Chapter 19

Single-Cell RNA Sequencing Analysis: A Step-by-Step Overview

Shaked Slovin, Annamaria Carissimo, Francesco Panariello, Antonio Grimaldi, Valentina Bouché, Gennaro Gambardella, and Davide Cacchiarelli

Abstract

Thanks to innovative sample-preparation and sequencing technologies, gene expression in individual cells can now be measured for thousands of cells in a single experiment. Since its introduction, single-cell RNA sequencing (scRNA-seq) approaches have revolutionized the genomics field as they created unprecedented opportunities for resolving cell heterogeneity by exploring gene expression profiles at a single-cell resolution. However, the rapidly evolving field of scRNA-seq invoked the emergence of various analytics approaches aimed to maximize the full potential of this novel strategy. Unlike population-based RNA sequencing approaches, scRNA seq necessitates comprehensive computational tools to address high data complexity and keep up with the emerging single-cell associated challenges. Despite the vast number of analytical methods, a universal standardization is lacking. While this reflects the fields' immaturity, it may also encumber a newcomer to blend in.

In this review, we aim to bridge over the abovementioned hurdle and propose four ready-to-use pipelines for scRNA-seq analysis easily accessible by a newcomer, that could fit various biological data types. Here we provide an overview of the currently available single-cell technologies for cell isolation and library preparation and a step by step guide that covers the entire canonical analytic workflow to analyse scRNA-seq data including read mapping, quality controls, gene expression quantification, normalization, feature selection, dimensionality reduction, and cell clustering useful for trajectory inference and differential expression. Such workflow guidelines will escort novices as well as expert users in the analysis of complex scRNA-seq datasets, thus further expanding the research potential of single-cell approaches in basic science, and envisaging its future implementation as best practice in the field.

Key words Single-cell RNA-seq, Experimental workflow, Data analysis tutorial, Computational pipelines, Clustering, Monocle, Seurat, gf-icf, Scanpy

Shaked Slovin, Annamaria Carissimo and Francesco Panariello contributed equally with all other contributors.

Ernesto Picardi (ed.), *RNA Bioinformatics*, Methods in Molecular Biology, vol. 2284, https://doi.org/10.1007/978-1-0716-1307-8_19,

1 Introduction

Throughout the last decade, population-based RNA sequencing approaches (aka bulk RNA-seq) have played a significant role in deciphering genome-wide transcriptome variations across a broad range of fields, including cancer biology, developmental biology, and cellular homeostasis [1–3]. However, as bulk RNA-seq data represents an average of gene expression across individual cells, it may mask the transcriptional trends of distinct subpopulations with the most abundant cell types or states (Simpson's paradox [4]).

Single-cell RNA sequencing (scRNA-seq) bridged over this hurdle, providing unprecedented opportunities for exploring gene expression profiles at a single-cell resolution. Since its first introduction in 2009 [5, 6], scRNA-seq opened a new avenue to uncover the underlying cellular heterogeneity of composite systems. However, the practical procedures were arduous, time-consuming, cost-intensive, and heavily relied on a single-sourced set of equipment. At present, with the emergence of efficient and low-cost technologies (Table 1 [7]), a typical lab bench suffices for building sequencing libraries amounting to thousands of cells [8–10], thus encouraging the use of single-cell technology as a standard procedure.

These technical advancements enabled the discovery of novel cell types [11, 12] and the study of cellular dynamic processes at a previously unattainable spatial and temporal resolution [13–16], featuring in-cell variation such as gene interaction, allelic expression, and novel RNA processing in the field of molecular cell biology [17, 18]. Moreover, scRNA-seq became a key ingredient in the rapidly evolving field of precision medicine [19, 20]. The profound amount of new information obtained with scRNA-seq holds the potential of reshaping our understanding of developmental biology, gene regulation, and cell heterogeneity in health and disease.

2 The Laboratory Workflow of scRNA-seq

At present, all scRNA-seq laboratory methods rely on six main steps (Fig. 1): (*I*) preparation of a viable single-cell suspension, (*II*) assessment of cell viability, (*III*) lysed cell removal, (*IV*) individual transcriptome barcoding, (*V*) cDNA generation, and (*VI*) sequencing library generation [21]. As for instrument implementation, one of the most popular sequencing platforms is the Illumina® series due to its cost-effectiveness and high-quality outputs. A relatively new introduction in the field is the BGI sequencing portfolio, which allows equipotential sequencing results even in single-cell studies [22].

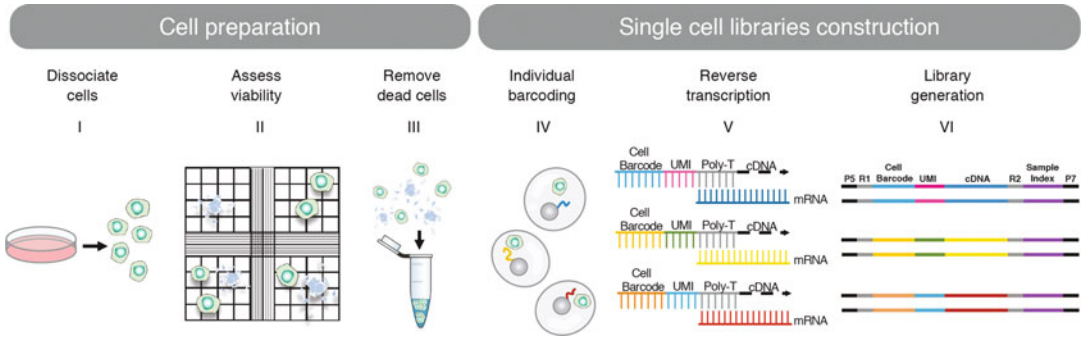

Fig. 1 Single-cell RNA sequencing workflow. The scRNA-seq procedure consists of six key steps. (*I*) Samples are dissociated into a single-cell suspension. (*II*) As lysed cells might bias the data and cause high noise interference, it is essential to maximize the quality of the input material and assess cell viability. (*III*) If the viability is lower than 90%, dead cells should be filtered either by centrifugation (i.e., density gradient) or immunodepletion (*i.e* FACS or magnetic sorting). (*IV*) Single cells are captured and isolated in different ways, depending on the technique of choice. Microfluidics-based scRNA-seq technologies encapsulate single cells within water-in-oil droplets together with unique primers attached to microparticle surface and lysis buffer. Then, each lysed cell's mRNA content is captured by the poly-A tail domain of a single primer and labeled with UMI and cell-specific barcodes. Several errors can occur during this step, like multiple cells or microparticles captured in a single droplet (i.e., multiplets), and sub-Poisson loading trade-offs, such as empty barcoded drops. (*V*) Captured mRNA transcripts from droplets are then collected, reverse-transcribed, and (*VI*) amplified in pools to be used for standard sequencing platforms. During library construction, cDNA molecules are tagged with sample-specific indexes allowing multiplexing of different captures in the same sequencing run. Further computational demultiplexing will use such barcode information to sort samples, cells, and transcripts

Barcoding the transcriptome of individual cells is a key step in all available single-cell protocols, and exemplifies the main difference from bulk RNA-seq. Two barcoding strategies are suggested, either (1) the addition of a cell-specific barcode to each transcriptome following cell isolation, or alternatively (2) the addition of a unique index combination to each cell transcriptome without physical partitioning (e.g., split-Seq [23]). Both strategies can be further classified into subcategories with different advantages and drawbacks (Table 1, Supplementary). However, all scRNA-seq strategies rely on high-quality input material, requiring the optimization of any dissociation and thawing protocol to maximize cell viability [24–26].

Among the more recent advancements in the field, microfluidic-based scRNA-seq technologies have gained popularity due to their cost-effectiveness, high efficiency, and moderate data size requirements for preserving data integrity and coherence [27, 28]. Generally, microfluidic technologies, such as Chromium [10], inDrop [9], and Drop-seq [8], rely on passive coflow of cells, microparticles (i.e., beads) and a lysis buffer that produces water-in-oil droplets, thus encapsulating precisely one cell and one bead. The transcriptional content of each cell is captured and amplified by unique primers attached to the surface of a single microparticle.

Such primers share the same underlying three-tuple structure, including (1) a cellular barcode, a short sequence common to all primers on a single microparticle, with the purpose of identifying all transcripts belonging to the same cell; (*2*) a unique molecular identifier (UMI), a molecular transcript-specific tag which secures read's integrity by identifying PCR duplicates [29]; (iii) a poly-T tail, for the capture and amplification of the 3′END of each transcript.

Ideally, each droplet should encapsulate a single cell and a single bead. However, as in practice the encapsulation step follows a Poisson distribution, the capture rate of one bead and one cell within a single droplet follows a double Poisson distribution. Ergo, many droplet-based approaches yielded large numbers of empty droplets and inefficient data assemblage.

The limitation of Poisson statistics has been tackled by inDrop and Chromium technologies. Through close-packed ordering of deformable particles, both methods instrument a sub-Poisson distribution [30], thus achieving controllable encapsulated particle quantities, with a single bead occupancy of about 80%. Hence, the main differences among the three platforms, inDrop, Chromium, and Dop-seq, is their respective capture efficiency, largely dependent on beads types at use [31]. While Drop-seq, inDrops, and Chromium capture about 5–12%, 75%, and 65% of the input cells, they also require $>2 \times 10^5$, 2×10^3–10^4, and $>10^3$ input cells, respectively.

Hence, choosing the appropriate technique is crucial, and pends on a particular field of study and research requirements. When investigating highly heterogeneous samples, like tumors and tissues, high-throughput methods are advisable. Nevertheless, high-sensitivity strategies are best suited either when analyzing low expressed genes or classifying rare cell populations [32].

However, if on one side scRNA-seq allows to dissect cellular heterogeneity at high-resolution, it also carries two key drawbacks. The first one is a low gene retrieval yield, with usually a 1–5% of transcripts per cell representing highly expressed genes (about 5000 genes per cell), thus leading to significant observational uncertainty. This drop-out effect introduces a high cell-to-cell variability and low signal-to-noise ratio (SNR) [33]. A further drawback is evidently the cost burden of scRNA-seq commercial technologies, while noncommercial platforms (inDrop, Drop-seq) require considerable operator expertise. Consequently, the implementation of scRNA-seq techniques is still not broadly accessible for many laboratories in the field [34, 35]. However, these obstacles shall not hold the promise of scRNA-seq to expand beyond the genomic research frontier, as overcoming the current challenges will widen the future outlooks for medicine and biological studies.

3 The Computational Workflow of scRNA-seq

Unlike previous genome-wide transcriptomic assays, scRNA-seq necessitates innovative analysis tools to address the emerging single-cell associated challenges, including large-scale data and high levels of noise interference due to dropout events [33, 35]. Indeed, more than 600 standalone tools are available to analyze and explore single-cell transcriptomic data [36], but with the lack of universal standardization [37, 38], a newcomer to blend in.

The challenge in achieving standardized pipelines stems from several reasons, including the relative immaturity of the field. Depending on the platform of choice, individual procedural steps may be processed differently, resulting in inconsistent downstream analysis outputs for the same entry dataset [6]. In addition, the choice of a specific analytic tool is largely swayed by a programming language preference such as R or Python, and thus restricting their usage to a narrower audience specialized in a specific programming language. A further, and probably the most significant hurdle, is the need to find a common analytic strategy that could fit various biological data types (cell lines, cancer cells, stem cells, etc.). However, due to their high diversity and distinct biological inquiries at hand, ad hoc computational strategies might be needed.

In this review we aim to address all the above-mentioned challenges, outlining a standardized workflow that will guide the reader through the key steps of scRNA-seq data analysis, regardless of specific tools and different biological data types. Herein, we propose four ready-to-use computational pipelines, which include raw counts normalization, feature selection, dimensionality reduction, and clustering (Fig. 2). Completing these steps enables the users to analyze their respective data without any loss of information. The proposed pipelines cover both R and Python programming languages, and employ Seurat (R) [39], Scanpy (Python) [40], Monocle (R) [41, 42], and gf-icf (R) [43] platforms, which are all easily accessible for a newcomer.

A case study employing the four proposed pipelines is demonstrated using a subset of Tabula Muris [44] public dataset retrieved by the Chromium technology, outlining the different steps with plots and command lines, all available on github [45]. As the proposed pipelines might be permissive or too restrictive for a given assay, we offer guidelines for tailoring the analytic settings to meet user's data requisites.

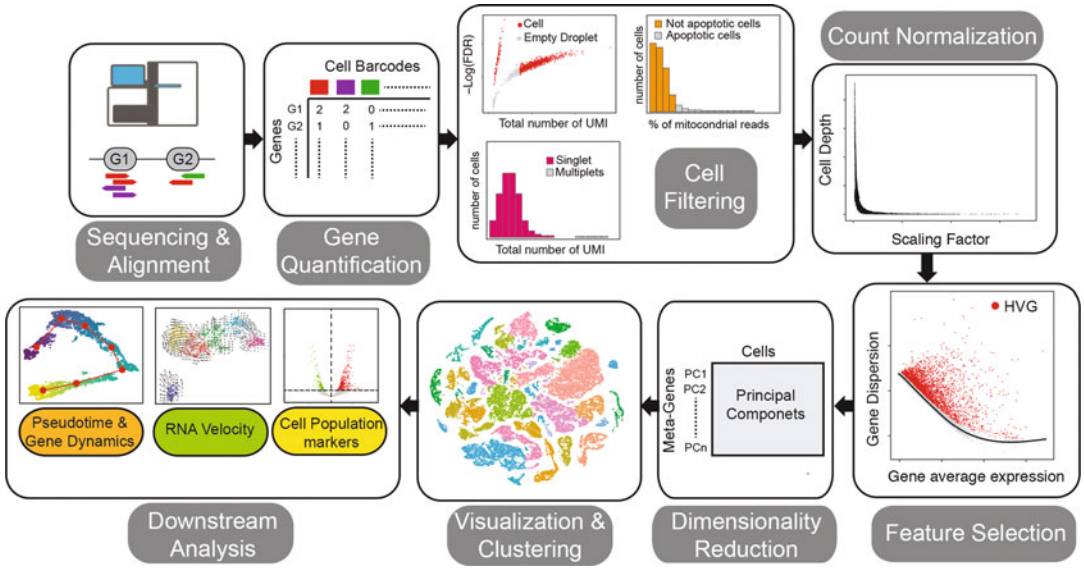

Fig. 2 Computational analysis of single-cell RNA sequencing. ScRNA-seq analysis embraces six underlying steps, including raw-data preprocessing, filtering via QC covariants, normalization, feature selection, linear dimensional reduction, visualization, and clustering: (*I*) Raw reads are processed and quantified to generate gene/barcode matrices. (*II*) Cells in the count matrix are then filtered to avoid misinterpretation of ambient gene expression, apoptotic cells, and multiplets. (*III*) Count reads normalization is required, as the analysis is disrupted by low input and weak SNR, following which data is primed for downstream analysis. (*IV*) A lesser number of highly variable features are selected for the purpose of realizing a faster and accurate procedure. (*V*) Based on the designated genes, a PCA is performed to lower data dimensionality. (*VI*) Clustering and nonlinear dimensionality reduction steps utilize a subset of significant principal components to overcome data noisiness. Subsequently, cells are clustered and visualized based on their PCA scores

4 Raw Reads Demultiplexing, Alignment, and Expression Quantification

Captured transcript fragments that are processed by sequencers, termed "reads," are stored into a text-based format called FASTQ [46]. FASTQ files contain both nucleotide sequence reads and their corresponding quality scores encoded as ASCII (American Standard Code for Information Interchange) characters.

While the majority of bulk approaches are suitable for the preprocessing of full-length scRNA-seq datasets, 3′-end scRNA-seq protocols require distinctive analytic tools. The preprocessing workflow of 3′-end scRNA-seq raw data includes three steps, (1) assigning captured RNA fragments to their associated sample and store them in FASTQ files (i.e., demultiplexing); (2) aligning the reads to a reference genome; (3) quantifying UMI per gene and assigning them to their associated barcode (i.e., cell). Eventually, each sample compiles into gene/barcode matrices that can be further filtered and analyzed.

4.1 Demultiplexing

Herein, we employ a commonly used subpipeline of the CellRanger platform [10], namely, **mkfastq**, exclusively designed for preprocessing raw data obtained from the 10x-Genomics® platform. Although CellRanger offers additional analytical tools for clustering and gene expression analysis, we have narrowed its use only to the preprocessing steps.

As input, CellRanger **mkfastq** uses raw sequencer's reads in the form of BCL files. Providing sample index sequences, **mkfastq** will demultiplex the raw data into sample-specific FASTQ files using the sample indexes.

4.2 Mapping and Expression Quantification

Before quantifying gene expression, the raw reads are first aligned to a reference genome, grouped by genes, and assigned to their original cellular barcode. These steps can be applied either by CellRanger-count for data retrieved via the 10×-Genomic® platform, or through the STARsolo tool [47, 48] for all other protocols.

Both tools require the raw FASTQ files obtained by the demultiplex step as input, and perform: (1) error correction of cell barcodes using a predefined whitelist; (2) mapping using STAR aligner; (3) correction and deduplication of UMI, and finally (4) quantification of gene expression per cell by counting the number of unique UMI per gene (i.e., transcripts).

Through the mapping step, read alignment assigns raw sequences to the most proper position in a reference genome. Although the alignment can employ a transcriptional reference, it is preferable to use a whole-genome reference, as it allows easier removal of "off-target" captured sequences that are not forced to be aligned on a transcriptional reference, but filtered out (*see* **Notes 1** and **2**).

Next, inconsistent cell barcodes and UMIs are filtered to avoid data misrepresentation. During this step, the presence of each barcode is verified in a predefined list of known cell barcode sequences provided by the single-cell platform. Accordingly, incompatible cell barcodes are either discarded or corrected by the most abundant barcode separated with a single editing distance. Similarly, CellRanger and STARsolo will assess the quality of UMIs and correct a single mismatch to a higher count UMI sequence if they both share a cell barcode and gene sequence.

Both CellRanger and STARsolo output two count matrices, filtered and unfiltered, so the user can choose which to include in the downstream analysis. The filtered count matrix consists of barcodes/identifiers that represent genuine cells and the expression levels for each gene. Differently from STARsolo, last CellRanger versions (above 3.1) employ a statistical method called EmptyDrop [49] to distinguish cells from empty barcoded drops. In this review, we will demonstrate how to apply EmptyDrop autonomously using the unfiltered count matrix, as it is common to both STARsolo and Cellranger outputs.

5 Quality Control and Cell Filtering: How to Identify Viable Cells

Current limitations of scRNA-seq are mainly related to low capture efficiency that can result in an increased level of technical noise. As of today, even the highly sensitive scRNA-seq protocols produce a small portion of low-quality barcodes due to lysed or apoptotic cells. Therefore, before proceeding with the downstream analysis, cellular barcodes that do not correspond to viable cells must be filtered out. These cells are usually recognized by detecting outliers in the distribution of QC covariates and filtered out by thresholding (*see* **Notes 3** and **4**). This step is common to all scRNA-seq pipelines and based on the analysis of three QC covariates distribution: (1) the number of captured genes per cell barcode; (2) the fraction of mitochondrial reads per barcode to identify dying cells; and (3) the number of unique UMIs per barcode (i.e., coverage depth of a cell).

5.1 Identify Empty Barcoded Drops

It is common to have empty drops when using droplet-based technology, as cells are highly diluted in order to yield a single-cell scaling. Empty drops might be contaminated with free RNA molecules, also called "ambient" RNA [37], that originated from cell lysis, which can be wrongly considered as cell-specific transcripts. To avoid misleading results, empty barcoded drops should not be included in downstream analysis. A recent method for identifying and filtering out empty drops is through the aforementioned emptyDrops function, provided by the DropletUtils package [49]. EmptyDrops is a function designed to test how significantly the barcode expression profile deviates from the ambient one using a Dirichlet-multinomial model. As input, it takes an unfiltered feature-barcode matrix and returns a data frame, where each barcode is associated with a p-value, obtained by permutation testing, and its relative FDR correction. Putting a threshold to this latter parameter allows the identification of ambient profiles with a significant deviation from cell-containing droplets, which are then considered as genuine cells. Here we show how to read data generated from the cellRanger count pipeline and detect empty droplets in the case of Tabula Muris dataset (Fig. 3a), where read data have been originally generated with the cellRanger count pipeline. Notably, since significance is retrieved by using permutations, a seed needs to be set.

5.2 Multiplet Identification

Multiplets occur when two or more cells are captured in a single drop and thus assigned to the same cell barcode [50]. This error may be misinterpreted as higher gene counts in an individual cell. Thereby, doublets can be simply filtered by identifying outliers in the count depth distribution. In the case of datasets generated by the aggregation of different samples and with different depth of

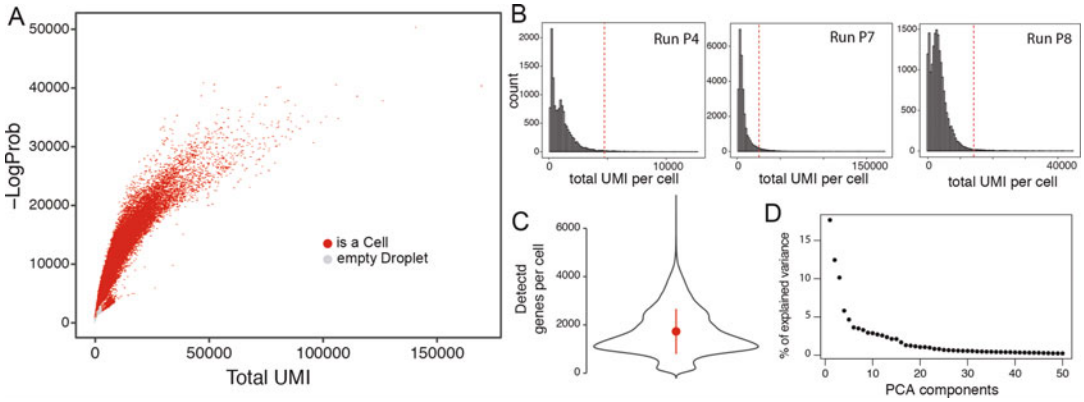

Fig. 3 Cell QC on Tabula Muris dataset. (**a**) Detection of empty droplet by using emptyDrop function from DropUtils R package on Tabula Muris dataset. (**b**) Identification of cell multiplets in each independent run of Tabula Muris Dataset. (**c**) Distribution number of detected genes across the cells in the Tabula muris dataset. (**d**) PCA components as a function of their percentage of explained variance on Tabula Muris dataset (elbow plot)

coverage among them, it is essential to perform multiplets filtered separately (Fig. 3b, *see* **Note 5**). Although a thresholding approach is usually sufficient to identify cell multiplets, new specific tools have been developed recently, offering more elegant and potentially better solutions [50–52].

5.3 Cells Lysis

Cell barcodes associated with transcripts originated by lytic cells are usually characterized by low count depth with few detected genes (Fig. 2c) and a high fraction of mitochondrial reads. In this case, unlike cytoplasmic RNA, most mitochondrial RNA is conserved thanks to undamaged mitochondrial membranes. Hence, it is acceptable to filter out barcodes with more than 10% of mitochondrial-associated reads. However, when setting a threshold, the biological property of the dataset should always be considered, therefore the threshold for an acceptable percentage of mitochondrial reads may vary according to the biological model of study. For cancer cells or specific cell types with increased respiratory or metabolic processes, the high levels of mitochondrial RNAs are inherent to the model itself [53] (*see* **Note 6**).

6 Start Working with the Scanpy, Seurat, Monocle, and gf-icf Pipelines

With the outburst of single-cell sequencing technologies, numerous statistical methods have been developed to address distinct steps of scRNA-seq analysis. Different toolkits like Seurat, Monocle 3, Scanpy, and gf-icf assembled these standalone methods to offer a single workflow. One of the most popular code-based platforms is Seurat, which offers a wide range of tutorials and analytical tools

[54]. An additional well-adapted platform is Monocle, which largely facilitated the trajectory inference field since its first introduction [41]. The latest version, Monocle 3, provides both pseudotemporal ordering and the basic scRNA-seq clustering pipeline for user convenience [42]. Scanpy, a relatively new addition in the field, allows for analyzing large size datasets up to one million cells and more, as it has improved the computational scaling. Here we also tested a recently introduced method named gf-icf, which is based on a data transformation model called term frequency-inverse document frequency (TF-IDF) that has been extensively used in the field of text mining, where sparse and zero-inflated data are common [55]. For downstream analysis, each pipeline employs either R or Python programming language. In order to interpret outputs and understand the basics of the analytical tools, each step will be examined and compared in all four pipelines.

7 Gene Filtering: How to Remove "Noisy" Genes

A scRNA-seq dataset often includes over 25,000 genes measured across thousands of cells, many of which might be uninformative as they mostly contain zero counts, and should be filtered out before starting the downstream analysis. Gene filtering can help to speed up data processing by dipping its dimensions and reducing the excess of zeros counts, consequently improving the data normalization step and all downstream analysis. Usually, a fixed threshold is defined, whereby genes detected in a small number of cells are removed (*see* **Notes 7** and **8**).

8 Data Normalization: How to Make Gene Expression Comparable Across Individual Cells

Data normalization addresses the unwanted biases arisen by count depth variability while preserving true biological differences. The quantity of mRNA captured from each cell may diversify due to either biological variability or technical effects inherited throughout the scRNA-seq procedure, including single-cell preparation, library construction, and sequence steps [56–58]. With normalization, the expression of each gene is rescaled, considering the abundance of mRNA molecules that have been captured for each cell, in order to make gene expression comparable across individual cells. The way in which this scaling factor is estimated for each cell mainly differs across the plethora of currently available normalization methods.

As discussed above, scRNA-seq data is usually sparse due to both biological and technical reasons (dropouts). Hence, normalization methods adopted from bulk RNA-seq, such as TMM [59]

and DESeq [60], might be biased by zero inflation. To address this issue, single-cell normalization procedures have evolved in recent years [61, 62]. However, at present, the most commonly used method for scRNA-seq data normalization is count per million (CPM), a linear global scaling approach that has been inherited from bulk RNA-seq.

An additional source of variation not related to the biological process under study can result from handling samples in different batches. The batch effect may arise when an experiment with identical cells is repeated independently, for example, by different operators or sampling different experimental time-lines. Standard normalization procedures do not correct for batch effect, compromising the analysis of the real biological effects. Several methods have been recently developed to account for the batch effects in scRNA-seq data [63], although ComBat [64], a method originally developed for microarray data, performs well also for single-cell experiments of low-to-medium complexity [65].

All four pipelines proposed here account for the normalization step through the CPM method. Seurat, Monocle, and Scanpy use log transformation of the CPM to reduce cell depth variability (*see* **Notes 9–11**) and few advanced options to rescale the data for some sources of variation, including the effect cell cycle [39]. With the gf-icf pipeline, genes are rescaled by their inverse cell frequency and cells are rescaled to have Euclidean norm equal to one (L2 normalization), in order to account for cell depth variability.

9 Feature Selection: How to Discard "Uninformative" Genes

A large-scale scRNA-seq dataset can easily include over 25,000 genes measured across more than 10,000 cells, with many of these genes being uninformative because mostly containing zero counts.

Feature selection aims to detect genes with relevant biological information, while excluding the uninformative ones. ScRNA-seq data dimensions can remain quite high, with a large number of genes (>10,000) still retained even after gene filtering. Feature selection can largely speed up the processing as it reduces data dimensionality by filtering "uninformative" genes. This is usually accomplished by selecting a limited number of highly variable genes (HVG) to direct further analysis. HVG are highly informative as they have a significant impact on the data configurations, and therefore allow to preserve the integrity and reproducibility of the data. Usually, 1000–5000 HVG are selected depending on the size of the assay (*see* **Note 12**). Each pipeline implements a unique method for the detection and selection of HGVs. Using Scanpy, genes are binned by their mean expression, and genes with the highest variance-to-mean ratio are selected as HVGs in each bin.

Seurat, on the other hand, first modeled the mean-variance relationship using a local polynomial regression function. Then, given the expected variance by the fitted curve and the observed mean, the feature values are standardized, and for each gene, the variance across all cells is computed [66] (*see* **Note 13**). Unlike Seurat and Scanpy, Monocle does not cover this step, while gf-icf feature selection is performed only when differentially expressed genes across clusters need to be identified. Although with few differences, also in gf-icf the feature selection is performed by modeling the mean/variance relationship as proposed by Chen et al. [67]. The feature selection in gf-icf pipeline is built in the normalization step when gene expression is rescaled by their inverse cell frequency [43], and the total number of filtered genes is considered for the dimensionality reduction step.

10 Dimensionality Reduction: How to Summarize and Visualize scRNA-seq Data

10.1 Linear Dimensional Reduction: For the Summarization of scRNA-seq Data

Dimensionality reduction aims to condense the complexity of the data into a lower-dimensional space by optimally preserving its key properties. Dimensionality reduction methods are essential for clustering, visualization, and summarization of scRNA-seq data. Linear dimensionality reduction methods are commonly used as a preprocessing step for nonlinear dimensionality reduction methods. The most popular linear dimensionality reduction algorithm is the PCA (Principal Component Analysis) [68]. Usually, 10–50 significant principal components are selected and later used as input for nonlinear dimensionality reduction methods. Principle components are highly indicative of primary sources of heterogeneity in the dataset.

PCA is used to summarise a dataset throughout the top N principal components (*see* **Note 14**). The number of PCA to use is usually determined by manually inspecting the elbow plot (Fig. 3d), in which principal components are plotted as a function of the variability they account for, and the number of PCA to use is determined by the point in which an "elbow" is observed. Additional methods can be used, including jackstraw [69] and heat maps of leading genes in each principal component. However, when choosing the significant principal components to use, it is better to err on the higher side to avoid information loss.

10.2 Nonlinear Dimensionality Reduction for the Visualization of scRNA-seq Data

Dimensionality reduction for visualization of scRNA-data uses methods that capture the nonlinearity of the scRNA-seq data, avoiding the overcrowding of the representation (*see* **Note 15**). The two most commonly used methods are the t-Distributed Stochastic Neighbor Embedding (t-SNE) [70] and the Uniform Manifold Approximation and Projection (UMAP) [71]. t-SNE is a stochastic method that efficiently highlights local data structure in

low dimensions, representing cell populations as distinct clusters. However, t-SNE is not able to preserve the global structure, so the distance between clusters is meaningless. UMAP is a more recent nonlinear dimensionality-reduction technique, that is instead able to preserve both local and global structure of the data outperforming t-SNE also with a shorter run time for really large-scale scRNA-datasets.

Several additional methods exist for both linear and nonlinear data dimensionality reduction, but it is out of the scope of this tutorial to review all the existing methods, while we prefer to focus on best practices and methods currently accepted by the scRNA-seq community. However, a detailed review of linear and nonlinear methods for dimensionality reduction of single-cell transcriptomic data can be found in Moon et al. [72].

11 Clustering Analysis: How to Identify Cellular Subpopulations

As transcriptionally distinct populations of cells usually correspond to distinct cell types, a key goal of scRNA-seq consists in the identification of cell subpopulations based on their transcriptional similarity [73]. Thus, organizing cells into groups (i.e., clusters) can allow for de novo detection of cell types or identification of different subpopulations in a single cell state (*see* **Note 16**).

Clustering is an old unsupervised machine learning problem, which aims to determine the intrinsic grouping in a set of unlabelled objects by knowing their similarity score (i.e., distance). A plethora of distance measures has been proposed in the literature to compute similarity scores among objects of interest, including Euclidean distance, Cosine distance, and correlation-based distances.

Several unsupervised clustering methods have been applied to partition single-cell data and can be further divided into three groups: (1) *k*-means, (2) hierarchical clustering, and (3) community detection approaches. For single-cell data analysis, all methods are applied after feature selection and data dimensionality reduction on the PC-reduced space. The identified clusters of cells are then overlaid onto the visualization space.

The k-means algorithm uses an iterative approach to partition cells into a predefined number of clusters (k). At each iteration, cells are assigned to the closest centroid using the Euclidean distance. Alternative distances, like correlation-based or cosine distances, can also be used for single-cell data analysis [74]. The position of the centroids is recomputed at the end of each iteration, and since the starting position of centroids is randomly selected, it is common to run the *k*-means algorithm multiple times [75]. Although fast, *k*-

means requires to know the initial number of clusters (k) in which to partition cells, which is usually unknown and must be settled with additional complex analyses.

Hierarchical clustering is a partitioning method that seeks to build a hierarchy of clusters, and it is generally divided into two types, namely agglomerative or divisive. An agglomerative hierarchical clustering technique follows the "bottom-up" approach, where initially each cell represents an individual cluster, and gradually similar clusters are merged until getting a unique cluster. On the other hand, a divisive hierarchical clustering follows the "top-down" approach, where all cells start from a single cluster and are then progressively split. Hierarchical clustering produces a dendrogram where clusters are obtained by cutting the tree at a predetermined distance that can heuristically be settled using bootstrap approaches [76]. Examples of the application of hierarchical clustering in scRNA-seq data can be found in CIDR [77], SINCERA [78], and pcaReduce [79]. However, hierarchical clustering methods generally work slower than k-means, and do not perform well on a large-scale scRNA-seq dataset.

Community detection techniques are scalable clustering approaches, which are appropriate for large-scale graphs and can be used to cluster a hundred thousand or even millions of cells efficiently. By definition, a graph $G = (V,E)$ consists of a collection of nodes V (i.e., cells) and edges representing the degree of similarity between pairs of cells. This graph of cells can be built using the K-Nearest Neighbors (KNN) algorithm [80] applied on the PC-reduced space, where each cell is connected to its K most similar cells. Then, edge weight between any two cells is refined by Jaccard similarity, by using the proportion of neighbors they share.

Finding communities means gathering cells into groups, with a higher density of edges within groups than between them [81]. A measure of the community structure of a graph is modularity [82], namely, the fraction of edges that fall within the given groups minus the expected fraction if edges were randomly distributed. Modularity is based on the idea that a random graph is not expected to have a cluster structure. The most popular detection algorithm based on modularity is Louvain, which was introduced by PhenoGraph and also used by Seurat, Scanpy, and gf-icf.

When running a graph-based clustering, it is necessary to set the resolution parameter for the community detection algorithm based on modularity optimization. The resolution parameter is correlated to the scale of observing communities. In particular, the higher is the resolution parameter, the larger is the number of smaller communities. In our pipelines, we set the resolution parameter to 0.5, which usually represents a good trade-off.

12 Differential Expression: How to Annotate Cell Populations

Characterization and annotation of the groups of cells identified by a clustering algorithm can be managed by identifying marker genes (i.e. cluster gene signature) via differential expression analysis. Marker genes are identified by comparing cells of every single cluster to all other cells. Some differentially expressed testing methods have been developed specifically for handling the presence of dropout elements in scRNA-seq data, including the Bayesian approach [83] and MAST [84], but they are not computationally efficient when considering large-scale scRNA-seq datasets. Hence, faster tests are used for detecting differentially expressed genes, like Wilcoxon rank-sum test implemented in Seurat, Scanpy, and gf-icf, while Monocle uses a generalized additive model (VGAM). Additional complex tests are also provided by Seurat, Scanpy, and Monocle. Once gene signatures of each cluster have been identified, additional analysis including Gene Ontology Enrichment Analysis (GOEA) and Gene Set Enrichment Analysis (GSEA) [85] can be used to identify the biological processes active in each cell's cluster.

13 Results Evaluation and Comparison Among the Implemented Pipelines

To evaluate the performance of the four pipelines in identifying groups of cells (Fig. 4a–d), we calculated the agreement across clusters produced by the different methods, by using the average Jaccard coefficient [86] among each pair of clusters (Fig. 4e). We then used the retrieved clusters from each method to hierarchically cluster cell types (Fig. 4f), and showed that the different methods produce biologically meaningful partitions. We also observed that cells in the same cluster belong to the same lineage but with different levels of granularity, which can be tuned by changing the resolution parameter used to identify cell clusters.

14 Additional Analyses: How to Reconstruct Cell Transcriptional Dynamics

Depending on the biological question to address, one may think to investigate further single-cell data leveraging other existing tools that may provide other levels of information. Biological mechanisms are highly dynamic processes and thus cannot always be well described by using a discrete approach, such as clustering. Cells can transit across several transcriptional states governed by environmental changes and external perturbations. Thus, to model such continuance biological systems, including developmental processes, a new class of computational methods, called trajectory inference, have been developed in the last few years. These methods

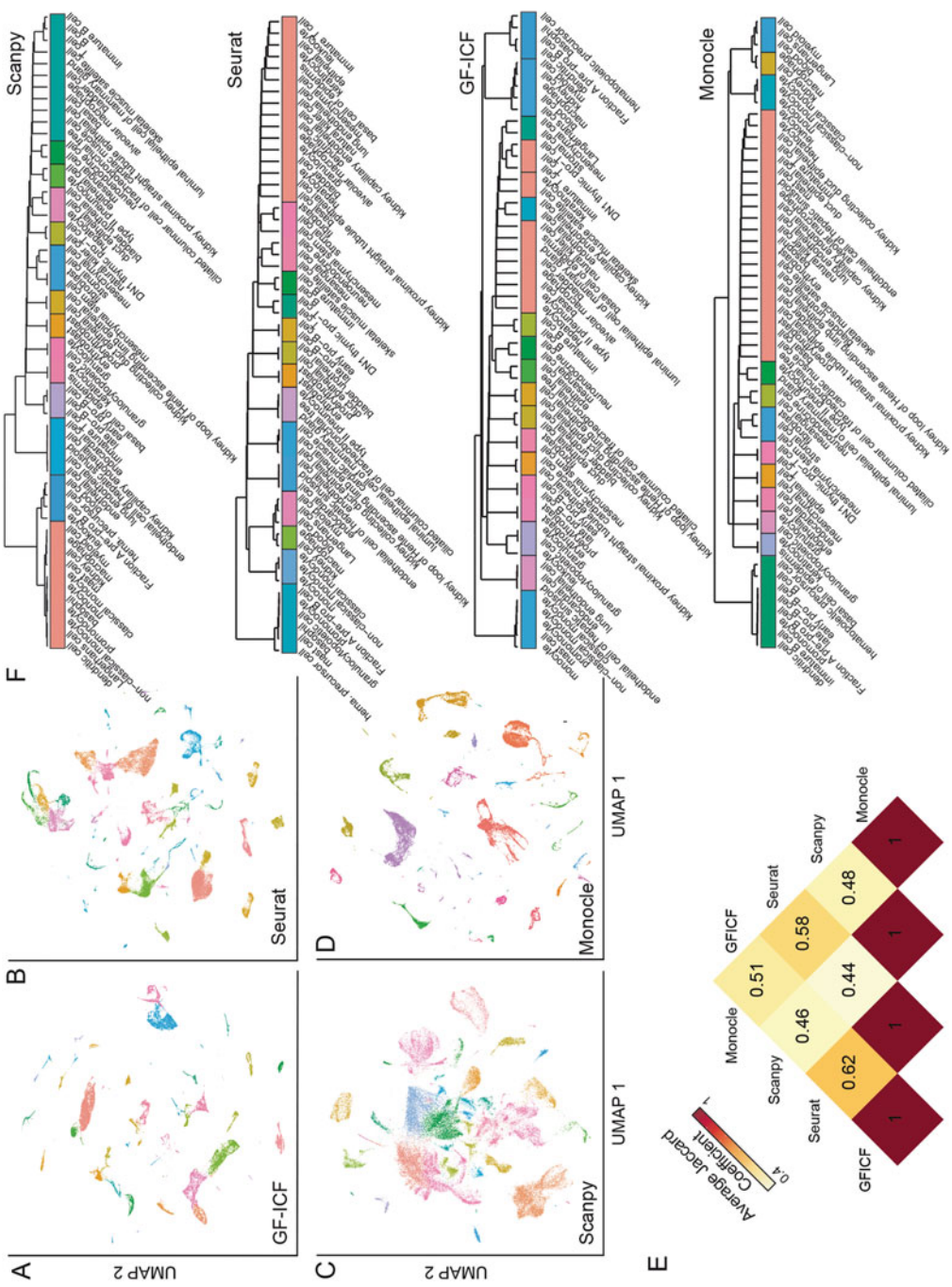

Fig. 4 Cell clustering, comparison and evaluation of the implemented pipelines. (**a**–**d**) UMAP visualization produced by each one of the four implemented pipelines where cells are colored according to the cluster in which they fall. (**e**) Agreement of identified cell clusters among the four implemented pipelines. (**f**) Cluster results of each independent pipeline is used to hierarchically cluster cell types and reconstruct cell lineage

use scRNA-seq data generated from a population of cells underlying a biological process that were collected at different time-points, and try to computationally order them along an evolutionary trajectory, which can have different topologies (i.e. linear, bifurcating or even more complex graph structure). Once cells have been ordered, gene expression patterns throughout the inferred trajectories can be used to identify key regulator genes governing cell fate decisions.

We first introduced the concept of pseudotime with Monocle as a robust methodology to describe developmental systems [41]. Since Monocle, which is at its third version now [42], the number of available methods has grown exponentially [87]. Recently, a newly proposed method to infer developmental trajectories that substantially differ from others, was modeling cellular processes using the optimal transport problem [88]. Interestingly, to date, more than 100 methods have been developed to infer cell trajectory [87].

Once trajectories have been reconstructed, RNA velocities [89] can be overlaid onto the inferred trajectory to add directionality to the reconstructed dynamical process.

15 Discussion and Future Directions

With the outburst of scRNA-seq technology, an increasing number of analytic methods have been introduced to the scientific community. Despite the wide range of analytic options, the absence of standardization leads to high entry barriers. In the present review, we propose four ready-to-use pipelines for the analysis of scRNA-seq data that could fit various biological data types. With a novice in mind, these computational pipelines provide an effective and simple workflow, including normalization of raw counts, feature selection, dimensionality reduction, and data clustering. The proposed pipelines cover both R and Python programming languages, and employ Seurat (R), Scanpy (Python), Monocle (R), and gf-icf (R) platforms.

As it is important to have the ability to interpret outputs in order to ensure data coherence, we reviewed the key steps of scRNA-seq analysis. We also highlighted guidelines and offered standardized values to filter and reduce data dimensionality. It is the user's responsibility to carefully assess the output of their analysis, and if necessary, adjust the pipeline default settings to fit source data. Furthermore, as the field evolves rapidly, this review might lag behind the up-to-date tools. Therefore, we recommend referring to this review as a basic workflow guideline while keeping in line with innovations in the field.

As single-cell sequencing is no longer limited to transcriptional experiments but allows for capturing also other data types, including DNA, ChIP, and ATAC, we presume that future pipelines must be able to cope with multiomics data integration. Single-cell multiomics will simultaneously allow gaining information on all levels of the living cells, including DNA, RNA, proteins, and epigenetic modification [90–92]. Integration of these different "omics" information into a single dimension will allow having a more comprehensive understanding of the cell fate regulations and phenotypes.

Interestingly, another new technology that necessitates high scaling computational tools is the spatial transcriptomics, which allows to identify the cell type spatial composition of tissues [93, 94]. This approach may help to increase the accuracy of the investigated system by adding another guiding dimension to the data. By positionally annotating the cells, it would be possible to precisely cluster different subpopulations in highly heterogeneous systems, such as organoids, and track the spatiotemporal dynamics between them. Therefore, the ability to conserve the spatial position will provide a better perception of tissue organization, functionality, and development.

An additional perspective is the use of high-throughput scRNA-seq technology for personalized medicine. Several efforts have been made to screen different cell types and tissues via scRNA-seq to tailor appropriate medical treatment to patients' individual characteristics [95, 96]. Developing new tools that incorporate machine learning approaches may increase the advancement in the field of precision medicine, and bring it closer to clinical usage. We believe that innovative tools occupying the aforementioned properties will stand at the forefront of science.

16 Notes

1. Before proceeding with further analyses, it is recommendable to go through sequencing and mapping statistics, which are often provided by the bioinformatic tool used for preprocessing. For instance, less than 70% of barcode-associated reads might suggest high levels of ambient RNA (due to a significant level of lysed cells or insufficient washes after tissue dissociation).
2. Reads confidently mapped to the genome should exceed 80% of the total.
3. QC-based outlier detection, that is, multiplet and lytic cell filtering, should be performed taking these covariates concomitantly.
4. The threshold for filtering outlier cells should be as permissive as possible to avoid excessive dropout effect. It could be further

adjusted once downstream analyses have been performed to better interpret data.

5. As transcripts coverage may differ between samples, it is essential to set the threshold for each one separately.
6. When setting a threshold, the biological property of the dataset should be considered, as increased respiratory or metabolic processes may also cause high mitochondrial reads.
7. The selected threshold should be as permissible as possible to avoid a dropout effect or removal of a rare cell population.
8. An acceptable guideline is to adjust the threshold to the smallest cluster size or to the number of genes expressed in more than 1–5% in the dataset.
9. Despite the normalization method of choice, data transformation (e.g., log transformation) should always be applied since most tools for downstream analyses expect normally distributed data.
10. Try to avoid correcting biological batches, unless you want to infer trajectories and such correction does NOT mask other biological information of interest.
11. When performing batch correction on technical as well as biological covariates, it should be done simultaneously.
12. The choice of HVGs may influence downstream analysis, although it has been shown that choosing between 200 and 2400 HVGs does not affect representation in lower dimensions (i.e., PCA space) [9].
13. Feature selection based on mean and variance cannot be performed on data scaled to zero mean and unit variance.
14. Principal components can also be used to inspect the effect of technical covariates on data [65], or to address the role of specific genes across the dataset [69].
15. Nonlinear dimensionality reduction methods are a powerful tool for data visualization, NOT summarization.
16. Downstream analyses require summarized data, for example, PCA or diffusion maps.

Acknowledgments

We thank Chiara Reggio and Bashir Sadet (10× Genomics) for their feedbacks. This work was supported by Fondazione Telethon Core Grant, Armenise-Harvard Foundation Career Development Award, European Research Council (grant agreement 759154, CellKarma), the Rita-Levi Montalcini program from MIUR (to D.C.) and by the STAR (Sostegno Territoriale alle Attività di Ricerca) grant of University of Naples Federico II and My First

AIRC Grant (MFAG, 23162) (to G.G.). Conflict of interest – Gennaro Gambardella declares that he has no conflict of interest. Davide Cacchiarelli is the founder, shareholder, and consultant of Next Generation Diagnostic srl. Figures were created with BioRender software,©biorender.com.

References

1. Wang ET, Sandberg R, Luo S et al (2008) Alternative isoform regulation in human tissue transcriptomes. Nature 456:470–476
2. Cacchiarelli D, Trapnell C, Ziller MJ et al (2015) Integrative analyses of human reprogramming reveal dynamic nature of induced pluripotency. Cell 162:412–424
3. Mitelman F, Johansson B, Mertens F (2007) The impact of translocations and gene fusions on cancer causation. Nat Rev Cancer 7:233–245
4. Trapnell C (2015) Defining cell types and states with single-cell genomics. Genome Res 25:1491–1498
5. Hedlund E, Deng Q (2018) Single-cell RNA sequencing: technical advancements and biological applications. Mol Asp Med 59:36–46
6. Hwang B, Lee JH, Bang D (2018) Single-cell RNA sequencing technologies and bioinformatics pipelines. Exp Mol Med 50(96)
7. Supplementary Table 1: https://github.com/gambalab/scRNAseq_chapter/blob/master/tables/table1.xlsx
8. Macosko EZ, Basu A, Satija R et al (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 161:1202–1214
9. Klein AM, Mazutis L, Akartuna I et al (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 161:1187–1201
10. Zheng GXY, Terry JM, Belgrader P et al (2017) Massively parallel digital transcriptional profiling of single cells. Nat Commun 8:14049
11. Plasschaert LW, Žilionis R, Choo-Wing R et al (2018) A single-cell atlas of the airway epithelium reveals the CFTR-rich pulmonary ionocyte. Nature 560:377–381
12. Suo S, Zhu Q, Saadatpour A et al (2018) Revealing the critical regulators of cell identity in the mouse cell atlas. Cell Rep 25:1436–1445.e3
13. Velasco S, Kedaigle AJ, Simmons SK et al (2019) Individual brain organoids reproducibly form cell diversity of the human cerebral cortex. Nature 570:523–527
14. Fischer DS, Fiedler AK, Kernfeld EM et al (2019) Inferring population dynamics from single-cell RNA-sequencing time series data. Nat Biotechnol 37:461–468
15. Liu Z, Wang L, Welch JD et al (2017) Single-cell transcriptomics reconstructs fate conversion from fibroblast to cardiomyocyte. Nature 551:100–104
16. Cacchiarelli D, Qiu X, Srivatsan S et al (2018) Aligning single-cell developmental and reprogramming trajectories identifies molecular determinants of myogenic reprogramming outcome. Cell Syst 7:258–268.e3
17. van Dijk D, Sharma R, Nainys J et al (2018) Recovering gene interactions from single-cell data using data diffusion. Cell 174:716–729.e27
18. Hayashi T, Ozaki H, Sasagawa Y et al (2018) Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. Nat Commun 9:619
19. Savas P, Virassamy B, Ye C et al (2018) Single-cell profiling of breast cancer T cells reveals a tissue-resident memory subset associated with improved prognosis. Nat Med 24:986–993
20. Moghe I, Loupy A, Solez K (2018) The human cell atlas project by the numbers: relationship to the Banff classification. Am. J. Transplant 18:1830
21. Ziegenhain C, Vieth B, Parekh S et al (2017) Comparative analysis of single-cell RNA sequencing methods. Mol Cell 65:631–643.e4
22. Senabouth A, Andersen S, Shi Q et al (2020) Comparative performance of the BGI and Illumina sequencing technology for single-cell RNA-sequencing. NAR Genom Bioinform 2. https://doi.org/10.1093/nargab/lqaa034
23. Rosenberg AB, Roco CM, Muscat RA et al (2018) Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science 360:176–182
24. Tasic B, Yao Z, Graybuck LT et al (2018) Shared and distinct transcriptomic cell types across neocortical areas. Nature 563:72–78

25. Guillaumet-Adkins A, Rodríguez-Esteban G, Mereu E et al (2017) Single-cell transcriptome conservation in cryopreserved cells and tissues. Genome Biol 18:45
26. Wohnhaas CT, Leparc GG, Fernandez-Albert F et al (2019) DMSO cryopreservation is the method of choice to preserve cells for droplet-based single-cell RNA sequencing. Sci Rep 9:10699
27. Baran-Gale J, Chandra T, Kirschner K (2018) Experimental design for single-cell RNA sequencing. Brief Funct Genomics 17:233–239
28. Salomon R, Kaczorowski D, Valdes-Mora F et al (2019) Droplet-based single cell RNAseq tools: a practical guide. Lab Chip 19:1706–1727
29. Islam S, Zeisel A, Joost S et al (2014) Quantitative single-cell RNA-seq with unique molecular identifiers. Nat Methods 11:163–166
30. Abate AR, Chen C-H, Agresti JJ, Weitz DA (2009) Beating Poisson encapsulation statistics using close-packed ordering. Lab on a Chip 9:2628
31. Zhang X, Li T, Liu F et al (2019) Comparative analysis of droplet-based ultra-high-throughput single-cell RNA-Seq systems. Mol Cell 73:130–142.e5
32. Brazovskaja A, Treutlein B, Camp JG (2019) High-throughput single-cell transcriptomics on organoids. Curr Opin Biotechnol 55:167–171
33. Stegle O, Teichmann SA, Marioni JC (2015) Computational and analytical challenges in single-cell transcriptomics. Nat Rev Genet 16:133–145
34. Haque A, Engel J, Teichmann SA, Lönnberg T (2017) A practical guide to single-cell RNA--sequencing for biomedical research and clinical applications. Genome Medicine 9
35. Lähnemann D, Köster J, Szczurek E et al (2020) Eleven grand challenges in single-cell data science. Genome Biol 21:31
36. scRNA-tools table page. https://www.scrna-tools.org/. Accessed 22 June 2020
37. Luecken MD, Theis FJ (2019) Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol 15
38. Neu KE, Tang Q, Wilson PC, Khan AA (2017) Single-cell genomics: approaches and utility in immunology. Trends Immunol 38:140–149
39. Butler A, Hoffman P, Smibert P et al (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol 36:411–420
40. Wolf FA, Angerer P, Theis FJ (2018) SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 19:15
41. Trapnell C, Cacchiarelli D, Grimsby J et al (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol 32:381–386
42. Cao J, Spielmann M, Qiu X et al (2019) The single-cell transcriptional landscape of mammalian organogenesis. Nature 566:496–502
43. Gambardella G, di Bernardo D (2019) A tool for visualization and analysis of single-cell RNA-Seq data based on text mining, Front Genet:10
44. Tabula Muris Consortium, Overall Coordination, Logistical Coordination, et al (2018) Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature 562:367–372
45. scRNAseq_chapter. Github. https://github.com/gambalab/scRNAseq_chapter
46. Cock PJA, Fields CJ, Goto N et al (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38:1767–1771
47. Dobin A, Davis CA, Schlesinger F et al (2013) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29:15–21
48. Du Y, Huang Q, Arisdakessian C, Garmire LX (2020) Evaluation of STAR and Kallisto on single cell RNA-Seq data alignment. G3 10:1775–1783
49. Lun ATL, participants in the 1st Human Cell Atlas Jamboree, Riesenfeld S, et al (2019) EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. Genome Biol 20
50. Wolock SL, Lopez R, Klein AM (2019) Scrublet: computational identification of cell doublets in single-cell transcriptomic data. Cell Syst 8:281–291.e9
51. DePasquale EAK, Schnell DJ, Van Camp P-J et al (2019) DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data. Cell Rep 29:1718–1727.e8
52. McGinnis CS, Murrow LM, Gartner ZJ (2019) DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. Cell Syst 8(4):329–337.e4
53. Rogalinska M (2016) The role of mitochondria in cancer induction, progression and changes in metabolism. Mini Rev Med Chem 16:524–530
54. Düchting H, Seurat G (2000) Seurat. Taschen
55. Robertson SE, Jones KS (1976) Relevance weighting of search terms. J Am Soc Inf Sci 27:129–146
56. Marinov GK, Williams BA, McCue K et al (2014) From single-cell to cell-pool transcriptomes: Stochasticity in gene expression and RNA splicing. Genome Res 24:496–510

57. Grün D, Kester L, van Oudenaarden A (2014) Validation of noise models for single-cell transcriptomics. Nat Methods 11:637–640
58. Wu AR, Neff NF, Kalisky T et al (2014) Quantitative assessment of single-cell RNA-sequencing methods. Nat Methods 11:41–46
59. Robinson MD, Oshlack A (2010) A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol 11:R25
60. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11(10):R106
61. Lun ATL, Bach K, Marioni JC (2016) Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol 17:75
62. Vallejos CA, Risso D, Scialdone A et al (2017) Normalizing single-cell RNA sequencing data: challenges and opportunities. Nat Methods 14:565–571
63. Tran HTN, Ang KS, Chevrier M et al (2020) A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biol 21:12
64. Johnson WE, Li C, Rabinovic A (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics 8:118–127
65. Büttner M, Miao Z, Wolf FA et al (2019) A test metric for assessing single-cell RNA-seq batch correction. Nat Methods 16:43–49
66. Stuart T, Butler A, Hoffman P et al (2019) Comprehensive integration of single-cell data. Cell 177:1888–1902.e21
67. Chen H-IH, Jin Y, Huang Y, Chen Y (2016) Detection of high variability in gene expression from single-cell RNA-seq profiling. BMC Genomics 17(Suppl 7):508
68. Pearson K (1901) LIII. On lines and planes of closest fit to systems of points in space. London Edinburgh Dublin Philos Mag J Sci 2:559–572
69. Chung NC, Storey JD (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics 31:545–554
70. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
71. McInnes L, Healy J, Melville J (2018) UMAP: Uniform Manifold Approximation and Projection for dimension reduction. arXiv [stat.ML]
72. Moon KR, Stanley JS, Burkhardt D et al (2018) Manifold learning-based methods for analyzing single-cell RNA-sequencing data. Curr Opin Syst Biol 7:36–46
73. Andrews TS, Hemberg M (2018) Identifying cell populations with scRNASeq. Mol Aspects Med 59:114–122
74. Kim T, Chen IR, Lin Y et al (2019) Impact of similarity metrics on single-cell RNA-seq data clustering. Brief Bioinform 20:2316–2326
75. Kiselev VY, Kirschner K, Schaub MT et al (2017) SC3 – consensus clustering of single-cell RNA-Seq data. Nat Methods 14 (5):483–486
76. Langfelder P, Zhang B, Horvath S (2008) Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics 24:719–720
77. Lin P, Troup M, Ho JWK (2017) CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. Genome Biol 18:59
78. Guo M, Wang H, Potter SS et al (2015) SINCERA: a pipeline for single-cell RNA-Seq profiling analysis. PLoS Comput Biol 11: e1004575
79. Žurauskienė J, Yau C (2016) pcaReduce: hierarchical clustering of single cell transcriptional profiles. BMC Bioinformatics 17:140
80. Levine JH, Simonds EF, Bendall SC et al (2015) Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell 162:184–197
81. Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E Stat Nonlin Soft Matter Phys 70:066111
82. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci U S A 103:8577–8582
83. Kharchenko PV, Silberstein L, Scadden DT (2014) Bayesian approach to single-cell differential expression analysis. Nat Methods 11:740–742
84. Finak G, McDavid A, Yajima M et al (2015) MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. Genome Biol 16:278
85. Subramanian A, Tamayo P, Mootha VK et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102:15545–15550
86. Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaud sci nat 37:547–579
87. Saelens W, Cannoodt R, Todorov H, Saeys Y (2019) A comparison of single-cell trajectory inference methods. Nat Biotechnol 37:547–554

88. Schiebinger G, Shu J, Tabaka M et al (2019) Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. Cell 176:1517
89. La Manno G, Soldatov R, Zeisel A et al (2018) RNA velocity of single cells. Nature 560:494–498
90. Argelaguet R, Clark SJ, Mohammed H et al (2019) Multi-omics profiling of mouse gastrulation at single-cell resolution. Nature 576:487–491
91. Angermueller C, Clark SJ, Lee HJ et al (2016) Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. Nat Methods 13:229–232
92. Han KY, Kim K-T, Joung J-G et al (2018) SIDR: simultaneous isolation and parallel sequencing of genomic DNA and total RNA from single cells. Genome Res 28:75–87
93. Moncada R, Barkley D, Wagner F et al (2020) Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. Nat Biotechnol 38:333–342
94. Ståhl PL, Salmén F, Vickovic S et al (2016) Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science 353:78–82
95. Valdes-Mora F, Handler K, Law AMK et al (2018) Single-cell transcriptomics in cancer immunobiology: the future of precision oncology. Front Immunol 9:2582
96. Shalek AK, Benson M (2017) Single-cell analyses to tailor treatments. Sci Transl Med 9. https://doi.org/10.1126/scitranslmed.aan4730