transcriptional loss of class I HLA. *Nat. Commun.* **9**, 3868 (2018).
17. Zeisel, A. et al. Brain structure: cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).
18. Deng, Q., Ramsköld, D., Reinius, B. & Sandberg, R. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343**, 193–196 (2014).
19. Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.* **20**, 273–282 (2019).
20. Cannoodt, R., Saelens, W. & Saeys, Y. Computational methods for trajectory inference from single-cell transcriptomics. *Eur. J. Immunol.* **46**, 2496–2506 (2016).
21. Regev, A. et al. The Human cell atlas. *eLife* **6**, e27041 (2017).
22. Rozenblatt-Rosen, O., Stubbington, M. J. T., Regev, A. & Teichmann, S. A. The human cell atlas: from vision to reality. *Nature* **550**, 451–453 (2017).
23. Han, X. et al. Mapping the mouse cell atlas by microwell-seq. *Cell* **173**, 1307 (2018).
24. McDavid, A. et al. Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments. *Bioinformatics* **29**, 461–467 (2013).
25. Hicks, S. C., Townes, F. W., Teng, M. & Irizarry, R. A. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics* **19**, 562–578 (2018).
26. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat. Methods* **11**, 740–742 (2014).
27. Finak, G. et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16**, 278 (2015).
28. Lun, A. T. L., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).
29. Ji, Z. & Ji, H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res.* **44**, e117 (2016).
30. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.* **9**, 284 (2018).
31. Chambers, J. M. Object-oriented programming, functional programming and R. *Stat. Sci.* **29**, 167–180 (2014).
32. Tian, L. et al. scPipe: a flexible R/Bioconductor preprocessing pipeline for single-cell RNA-sequencing data. *PLoS Comput. Biol.* **14**, e1006361 (2018).
33. Wang, Z., Hu, J., Johnson, W. E. & Campbell, J. D. scruff: an R/Bioconductor package for preprocessing single-cell RNA-sequencing data. *BMC Bioinform.* **20**, 222 (2019).
34. Lun, AaronT. L. et al. Emptydrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol.* **20**, 63 (2019).
35. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
36. Melsted, P. et al. Modular and efficient pre-processing of single-cell rna-seq. Preprint at *bioRxiv* https://doi.org/10.1101/673285 (2019).
37. Srivastava, A., Malik, L., Smith, T., Sudbery, I. & Patro, R. Alevin efficiently estimates accurate gene abundances from dscRNA-seq data. *Genome Biol.* **20**, 65 (2019).
38. Griffiths, J. A., Richard, A. C., Bach, K., Lun, A. T. L. & Marioni, J. C. Detection and removal of barcode swapping in single-cell RNA-seq data. *Nat. Commun.* **9**, 2667 (2018).
39. Bais, A. S. & Kostka, D. scds: computational annotation of doublets in single cell RNA sequencing data. *Bioinformatics* https://doi.org/10.1093/bioinformatics/btz698 (2019).
40. Ilicic, T. et al. Classification of low quality cells from single-cell RNA-seq data. *Genome Biol.* **17**, 29 (2016).
41. McCarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33**, 1179–1186 (2017).
42. Vallejos, C. A., Risso, D. R., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods* **14**, 565–571 (2017).
43. Vallejos, C. A., Richardson, S. & Marioni, J. C. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biol.* **17**, 70 (2016).
44. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
45. Li, W. V. & Li, J. L. An accurate and robust imputation method scImpute for singlecell RNA-seq data. *Nat. Commun.* **9**, 997 (2018).
46. Svensson, V. Droplet scRNA-seq is not zero-inflated. Preprint *bioRxiv* https://doi.org/10.1101/582064 (2019).
47. Vieth, B., Ziegenhain, C., Parekh, S., Enard, W. & Hellmann, I. powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics* **33**, 3486–3488 (2017).
48. Townes, F. W., Hicks, S. C., Aryee, M. J. & Irizarry, R. A. Feature selection and dimension reduction for single cell RNA-seq based on a multinomial model. Preprint at *bioRxiv* https://doi.org/10.1101/574574 (2019).
49. Andrews, T. & Hemberg, M. False signals induced by single-cell imputation. *F1000Res.* https://doi.org/10.12688/f1000research.16613.2 (2019).
50. Andrews, T. & Hemberg, M. M3Drop: Dropout-based feature selection for scRNASeq. *Bioinformatics* **35**, 2865–2867 (2019).
51. Yip, S. H., Sham, P. C. & Wang, J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief. Bioinform.* **20**, 1583–1589 (2018).
52. Lun, A. T. L., McCarthy, D. J. & Marioni, J. C. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Res.* **5**, 2122 (2016).
53. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
54. Melville, J., McInnes, L. & Healy, J. UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at *arXiv* https://arxiv.org/abs/1802.03426 (2018).
55. Angerer., P. et al. Destiny: diffusion maps for large-scale single-cell data in R. *Bioinformatics* **32**, 1241–1243 (2016).

56. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
57. Lin, Y. et al. scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets. *Proc. Natl. Acad. Sci. USA* **116**, 9775–9784 (2019).
58. Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods* **15**, 359–362 (2018).
59. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).
60. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Methods* **14**, 414–416 (2017).
61. Kiselev, V. Y. et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods* **14**, 483–486 (2017).
62. Risso, D. et al. clusterExperiment and RSEC: a bioconductor package and framework for clustering of singlecell and other large gene expression datasets. *PLoS Comp. Biol.* **14**, e1006378–16 (2018).
63. Van den Berge, K. et al. Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol.* **19**, 24 (2018).
64. Korthauer, K. D. et al. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol.* **17**, 222 (2016).
65. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods* **15**, 255–261 (2018).
66. Wang, T., Li, B., Nelson, C. E. & Nabavi, S. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinform.* **20**, 40 (2019).
67. Crowell, H. L. et al. On the discovery of population-specific state transitions from multi-sample multi-condition single-cell RNA sequencing data. Preprint at *bioRxiv* https://doi.org/10.1101/713412 (2019).
68. Andrews, T. S. & Hemberg, M. Identifying cell populations with scRNASeq. *Mol. Asp. Med.* **59**, 114–122 (2018).
69. Qiu, X. et al. Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14**, 979–982 (2017).
70. Campbell, K. R. & Yau, C. switchde: inference of switch-like differential expression along single-cell trajectories. *Bioinformatics* **33**, 1241–1242 (2017).
71. Street, K. et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19**, 477 (2018).
72. duVerle, D. A., Yotsukura, S., Nomura, S., Aburatani, H. & Tsuda, K. CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinform.* **17**, 363 (2016).
73. Campbell, K. R. & Yau, C. Probabilistic modeling of bifurcations in single-cell gene expression data using a bayesian mixture of factor analyzers. *Wellcome Open Res.* **2**, 19 (2017).
74. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547 (2019).
75. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* **102**, 15545–15550 (2005).
76. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**, 353–361 (2017).
77. Fabregat, A. et al. The reactome pathway knowledgebase. *Nucleic Acids Res.* **44**, 481–487 (2015).
78. Ashburner, M. et al. Gene ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29 (2000).
79. Geistlinger, L., Csaba, G. & Zimmer, R. Bioconductor's EnrichmentBrowser: seamless navigation through combined results of set and network-based enrichment analysis. *BMC Bioinform.* **17**, 45 (2016).
80. Alhamdoosh, M. et al. Combining multiple tools outperforms individual methods in gene set enrichment analyses. *Bioinformatics* **33**, 414–424 (2017).
81. Aibar, S. et al. SCENIC: single-cell regulatory network inference and clustering. *Nat. Methods* **14**, 1083–1086 (2017).
82. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. fscLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.* **18**, 212 (2017).
83. Aran, D. et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.* **20**, 163–172 (2019).
84. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017).
85. Kimes, P. K. & Reyes, A. Reproducible and replicable comparisons using SummarizedBenchmark. *Bioinformatics* **35**, 137–139 (2019).
86. Tian, L. et al. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat. Methods* **16**, 479–487 (2019).
87. Rue-Albrecht, K., Marini, F., Soneson, C. & Lun, A. T. L. iSEE: interactive SummarizedExperiment Explorer. *F1000Res.* **7**, 741 (2018).
88. Peterson, V. M. et al. Multiplexed quantification of proteins and transcripts in single cells. *Nat. Biotechnol.* **35**, 936–939 (2017).
89. Dey, S. S., Kester, L., Spanjaard, B., Bienko, M. & van Oudenaarden, A. Integrated genome and transcriptome sequencing of the same cell. *Nat. Biotechnol.* **33**, 285–289 (2015).
90. Macaulay, IainC. et al. Separation and parallel sequencing of the genomes and transcriptomes of single cells using GT-seq. *Nat. Protoc.* **11**, 2081–2103 (2016).
91. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
92. Shahi, P., Kim, S. C., Haliburton, J. R., Gartner, Z. J. & Abate, A. R. Abseq: ultrahighthroughput single cell protein profiling with droplet microfluidic barcoding. *Sci. Rep.* **7**, 44447 (2017).
93. Angermueller, C. et al. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nat. Methods* **13**, 229–232 (2016).
94. Cao, J. et al. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science* **361**, 1380–1385 (2018).
95. Clark, S. J. et al. scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. *Nat. Commun.* **9**, 781 (2018).
96. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).
97. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
98. Eddelbuettel, D. & François, R. Rcpp: seamless R and C++ integration. *J. Stat. Softw.* **40**, 1–18 (2011).

Acknowledgements

Bioconductor is supported by the National Human Genome Research Institute (NHGRI) and National Cancer Institute (NCI) of the National Institutes of Health (NIH) (grant no. U41HG004059, U24CA180996), the European Union (EU) H2020 Personalizing Health and Care Program Action (contract number 633974) and the SOUND Consortium. In addition, M.M., S.C.H., R.G., W.H., A.T.L.L. and D.R. are supported by the Chan Zuckerberg Initiative (CZI) DAF (grant no. 2018-183201, 2018-183560), an advised fund of Silicon Valley Community Foundation. D.R., W.H., M.M. and S.C.H. are supported by 2019-002443 from the CZI. S.C.H. is supported by the NIH/NHGRI (grant no. R00HG009007). R.A.A. and R.G. are supported by the Integrated Immunotherapy Research Center at Fred Hutch. M.M. is supported by the NCI/NHGRI (grant no. U24CA232979). L.G. is supported by a research fellowship from the German Research Foundation (grant no. GE3023/1-1). L.W. and V.J.C. are supported by the NCI (grant no. U24CA18099). V.J.C. is additionally supported by NCI U01 CA214846 and Chan Zuckerberg Initiative DAF (grant no. 2018-183436). ATLL received support from CRUK (grant no. A17179) and the Wellcome Trust (grant no. WT/108437/Z/15). F.M. is supported by the German Federal Ministry of Education and Research (grant no. BMBF 01EO1003). M.L.S. is supported by the German Network for Bioinformatics Infrastructure (grant no. 031A537B). D.R. is supported by the Programma per Giovani Ricercatori Rita Levi Montalcini from the Italian Ministry of Education, University and Research. H.P. is supported by the NIH Bioconductor grant (no. U41HG004059).

Author contributions

E.B., V.J.C., L.N.C., L.G., F.M., K.R., D.R., C.S. and L.W. contributed equally to this work. S.C.H. and R.G. contributed equally to the supervision of this work. S.C.H. and R.G. conceptualized the manuscript. R.A.A., A.T.L.L., S.C.H. and R.G. wrote the manuscript with contributions and input from all authors. All authors read and approved the final manuscript.

Competing interests

R.G. declares ownership in CellSpace Biosciences.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0654-x.

Correspondence should be addressed to R.G. or S.C.H.

Peer review information Lei Tang was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

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