Check for updates

Tools for the analysis of high-dimensional single-cell RNA sequencing data

Yan Wu and Kun Zhang

Department of Bioengineering, University of California at San Diego, La Jolla, CA, USA.

e-mail: kzhang@bioeng.ucsd.edu

https://doi.org/10.1038/s41581-020-0262-0

Abstract | Breakthroughs in the development of high-throughput technologies for profiling transcriptomes at the single-cell level have helped biologists to understand the heterogeneity of cell populations, disease states and developmental lineages. However, these single-cell RNA sequencing (scRNA-seq) technologies generate an extraordinary amount of data, which creates analysis and interpretation challenges. Additionally, scRNA-seq datasets often contain technical sources of noise owing to incomplete RNA capture, PCR amplification biases and/or batch effects specific to the patient or sample. If not addressed, this technical noise can bias the analysis and interpretation of the data. In response to these challenges, a suite of computational tools has been developed to process, analyse and visualize scRNA-seq datasets. Although the specific steps of any given scRNA-seq analysis might differ depending on the biological questions being asked, a core workflow is used in most analyses. Typically, raw sequencing reads are processed into a gene expression matrix that is then normalized and scaled to remove technical noise. Next, cells are grouped according to similarities in their patterns of gene expression, which can be summarized in two or three dimensions for visualization on a scatterplot. These data can then be further analysed to provide an in-depth view of the cell types or developmental trajectories in the sample of interest.

In a single organism, most cells have the same genome, but specific gene expression varies across different tissues and cell types. Any given tissue or cell type expresses ~11,000–13,000 genes, of which ~3,000–5,000 have a cell-type-specific expression pattern, whereas the remaining genes are ubiquitously expressed[1]. These unique patterns of gene expression translate to differences at the protein level between different cell types and result in the vast array of cellular phenotypes found throughout the body. Therefore, a snapshot of the gene expression profile of a cell can be indicative of its phenotype. Owing to the limited amount of RNA present in each cell, gene expression profiling was historically performed on pooled cells, but this bulk sequencing approach obscured the potential cell heterogeneity in a sample or tissue[2]. For example, in a pool of developing progenitor cells, different cells might be primed to make distinct fate decisions but these transcriptional programmes are indistinguishable in a bulk analysis of the average gene expression in the progenitor pool.

The development of technologies that can isolate thousands to tens of thousands of cells and assess their gene expression profiles at the single-cell level has enabled researchers to dissect this cellular heterogeneity and work towards a better understanding of physiology, biological development and disease[2–6]. For example, researchers generated an improved quantitative map of the cell types present in the developing human kidney, which has provided insights into renal physiology[7]. Another single-cell study demonstrated the similarities between fetal human kidney and human kidney organoids, reaffirming the utility of kidney organoids as a model for the study of disease and for drug screening[8].

However, deriving biological insights from single-cell RNA sequencing (scRNA-seq) methods demands that researchers handle the large volume of data generated by these technologies and their accompanying sources of technical noise[9]. Addressing the scale and complexity of these datasets thus requires a complex ecosystem of computational methods.

Beyond scRNA-seq analysis, other available technologies can profile genomes[10], methylation patterns[11] and chromatin accessibility patterns[12,13] at the single-cell level. Each type of single-cell profiling comes with its own challenges in terms of data analysis. Additionally, the development of 'multi-omics' approaches, in which multiple types of biological molecules are profiled in the same cell, has advanced substantially in recent years.

Key points

- As single-cell RNA sequencing datasets increase in scale and complexity, faster and more efficient computational tools for processing and analysis are required.
- New computational tools that correct technical and batch effects can unlock additional heterogeneity and enable higher-resolution clustering and trajectory inference.
- Graph-based methods for clustering and trajectory inference allow for the scalable analysis of large single-cell RNA sequencing datasets.
- Visualization methods can distort the structure of the data and batch correction methods can reduce cell-type resolution; both methods should therefore be used with care and might require specific parameter tuning for each dataset.
- High-level biological interpretation, such as cell-type annotation, remains challenging and time-consuming — new automated methods, alongside the creation of single-cell reference atlases, promise to address these issues.

For example, some methods simultaneously profile RNA and chromatin accessibility[14], RNA and methylation[15], or even a combination of chromatin accessibility, RNA and methylation, albeit at a lower throughput[16].

In this Review, we provide the non-expert reader with a broad overview of the different steps required for scRNA-seq analysis, including pre-processing of data and downstream analysis (FIG. 1). We discuss challenges that are typically encountered in every step of scRNA-seq data analysis and examine the different computational tools and approaches developed to address these issues, including their strengths and their limitations. We also explore how experimental design choices can affect downstream data analyses. In-depth, technical explanations of specific scRNA-seq analysis steps are available elsewhere[17–19].

Data pre-processing

The raw data obtained from scRNA-seq platforms must first go through several pre-processing steps before it can be used to assess biologically relevant changes in gene expression. These pre-processing steps transform the raw data into a more usable format and address issues related to sample quality, the wide range of gene expression levels and variance. Additionally, these steps can reduce the impact of technical batch effects if multiple datasets are to be analysed simultaneously.

Generating a gene expression matrix

The initial output FASTQ file (or files) generated in an scRNA-seq experiment consists of complimentary DNA (cDNA) reads. Each read contains an RNA sequence, a cell barcode that identifies the cell from which the read was generated and a unique molecular index (UMI) that identifies the exact mRNA molecule[3–6]. The first step of scRNA-seq analysis is to process these reads into a counts matrix that summarizes the number of molecules of each gene detected in each cell in the dataset[4,20,21]. The counts matrix serves as the input for the remaining analysis steps and is also an efficient way of storing and sharing information on gene expression (BOX 1). The creation of a counts matrix typically involves aligning the cDNA sequence in each read to a reference genome to identify the specific gene that the read originated from and then assigning each read to its cell of origin through its cell barcode[4,20,21] (FIG. 2a).

scRNA-seq technologies use PCR to exponentially amplify cDNA molecules and UMIs enable users to identify and collapse duplicate reads that might be generated during this amplification step, thus reducing technical noise[22]. Of note, sequencing errors in the UMI can artificially inflate gene expression, as duplicate reads that should be collapsed are treated as distinct molecules[20,23]. Conversely, distinct molecules might be incorrectly labelled with the same UMI sequence and thus be treated as one molecule[20].

For most sequencing technologies, background RNA contamination and sequencing errors result in a large number of cell barcodes that have a low number of reads but do not correspond to real cells. These empty barcodes can be detected and removed by setting a minimum number of reads or a UMI threshold for cell barcodes. More sophisticated methods such as dropEst are also available[4,20].

Several tools can be used for read processing (TABLE 1), including CellRanger, which accompanies the 10X genomics Chromium scRNA-seq platform. CellRanger handles cDNA reads, runs sequence alignment, collapses duplicate reads by their UMIs and outputs a counts matrix along with quality control (QC) statistics[4]. CellRanger can also perform secondary analyses such as clustering (that is, grouping cells according to similarities in their patterns of gene expression) and visualization (discussed in more detail later), albeit using a rather basic pipeline[4]. However, CellRanger can be fairly slow and memory intensive, using a maximum of 30 GB of RAM and taking ~22 h to process 784 million reads (equivalent to ~50,000 cells at a depth of 15,000 reads per cell)[21]. Nevertheless, the integration of CellRanger with the Loupe Cell Browser, another piece of 10X genomics software, offers non-expert users an interactive browser that can be used to visualize the results of clustering and the expression of marker genes[4].

In the past few years, researchers have developed scRNA-seq methods that can profile hundreds of thousands to millions of cells in a single experiment by using combinatorial indexing. Such methods include split pool ligation-based transcriptome sequencing and single-cell combinatorial indexing RNA-seq[5,6]. Given these technological advances and considering the amount of memory and processing time required by CellRanger[21], alternative computational pipelines for processing cDNA reads into single-cell gene expression counts have also been developed[5,6]. The dropEst pipeline, for example, has faster runtimes and lower memory usage than CellRanger, and provides more accurate gene expression estimates by correcting sequencing errors in the cell barcodes and UMIs[20] (TABLE 1). DropEst also improves data recovery by using a machine learning model to identify empty barcodes, enabling the recovery of cell types that are smaller than average in size, and cell types with low RNA content that might otherwise be excluded from the analysis[20]. UMI-Tools is another pipeline that corrects sequencing errors in the cell barcodes and UMIs to provide more accurate quantification of gene expression[23].

One of the slowest steps in the CellRanger pipeline is the alignment of cDNA reads to the reference genome[24]. The Kallisto pseudo-aligner, used alongside the BUStools suite of methods for storing and manipulating scRNA-seq data, is a highly efficient alternative

FASTQ file
A text file that stores DNA sequences and their associated quality metrics and metadata; a single sequence in a FASTQ file is called a 'read'.

Counts matrix
An integer matrix (that is, numerical data arranged in a set of columns and rows) in which the columns typically correspond to cells, whereas the rows correspond to genes; each entry represents the number of molecules of that gene expressed in that cell.

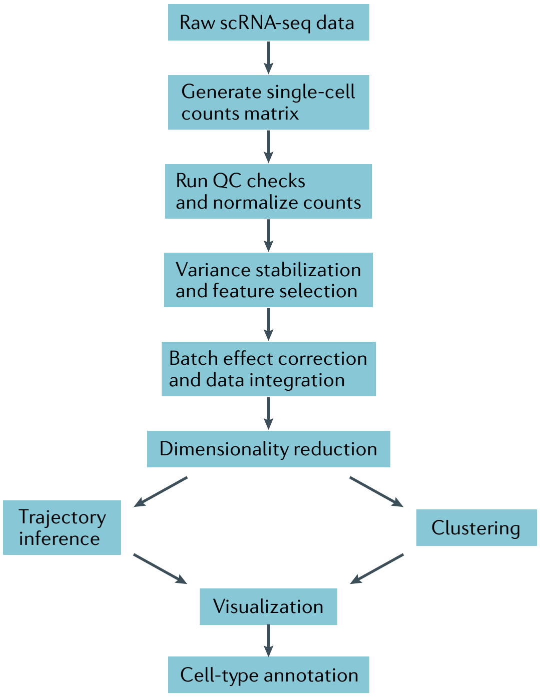

Raw scRNA-seq data
Generate single-cell counts matrix
Run QC checks and normalize counts
Variance stabilization and feature selection
Batch effect correction and data integration
Dimensionality reduction
Trajectory inference
Clustering
Visualization
Cell-type annotation

Fig. 1 | **Overview of the single-cell RNA sequencing analysis pipeline.** The raw data generated by single-cell RNA sequencing (scRNA-seq) contain all sequenced complementary DNA reads and the first analysis step consists of assigning individual reads to their cell of origin to generate a single-cell counts matrix. The next step involves filtering cells and genes according to quality control (QC) metrics. Data normalization, scaling and variance stabilization are used to address technical biases and facilitate the selection of the most biologically relevant genes, ensuring that the downstream analysis is driven by relevant biological phenomena and not technical noise. The comparison of datasets acquired from different experiments also requires the correction of batch effects to enable appropriate data integration. Dimensionality reduction summarizes the expression patterns of thousands of genes in fewer dimensions, which are used to create clusters of cells with similar patterns of gene expression. In developmental datasets, cells often do not group into discrete clusters but instead follow continuous trajectories, requiring a continuous model of cell states — trajectory inference aims to identify the location of cells along the developmental continuum. Finally, the dataset can then be visualized in two dimensions and analysed to identify key marker genes in each cluster. Any unknown cell types or states are then annotated using these key marker genes or through comparisons with existing reference datasets.

to CellRanger because it creates a list of compatible transcripts for each read (pseudo-alignment) instead of aligning individual reads to an exact position in the transcriptome (alignment)[21,24]. The combined Kallisto–BUStools method is up to 51 times faster than CellRanger and uses a maximum of ~12 GB of RAM when processing 50,000 cells[21] (TABLE 1). However, Kallisto–BUStools does not remove empty cell barcodes. STARSolo and Alevin are extensions of two alignment and pseudo-alignment methods, respectively, that can also be used for processing of scRNA-seq data[25,26] (TABLE 1). Both STARSolo and Alevin have significantly faster runtimes than CellRanger, but STARSolo has a higher maximum RAM usage[21].

In summary, the first step of scRNA-seq analysis is to process raw reads into a matrix of single-cell gene expression counts. For users of the 10X genomics scRNA-seq platform, CellRanger offers a convenient, albeit slow and memory-intensive method for this processing. CellRanger also runs basic clustering and marker gene analysis that can be visualized with the Loupe Cell Browser. DropEst, Kallisto–BUStools, UMI-Tools, STARSolo and Alevin are alternative read processing methods that offer substantial runtime and memory improvements, enabling users to process their scRNA-seq runs without having to invest as much in computational infrastructure. Additionally, the enhanced correction of UMI and cell barcode errors available with DropEst, UMI-Tools and Kallisto–BUStools can improve gene expression estimates compared with CellRanger.

Quality control and doublet detection

All scRNA-seq methods generate technical biases and noise — some basic QC addresses these issues before downstream analysis. Protocols used for single-cell dissociation and sequencing, for example, can induce cellular stress and result in cell death, which biases gene expression and can result in artificial clusters of dead cells in downstream analyses[27]. Filtering out cells with either a low cDNA read or UMI count, as well as cells with a large number of mitochondrial reads per total number of UMIs (also known as mitochondrial fraction) can help to remove dead cells[2]. Unlike cytoplasmic RNA, the presence of mitochondrial RNA is indicative of cell death. The appropriate threshold for the number of reads or UMIs, and mitochondrial read fraction depends on the cell types present in the dataset and the scRNA-seq method being used. Setting a threshold for the minimum number of cells in which a gene is detected can also help to exclude genes that are only expressed in a small number of cells and are unlikely to be informative. However, users should ensure that this threshold is not too high, as rare cell types might be otherwise missed in the downstream analysis.

For most scRNA-seq methods, the presence of doublets, generated when two or more cells are assigned to the same cell barcode, can create artificial clusters in the downstream analysis, as merging the gene expression patterns of two distinct cell types might create a unique expression signature that is not found in any real cell type. However, manually differentiating doublet clusters from true clusters can be challenging, especially for large datasets with many cell types[28]. One common strategy for identifying doublets involves generating simulated doublets by combining cells from different clusters in the dataset and assessing which cells have similar expression profiles to the simulated doublet cells[28]. However, this strategy is only feasible when the dataset contains discrete cell types, rather than continuous cellular trajectories[28].

QC thresholds might differ between datasets and some exploratory data analysis, such as histograms of the distribution of UMIs per cell or gene, can help to set thresholds for each dataset. In some cases, such as when an artificial cluster of dead or dying cells becomes apparent in the downstream analysis, modifying these thresholds after running the entire analysis pipeline and repeating the analysis can also be helpful (FIG. 2b). Seurat[29] and SCANPY[30] are scRNA-seq analysis pipeline packages that include functions for computing QC metrics, such as the fraction of genes expressed per cell, mitochondrial fraction and total counts; users determine the thresholds with which to filter genes and cells in the dataset. Scater[31] also offers a suite of tools for computing key QC metrics.

Data normalization

The fraction of RNA captured in each cell can vary owing to factors such as reverse transcription efficiency, primer capture efficiency and errors associated with collapsing UMIs[2,32,33]. Differences in the total amount of UMIs or reads in each cell might thus result from technical factors rather than biological variation (FIG. 2c). If not normalized, technical differences in total UMIs or reads can dominate the downstream analysis. For example, cells with similar amounts of total UMIs or reads cluster together instead of cells with similar gene expression patterns[33]. Normalization is therefore crucial to revealing the true biological heterogeneity of a dataset. Most normalization methods attempt to estimate the bias for each cell (also known as a size factor). The UMI or read counts of all cells can then be normalized by dividing those values by the size factor, enabling the comparison of gene expression levels across different cells. Total counts normalization is a simple normalization strategy, in which the size factors consist of the total number of UMIs or reads in each cell. However, total counts normalization can be dominated by highly expressed genes and results in biased size factor estimation when strong cell-type-specific gene expression exists, which can occur when very different cells or tissue types are present in the dataset[33,34]. Also, some cell types are larger and have more RNA molecules than others, a biological factor that is obscured when simply dividing the number of UMIs or reads by the total counts[34].

The scran package pools cells with similar expression patterns before estimating size factors, therefore addressing normalization issues due to cell-type-specific gene expression or UMI counts[34]. However, scaling genes with high expression and low expression using the same size factor can lead to overcorrection of genes with low expression, such as transcription factors, and under-correction of genes with high expression, such as housekeeping genes[35,36]. SCnorm addresses this issue by pooling genes with similar dependencies on total UMI or read count and computing size factors within each pool[35]. sctransform (implemented in the Seurat package) uses a probabilistic model to compute the effect of total UMI or read count on each gene, which also enables it to stabilize gene variances (discussed later in more detail) and identify over-dispersed genes[36].

Overall, some type of normalization is crucial for scRNA-seq analysis and, although total count normalization successfully mitigates technical bias, it can partially obscure true biological heterogeneity. Using specialized normalization methods, such as SCnorm and sctransform, can unlock additional heterogeneity in a dataset[33].

Total counts
The total number of reads or UMIs in a given cell.

Size factor
An estimate of how much variation in sequencing depth or RNA capture efficiency affects the overall quantification of gene expression in a cell.

Over-dispersed genes
Genes that show a greater than expected variance between cells given their average expression, which suggests that they are expressed in a cell-type-specific manner.

Box 1 | **Dataset uploading and exploration**

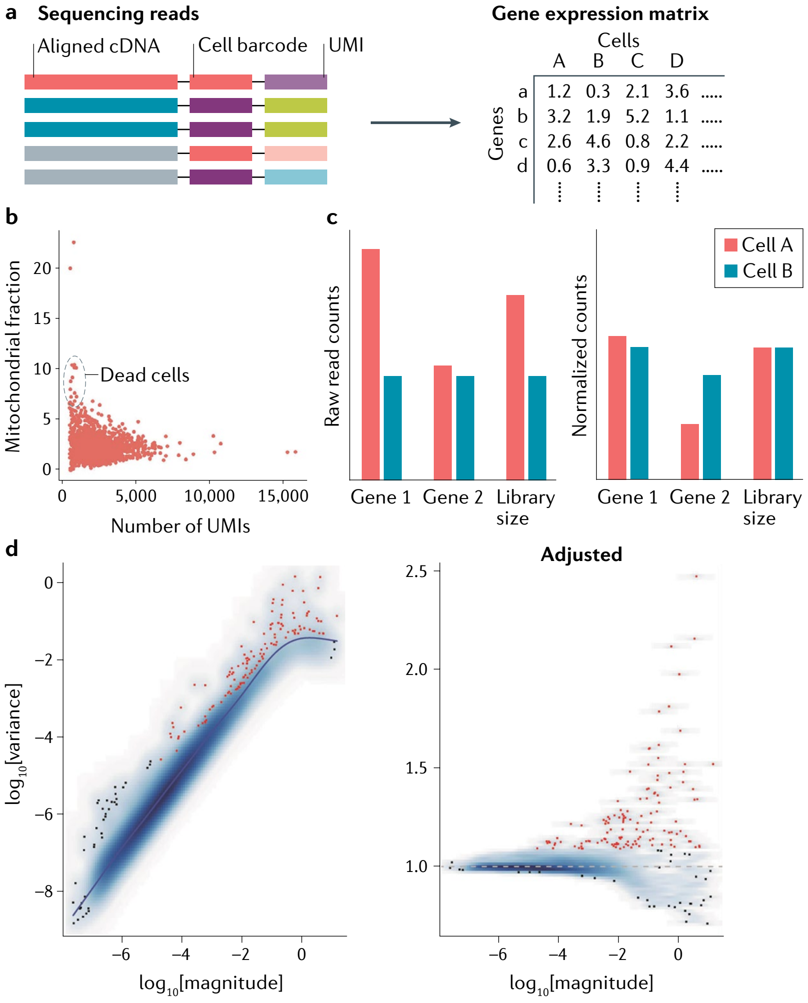
A crucial step in the analysis of single-cell RNA sequencing (scRNA-seq) data is the assessment of related available datasets, not only to enable a better understanding of existing work but also to assist in the formulation of novel hypotheses. Online data browsers are particularly useful for the investigation of existing scRNA-seq datasets. University of California Santa Cruz (UCSC), the Broad Institute and European Molecular Biology Laboratory-European Bioinformatics Institute (EMBL-EBI) have single-cell data browsers that enable users to interactively visualize the different cell types and marker genes present in these datasets. The convenient user interfaces of these browsers allow users to quickly explore the datasets without the need to download the data or previous programming experience. Additionally, uploading a dataset to these browsers increases the visibility of the study that generated the data and maximizes the potential for new insights from the dataset.

Variance stabilization

Gene expression levels can vary enormously and the average expression (or magnitude) of a gene is strongly associated with its variance[37], an effect known as the mean–variance relationship (FIG. 2d). Variance stabilization adjusts the data to remove the influence of gene expression magnitude on gene variance. This step ensures that downstream analyses are focused on the most biologically relevant genes (that is, the genes that are expressed in specific cell types in the dataset) rather than simply focusing on the genes with the highest expression. For example, variance stabilization might facilitate the separation of two subpopulations of a developmental progenitor, which might otherwise be merged, by enabling genes with low average levels of expression, such as transcription factors, to still contribute to the analysis. Despite having low overall expression levels within a cell, such genes might be important in uncovering the fate of that cell.

One simple variance-stabilizing approach is to log-transform normalized counts, which reduces the difference between genes with high and low expression[38] (FIG. 2d). Pipelines that can be used to remove the effect of average gene expression on gene variance include Seurat, Pagoda2 (FIG. 2d) and SCANPY, which explicitly fit a mean–variance relationship and apply a scaling factor[30,37,39,40]. ZINB-Wave, single-cell variational inference (scVI) and deep count autoencoder (DCA) are alternative methods that use a different approach (negative binomial distribution) to model single-cell count data[36,41–43].

Overall, although variance stabilization is not strictly necessary for scRNA-seq analysis, adjusting the dataset for the wide variation in average gene expression enhances the contribution of biologically relevant genes to downstream analyses. This approach removes the influence of genes, such as housekeeping genes, which are abundantly expressed but at similar levels in all cells



	A	B	C	D	
a	1.2	0.3	2.1	3.6	.....
b	3.2	1.9	5.2	1.1	.....
c	2.6	4.6	0.8	2.2	.....
d	0.6	3.3	0.9	4.4	.....
	⋮	⋮	⋮	⋮	

Fig. 2 | **Pre-processing of single-cell RNA sequencing data. a** | The first step of single-cell RNA sequencing (scRNA-seq) data pre-processing involves the generation of a gene expression counts matrix from raw sequencing reads. These reads contain a cDNA sequence, a cell barcode that identifies the cell from which the cDNA was amplified, and a unique molecular identifier (UMI) that identifies the RNA molecule. The matrix comprises the gene expression values for the complete dataset, organized by gene (rows) and single cell (columns). **b** | Common quality control metrics include mitochondrial fraction and UMI count. These metrics can be used, for example, to identify and exclude cells with a high mitochondrial fraction and low UMI count, which might correspond to dead cells. An example plot was generated using the Seurat scRNA-seq analysis pipeline. **c** | Normalization adjusts data for cell-specific differences in total UMI count and reveals true gene expression differences between cells. In this example, discrepancies in library size (that is, the total number of reads in a cell) masked the variation in the expression of gene 2 between cells A and B. **d** | Variance stabilization facilitates the identification of the genes with the highest variance in a dataset by transforming the data to ensure that the analysis is not dominated by genes that, despite being expressed at high levels, do not vary greatly across the dataset. The left panel shows a mean–variance fit from Pagoda2, which demonstrates the relationship between average gene expression (*x* axis) and gene variance (*y* axis). The right panel shows residual variances after adjusting for the mean–variance relationship (that is, the correlation between the magnitude of expression of a gene and its variance). Data depicted in parts **b** and **d** were obtained from a dataset of peripheral blood mononuclear cells sequenced using the 10X genomics Chromium scRNA-seq platform.

and are therefore not useful to investigations of cellular heterogeneity. After variance stabilization, identifying and selecting highly variable genes can improve the resolution of cell types in downstream analyses, especially if the cell types being assayed are fairly similar[44]. This optional processing step involves selecting the genes with the highest residual variance after adjusting for the differences in average gene expression.

Batch effects and data integration

Joint analysis of multiple scRNA-seq datasets generated using different technologies, obtained from different patients or samples, or from different experiments, increases the total number of cells analysed. This approach can improve the resolution of cellular subtypes and the detection of rare cell phenotypes, and also enables direct comparisons of patients, samples or technologies. However, this type of analysis is often challenging owing to batch effects — technical differences in gene expression can mask relevant biological phenomena[29,45–48]. Intra-batch variation is typically due to differences between cell types and biologically relevant factors, whereas inter-batch variation might also result from technical factors. These batch effects can arise from variability in patients, samples or protocols (including operator-driven variation) that affect RNA capture efficiency or cell viability[45]. The strength of a batch effect depends on the type of dataset and can be difficult to predict before running the analysis.

A simple approach for eliminating technical batch effects is to essentially assume that each batch must have the same average gene expression across all cells and remove any differences across batches using a regression model[49]. Although this batch correction approach works for bulk RNA-seq data, it can over-correct when the different batches are not identical[29,46,47,50]. Specifically, if the cell-type proportions differ between batches, this type of crude batch correction might have an impact on the ability to resolve cell types[40,46]. For example, if one kidney sample contains more collecting ducts than another sample that is enriched for proximal tubular cells, then the average gene expression across the cells from each sample is different owing to the differences in cell-type composition. Applying a batch correction that forces the average gene expression across cells from both samples to be the same reduces the magnitude of the gene expression differences between collecting ducts and tubules, reducing the ability to resolve those cell types.

Methods that are tailored specifically for the integration of scRNA-seq data enable the preservation of differences in cell-type proportions between batches while eliminating batch effects[50]. Most of these methods rely on the concept of finding pairs of cells that correspond to the same cell type or state across different batches[40,46,47,51] (FIG. 3a). Once these pairs, also known as mutual nearest neighbours (MNNs), are identified, any remaining gene expression differences between MNNs are assumed to be due to batch effects and can be corrected[40,46,47]. An advantage of the MNN approach is that if a cell type or state is unique to a specific batch, it is not identified as an MNN, thus preserving the unique biological properties of each batch[40,46,47] (FIG. 3b). For example, when using an MNN approach to integrate kidney scRNA-seq data from two mice, one wild-type control and one genetic knockout that lacks podocytes, the podocytes from the control mouse would remain in

Table 1 | **FASTQ processing tools**

Method	Description	Documentation	Detects empty barcodes	Ref.
CellRanger	Default 10X genomics software package for processing data generated on the 10X platform	https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger	Yes	4
DropEst	Improves on quantification accuracy compared with CellRanger. Supports 10X, Split-seq, Drop-seq, inDrop, iCLIP and Seq-Well	https://github.com/hms-dbmi/dropEst	Yes	20
Kallisto–BUStools	Extremely efficient memory and CPU usage through the use of the BUStools file formats. Supports any platform that uses cell barcodes	https://www.kallistobus.tools/getting_started	No	21
Alevin	Extension of the Salmon pseudo-aligner for scRNA-seq data. Supports 10X and Drop-seq platforms	https://salmon.readthedocs.io/en/latest/alevin.html	Yes	26
STARSolo	Extension of the STAR read aligner for processing single-cell data. Supports the 10X platform	https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf	Yes	25
UMI-Tools	Models potential errors in UMIs and corrects them to improve gene expression accuracy	https://github.com/CGATOxford/UMI-tools	Yes	23

CPU, central processing unit; scRNA-seq, single-cell RNA sequencing; UMI, unique molecular index.

a separate cluster after integration. Of note, if the knockout caused a uniform shift in gene expression across all podocytes rather than podocyte loss, that shift might be lost after integration as it would be indistinguishable from a batch effect.

Identifying MNNs can be difficult if the batch effects are stronger than the differences in gene expression between cell types. To overcome this challenge, canonical correlation analysis can be applied to focus the analysis on intra-batch variation and not on inter-batch variation[29], even if the differences between batches are stronger than the differences between cell types[40].

One caveat of these methods of data integration is that a compromise between reducing the size of the batch effect and resolving cell types might be required; this parameter can be explicitly tuned in methods such as clustering on network of samples (CONOS)[47]. For example, completely removing the batch effect from kidney cells collected from two different patients might reduce the ability to resolve podocyte subtypes. The extent of this compromise depends on the specific datasets being integrated and the strength of the batch effects.

Overall, the advent of MNN-based methods has enabled scRNA-seq users to analyse and compare samples across platforms, patients or samples, and even across species, improving the capacity of scRNA-seq to resolve cell types and trajectories[29,40,46,51].

Downstream analyses

Once the pre-processing steps are completed, downstream analysis steps, which include dimensionality reduction, clustering and trajectory inference, focus on identifying patterns in the data that provide biological insight. Dimensionality reduction involves transforming the dataset into a more compact, and possibly more interpretable, representation that captures the primary biological axes of variation and improves the performance of clustering and trajectory inference. Clustering refers to partitioning cells into groups based on similar patterns of gene expression; these groups (also known as clusters) usually correspond to distinct biological cell types or states[52]. Trajectory inference is usually applied to cells that are dynamically transitioning across a continuum of cellular states[52,53].

Upstream analysis choices, such as QC filtering and normalization, can have a substantial impact on both clustering and trajectory inference. For example, data normalization is a critical step, otherwise clusters are almost entirely based on the number of UMIs or reads of the cell rather than on similarities in gene expression profiles. Dead or dying cells, as well as doublets, can also generate artificial clusters that might be difficult to distinguish from real clusters if not removed.

Dimensionality reduction and imputation

The dimensionality of a dataset refers to the number of variables being measured for each data point. In the context of scRNA-seq, each data point corresponds to a cell and the variables are the genes. scRNA-seq experiments are characterized as 'high dimensional' as they typically measure the expression of ~20,000 variables (genes). Even after selecting only a subset of highly variable and/or biologically relevant genes, users often still have a dataset with thousands of genes, many of which are highly correlated and provide redundant information, potentially masking more subtle biological patterns[52,54,55]. Additionally, the metrics used to measure similarity in gene expression patterns between cells become less reliable in a high-dimensional space, a phenomenon known as the 'curse of dimensionality'[52,54]. Therefore, applying dimensionality reduction to scRNA-seq datasets can improve downstream analyses. The reduced dimensions are typically called an embedding of the dataset. Dimensionality reduction has the added benefit of

Regression model
A model that compares the relationship between two variables. In the context of single-cell RNA sequencing, regression can assess relationships between observed gene expression, and technical and/or biological factors.

Mutual nearest neighbours
(MNNs). Cells from different batches that belong to each other's set of *k*-nearest neighbours (that is, cells with the most similar gene expression patterns).

Dimensionality reduction
Summarizing a large set of variables with a smaller set of variables, while retaining as much information as possible.

Embedding
The set of variables that remains after running some form of dimensional reduction.

Dropout
The absence of a detectable gene or transcript in a cell.

improving the speed of most downstream analyses. However, although it is extremely helpful for most datasets, dimensionality reduction is not strictly necessary for downstream analyses.

Linear methods. A linear relationship between two variables exists when both variables change at the same rate (direct proportion). The most common dimensionality reduction method for scRNA-seq analysis is principal component analysis (PCA), which creates a linear combination of genes that best capture the variance in the data[56] (TABLE 2). The ability of PCA to reduce the dimensionality of the data while finding the dimensions of highest variance makes it a very useful dimensionality reduction tool before clustering.

Only a relatively small fraction of the total RNA of a cell is captured and reverse transcribed in an scRNA-seq experiment. Consequently, no molecules are detected for many genes in most cells, resulting in a large amount of zeros in the single-cell counts matrix, which is known as zero inflation[3,4]. Zero-inflated factor analysis (ZIFA) is a variation of PCA that is designed to explicitly model the expected high amount of zero values in scRNA-seq count data[57] (TABLE 2).

One downside of PCA is that the principal components themselves can be difficult to interpret biologically. Ideally, each dimension obtained after dimensionality reduction would correspond to a biological process. For example, for a developmental kidney dataset, each dimension would correspond to a developing kidney compartment (for example, the collecting duct or the tubule). The factorial single-cell latent variable model (f-scLVM) addresses this interpretability issue by explicitly modelling annotated gene sets as the reduced dimensions[58] (TABLE 2). Therefore, after running f-scLVM, each reduced dimension corresponds to a pre-annotated gene set. Pagoda and Pagoda2 also create highly interpretable dimensions by running PCA within pre-annotated gene sets and selecting the dimensions that show significant variance in the dataset[37,39] (TABLE 2). Non-negative matrix factorization (NMF) is another linear matrix factorization method that generates more interpretable dimensions by attempting to find discrete components (such as a collecting duct or tubule) that underlie the dataset[59,60].

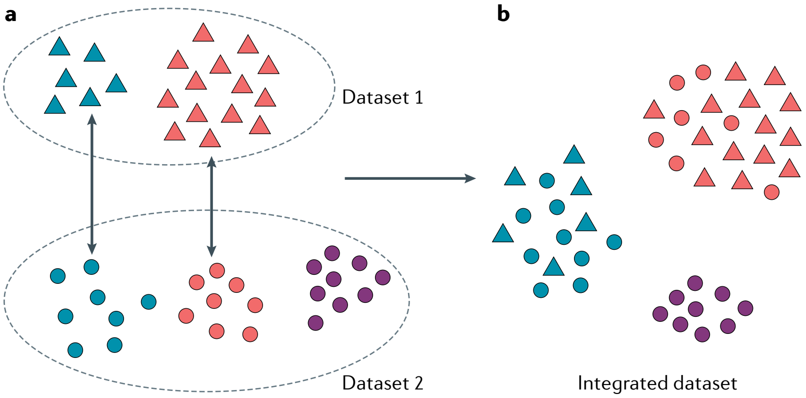


Fig. 3 | **Integration of single-cell RNA sequencing data. a** | The first step of data integration based on mutual nearest-neighbour data involves the identification of matching cell types across datasets. **b** | These matching cell types can then be grouped together and integrated into one dataset, while preserving any biologically relevant cell types that are unique to different datasets.

Non-linear methods. The relationship between genes can be highly non-linear, which affects the ability of linear models such as PCA to analyse scRNA-seq data[42]. Methods that can generate a non-linear transformation of the dataset can thus outperform linear methods in certain cases. Specifically, locally linear embedding (LLE) and diffusion maps (Dmaps) were shown to be effective when the dataset follows a continuous trajectory, such as with datasets from developmental time series[61–63] (TABLE 2).

Another approach to non-linear dimensionality reduction is the use of deep neural networks, which are models that apply iterative, non-linear transformations to the dataset[64]. By layering these iterative transformations, deep neural networks can learn complex features of a dataset, which enables them to represent the data using fewer dimensions[64]. scScope and DCA use neural networks that can outperform linear dimensional reduction methods such as PCA[43,65] (TABLE 2). scVI also uses neural networks to create a framework for modelling gene expression in a way that enables the quantification of uncertainty for each gene expression estimate, while accounting for technical effects such as batch effects and zero inflation[42] (TABLE 2).

For users who are interested in simply reducing the dimensionality of the data and proceeding to clustering and visualization, PCA is a good default approach, but more specialized methods such as f-scLVM or scVI can generate low-dimensional embeddings that are either more interpretable or capture the non-linear structure of the data more faithfully[43].

Zero inflation and imputation. Zero inflation is a technical limitation of more recent high-throughput scRNA-seq methods and is driven by several factors, including incomplete reverse transcription or RNA capture. Total efficiency calculations estimate that only 10–15% of the total RNA in a cell is captured and transcribed[3–5]. Of note, some researchers argue that the zero inflation for droplet-based methods is mostly due to biological variance and not due to technical noise[66]. However, the newer generation of combinatorial indexing methods tends to capture even fewer molecules per cell than droplet-based methods and technical zero inflation might thus be present in those datasets[5,6]. Several methods have been developed to impute these missing values (that is, to replace the zeros in the counts matrix with estimated values). One class of methods, including MAGIC and kNN-smoothing, uses information from neighbouring cells to impute missing values for any given cell[67,68]. Another class of methods such as single-cell analysis via expression recovery, clustering through imputation and dimensionality reduction (CIDR) and scImpute use probabilistic models and relationships between genes to distinguish technical from biological dropout[69–71]. However, these imputation methods should be used with care as they can introduce false-positive results when analysing differential gene expression[72].

Table 2 | **Methods of dimensionality reduction**

Method	Description	Documentation	Ref.
PCA	Default dimensionality reduction method for most single-cell pipelines	Implemented in Seurat, SCANPY and Pagoda2; https://github.com/ujjwalkarn/DataScienceR/blob/master/PCA.R	56
ZIFA	Variation of PCA that accounts for zero inflation in the counts matrix	https://github.com/epierson9/ZIFA	57
f-scLVM	Uses latent variable modelling and gene sets to generate interpretable lower dimensional factors	https://github.com/bioFAM/slalom	58
Pagoda2	Runs PCA on gene sets to identify interpretable components and find the ones with the highest variability for the given dataset	https://github.com/hms-dbmi/pagoda2	39
NMF	Generates a more interpretable dimensional reduction in which each dimension typically corresponds to a group of genes expressed in a group of cells	https://github.com/linxihui/NNLM	60
LLE	Generates a piecewise locally linear dimensional reduction that can capture non-linearity in the data. Works well for capturing trajectories	https://github.com/jw156605/SLICER	63
Dmaps	Generates a smooth dimensional reduction under the assumption that the cells follow a continuous path	https://github.com/theislab/destiny	62
DCA	Uses a deep neural network to encode the dataset into lower dimensions	https://github.com/theislab/dca	43
scScope	Uses a recurrent neural network to remove technical noise and then encode the dataset into lower dimensions	https://github.com/AltschulerWu-Lab/scScope	65
scVI	Uses probabilistic modelling with deep neural networks to generate a lower dimensional embedding of the dataset	https://github.com/YosefLab/scVI	42

DCA, deep count autoencoder; Dmaps, diffusion maps; f-scLVM, single-cell latent variable model; LLE, locally linear embedding; NMF, non-negative matrix factorization; PCA, principal component analysis; scVI, single-cell variational inference; ZIFA, zero-inflated factor analysis.

Therefore, users should be cautious when analysing differences in genes with low levels of expression and high levels of dropout.

Clustering

Generally, most scRNA-seq datasets either comprise discrete cell types or reflect a continuous trajectory of development or differentiation. For datasets in which individual cells can be grouped into discrete cell types, clustering needs to be applied to resolve those cell types. Each cluster generally expresses a set of genes (marker genes) that are not expressed in cells from other clusters (FIG. 4). *k*-means clustering is a simple and popular clustering method that iteratively assigns cells to clusters[73]. However, *k*-means clustering requires users to pre-specify the number of cell clusters present in the dataset and determining the number of biologically relevant clusters in an scRNA-seq dataset remains a challenge[52,73]. One strategy for dealing with this problem is to generate more clusters than those expected to be found in the dataset and then iteratively either merge neighbouring clusters or divide larger clusters based on a similarity threshold, such as the number of differentially expressed genes between clusters. CIDR, BackSPIN and pcaReduce use this hierarchical clustering approach[70,74,75]. Users can then select the groupings that best match the required level of cluster granularity. Hierarchical analysis with multiple stages of clustering might be necessary for extremely large datasets (>100,000 cells) with many different cell types. This approach, used for example in a study of the mouse nervous system[76], requires an initial broad identification of well-defined cell types, which are then sub-clustered to further resolve their heterogeneity.

Both *k*-means and hierarchical clustering methods are slow to run for large datasets and are limited in the types of clusters they can detect[77]. Seurat, Pagoda2, SCANPY and CellRanger use graph-based clustering algorithms, which tend to run quickly and generate biologically relevant clusters for larger datasets[4,29,30,39]. Graph clustering requires building a graph by connecting each cell to its nearest neighbours. The Louvain clustering algorithm, for example, can be applied to cells that have been connected in a graph. Starting with each single cell as its own cluster, the algorithm iteratively merges clusters as long as the merging increases the modularity of the graph (the higher the modularity, the lower the likelihood that cells were connected in the network by random chance)[78]. However, the Louvain method can sometimes generate erroneous clusters composed of cells that are not well connected[79]. Leiden clustering improves on Louvain clustering by guaranteeing well-connected clusters and improving runtime[79].

Other approaches include SC3 Consensus Clustering, which uses the consensus of multiple clustering methods to improve clustering accuracy[80]. Reference component analysis projects single cells onto a low-dimensional space defined by existing bulk RNA-seq datasets, which can be very useful for cell populations that are highly heterogeneous and difficult to interpret such as those found in cancer[81]. Overall, graph clustering methods such as Leiden or Louvain have a strong clustering performance with fairly fast running times.

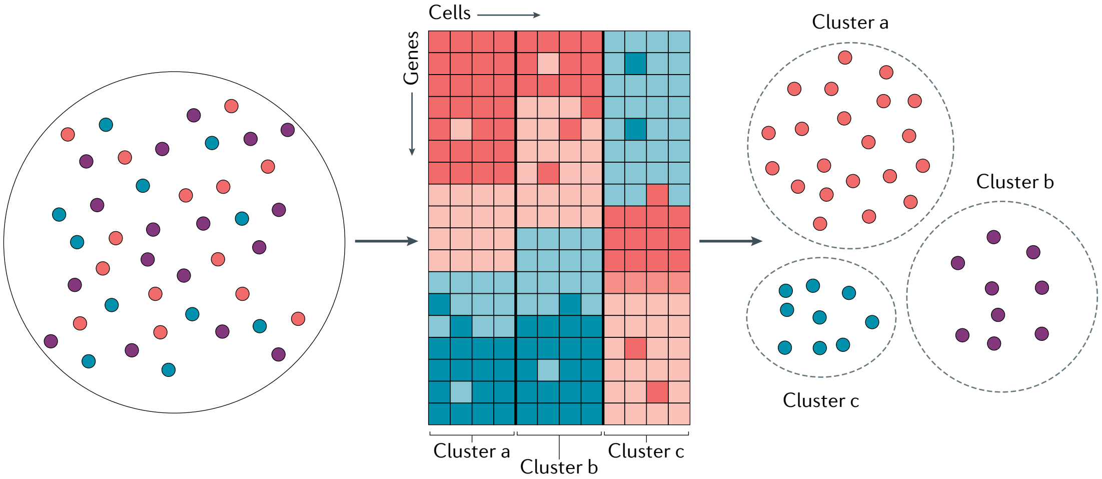


Fig. 4 | **Cell clustering in datasets with discrete cell types.** An important objective of single-cell RNA sequencing analysis is to resolve the cellular heterogeneity of a dataset by identifying the different subpopulations present. Cell clustering identifies and groups cells from a heterogeneous dataset into clusters, according to similarities in their patterns of gene expression, as illustrated in the heatmap. These cell clusters usually correspond to different cell types present in a dataset.

Trajectory inference

Although clustering is useful for grouping cells into discrete cell types, in many cases the gene expression patterns of cells form a continuum as they transition between cell states[52,53] (FIG. 5). For these datasets, clustering is generally performed first to identify the cell states that the trajectory runs through, as well as any cell states that are not part of the trajectory. Compared with a dataset that mainly comprises discrete cell types, a continuum of cell states is generally characterized by the presence of fewer discrete marker genes and more genes that are expressed along a continuous gradient (FIG. 5b). For example, during kidney development in mice, cells differentiate from nephron progenitor cells to proximal and distal tubules in a continuous manner[82]. In these types of studies, assigning cells to a specific point in development along this continuum is an important analysis objective; this approach is known as pseudotime estimation[52,53]. Identifying the point where a continuum splits into different branches is also important, as these branch points represent key fate decisions[53,83,84] (FIG. 5a). In one study, identifying the branch point where progenitor cells separate and either become proximal or distal tubular cells enabled researchers to identify key regulators of tubule development in mice[82]. Analysing this type of continuous cell state data is generally known as trajectory inference and is a highly active area of research. A study that compared 45 different trajectory inference methods concluded that, owing to large differences in the structure of these continuous datasets, no single method performs well in all cases[53], suggesting that multiple methods should be tested for any given dataset. Additionally, many datasets include both discrete and continuous components; thus, it might be necessary to use both clustering and trajectory inference during analysis.

One common issue with trajectory inference is that biologically dissimilar cells might be placed close to each other on this continuum owing to technical or biological noise, a phenomenon known as 'short circuiting'[53,85]. One interesting approach to dealing with this issue is partition-based graph abstraction (PAGA), which was one of the few methods that performed well on most datasets in the aforementioned comparison study, while maintaining a reasonable computational runtime[53,85]. In an approach that resembles graph clustering, PAGA generates a nearest-neighbours graph of the data and then generates a grouping of the cells, connecting groups that have more connections between cells than one would expect by random chance, to construct a summary graph of the data[85]. As 'short circuits' are more easily identified in connections between groups of cells than in connections between individual cells, PAGA prunes spurious connections between groups[85]. Monocle3 builds on this approach by constructing a cell-level graph with connections between individual cells, where any connections between groups of cells that are not connected in the summary graph are pruned away[84].

Many methods have been developed to identify the position of a cell along a developmental trajectory, but they do not provide information on the direction of the trajectory. One approach to predicting the transcriptional direction of a cell is to estimate RNA velocity[86]. This method is based on an assessment of whether the RNA molecule is spliced or unspliced (that is, nascent RNA that still contains intronic sequences)[86]. A high ratio of unspliced to spliced RNA for a given gene indicates that the expression of the gene is increasing, as the higher amount of unspliced RNA suggests that more of the RNA is being transcribed than degraded. Conversely, a high ratio of spliced to unspliced RNA for a given gene is indicative of decreasing gene expression[86]. RNA velocity is thus able to predict the future gene expression state of a given cell[86] and can help to determine, for example, whether a nephron progenitor is primed to become a proximal or a distal tubular cell.

Visualization

After clustering and/or trajectory inference, the next step is to generate a 2D or 3D scatterplot of the cells to visualize major trends and trajectories in the data.

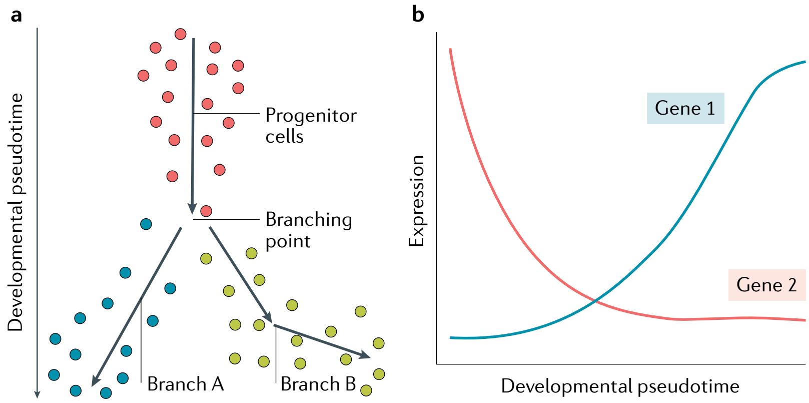


Fig. 5 | **Modelling continuous cellular states.** In datasets that are characterized by the presence of a continuum of cell states, the objective of the single-cell RNA sequencing analysis is to model the cellular trajectory. This process includes computing the developmental pseudotime (that is, the approximation of how far a cell has progressed along a developmental or differentiation pathway), the branch identities (for example, distal and proximal tubule branches for cells in these distinct developmental pathways) and the location of the branching point (for example, the point where nephron progenitors split into cells that will either develop as distal or proximal tubule cells). **a** | Illustration of a dataset with a continuous developmental trajectory that branches off into two lineages. The arrow represents the direction of developmental pseudotime. **b** | Example of a typical pattern of gene expression in a dataset with a continuous trajectory.

Although this step is conceptually identical to dimensionality reduction, visually separating closely related cell types (maintaining the local structure of the data) in just two or three dimensions, while also ensuring that the relative distances between cell types and trajectories reflects the magnitude of the gene expression differences between those cell types (maintaining the global structure of the data), is a complex task. Many linear dimensional reduction methods, such as PCA, are unable to generate accurate visual representations of the data in two or three dimensions[87,88]. Thus, visualization methods tend to transform the data using a non-linear method, which can distort the structure of the data if not used correctly[87,89–91].

t-stochastic neighbour embedding (t-SNE) is one of the most popular visualization methods and uses pairwise similarities of cells to embed them in a low-dimensional space, ensuring that cells with similar gene expression profiles are close in the embedding[88,92]. t-SNE thus prioritizes the local structure of the data, essentially ensuring that neighbouring cells remain together in the 2D visualization[88,92] (TABLE 3). This feature enables t-SNE to visually separate complex datasets with closely related cell types. However, t-SNE, as traditionally implemented, does not visualize global properties of the datasets, such as relative distances between cell types, as effectively[88,92] (FIG. 6). t-SNE is currently implemented in Seurat, Pagoda2, SCANPY and CellRanger–Loupe Cell Browser.

In the past few years, uniform manifold approximation and projection (UMAP) has overtaken t-SNE as the default visualization method for scRNA-seq data[89,90]. Similar to graph clustering, UMAP generates a nearest-neighbours graph of the cells, weighting each cell–cell connection by the strength of similarity; the graph is then embedded in two dimensions[89]. UMAP can also be initialized with the PAGA graph to generate highly accurate visualizations of continuous developmental datasets[85]. In practice, UMAP has been found to perform as well as t-SNE in visualizing the local structure of datasets (FIG. 6a), including separating closely related cell types, while performing vastly better in terms of visualizing the global properties of the data (FIG. 6b). Thus, UMAP is a very useful default visualization option for most users[89,90]. Additional testing of UMAP and t-SNE has suggested that the way in which these methods are initialized is very important to their overall performance[93,94]. In fact, t-SNE and UMAP seem to perform equally well in terms of preserving global structure when initialized using PCA[93,94].

Similarity-weighted non-negative embedding (SWNE) uses NMF (TABLE 2) to reduce the dimensionality of the data and then uses the dimensions as a framework with which to project the cells in two dimensions, adjusting the relative positions of the cells using a weighted nearest-neighbours graph[87]. This framework also enables genes to be visualized alongside the cells, adding biological context and interpretability to the visualizations[87] (TABLE 3). SWNE performs better than t-SNE and is similar to UMAP in terms of capturing global structure, although its representation of local structure is inferior to both t-SNE and UMAP[87].

Potential of heat diffusion for affinity-based transition embedding (PHATE) uses a diffusion-based distance metric that is accurate for both local and global structure[95]. PHATE first computes local distances between neighbouring cells and then propagates those distances (in a manner similar to that of Dmaps) to compute global distances between all cells. PHATE seems to perform very well for datasets with developmental trajectories, outperforming both t-SNE and UMAP in capturing global and local structure.

Deep learning methods can also capture the structure of high-dimensional data in a 2D embedding owing to their ability to capture non-linearity in the data[96]. scvis uses a deep neural network to condense high-dimensional data into a low-dimensional embedding, which results in better cell-type separation (the ability to capture local structure) than t-SNE (as measured by classification accuracy), as well as faster runtimes[96] (TABLE 3). Other deep learning-based methods such as scScope, DCA and scVI can also be used to encode high-dimensional data in two dimensions[42,43,65,96].

Overall, visualization is critical for understanding and communicating the properties of a dataset. One common misconception is that clustering and visualization are identical analyses. Although clusters can be created based on UMAP or t-SNE coordinates, using more dimensions with a generalized method such as PCA to create cell clusters is typically more useful, because all the structure and nuances of a dataset cannot be accurately compressed into two or three dimensions. In fact, a benchmarking study found that dimensionality reduction methods that work well for clustering often do not work as well for visualization[55]. However, for trajectory inference, methods that are used for visualization, such as UMAP, Dmaps and LLE, generally work well as a basis for building trajectory graphs[63,84].

Classification
A machine learning task in which an algorithm learns the relevant features that distinguish the different classes of a training dataset to predict the classes of an unknown test dataset.

As a starting point, UMAP is a very useful default method that faithfully visualizes most datasets and requires less parameter adjusting than t-SNE or SWNE to work well. However, users still need to take care not to over-interpret visualizations, as all methods result in a certain degree of data distortion. Additionally, more research is needed on how different initializations of these non-linear methods can affect their overall performance.

Cell-type annotation

Often, the most time-consuming step of scRNA-seq analysis is the identification of the biological cell types present in the dataset. The standard protocol for this cell-type annotation is to find the genes that are uniquely expressed in each cluster and match those genes to lists of canonical cell-type markers[52,97]. Tools for marker gene discovery and visualization are included in Seurat[29], Pagoda2 (REF.[39]), SCANPY[30] and the Loupe Cell Browser[4]. An evaluation of marker gene discovery methods found that most methods developed for bulk RNA-seq, such as edgeR and limma, perform just as well as scRNA-seq-specific methods[98]. Nonetheless, the Wilcoxon method, which is the default method for both Seurat and Pagoda2, performed relatively well[98].

Interpreting the output of these marker gene discovery methods can be challenging for new users. For single-cell methods such as the Wilcoxon test, the *P* values are often extremely low as the test treats each cell as an independent replicate. In these cases, the log fold change of gene expression can be a helpful metric, as it is indicative of the magnitude of the difference in gene expression. When an experiment contains multiple biological or technical replicates, one useful approach is to create a pseudo-bulk counts matrix after clustering by summing or averaging the counts from cells in a single replicate and a single cluster[99]. Bulk approaches such as edgeR or limma can then be used to assess differential gene expression.

Manually examining lists of marker genes can be extremely time-consuming and requires knowledge of the biological system being studied. Close collaboration between biologists and bioinformaticians can thus be extremely helpful at this stage. One class of methods that can help to accelerate this process uses enrichment of marker genes in functional pathways and gene ontology terms, which can greatly enhance interpretability[100]. For example, a cell-type cluster with marker genes that are highly enriched for the gene ontology term ‘nephron epithelium development’ is likely to contain cells related to the nephron epithelium.

A second class of methods matches individual cells or clusters to either a single-cell or bulk reference RNA-seq dataset for automated cell-type classification. A benchmarking analysis of these automated classification methods[97] found that the best-performing method was the support-vector machine, a common type of machine learning classifier[101]. The analysis also found that methods that use previously known sets of canonical marker genes, such as Garnett[102], do not outperform unbiased methods[97]. Other automated cell-type annotation methods include scmap, which classifies scRNA-seq clusters using correlations with reference datasets and a feature selection approach based on machine learning[103], and scPred, which uses a combination of dimensionality reduction and classification[104].

Data integration methods such as Seurat, CONOS and Scanorama also offer automated cell-type classification methods[40,47,51]. These methods find the MNNs across datasets, which enables them to classify cell types in a dataset without pre-set cell-type labels, based on the labels of a reference dataset. For example, if a cell of unknown type has ten MNNs in the reference dataset, and nine of them are podocytes, the unknown cell is most likely to be a podocyte too.

Although automated cell-type annotation methods can be convenient, they require existing reference scRNA-seq datasets. If a dataset contains novel cell types or cell states, manual annotation with marker genes is still necessary. Of note, even with a reference dataset, manual inspection of marker genes is critical to validating the identified cell types. Nonetheless, as single-cell atlases such as the human cell atlas and other reference catalogues of single-cell gene expression become more widely available, the use of automated cell-type classification will become more widespread[105].

Table 3 | **Visualization methods for single-cell RNA sequencing data**

Method	Description	Documentation	Refs
t-SNE	Generates 2D visualizations that maintain cell-type separation but sacrifice global structure	Implemented in most single-cell pipelines; https://github.com/jkrijthe/Rtsne	88,92
UMAP	Creates a low-dimensional embedding of a nearest-neighbours graph that maintains cell-type separation and global structure	Implemented in Seurat and SCANPY; https://github.com/lmcinnes/umap	90
SWNE	Uses a combination of NMF and nearest-neighbours smoothing to generate a highly interpretable 2D embedding	https://github.com/yanwu2014/swne	87
PHATE	Specifically encodes local and global information by first learning local relationships and then uses data diffusion to learn global structure	https://github.com/KrishnaswamyLab/PHATE	95
scvis	Uses a deep neural network to encode data in two dimensions	https://bitbucket.org/jerry00/scvis-dev	96

NMF, non-negative matrix factorization; PHATE, potential of heat diffusion for affinity-based transition embedding; SWNE, similarity weighted non-negative embedding; t-SNE, t-stochastic neighbour embedding; UMAP, uniform manifold approximation and projection.

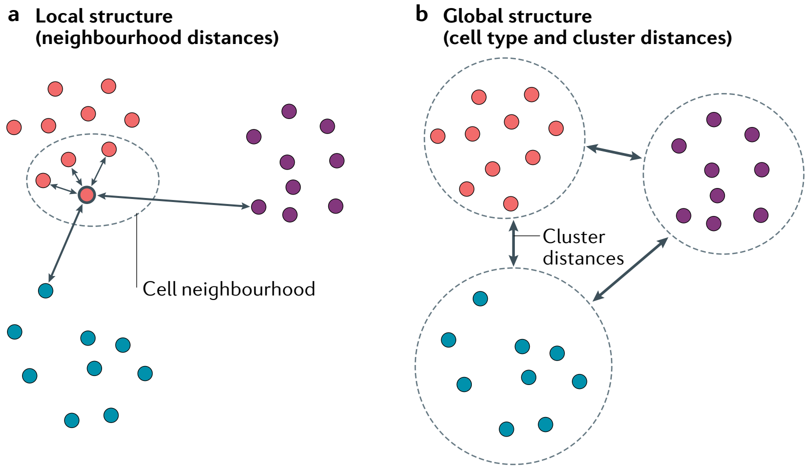


Fig. 6 | **Local and global structure in a dataset. a** | Preserving the local structure of a dataset ensures that the neighbouring cells of each cell remain together in the visualization, rather than preserving the original gene expression space. The distance between cells in the final visualization is therefore representative of the degree of similarity between their gene expression patterns. **b** | Preserving the global structure of a dataset ensures that large-scale distances, such as the distances between cell types, are maintained.

Experimental design considerations

Experimental design can have a substantial impact on analysis. For example, if multiple biological samples are to be collected and analysed, cells from each sample should ideally be tagged to allow multiplexing, using methods such as cell hashing, and then analysed on the same scRNA-seq run[106]. For example, in an analysis of kidney samples from five different patients across three scRNA-seq runs, each run would ideally contain tagged cells from each patient. This approach enables the distinction between sample-specific effects and experimental batch effects, which is especially crucial if the samples are from a case–control study[32]. For example, when comparing gene knockout mice with wild-type controls, cells from both types of mice are ideally run on the same experiment. Combinatorial indexing methods facilitate this approach as cells from different samples can be positioned in different wells during the first round of barcoding[5,6]. For droplet-based methods, some form of sample-specific cell tag is necessary to identify the sample source of a cell[106]. However, from a logistics standpoint, gathering all samples for processing in the same experimental batch is not always possible, especially for animal experiments across various conditions and/or timepoints, or for patient samples that are collected during clinical procedures.

The choice of scRNA-seq methodology also has an effect on the number of molecules captured per cell and the total number of cells analysed. In general, the combinatorial indexing methods capture fewer UMIs per cell than the droplet-based methods, which can affect their ability to resolve some closely related cell subtypes[4–6]. However, combinatorial indexing methods can capture far more cells per experiment, potentially enabling the identification of rare cell populations[4–6]. For all of these methods, the user can generally control the number of cells that are loaded onto the scRNA-seq platform. Loading more cells enables greater throughput at the cost of a potential increase in cell doublets[4].

Cell hashing
A technique that attaches unique molecular barcodes to multiple batches of samples for pooling and processing in one batch, which not only improves the experimental throughput but also reduces technical batch differences.

The choice of tissue dissociation methodology can also have a substantial impact on the types of cells available for analysis[107]. One key choice is whether to dissociate the sample into single cells or single nuclei. Dissociation of single cells has been widely applied to fresh tissue samples[107]. For frozen tissues, single-nucleus isolation and sequencing is a more viable option[107–109]. Both types of protocol seem to have their own specific biases, although for some sample types, such as human neurons, only single-nuclei dissociation has been shown to work well[107–109]. One limitation of single-nuclei methods is that they generally result in fewer molecules captured per cell, as most RNA is in the cytoplasm. However, information captured from nuclei alone can often be sufficient for accurate classification of cell types and subtypes[7].

Conclusions

Technical advances in scRNA-seq technologies have led to the generation of datasets of increasing scale and complexity. In response, an ecosystem of computational methods has been developed to deal with the challenges involved in analysing these datasets. Methods based on the identification of MNNs successfully integrate datasets across patients, conditions and technologies, addressing the crucial issue of batch effects in scRNA-seq data. Additionally, a number of methods have been developed to model cellular trajectories and identify cell clusters. However, one remaining limitation is that most clustering methods require users to specify the number of clusters and finding the optimal number of clusters for a given dataset is challenging. A second limitation is that manually annotating cell types using marker genes can be extremely time-consuming. Fortunately, new automated and semi-automated cell-type classification methods are being developed to address this issue, although novel cell types and states will still need to be manually annotated.

The ability to integrate datasets across samples, along with the increased throughput of the latest scRNA-seq methods, will increase our ability to resolve cell subtypes and discover rare cell types. Additionally, many newer methods, especially those used for low-level data pre-processing, take into account memory and central processing unit usage, which is critical, as the size of single-cell datasets continues to increase. Further development of these computational methods will help researchers to unlock additional biological insights. Despite these advances in computational methodology, validation of any computational findings by testing multiple biological replicates or conducting additional experiments, such as immunostaining or RNA-FISH, is still required.

The advent of multi-omics approaches will require a new set of tools that can link data on different cellular parameters, such as protein expression or epigenetic data, to provide additional biological insight. For example, analysing the relationship between gene expression and enhancer and/or promoter accessibility might delineate cell-type-specific maps of gene regulation, maximizing the utility of scRNA-seq datasets.

Published online 27 March 2020

1. Ramsköld, D., Wang, E. T., Burge, C. B. & Sandberg, R. An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Comput. Biol.* **5**, e1000598 (2009).
2. Potter, S. S. Single-cell RNA sequencing for the study of development, physiology and disease. *Nat. Rev. Nephrol.* **14**, 479–492 (2018).
3. Macosko, E. Z. et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214 (2015).
4. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
5. Rosenberg, A. B. et al. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **12**, eaam8999 (2018).
6. Cao, J. et al. Comprehensive single cell transcriptional profiling of a multicellular organism by combinatorial indexing. *Science* **357**, 661–667 (2017).
7. Lake, B. B. et al. A single-nucleus RNA-sequencing pipeline to decipher the molecular anatomy and pathophysiology of human kidneys. *Nat. Commun.* **10**, 2832 (2019).
8. Combes, A. N., Zappia, L., Er, P. X., Oshlack, A. & Little, M. H. Single-cell analysis reveals congruence between kidney organoids and human fetal kidney. *Genome Med.* **11**, 3 (2019).
9. Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331–338 (2017).
10. Chen, C. et al. Single-cell whole-genome analyses by linear amplification via transposon insertion (LIANTI). *Science* **356**, 189–194 (2017).
11. Smallwood, S. A. et al. Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nat. Methods* **11**, 817–820 (2014).
12. Cusanovich, D. A. et al. Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science* **348**, 910–914 (2015).
13. Buenrostro, J. D. et al. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* **523**, 486–490 (2015).
14. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* **37**, 1452–1457 (2019).
15. Linker, S. M. et al. Combined single-cell profiling of expression and DNA methylation reveals splicing regulation and heterogeneity. *Genome Biol.* **20**, 30 (2019).
16. Gu, C., Liu, S., Wu, Q., Zhang, L. & Guo, F. Integrative single-cell analysis of transcriptome, DNA methylome and chromatin accessibility in mouse oocytes. *Cell Res.* **29**, 110–123 (2019).
17. Amezquita, R. A. et al. Orchestrating single-cell analysis with Bioconductor. *Nat. Methods* **17**, 137–145 (2020).
A useful stepwise practical tutorial on how to perform scRNA-seq analysis in the R programming language using the Bioconductor suite of tools.
18. 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).
19. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
This tutorial discusses scRNA-seq analysis steps using the latest methods developed for each step.
20. Petukhov, V. et al. Accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome Biol.* **19**, 78 (2018).
21. Melsted, P. et al. Modular and efficient pre-processing of single-cell RNA-seq. Preprint at https://doi.org/10.1101/673285 (2019).
22. Islam, S. et al. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat. Methods* **11**, 163–166 (2014).
23. Smith, T. & Sudbery, I. UMI-tools: modelling sequencing errors in unique molecular identifiers to improve quantification accuracy. *Genome Res.* **27**, 491–499 (2017).
24. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34**, 525–527 (2016).
25. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
26. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).
27. van den Brink, S. et al. Single-cell sequencing reveals dissociation-induced gene expression in tissue subpopulations. *Nat. Methods* **14**, 935–936 (2017).
28. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell Syst.* **8**, 329–337.e4 (2019).
29. 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).
30. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
31. 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).
32. Wagner, A., Regev, A. & Yosef, N. Uncovering the vectors of cellular states with single cell genomics. *Nat. Publ. Gr.* **34**, 1–53 (2016).
33. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods* **14**, 565–571 (2017).
34. L. Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).
35. Bacher, R. et al. SCnorm: robust normalization of single-cell RNA-seq data. *Nat. Methods* **14**, 584–586 (2017).
36. Hafemeister, C. & Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol*. **20**, 296 (2019).
37. Fan, J. et al. Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nat. Methods* **13**, 241–244 (2016).
38. Brennecke, P. et al. Accounting for technical noise in single-cell RNA-seq experiments. *Nat. Methods* **10**, 1093–1095 (2013).
39. Barkas, N. et al. pagoda2: a package for analyzing and interactively exploring large single-cell RNA-seq datasets. *GitHub* https://github.com/hms-dbmi/pagoda2 (2018).
40. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (2019).
41. 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).
42. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
43. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. DCA: single cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
44. 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).
A benchmark analysis of methods available for selecting over-dispersed genes.
45. Leek, J. T. et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.* **11**, 733–739 (2010).
46. 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).
47. Barkas, N. et al. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nat. Methods* **16**, 695–698 (2019).
48. Tran, H. T. N. et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol.* **21**, 12 (2020).
A benchmark study of methods available for batch correction during analysis of scRNA-seq data.
49. Leek, J. T. Svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res.* **42**, e161 (2014).
50. Stuart, T. & Satija, R. Integrative single-cell analysis. *Nat. Rev. Genet.* **20**, 257–272 (2019).
51. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* **37**, 685–691 (2019).
52. 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).
53. Saelens, W., Cannoodt, R., Todorov HelenaSaeys, Y., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods: towards more accurate and robust tools. *Nat. Biotechnol.* **37**, 547–554 (2019).
A benchmark analysis of methods for single-cell trajectory inference.
54. Bellman, R. On the theory of dynamic programming. *Proc. Natl Acad. Sci. USA* **38**, 716–719 (1952).
55. Sun, S., Zhu, J., Ma, Y. & Zhou, X. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome Biol.* **20**, 269 (2019).
A benchmark study of methods used for dimensionality reduction of scRNA-seq data.
56. Abdi, H. & Williams, L. J. Principal component analysis. *Chemom. Intell. Lab. Syst.* **2**, 433–459 (2010).
57. Pierson, E. & Yau, C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16**, 241 (2015).
58. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.* **18**, 212 (2017).
59. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788–791 (1999).
60. Lin, X. & Boutros, P. C. Optimization and expansion of non-negative matrix factorization. *BMC Bioinformatics* **21**, 7 (2020).
61. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326 (2000).
62. Angerer, P. et al. destiny: diffusion maps for large-scale single-cell data in R. *Bioinformatics* **32**, 1241–1243 (2015).
63. Welch, J. D., Hartemink, A. J. & Prins, J. F. SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. *Genome Biol.* **17**, 106 (2016).
64. Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* **20**, 389–403 (2019).
65. Deng, Y., Bao, F., Dai, Q., Wu, L. F. & Altschuler, S. J. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat. Methods* **16**, 311–314 (2019).
66. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol.* **38**, 147–150 (2020).
67. Wagner, F., Yan, Y. & Yanai, I. K-nearest neighbor smoothing for single-cell RNA-seq data. Preprint at https://doi.org/10.1101/217737 (2017).
68. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729.e27 (2018).
69. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
70. Lin, P., Troup, M. & Ho, J. W. K. CIDR: ultrafast and accurate clustering through imputation for single cell RNA-seq data. *Genome Biol.* **18**, 59 (2017).
71. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat. Commun.* **9**, 997 (2018).
72. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Res.* **7**, 1740 (2019).
73. Lloyd, S. P. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**, 129–137 (1982).
74. Žurauskiene, J. & Yau, C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics* **17**, 140 (2016).
75. Zeisel, A. et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).
76. Zeisel, A. et al. Molecular architecture of the mouse nervous system. *Cell* **174**, 999–1014.e22 (2018).
77. Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Res.* **7**, 1141 (2018).
A benchmark analysis of methods available for clustering in scRNA-seq data analysis.
78. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).
79. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep*. **9**, 5233 (2019).
80. Kiselev, V. Y. et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods* **14**, 483–486 (2017).

81. Li, H. et al. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nat. Genet.* **49**, 708–718 (2017).
82. Combes, A. N. et al. Single cell analysis of the developing mouse kidney provides deeper insight into marker gene expression and ligand-receptor crosstalk. *Development* **146**, dev178673 (2019).
83. Qiu, X. et al. Single-cell mRNA quantification and differential analysis with Census. *Nat. Methods* **14**, 309–315 (2017).
84. Cao, J. et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496–502 (2019).
85. Wolf, F. A. et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 59 (2019).
86. La Manno, G. et al. RNA velocity of single cells. *Nature* **560**, 494–498 (2018).
87. Wu, Y., Tamayo, P. & Zhang, K. Visualizing and interpreting single-cell gene expression datasets with similarity weighted nonnegative embedding. *Cell Syst.* **7**, 656–666.e4 (2018).
88. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
89. McInnes, L. & Healy, J. UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at https://arxiv.org/abs/1802.03426 (2018).
90. Becht, E. et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **37**, 38–44 (2018).
91. Wattenberg, M., Viegas, F. & Johnson, I. How to use t-SNE effectively. *Distill* https://doi.org/10.23915/distill.00002 (2016).
92. van der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **15**, 3221–3245 (2014).
93. Kobak, D. & Linderman, G. C. UMAP does not preserve global structure any better than t-SNE when using the same initialization. Preprint at https://doi.org/10.1101/2019.12.19.877522 (2019).
94. Kobak, D. & Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **10**, 5416 (2019).
95. Moon, K. R. et al. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
96. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.* **9**, 2002 (2018).
97. Abdelaal, T. et al. A comparison of automatic cell identification methods for single-cell RNA-sequencing data. *Genome Biol.* **20**, 194 (2019).
A benchmark study of methods available for automated cell-type classification in scRNA-seq data.
98. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods* **15**, 255–261 (2018).
99. Lun, A. T. L. & Marioni, J. C. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics* **18**, 451–464 (2017).
100. 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).
101. Suykens, J. A. K. & Vandewalle, J. Indefinite kernels in least squares support vector machines and principal component analysis. *Neural Process. Lett.* **43**, 162–172 (2017).
102. Pliner, H. A., Shendure, J. & Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *Nat. Methods* **16**, 983–986 (2019).
103. Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods* **15**, 359–362 (2017).
104. Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. ScPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biol.* **20**, 264 (2019).
105. Regev, A. et al. The human cell atlas. *eLife* **6**, e27041 (2017).
106. Stoeckius, M. et al. Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome Biol.* **19**, 224 (2018).
107. Denisenko, E. et al. Systematic bias assessment in solid tissue 10x scRNA-seq workflows. Preprint at https://doi.org/10.1101/832444 (2019).
108. Lake, B. et al. Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. *Science* **352**, 1586–1590 (2016).
109. Krishnaswami, S. R. et al. Using single nuclei for RNA-seq to capture the transcriptome of postmortem neurons. *Nat. Protoc.* **11**, 499–524 (2016).

Acknowledgements

The authors were supported by NIH grants U01MH098977, R01HL123755, U54HL145608, UH3DK114933 and R01HG009285.

Author contributions

All authors researched data for the article, wrote the manuscript, made substantial contributions to discussions of the content and reviewed or edited the manuscript before submission.

Competing interests

Y.W. declares no competing interests. K.Z. is a co-founder, equity holder, scientific advisory board member and paid consultant of Singlera Genomics, which has no commercial interests related to this article.

Peer review information

Nature Reviews Nephrology thanks B. J. Aronow and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Publisher's note

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

RELATED LINKS

Broad Institute online single-cell data browser: https://portals.broadinstitute.org/single_cell

EMBL-EBI online single-cell data browser: https://www.ebi.ac.uk/gxa/sc/home

UCSC online single-cell data browser: https://cells.ucsc.edu/

© Springer Nature Limited 2020