nature protocols

REVIEW ARTICLE

https://doi.org/10.1038/s41596-020-00409-w

Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data

Tallulah S. Andrews[1], Vladimir Yu Kiselev[1], Davis McCarthy[2,3] and Martin Hemberg[1]✉

Single-cell RNA sequencing (scRNA-seq) is a popular and powerful technology that allows you to profile the whole transcriptome of a large number of individual cells. However, the analysis of the large volumes of data generated from these experiments requires specialized statistical and computational methods. Here we present an overview of the computational workflow involved in processing scRNA-seq data. We discuss some of the most common tasks and the tools available for addressing central biological questions. In this article and our companion website (https://scrnaseq-course.cog.sanger.ac.uk/website/index.html), we provide guidelines regarding best practices for performing computational analyses. This tutorial provides a hands-on guide for experimentalists interested in analyzing their data as well as an overview for bioinformaticians seeking to develop new computational methods.

scRNA-seq has emerged as a transformative technology to characterize complex tissues and answer questions that could not be addressed using bulk RNA sequencing. Since the first scRNA-seq protocol was published in 2009[1], many protocols and commercial platforms have been released[2,3]. Today, there are two main paradigms for scRNA-seq experiments. The most common approach is to use microscopic droplets or wells to isolate a large number of cells and then sequence the libraries relatively shallowly[4,5]. To identify which cell a given transcript came from, these methods use cellular barcodes (short nucleotide tags attached to each read that are unique to a droplet or well). This high-throughput, low-depth paradigm is typical for experiments using the popular 10× Chromium platform. An important advantage of this technology is that it supports unique molecular identifiers (UMIs). UMIs are short barcodes that are attached to transcripts before amplification, making it possible to remove polymerase chain reaction duplicates and to obtain more accurate estimates of expression levels. A major shortcoming is that the platforms only allow for the 5′ or 3′ end of each messenger RNA (mRNA) to be sequenced. Many studies take the opposite approach of isolating relatively few cells but sequencing them much more deeply. These low-throughput, high-depth experiments typically isolate cells into individual wells and apply the Smart-seq2 protocol[6]. With the exception of the recently introduced Smart-seq3 protocol[7], these methods do not support UMIs, but they typically show higher sensitivity than droplet-based technologies[2,3], and they also allow for the entire transcript to be profiled. For an in-depth overview of different platforms, see recent reviews and benchmarks[2,3,8,9].

In addition to facilitating the experimental workflow, recent innovations have also provided a substantial reduction in the cost per cell of scRNA-seq. Consequently, there has been an exponential growth in terms of the number of cells profiled[2]. Given the large volumes of data generated, efficient computational and statistical methods are required for single-cell data analysis. As experimental protocols have improved rapidly, computational workflows for processing the data have also been refined. The purpose of this Tutorial is to provide an overview of the most common types of analyses for scRNA-seq data. This article is meant to serve as a companion to the course material (https://scrnaseq-course.cog.sanger.ac.uk/website/index.html) that we have developed for teaching computational analysis of scRNA-seq data. The website was first launched in 2016, and it has been continuously updated to include new methods and provide up-to-date recommendations on best practices.

A central component of scRNA-seq analysis is the expression matrix, which represents the number of transcripts observed for each gene and cell. The workflow can be split into two main sections: 1) generation of the expression matrix and 2) analyses of the expression matrix (Fig. 1 and Table 1). Although our online tutorial covers both aspects, here we focus on the types of analyses that are carried out once the expression matrix has been obtained. Most genes are used only in a subset of cell types, but, owing to the low amounts of starting material and the low sequencing depth commonly used in scRNA-seq experiments, some genes will fail to be detected, even if they are expressed. The result is a large number of zero values in the gene expression matrix, which is problematic because some of the zeros can represent actual low or zero expression in the cell as well as variation from the measurement process[10]. The difficulty in distinguishing and appropriately modeling these sources of observed zeros is one of the

[1]Wellcome Sanger Institute, Hinxton, UK. [2]Bioinformatics and Cellular Genomics, St Vincent's Institute of Medical Research, Fitzroy, Victoria, Australia. [3]Melbourne Integrative Genomics, Faculty of Science, University of Melbourne, Melbourne, Victoria, Australia. ✉e-mail: mh26@sanger.ac.uk

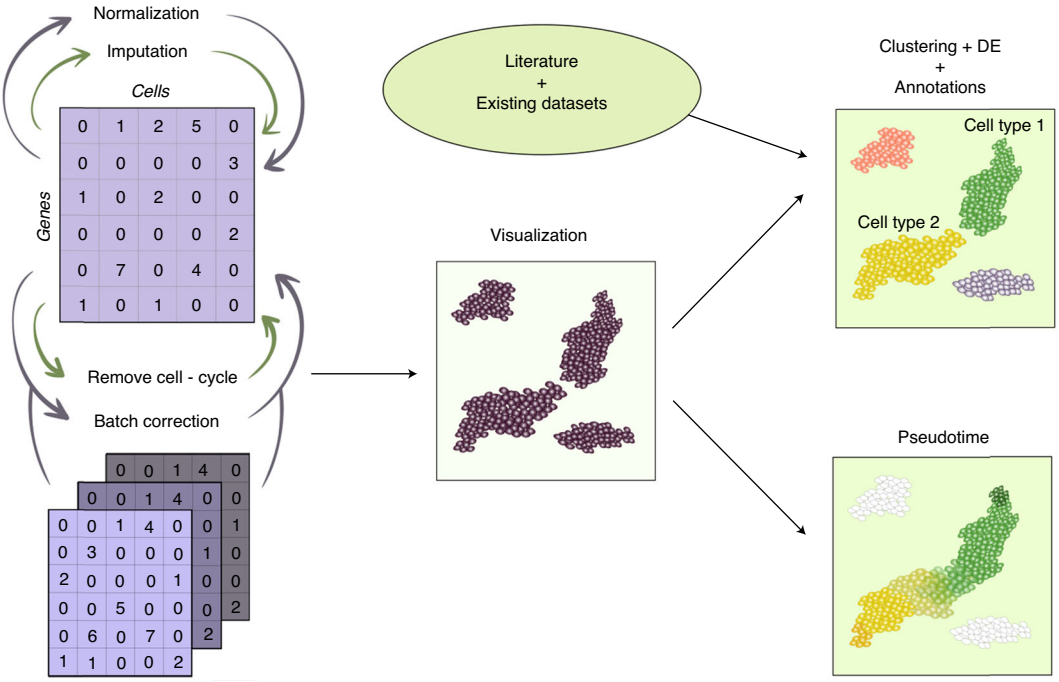

Fig. 1 | Overview of the workflow. In a typical scenario, the researcher must first combine expression matrices from multiple experiments to obtain a combined expression matrix, which is corrected for sequencing depth, cell cycle stage and other confounders. Next, the data are visualized, and biologically meaningful patterns are identified through clustering, pseudotime and differential expression analysis. Finally, the results are compared to the literature and existing datasets.

main challenges for the computational analysis. Even deeply sequenced datasets might have ~50% zeros, whereas shallowly sequenced datasets can have 99% zeros. By contrast, in a typical bulk RNA sequencing dataset, <20% of data entries are zeros.

Quality control

The first step in analyzing scRNA-seq is to exclude cell barcodes that are unlikely to represent intact individual cells. For high-throughput methods, a key step is filtering out cell barcodes that do not represent a cell. The most straightforward approach is to calculate a dataset-specific threshold of the minimum number of UMIs required to consider a barcode as a cell[11]. Alternatively, several recently developed tools, such as EmptyDrops[12], first estimate the background levels of RNA present in empty wells or droplets and then identify cell barcodes that significantly deviate from the background, which indicates the presence of a cell. The advantage of this strategy is that it enables the detection of cell types with low RNA content relative to other cells in the sample.

Unfortunately, neither of these approaches can distinguish intact live cells from damaged or dying cells. A second round of quality control that considers the number of detected genes, the proportion of RNA derived from the mitochondrial genome and the proportion of unmappable or multi-mapped reads per cell must be performed. Cells with high proportions of mitochondrially derived genes, few detected genes or high proportions of unmapped or multi-mapped reads are often damaged or dying cells[13]. The specific thresholds are typically determined from manual inspection of plots of the quality control metrics, as the optimal cutoff depends on the tissue, dissociation protocol and other technical factors. Defining outlier cells (in terms of median absolute deviations) for key metrics allows for straightforward construction of dataset-specific thresholds but should be applied with care, especially for samples comprising highly heterogeneous cell types[14].

In addition to some cell barcodes representing background noise, there is also the possibility that cell barcodes might correspond to more than one cell. Often, ~5% of cell barcodes are tagging multiple cells, known as doublets[15]. In addition, recent results suggest that, up to 20% of the time, multiple cell barcodes might be tagging the same single cell, called barcode multiplets[16]. Tools such as scrublet[15] and DoubletFinder [17] simulate possible doublets from the dataset itself and then calculate the similarity of real droplet barcodes to the simulated doublets and define a threshold to distinguish the inferred doublets from the assumed singlets. Other approaches can also be successfully applied (e.g., scds[18]), but doublet detection is a complicated issue, and no computational doublet detection method can be expected to perform perfectly across all experimental settings.

Normalization

The number of useful reads obtained from a sequencing experiment will vary between cells, and one must correct for this difference. For scRNA-seq data, this effect is pronounced, as the amount of RNA per cell can vary significantly owing to cell cycle stage and other biological factors, even within the same cell type[19]. Technical factors (e.g., differing droplet sizes) might further increase the variability in sequencing depth[20].

Table 1 | For the most common computational tasks, we list some of the most popular methods along with the theoretical framework they rely on

Task	Approach	Tool	Refs	Notes
Cell QC	Filter based on mt-RNA, number of detect genes and number of total UMIs	Seurat, scater	29,84	Thresholds are typically set manually after visual inspection
Doublet identification	Similarity to simulated doublets using a nearest-neighbor graph	DoubletFinder, scrublet	15,17	Can be sped up by pre-clustering data
Normalization	Use pools of cells to calculate size factors instead of doing it for each cell individually	scran	21	Parameters might need to be tuned to avoid negative size factors
	Gene-specific regularized negative binomial regression of library size	sctransform	24	Pearson residuals are used as normalized expression values
	Group genes into ten groups by expression levels and regress out effects due to sequencing depth	SCnorm	23	Computationally expensive to calculate
	Hierarchical empirical Bayes model for expression that applies a binomial model for mRNA capture	bayNorm	25	Provides a posterior normalized and smoothed gene expression matrix
Batch correction/merging datasets	Mutual nearest neighbors and alignment of components identified from CCA	mnnCorrect, Seurat	28,85	Identifies shared biology across datasets without any prior information
	Linear regression to remove batch factors	ComBat	27	Requires balanced experimental design
Imputation	Use external reference datasets as a guide	SAVER-X	36	Currently applicable to high-throughput UMI-tagged datasets only
	A variational autoencoder is used to project to low-dimensional subspace, and missing values are inferred from the inverted projection	scVI	35	A neural network is used to learn the parameters of a hierarchical negative binomial model describing the data-generative process
	Assumes a negative binomial distribution to infer true zeros from missing data	SAVER	32	Uses a computationally efficient Poisson LASSO strategy to estimate parameters
Cell cycle	Relative expression of a gene panel	cyclone	38	Cannot identify non-cycling cells
	Score each cell based on a panel of genes known to be specific to different stages of the cell cycle	Seurat	39	Designed for large UMI-tagged datasets
Visualization	Models the manifold as a fuzzy topological structure and finds an embedding by searching for a low-dimensional projection that is the closest equivalent to the topological structure using a nearest-neighbor network	UMAP	47	Sensitive to choice of hyperparameters
	Stochastic embedding combined with a transformation based on Student's *t*-distribution	t-SNE	48	Large-scale distances are not interpretable. Sensitive to choice of hyperparameters
Unsupervised clustering	Use multiple distance metrics and dimensionality reduction schemes to cluster cells many times and combine results using a consensus approach	SC3	51	Works well for datasets with <5,000 cells
	Build a *k*-nearest-neighbor network and use the Louvain algorithm to identify graph modules	Seurat, scanpy, scran	21,29,53,54	Scalable to millions of cells
Annotation/projection	Find the nearest cell or cell-type centroid by comparing high-dropout genes	scmap	74	Scales well to large references
	Classifier trained on user-defined set of marker genes	Garnett	86	Supports hierarchical cell-type definitions
pseudotime	Minimum spanning tree between clusters and principal curves	slingshot	87	Customizable for any dimensionality reduction or clusters
	Graph of clusters using inter-group version of modularity	PAGA	88	Output is summarized at the level of clusters, not single cells
	Diffusion maps to identify low-dimensional manifold	DPT/destiny	61	Pseudotime is calculated using random walks over cells
	Similar to PAGA, finds connected clusters in the UMAP space	Monocle3	5	Scales to datasets with millions of cells
DE	Hurdle model combining tests for difference in mean and detection rate	MAST	70	Appropriate for read counts or molecule/UMI counts
	Non-parametric, based on ranking expression	Wilcoxon		Robust and very easy to use
	Negative binomial model GLM, parametric	edgeR	89,90	Appropriate for molecule/UMI counts

Differences owing to uneven sequencing depth can be ameliorated by normalization.

For bulk RNA sequencing data, normalization amounts to calculating a quantity related to the sequencing depth of the sample, often called a 'size factor', and dividing the expression of all genes by this value. A similar approach can, in principle, be used for scRNA-seq, but the large number of zeros means that the strategy needs to be modified. The scran package achieves a robust outcome through the use of pools of cells to estimate size factors[21]. Alternatively, spike-in RNAs from the External RNA Control Consortium or housekeeping genes can be used to estimate size factors[22].

As a consequence of the large number of zeros, lowly expressed genes might behave differently from highly expressed genes in response to differing sequencing depths. To compensate for this behavior, one can use a normalization strategy specific to the expression level of each gene. For example, SCnorm can be used for low-throughput, high-depth data[23], and sctransform can be used for high-throughput, low-depth data[24]. In 2019, a novel Bayesian approach for scaling and inference of scRNA-seq counts called bayNorm was developed, which aims to estimate the underlying gene expression matrix after accounting for effects of mRNA capture[25].

Batch effect correction

Similar to differences in sequencing depth, batch effects are technical confounders that must be accounted for in order for the true biological signal to emerge. Batch effects are a common problem in biology, and they arise from differences in non-biological factors such as time of experiment, the person carrying out the experiment or differences in reagents. If not properly accounted for, batch effects can be mistaken for true biological signal, but, through careful experimental design, they can be avoided altogether[26]. To apply batch effect correction to a dataset, the experiment cannot be confounded (i.e., each batch must contain at least two biological conditions). Batch effect correction is most effective when all biological conditions are processed in all batches, known as a 'balanced design'. Unfortunately, it is often impossible to achieve a balanced design when samples cannot be processed simultaneously (e.g., if cells require immediate processing after collection).

Traditional batch correction methods, such as ComBat[27], assume that the biological condition of every cell is known a priori and leverage this information to separate biological effects from batch effects using a linear model. However, this assumption is often inappropriate for scRNA-seq data as the cell-type identity of individual cells might not be known. To address this challenge, mnnCorrect[28] uses mutual nearest neighbors between cells in different batches to identify common biological conditions across batches post hoc. This mutual-nearest-neighbor approach has also been adapted to find 'anchors' for the canonical correlation analysis (CCA) method of Seurat[29]. The main difference between these two tools is that mnnCorrect removes batch effects from the gene expression matrix using PCA, whereas CCA projects cells into a common gene correlation space and performs the correction on that space. However, even these single-cell-specific tools assume shared biological conditions across batches and will incorrectly remove real biological signals if applied to a confounded experiment.

Imputation and smoothing

Many normalization strategies do not change the zeros, so it is tempting to assume that they represent missing values and to fill in an estimate derived mathematically from the detected transcripts. In principle, removing zeros could reduce the noise and make it easier to identify the underlying structure of the data (e.g., gene–gene correlations, cell clusters, marker genes or developmental trajectories).

Several tools have been developed to 'impute' zero values found in scRNA-seq data, including scImpute[30], DrImpute[31] and SAVER[32]. DrImpute and scImpute perform similarly, whereas SAVER tends to have a smaller effect on the data and produces far fewer false signals. These tools all rely on finding a structure within the data that can be used to predict the expression level of missing values. However, these methods make the assumption that all genes in the dataset are determined by the identified structure, frequently resulting in a large number of false-positive signals being introduced[33]. Other tools, such as MAGIC[34], which uses a diffusion model, and scVI[35], which uses an autoencoder, apply smoothing algorithms to reduce the noise. Consequently, the methods take a data-driven approach in assuming that missing values can be inferred from other cells with a similar gene expression profile. Similarly to model-based imputation methods, they can make it easier to detect structures in the downstream analyses. One shortcoming is that the underlying model might distort the true structure (e.g., by amplifying random noise), which can be mistaken for a biological pattern[30,33]. With the increasing number of publicly available single-cell atlases, it is becoming feasible to use an external reference to impute missing values. Examples of such methods are SAVER-X[36] and netNMF-sc[37], which might not have the same drawbacks, as they are able to incorporate relevant information from other sources. Imputation can help improve visualization of scRNA-seq data, but any structure or pattern identified in imputed data (e.g., differential expressed genes or trajectories) must be verified with appropriate statistical tests applied to the pre-imputed data.

Cell cycle assignment

If the sample contains cells that are actively cycling, this can result in a biological confounder that might need to be removed for downstream analyses. Alternatively, the stage of the cell cycle might be of interest to the biological question under investigation. In either case, it is necessary to assign cells to their appropriate cell cycle stage. There are two widely used tools for identifying cell cycle stage: cyclone[38] and Seurat[29]. Cyclone analyzes pairs of genes that are expressed at different levels relative to each other to assign cells to G1, S or G2/M. Although cyclone is highly accurate regardless of normalization, it has difficulties distinguishing non-cycling cells. Seurat implements the method proposed by Tirosh et al.[39] to score cells based on the averaged normalized expression of known markers of G1/S and G2/M.

Once cells have been assigned a cell cycle stage, both tools use a general linear model to regress out differences. In addition, Seurat offers an option to regress out only the difference between the cells in G1/S and G2/M while retaining the difference between cycling and non-cycling cells. This latter case is important if one is interested in characterizing the differences between the cycling and non-cycling subpopulations.

Feature selection

In an scRNA-seq experiment, each gene represents a dimension, so, for a mouse or human dataset, there will be ~20,000 dimensions. However, many genes are not expressed in a given cell or cell type, and the number detected in an experiment depends on the protocol. High-throughput, droplet-based methods can identify up to ~5,000 genes, whereas more sensitive methods can detect twice as many genes[2,3]. However, many studies rely on shallow sequencing and, therefore, detect fewer genes, sometimes <1,000 genes per cell. The large number of genes can make the analysis difficult because distance estimates in high dimensions are unreliable, even when levels of noise are low.

Feature selection identifies genes with the strongest biological signal relative to the technical noise. By restricting downstream analysis to the most informative genes, the effect of dimensionality is diminished, noise is reduced and the analysis is simplified. Tools might identify a fixed number of features to use or try to establish which features contain a significant amount of biological information. There are two complicating factors to feature selection in scRNA-seq data: (i) the technical noise affecting each gene depends on the mean expression of that gene[40], and (ii) estimating variance is difficult for small sample sizes[41]. The most widely used strategy for feature selection is to consider highly variable genes (i.e., genes with a higher-than-expected variance)[42]. For datasets with thousands of cells quantified using UMIs, it has been shown that the noise follows a negative binomial distribution, which can be used to identify significant features[41,43]. Tools such as Seurat[29] use a non-parametric approach to identify highly variable genes by empirically fitting the relationship between variance and mean expression. An alternative feature selection strategy is to, instead, consider genes with a higher-than-expected number of observed zero values[41].

A limitation of many feature selection methods is that they consider the overall variability across the entire dataset. Thus, genes that are differentially expressed in a rare cell type might not be detected, as these cells provide only a small contribution to the total variability. In this case, alternative metrics, such as the Gini Index, which quantifies unequal distribution of transcripts, might be more appropriate, as demonstrated by the GiniClust method, which is designed to identify small clusters[44].

Dimensionality reduction and visualization

Another strategy for reducing the negative effect of the high dimensionality of the expression matrix is to perform dimensionality reduction on the reduced feature space. There are many methods available[45], but the most commonly used strategies involve principal component analysis (PCA), a linear transformation that preserves Euclidean distances between cells in the full PCA space and can be calculated efficiently even for very large datasets. The number of components retained for later analysis will depend on the complexity of the dataset, and various algorithms exist to identify the appropriate number[46]. As these are often computationally expensive to run, the most common approach is to plot the fraction of variance explained by each component and then visually identify the point where the curve makes a sharp bend, often referred to as the 'knee', and keep only those components above the knee.

Most scRNA-seq datasets are complex, and their structure cannot be captured by two or three principal components. Thus, visualization algorithms are used to create a two-dimensional plot summarizing an scRNA-seq dataset from a larger number of significant components. The current best-practice method is Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)[47]. This algorithm approximates the topology of the data using a cell–cell nearest-neighbor network and then estimates a low-dimensional embedding of the data that best preserves the structure. UMAP has largely replaced t-distributed stochastic neighbor embedding (t-SNE)[48] because of its ability to better preserve large-scale structures. A recent study showed that this can be attributed to the initialization strategy used by default in the popular implementations[49]. However, UMAP tends to favor fully connected representations of the data rather than the discrete clusters favored by t-SNE. One shortcoming of both t-SNE and UMAP is that they both require a user-defined hyperparameter, and the result can be sensitive to the value chosen. Moreover, the methods are stochastic, and providing a good initialization can significantly improve the results of both algorithms. It is important to note that neither visualization algorithm preserves cell–cell distances, so the resulting embedding should not be used directly by downstream analysis methods such as clustering or pseudotime inference.

Unsupervised clustering

Unsupervised clustering of scRNA-seq data is central for most analyses, as it identifies groups of cells with similar expression profiles. Some of these groups can represent distinct cell types, and others can be considered as intermediate cell states (e.g., cell cycle phases), depending on the biological system at hand. Various clustering methods were developed long before the advent of scRNA-seq, and existing tools are applications of classical methods. One example is the widely used *k*-means algorithm[50], which forms the basis of the single-cell consensus clustering (SC3) algorithm[51]. In addition to the basic *k*-means algorithm, SC3 uses a consensus approach to average multiple clustering outcomes. Another example is the Louvain algorithm[52] for network clustering, which was successfully adapted for single-cell datasets in Phenograph[53] and subsequently adopted by Seurat[29] and scanpy[54]. Methods based on the Louvain algorithm construct a nearest-neighbor network for the cells and then identify distinct communities in the network. The strength of these methods is their speed, even for very large datasets. Independent comparisons show that SC3 and Seurat perform similarly to each other overall, although, for

individual datasets, one or the other might perform better and outperform all other currently available methods[55,56]. These benchmarks are based on datasets where the cell type identity can be established through other means than transcriptome analysis (e.g., fluorescence-activated cell sorting analysis of known surface markers).

Every clustering algorithm has its own set of parameters that can significantly affect the results and the biological interpretation. For example, the Louvain algorithm has a resolution parameter that affects the size of the clusters—smaller resolutions recover smaller clusters. Similarly, for *k*-means-based methods, the value of *k* directly determines the number of clusters. Unfortunately, there are no set rules for determining the optimal parameters, and the user must typically make informed decisions that depend on the dataset at hand. It is important to consider both mathematical and biological aspects, as relying solely on one criterion can result in outcomes that either do not provide the best fit for the data or are not sensible given the context[57,58]. For instance, one might calculate the robustness of clusters to the input parameters and examine the resulting clusters for cell types that are already known to exist in the particular tissue.

manual examination.

Pseudotime

Clustering analysis assigns each cell to a group, which is not appropriate in some situations. For example, if the dataset represents a developmental process or is derived from a time-course experiment, then it is more appropriate to view the cells as drawn from a continuum. Such a continuous trajectory, which could represent spatial positions, chemical concentrations or time courses, is often referred to as 'pseudotime', and each cell can be assigned a specific position. Most tools cannot determine the direction or speed that the cells are moving along the trajectory. Instead, external information, such as sampling time for time-course experiments or marker genes for developmental trajectories, must be used to infer these quantities. A large number of pseudotime inference methods have been published, and they have been benchmarked recently[59,60]. The authors highlight that the methods are complementary, and they provide guidelines for which method to choose for different types of data.

Most tools take one of two approaches. The first approach is to use dimensionality reduction techniques to identify a low-dimensional 'manifold' that the cells lie upon and use a cell–cell graph to describe the topology of the manifold. Popular methods using this strategy include Monocle[5] and DPT[61]. The second approach is to use unsupervised clustering to group cells before linking the clusters and projecting individual cells onto the branches. Examples of such methods include TSCAN[62] and Mpath[63]. Cluster-based pseudotime methods tend to be more accurate when there is an unequal density of cells through the trajectory—for instance, cells from one state might be more frequent or more reliably captured than cells from other states and large-scale developmental hierarchies. On the other hand, manifold approaches perform best when there is an even sampling of cells across the transition and when examining details of singular transitions.

Relative abundances of exonic and intronic reads, representing spliced and unspliced transcripts, can be used to infer time dynamics in scRNA-seq experiments[64]. Tools such as RNAvelocity[64] and scVelo[65] infer whether each gene is increasing or decreasing in expression at the time when the cell was sampled. Although RNAvelocity uses a simply dynamical model, scVelo takes a probabilistic approach to account for the uncertainty in single-cell data. Although this approach is limited by sequencing depth and the number of reads mapped to introns, it enables the inference of the direction in which each cell is moving in expression space along with an estimate of the rate of change. The result can be visualized in a low-dimensional projection as an arrow indicating how each cell is moving, akin to a phase plane.

Differential expression

Differential expression (DE) has been one of the most important applications in bulk RNA sequencing as it provides a list of genes that are perturbed between two or more biological conditions. DE for scRNA-seq is more challenging, as we are not just comparing a single value for each gene, but, instead, we can compare distributions of expression levels. Another challenge unique to single-cell data is the fact that the groups of cells that we want to compare are not defined a priori. Instead, the groups are typically defined based on the expression levels that we want to compare, and this violates a central assumption in standard testing procedures. Indeed, it has been demonstrated that unsupervised clustering followed by differential expression analysis can result in artificially low *P*-values[66]. Because the expression values are used when defining groups of cells, this can introduce a bias, as the clustering and the DE analysis are no longer independent.

A recent comparison[67] concluded that the non-parametric Wilcoxon test performs remarkably well compared to purpose-built methods. The authors also concluded that the methods developed for bulk RNA sequencing performed well, in particular when combined with strategies to assign weights to each element of the expression matrix[68]. Another benchmarking study reached similar conclusions, adding that normalization can have an important effect on the outcome[69]. Of the methods specifically tailored for scRNA-seq, MAST[70], which uses a Gaussian hurdle model to combine both differences in detection rate and differences in mean expression into a single test, has been reported to have the best performance[67,71].

An interesting situation arises when a single-cell experiment contains multiple biological replicates (e.g., a comparison of cells from three healthy individuals versus those of three individuals with diabetes). Current single-cell differential expression tests treat each individual cell as a biological replicate and cannot account for shared genetic backgrounds or disease state. For such comparisons, the current options are (i) to calculate average expression among cells for each individual for each cell type and treat the result as bulk RNA sequencing samples or (ii) to perform all individual × individual comparisons and filter out results that are unique to a single individual[72]. The former method is similar to the recently proposed idea of a MetaCell[73]. The idea behind

MetaCell is to use a bootstrapping approach to identify the most stable and reproducible features of the original dataset. However, as scRNA-seq is applied to larger cohorts and comparison studies, we expect further developments to lead to more accurate statistical models for more complex experimental designs.

Comparing versus combining datasets

As the amount of scRNA-seq data continues to grow, an important challenge is establishing how best to combine datasets. Batch effects are a major challenge when combining experiments from different labs, and, even if they can be overcome, the re-analysis of the merged dataset might require substantial time, effort and storage. An alternative strategy to merging datasets is to, instead, compare them. This strategy is particularly useful when one of the datasets is very large (e.g., a cell atlas). scmap[74] builds a small index when given one or more datasets with known cell types. When given a new query dataset, scmap can quickly identify which cell type in the reference each cell of the new dataset is the closest to based on the transcriptional profile. Furthermore, scmap can predict the nearest cell in the reference, which means that it can be used when cells are assigned a pseudotime value rather than a discrete cluster label. Different methods for mapping cells to a reference have recently been benchmarked[75]. Another method, MetaNeighbor[76], is designed to test whether the cell types are consistent across multiple scRNA-seq datasets. It does so by calculating a cell–cell Spearman correlation across datasets, allowing MetaNeighbor to validate how reproducible the cell labels are across multiple experiments.

Conclusion

Computational scRNA-seq analysis is a rapidly evolving field. To a large extent, this is driven by the development of new platforms and protocols. However, researchers are also coming up with novel approaches for extracting information from the data. It is likely that there will be novel analysis tools over the coming years, further expanding the use of scRNA-seq. In addition, we also expect that there will be improvements to the software tools providing an integrated workflow (e.g., Seurat, scanpy and Bioconductor), making the analyses more accessible for users with limited bioinformatics expertise.

The rapid development of novel single-cell technologies, most notably multi-omics methods that can profile more than one aspect of a cell[77,78] and methods that provide spatial information[79,80], will require novel computational methods to take full advantage of the data. Moreover, the increasing volumes of data produced by various atlas projects[81–83] will require methods that scale more favorably with the number of cells analyzed.

References

1. Tang, F. et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* **6**, 377–382 (2009).
2. Svensson, V. et al. Power analysis of single-cell RNA-sequencing experiments. *Nat. Methods* **14**, 381–387 (2017).
3. Ziegenhain, C. et al. Comparative analysis of single-cell RNA sequencing methods. *Mol. Cell* **65**, 631–643 (2017).
4. Han, X. et al. Mapping the mouse cell atlas by microwell-seq. *Cell* **172**, 1091–1107 (2018).
5. Cao, J. et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496–502 (2019).
6. Picelli, S. et al. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nat. Methods* **10**, 1096–1098 (2013).
7. Hagemann-Jensen, M. et al. Single-cell RNA counting at allele- and isoform-resolution using Smart-seq3. *Nat. Biotechnol.* **38**, 708–714 (2020).
8. Zhang, X. et al. Comparative analysis of droplet-based ultra-high-throughput single-cell RNA-seq systems. *Mol. Cell* **73**, 130–142 (2019).
9. Wu, A. R. et al. Quantitative assessment of single-cell RNA-sequencing methods. *Nat. Methods* **11**, 41–46 (2014).
10. Sarkar, A. K. & Stephens, M. Separating measurement and expression models clarifies confusion in single cell RNA-seq analysis. Preprint at https://www.biorxiv.org/content/10.1101/2020.04.07.030007v1 (2020).
11. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
12. Lun, A. T. L. et al. EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol.* **20**, 63 (2019).
13. Ilicic, T. et al. Classification of low quality cells from single-cell RNA-seq data. *Genome Biol.* **17**, 29 (2016).
14. Amezquita, R. A. et al. Orchestrating single-cell analysis with bioconductor. *Nat. Methods* **17**, 137–145 (2020).
15. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell Syst.* **8**, 281–291 (2019).
16. Lareau, C. A., Ma, S., Duarte, F. M. & Buenrostro, J. D. Inference and effects of barcode multiplets in droplet-based single-cell assays. *Nat. Commun.* **11**, 866 (2020).
17. 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 (2019).
18. Bais, A. S. & Kostka, D. scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* **36**, 1150–1158 (2020).
19. Marinov, G. K. et al. From single-cell to cell-pool transcriptomes: stochasticity in gene expression and RNA splicing. *Genome Res.* **24**, 496–510 (2014).
20. Macosko, E. Z. et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214 (2015).
21. Lun, A. T. L., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).
22. Lun, A. T. L., Calero-Nieto, F. J., Haim-Vilmovsky, L., Göttgens, B. & Marioni, J. C. Assessing the reliability of spike-in normalization for analyses of single-cell RNA sequencing data. *Genome Res.* **27**, 1795–1806 (2017).
23. Bacher, R. et al. SCnorm: robust normalization of single-cell RNA-seq data. *Nat. Methods* **14**, 584–586 (2017).
24. Hafemeister, C. & Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol.* **20**, 296 (2019).
25. Tang, W. et al. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* **36**, 1174–1181 (2020).
26. Baran-Gale, J., Chandra, T. & Kirschner, K. Experimental design for single-cell RNA sequencing. *Brief Funct. Genomics* **17**, 233–239 (2018).
27. Stein, C. K. et al. Removing batch effects from purified plasma cell gene expression microarrays with modified ComBat. *BMC Bioinformatics* **16**, 63 (2015).

28. 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).
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. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat. Commun.* **9**, 997 (2018).
31. Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N. & Garry, D. J. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC Bioinformatics* **19**, 220 (2018).
32. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
33. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Research* **7**, 1740 (2018).
34. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729 (2018).
35. 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).
36. Wang, J. et al. Data denoising with transfer learning in single-cell transcriptomics. *Nat. Methods* **16**, 875–878 (2019).
37. Elyanow, R., Dumitrascu, B., Engelhardt, B. E. & Raphael, B. J. netNMF-sc: leveraging gene–gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res.* **30**, 195–204 (2020).
38. Scialdone, A. et al. Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods* **85**, 54–61 (2015).
39. Tirosh, I. et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
40. Grün, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat. Methods* **11**, 637–640 (2014).
41. Andrews, T. S. & Hemberg, M. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics* **35**, 2865–2867 (2019).
42. Yip, S. H., Sham, P. C. & Wang, J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief Bioinformatics* **20**, 1583–1589 (2019).
43. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.* **38**, 147–150 (2020).
44. Jiang, L., Chen, H., Pinello, L. & Yuan, G.-C. GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biol.* **17**, 144 (2016).
45. 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).
46. Peres-Neto, P. R., Jackson, D. A. & Somers, K. M. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Comput. Stat. Data Anal.* **49**, 974–997 (2005).
47. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: uniform manifold approximation and projection. *J. Open Source Software* **3**, 861 (2018).
48. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
49. Kobak, D. & Linderman, G. C. UMAP does not preserve global structure any better than t-SNE when using the same initialization. Preprint at https://www.biorxiv.org/content/10.1101/2019.12.19.877522v1 (2019).
50. Hartigan, J. A. & Wong, M. A. Algorithm AS 136: a *k*-means clustering algorithm. *Appl. Stat.* **28**, 100–108 (1979).
51. Kiselev, V. Y. et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods* **14**, 483–486 (2017).
52. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).
53. Levine, J. H. et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* **162**, 184–197 (2015).
54. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
55. Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research* **7**, 1141 (2018).
56. Freytag, S., Tian, L., Lönnstedt, I., Ng, M. & Bahlo, M. Comparison of clustering tools in R for medium-sized 10× Genomics single-cell RNA-sequencing data. *F1000Research* **7**, 1297 (2018).
57. 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).
58. Zappia, L. & Oshlack, A. Clustering trees: a visualization for evaluating clusterings at multiple resolutions. *Gigascience* **7**, giy083 (2018).
59. Cannoodt, R., Saelens, W. & Saeys, Y. Computational methods for trajectory inference from single-cell transcriptomics. *Eur. J. Immunol.* **46**, 2496–2506 (2016).
60. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
61. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).
62. Ji, Z. & Ji, H. TSCAN: pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res.* **44**, e117 (2016).
63. Chen, J., Schlitzer, A., Chakarov, S., Ginhoux, F. & Poidinger, M. Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development. *Nat. Commun.* **7**, 11988 (2016).
64. La Manno, G. et al. RNA velocity of single cells. *Nature* **560**, 494–498 (2018).
65. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-020-0591-3 (2020).
66. Zhang, J. M., Kamath, G. M. & Tse, D. N. Valid post-clustering differential analysis for single-cell RNA-seq. *Cell Syst.* **9**, 383–392 (2019).
67. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods* **15**, 255–261 (2018).
68. Van den Berge, K. et al. Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol.* **19**, 24 (2018).
69. Vieth, B., Parekh, S., Ziegenhain, C., Enard, W. & Hellmann, I. A systematic evaluation of single cell RNA-seq analysis pipelines. *Nat. Commun.* **10**, 4667 (2019).
70. Finak, G. et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16**, 278 (2015).
71. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
72. Crowell, H. L. et al. On the discovery of population-specific state transitions from multi-sample multi-condition single-cell RNA sequencing data. Preprint at https://www.biorxiv.org/content/10.1101/713412v1 (2019).
73. Baran, Y. et al. MetaCell: analysis of single-cell RNA-seq data using K-nn graph partitions. *Genome Biol.* **20**, 206 (2019).
74. Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods* **15**, 359–362 (2018).

75. Abdelaal, T. et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol.* **20**, 194 (2019).
76. Crow, M., Paul, A., Ballouz, S., Huang, Z. J. & Gillis, J. Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor. *Nat. Commun.* **9**, 884 (2018).
77. Macaulay, I. C., Ponting, C. P. & Voet, T. Single-cell multiomics: multiple measurements from single cells. *Trends Genet.* **33**, 155–168 (2017).
78. Stuart, T. & Satija, R. Integrative single-cell analysis. *Nat. Rev. Genet.* **20**, 257–272 (2019).
79. Rodriques, S. G. et al. Slide-seq: a scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).
80. Ståhl, P. L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
81. Regev, A. et al. The human cell atlas. *eLife* **6**, e27041 (2017).
82. Brunet Avalos, C., Maier, G. L., Bruggmann, R. & Sprecher, S. G. Single cell transcriptome atlas of the *Drosophila* larval brain. *eLife* **8**, e50354 (2019).
83. Tabula Muris Consortium. et al. Single-cell transcriptomics of 20 mouse organs creates a *Tabula Muris*. *Nature* **562**, 367–372 (2018).
84. 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).
85. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019).
86. Pliner, H. A., Shendure, J. & Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *Nat. Methods* **16**, 983–986 (2019).
87. Street, K. et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19**, 477 (2018).
88. 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).
89. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* **11**, R25 (2010).
90. McCarthy, D. J., Chen, Y. & Smyth, G. K. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res.* **40**, 4288–4297 (2012).

Acknowledgements

We thank J. Eliasova for help with Fig. 1. We also thank S. Ballerau, M. Büttner, M. Do Nascimentoo Lopes Primo, J. Lee, R. Lyu, E. Madissoon, R. Martinez Nunez, S. Y. Müller, K. Polanski, P. Qiao and J. Westoby for their contributions to teaching the course and developing the material.

Author contributions

T.S.A., V.Y.K., D.M. and M.H. planned the tutorial and wrote the text together.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.H.

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

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

Received: 30 January 2020; Accepted: 8 September 2020;
Published online: 7 December 2020