Single-Cell RNA-seq: Introduction to Bioinformatics Analysis

Fei Ji[1,2,4] and Ruslan I. Sadreyev[1,3]

[1]Department of Molecular Biology, Massachusetts General Hospital, Boston, Massachusetts
[2]Department of Genetics, Harvard Medical School, Boston, Massachusetts
[3]Department of Pathology, Massachusetts General Hospital and Harvard Medical School, Boston, Massachusetts
[4]Corresponding author: *ji@molbio.mgh.harvard.edu*

Quantitative analysis of single-cell RNA sequencing (RNA-seq) is crucial for discovering the heterogeneity of cell populations and understanding the molecular mechanisms in different cells. In this unit we present a bioinformatics workflow for analyzing single-cell RNA-seq data with a few current publicly available computational tools. This workflow is focused on the interpretation of the heterogeneity from single-cell transcriptomes as well as the identification of cell clusters and genes that are differentially expressed between clusters. © 2019 by John Wiley & Sons, Inc.

Keywords: bioinformatics • single-cell RNA-seq

How to cite this article:
Ji, F., & Sadreyev, R. I. (2019). Single-cell RNA-seq: Introduction to bioinformatics analysis. *Current Protocols in Molecular Biology*, *127,* e92. doi: 10.1002/cpmb.92

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) is rapidly becoming a mainstream experimental platform for the precise dissection of heterogeneous patterns of gene expression in cell populations and tissues, uncovering novel insights into cellular and molecular mechanisms that were impossible to gain from the bulk analysis of these cell populations and tissues. scRNA-seq has been instrumental in recent advances in the biology of cancer, cell differentiation and development, neuroscience, hematopoiesis, and other fields. The experimental design, technical protocols, and resulting next-generation sequencing data from scRNA-seq experiments are substantially different from those of the more basic RNA-seq analyses performed at the level of large cell populations, which presents specific challenges both for general conceptual approaches and for computational implementations of bioinformatics analyses of scRNA-seq data. In this article, we will discuss these approaches and implementations with respect to several bioinformatics methods in common use when this article was written in early 2019. We also guide the user through the corresponding step-by-step analysis protocols.

Typical bioinformatics workflows for scRNA-seq analysis include the following major steps: (1) performing quality control (QC) and filtering of unwanted genes and cells based on various quality metrics; (2) normalizing gene expression and estimating dispersion of expression values for all genes; and (3) identifying cells with similar gene expression patterns for clustering into distinct cell subtypes or for inferring pseudo-temporal trajectories that connect cells at different stages of a biological process: e.g., differentiation.

CURRENT PROTOCOLS
A Wiley Brand
Current Protocols in Molecular Biology e92, Volume 127
Published in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpmb.92

The endpoint of the workflow is the identification of gene expression signatures specific for each cell subtype.

In this article we focus on two widely used methods based on two software packages designed to run in the R environment. SEURAT (Butler, Hoffman, Smibert, Papalexi, & Satija, 2018) is designed for QC, analysis, and identification of distinct subpopulations of cells within scRNA-seq datasets. The Monocle (Qiu et al., 2017) toolkit provides a method for the analysis of transcriptional dynamics during a temporal process, such as cell differentiation or development. Monocle infers trajectories of progressing gene expression patterns by placing individual cells within a minimum-spanning tree (MST), so that cells with similar expression patterns represent immediately adjacent stages along a temporal or pseudotemporal trajectory of a process observed in the cell population. Monocle infers multiple different trajectories by first finding the longest trajectory of similar cells in the dataset, and then, after excluding the cells from this first trajectory, analyzing the rest of the cells to find other, shorter sequential paths, with each path being a part of the same tree and representing a potential alternative cell fate during differentiation. Monocle can also be applied to nontemporal single-cell studies, in which case trajectories are replaced by cell clusters that can be interpreted with no assumption of underlying temporal progression. However, building trees of cell trajectories that are changing is Monocle's strongest specialty, and it is best suited to the analyses of a cell population as a snapshot of differentiation, development, or other dynamic process that unfold over time.

BASIC PROTOCOL 1 ANALYSIS, VISUALIZATION, AND CLUSTERING OF SINGLE-CELL EXPRESSION PATTERNS USING SEURAT

The section describes the SEURAT workflow recommended by SEURAT's developers as applied to a specific scRNA-seq dataset as an example. A detailed SEURAT tutorial can be found at *https://satijalab.org/seurat/get_started.html*.

Necessary Resources

Software

R: *https://www.r-project.org*

Hardware

Computer with Unix, Linux, or Mac OS X operating systems with ≥1 Gb RAM

Processing sequencing data into read counts

1. Process the raw sequencing data into read counts per transcript in individual cells.

   Raw scRNA-seq data consist of files in the FASTQ format that contain millions of next-generation sequencing reads generated by a sequencing instrument, typically an instrument manufactured by Illumina. Each read includes a sequence of a short RNA fragment plus an add-on barcode tag for each specific cell sequenced. In addition, some scRNA-seq platforms introduce unique molecular identifiers (UMIs) to barcode individual RNA molecules, which helps account for PCR duplicates generated during library preparation. The total number of distinct UMIs observed in an individual cell can serve as an estimate of the number of distinct transcript copies that is independent of amplification biases (Islam et al., 2014; Stegle, Teichmann, & Marioni, 2015).

   There are multiple single-cell RNA-seq platforms that vary in their specific ways of barcoding cells and transcripts, so it would be untenable to list all platform-specific protocols for processing the raw sequencing data into read counts per transcript in individual cells. Instead, we give a general overview of this process as the first step of the protocol. The next, more specific steps of the downstream workflow assume that the

read counts are produced according to the specifications of the particular scRNA-seq platform that was employed.

In brief, FASTQ files with sequencing reads are first demultiplexed into smaller read sets for individual cells based on the cell-specific barcodes, redundant UMIs are filtered out, and the resulting reads are mapped to the reference genome taking alternative splicing into account; this is followed by counting of reads mapped to each annotated reference gene or transcript in order to quantitate gene expression. The computational tools used for scRNA-seq read mapping and counting are typically the same as the tools used for standard RNA-seq analyses of larger biological samples. For example, STAR (Dobin et al., 2013) or TopHat (Trapnell, Pachter, & Salzberg, 2009) can be used for mapping, and HTSeq (Anders, Pyl, & Huber, 2015), Cufflinks (Trapnell et al., 2012), or other methods (Li and Dewey, 2011; Patro, Duggal, Love, Irizarry, & Kingsford, 2017) can be used for the quantitation of gene expression. For more details, see published protocols of RNA-seq read mapping and quantitation (Dobin & Gingeras, 2015; Ji & Sadreyev, 2018).

Processing the scRNA-seq data should generate a single-cell gene expression table, with rows corresponding to genes and columns corresponding to individual single cells. Use this table as input for the downstream analysis workflows described in this protocol.

For the purpose of learning how to use this protocol, the user can download an example of a gene expression table generated from 2700 peripheral blood mononuclear cells (PBMC) using a standard 10X Genomics workflow: sample "3k PBMCs from a Healthy Donor" at https://support.10xgenomics.com/single-cell-gene-expression/datasets. The table can also be downloaded from https://github.com/MolBioBioinformatics/scRNA. Each step of the protocol can then be carried out using the data in this table so that users can verify that they are carrying out the protocol correctly by determining whether the results they obtain match the outputs shown in each of the figures.

Quality control and normalization

2. To start using SEURAT in R, install and open R, and within the R environment, download, install, and load the SEURAT package using the following commands:

```
source("https://bioconductor.org/biocLite.R")
biocLite("Seurat")
library(Seurat)
```

3. Load the preprocessed count table (step 1) from the input file in tab-delimited text format:

```
scRNA.data <- as.matrix(read.table("count.txt",
  sep="\t",header = T,row.names = 1))
```

4. Create a single-cell data object ("scRNA") that includes expression values only for genes that are expressed in at least 3 cells and only for cells with at least 200 expressed genes.

```
scRNA <- CreateSeuratObject(raw.data = scRNA.data,
  min.cells = 3, min.genes = 200, project =
  "scRNA")
```

An important challenge of scRNA-seq data is the high level of variance and noise among individual cells, which, in addition to the biological cell-to-cell variation, is caused by the experimental challenges of extracting fully intact single cells from a tissue sample and retrieving a representative set of transcripts from ultra-low amounts of RNA per cell. To address these challenges, various metrics have been introduced to remove "low-quality" cells and genes. Genes expressed in too few cells and cells with too few expressed genes are usually filtered out (Brennecke et al., 2013; Petropoulos et al., 2016).

For this purpose, the SEURAT workflow includes the step of reading the initial unfiltered count data, eliminating any gene that is expressed in only a few cells and any cell that has too few expressed genes. The resulting filtered table is used to create a specialized Seurat data object: i.e., the expression data is organized into a structure that is easily accessible for manipulation by specialized Seurat functions written as simple commands as opposed to extended pieces of code, for example in steps 5 and below.

5. Calculate the fraction of reads mapped to mitochondrial chromosome in each cell and include these fractions in the scRNA object created at step 4.

```
mito.genes <- grep(pattern = "^MT-", x = rownames
  (x = scRNA@data), value = TRUE)
percent.mito <- Matrix::colSums(scRNA@raw.data[mito.
  genes,])/Matrix::colSums(scRNA@raw.data)
scRNA <- AddMetaData(object = scRNA, metadata =
  percent.mito, col.name = "percent.mito")
```

The fraction of reads from the mitochondrial genome is often used as an additional metric of cell quality. Cells with an abnormally high percentage of mitochondrial gene reads, indicating increased cell death and mitochondrial leakage, are removed from further analysis.

6. Construct a violin plot illustrating the number of expressed genes and fraction of mitochondrial reads in all cells (Fig. 1).

```
VlnPlot(object = scRNA, features.plot = c("nGene",
  "percent.mito"), nCol = 2)
```

7. Based on the observed distributions of the number of detected expressed genes and the fraction of mitochondrial reads in a cell, apply additional filtering to remove all cells that have >2,500 or <200 expressed genes, and mitochondrial reads >0.05. Note that in cases when no lower or upper cutoff is applied, `-Inf` or `Inf` (infinity) is used as the nominal cutoff value:

```
scRNA <- FilterCells(object = scRNA, subset.names =
  c("nGene", "percent.mito"), low.thresholds =
  c(200, -Inf), high.thresholds = c(2500, 0.05))
```

8. Using the filtered sets of cells and genes from the previous step, normalize gene expression values using the command `LogNormalize`, which normalizes expression values in each cell by the total expression in the cell (divides by the sum of expression for all genes in the cell), multiplies the resulting fraction by a scaling factor (recommended `scale.factor` = 1e4), and log-transforms the scaled fraction:

```
scRNA <- NormalizeData(object = scRNA, normaliza-
  tion.method = "LogNormalize", scale.factor = 1e4)
```

Cell clustering

A key biological insight that can be obtained in scRNA-seq analysis is the characterization of cell types and cellular states among the cell population, which is unachievable in bulk RNA-seq experiments. A variety of classical unsupervised clustering methods have been used to classify cells into groups by their expression patterns. Basic principal-component analysis (PCA) has been widely applied to the visualization of scRNA-seq results (Fan et al., 2016; Satija, Farrell, Gennert, Schier, & Regev, 2015), in combination with distance-based hierarchical clustering algorithms (Shin et al., 2015; Yan et al., 2013). Over time, however, more sophisticated clustering approaches have been developed to overcome specific challenges presented by scRNA-seq data. The dropout event is a unique type of error occurring in scRNA-seq; the term refers to RNA molecules that are present in a

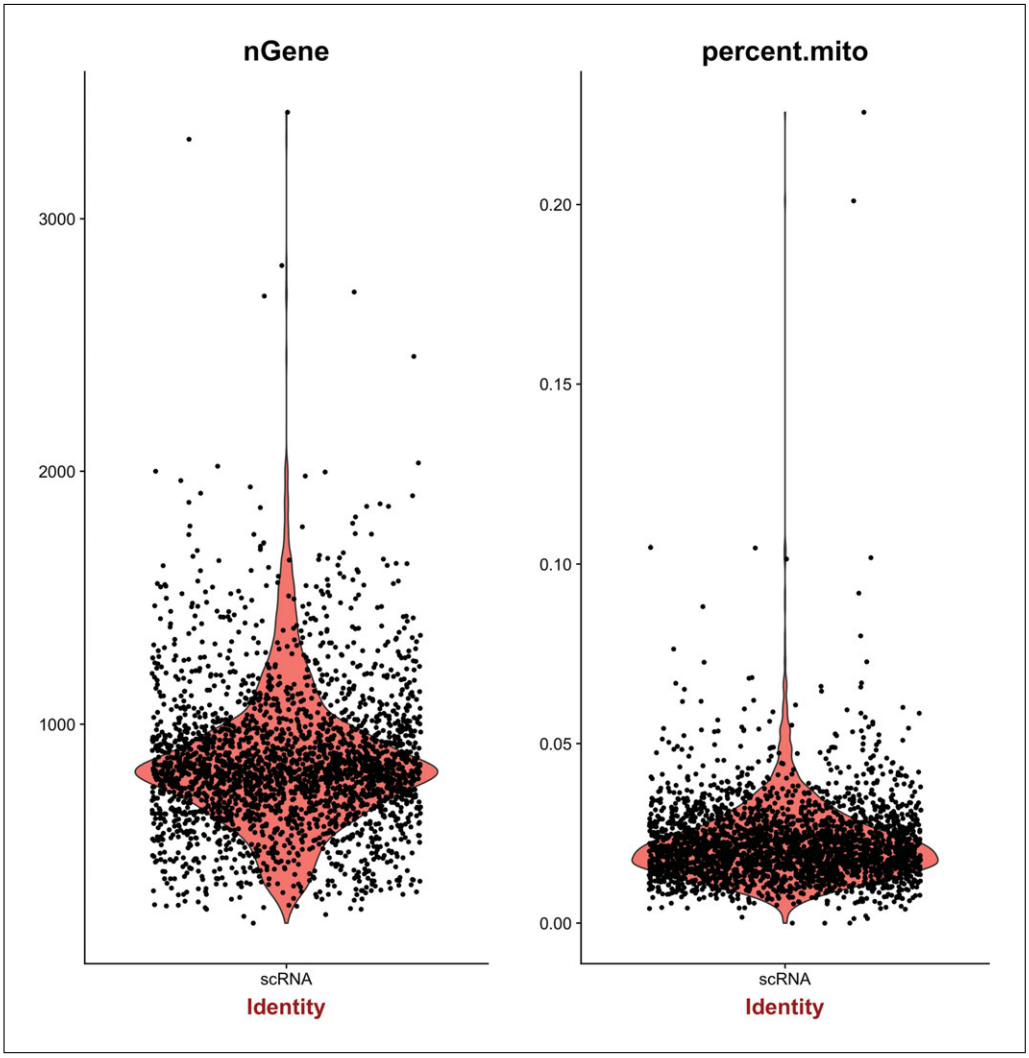

Figure 1 Violin plots showing the distribution of expressed gene numbers (nGene) and mitochondrial read fractions (percent.mito) in all individual cells within the sample. The majority of cells in this sample have <2000 detected expressed genes and <5% mitochondrial reads.

particular cell, but are not detected by scRNA-seq as a result of loss during cDNA library construction, most often at the steps of RNA extraction and reverse transcription. Hence, in an attempt to evaluate whether zero gene expression is due to technical dropout or is a consequence of biological variability, more sophisticated algorithms such as SCDE (Kharchenko, Silberstein, & Scadden, 2014) and PAGODA (Fan et al., 2016) have been developed to estimate the dropout probability using a mean-Poisson distribution model. The zero-inflated factor analysis (ZIFA) method (Pierson & Yau, 2015) models dropout rates based on a double exponential model to enhance the dimensionality reduction and cell clustering. Finally, because two- or three-dimensional visualization of clustering results is a key aspect to understanding the structure of the data, there have been efforts to develop nonlinear dimensionality reduction algorithms tailored to scRNA-seq data. A currently most popular example of a more sophisticated alternative to PCA, t-SNE (Maaten & Hinton, 2008), is a method aiming at visually separating distinct subgroups of cells in a stronger and clearer way.

9. The following command performs PCA, a computational technique for reducing the dimensionality of the dataset. Representing each cell as a point in the multi-dimensional space of expression values for each individual gene, PCA determines principal components as the directions of strongest variation among the population

of cells in this space of many genes (2500-dimensional space in our example). Plotting cells as points in the new space of a few strongest principal components often helps visualize strong patterns of variation among the cells using simple two- or three-dimensional plots. As a direction in the 2500-dimensional space, the orientation of each principal component is defined by contributions from the initial axes of all 2500 genes, with some genes having a stronger contribution to the direction of the principal component.

Execute the following command to perform PCA on the gene expression dataset normalized in the previous step, which outputs the top five genes contributing to each of the top three principal components (PC1 to PC3).

```
scRNA <- RunPCA(object = scRNA, pc.genes = scRNA
  @var.genes, do.print = TRUE, pcs.print = 1:3,
  genes.print = 5)
```

The command produces the following output showing the list of top genes for each principal component:

```
[1] "PC1"
[1] "CST3" "TYROBP" "LST1" "AIF1" "FCN1"
[1] ""
[1] "PTPRCAP" "IL32" "LTB" "CD2" "CTSW"
[1] ""
[1] ""
[1] "PC2"
[1] "NKG7" "GZMB" "PRF1" "CST7" "GZMA"
[1] ""
[1] "CD79A" "MS4A1" "TCL1A" "HLA-DQA1" "HLA-DQB1"
[1] ""
[1] ""
[1] "PC3"
[1] "HLA-DPA1" "HLA-DPB1" "CYBA" "CD37" "HLA-DRB1"
[1] ""
[1] "PPBP" "PF4" "SDPR" "GNG11" "SPARC"
[1] ""
[1] ""
```

This output suggests, for example, that *CST3*, *TYROBP*, and *LST1* are the top genes positively contributing to the first principal component (PC1), and *PTRCAP*, *IL32*, and *LTB* are the top genes contributing negatively to PC1.

At the following stages of the analysis, without running PCA again, one can use the command `PrintPCA` to display the top n genes (in the example below, $n = 5$, so we use `genes.print = 5`) contributing to the top m principal components of interest (PC1, PC2, ..., PCm; in the example below, $m = 3$, so we use `pcs.print = 1:3`).

```
PrintPCA(object = scRNA, pcs.print = 1:3, genes.
  print = 5, use.full = FALSE)
```

10. Plot all cells in a two-dimensional plane of PC1 and PC2 (Fig. 2). Cells can also be visualized in the planes of other principal components by choosing these components as dimensions — i.e., by setting dim.1 and dim.2 to combinations other than 1 and 2.

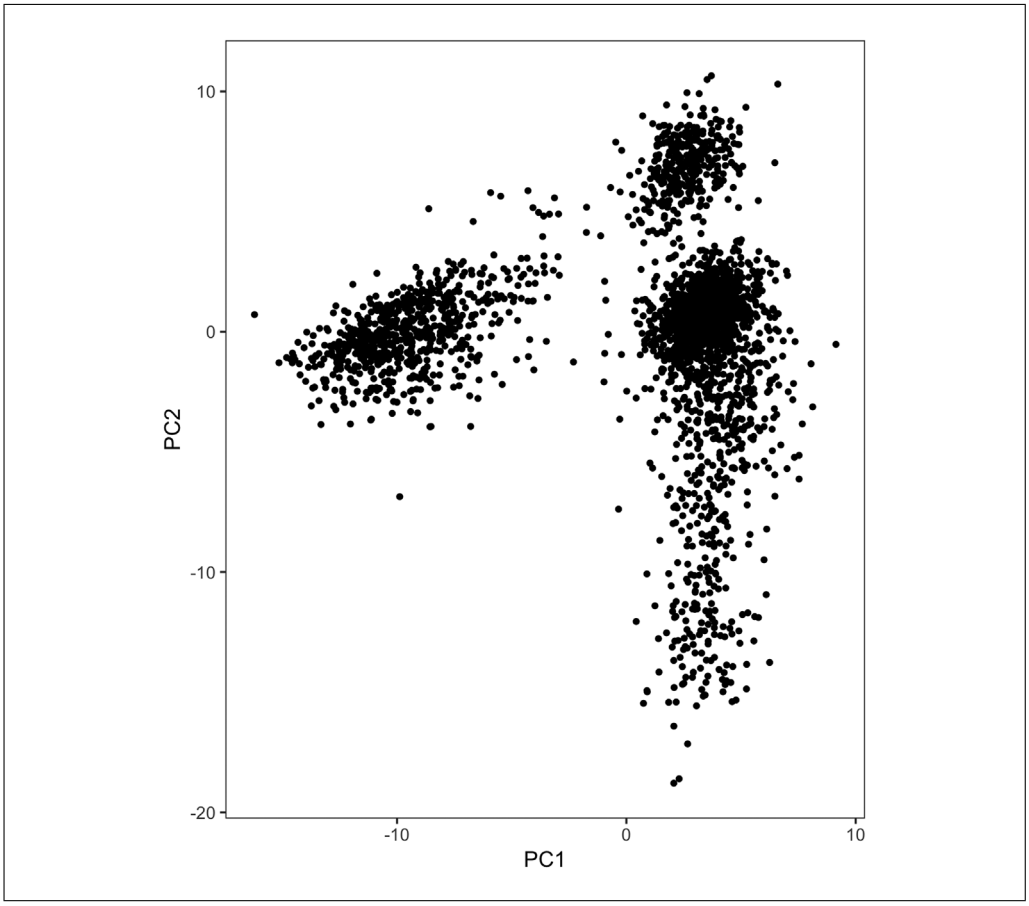

Figure 2 Principal-component analysis (PCA) plot of all single cells over first two principal components (PC1 and PC2).

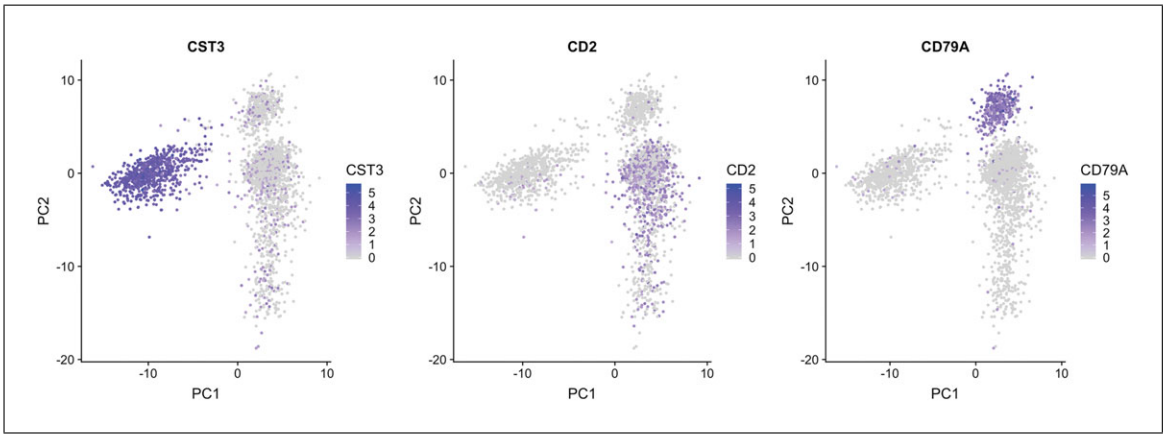

Figure 3 PCA plots. Each cell is color-coded by the expression level of *CST3*, *CD2*, and *CD79A* from low (grey) to high (blue).

```
PCAPlot(object = scRNA, dim.1 = 1, dim.2 = 2)
```

11. From the output message in step 9, *CST3* and *CD2* are the top two PC1 genes of opposite directions, and *CD79A* is one of the top genes contributing to PC2. Use the following command (`FeaturePlot`) to display the level of expression of these genes in each cell as a color shade in the PCA plot (Fig. 3). In this command, the argument `reduction.use` indicates the dimensionality reduction method used

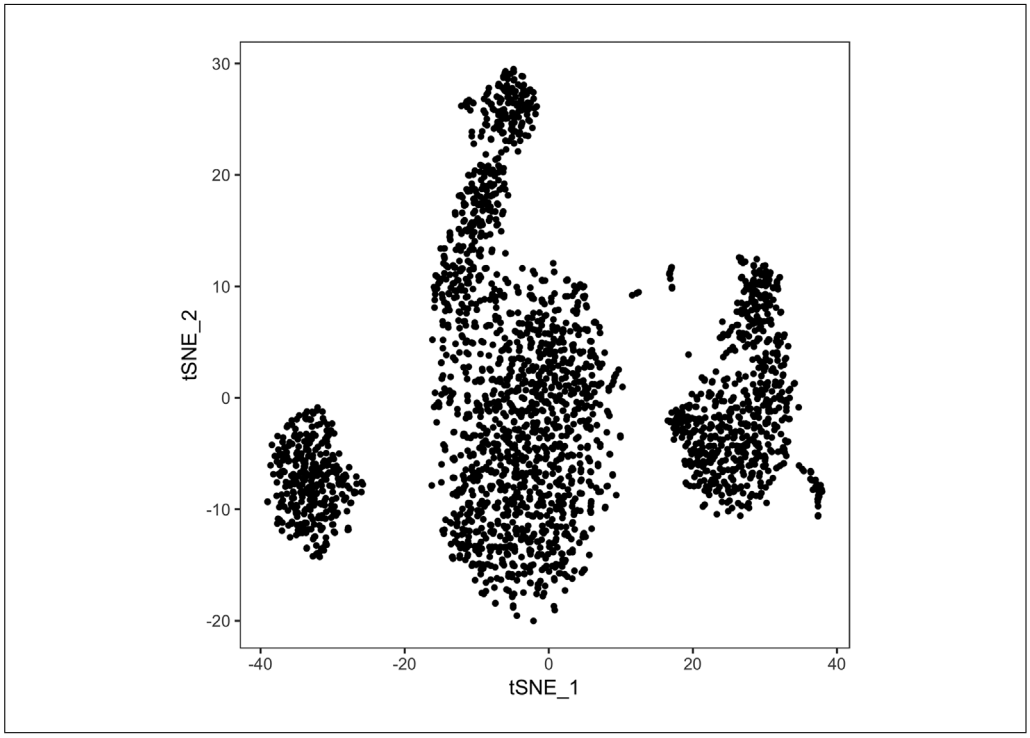

Figure 4 tSNE plot of all single cells.

(PCA in this example) and `features.plot` indicates the list of genes whose expression is to be displayed.

```
FeaturePlot(object = scRNA, features.plot =
  c("CST3","CD2","CD79A"), cols.use = c("lightgrey",
  "blue"),reduction.use="pca")
```

Consistent with their contributions to PC1 and PC2 (step 9), the genes CST3 and CD2 are highly expressed in the majority of cells located in the negative and positive areas along PC1 (x axis), respectively, whereas cells with high expression of CD79A are located in the area of positive PC2 (y axis). PC1 and PC2 correspond to the directions of the strongest variation among the cells, which in this case is largely determined by the separation between clusters of cells with similar expression patterns (Fig. 3). Thus the genes whose expression defines PC1 and PC2 also show a strong contrast in their expression between separate cell clusters: the expression of CBT3 is a marker of the leftmost cluster (Fig. 3, left panel), whereas CD2 (Fig. 3, middle panel) and CD79A (Fig. 3, right panel) are markers of two other clusters.

12. After carrying out the PCA, apply *t*-distributed stochastic neighbor embedding (t-SNE) analysis to the cells as points in the space of the first 20 principal components, PC1 to PC20, for a more sensitive clustering of all cells, and reduce the 20 PCA dimensions to 2 dimensions (tSNE1 and tSNE2).

```
scRNA <- RunTSNE(object = scRNA, dims.use = 1:20)
```

Plot the t-SNE results, with each cell as a point, using `TSNEPlot` (Fig. 4).

t-SNE rearranges the cells into more condensed clusters.

```
TSNEPlot(object = scRNA)
```

13. Overlay the expression levels of the same three genes (*CST3*, *CD2*, and *CD79A*) on the t-SNE plot with the same command as in step 11 by changing the

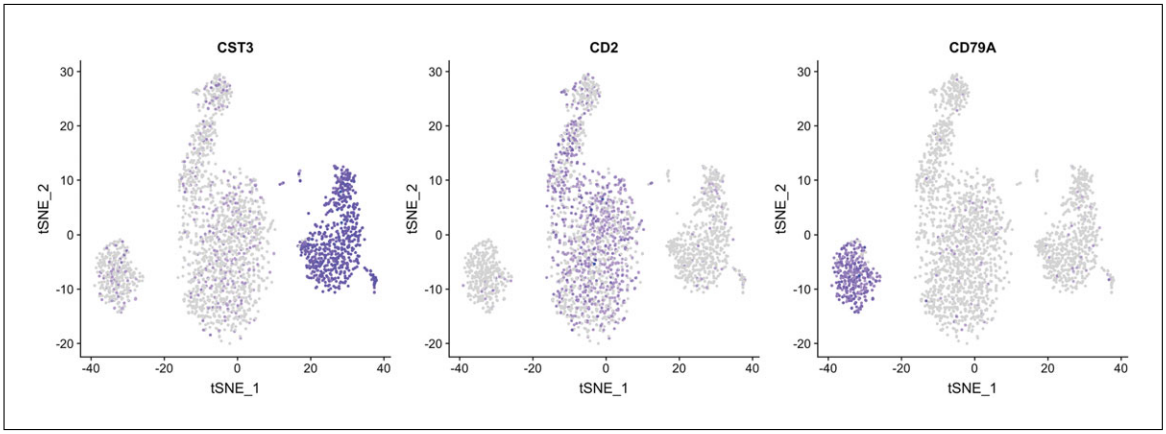

Figure 5 tSNE plots. Each cell is color-coded by the expression level of *CST3*, *CD2*, and *CD79A* from low (grey) to high (blue).

`reduction.use` option to "`tsne`". In this example, all cell clusters produced by the t-SNE analysis (Fig. 5) correspond to the clusters produced by PCA (Fig. 3).

```
FeaturePlot(object = scRNA, features.plot =
  c("CST3","CD2","CD79A"), cols.use = c("lightgrey",
  "blue"),reduction.use="tsne")
```

14. In addition to the general visualization of the cell population using PCA or t-SNE techniques and manual inspection of variation, grouping, and outliers within the cell population, another important part of the workflow is the explicit computational clustering of cells by assigning each cell to one of a few specific clusters. This clustering can be performed using one of various clustering algorithms that analyze the distances between expression patterns of individual cells. The SEURAT command FindClusters applies the shared-nearest-neighbor (SNN) clustering algorithm to a single-cell expression object (scRNA). Run this command to identify clusters of cells as points in the space of the first 20 principal components:

```
scRNA <- FindClusters(object = scRNA, reduction.
  type = "pca",
dims.use = 1:20, resolution = 0.6, print.output =
  0, save.SNN = TRUE)
```

After the clusters are identified, color-code each cluster in the t-SNE plot using the command `TSEPlot` (Fig. 6):

```
TSNEPlot(object = scRNA)
```

Three major clusters of cells are clearly separated by this clustering algorithm, and they are further separated into subclusters. For example, cluster 0 (pink) and 2 (green) in the middle, and cluster 1 (tan) and 4 (blue) on the right side.

15. Identify marker genes in each cluster, as the genes that show significant differential expression between the cells in each cluster compared to all of the remaining cells.

```
scRNA.markers <- FindAllMarkers(object = scRNA,
  only.pos = TRUE, min.pct = 0.25, thresh.use =
  0.25)
```

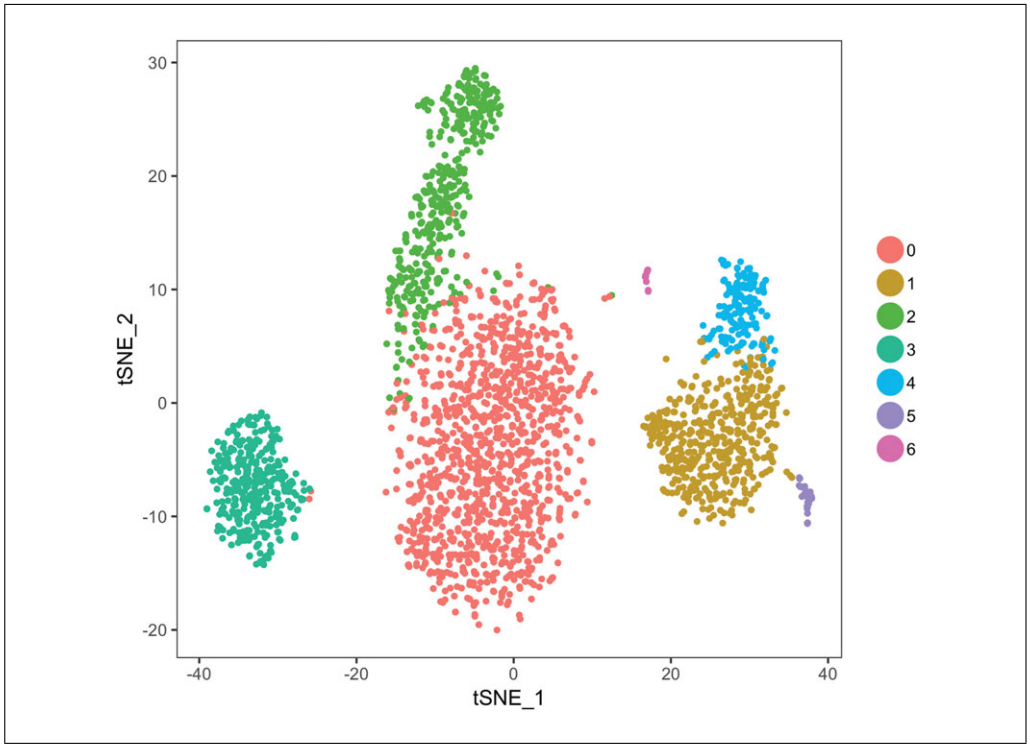

Figure 6 Unsupervised clustering of all single cells in tSNE plot.

16. Identify the top ten marker genes in each cell cluster and plot their expression as a heatmap across all clusters (Fig. 7):

```
top10 <- scRNA.markers %>% group_by(cluster) %>%
  top_n(10, avg_logFC)
DoHeatmap(object = scRNA, genes.use = top10$gene,
  slim.col.label = TRUE, remove.key = TRUE)
```

This heatmap illustrates the expression patterns of the top ten marker genes in each cluster. This heatmap provides more detailed information about cell clusters and their relationships to each other at the level of individual genes. For example, the marker genes of cluster 1 (e.g., S100A9) and cluster 4 (e.g., AIF1) are highly expressed in both clusters, consistent with the close positioning of these two clusters to each other as observed in t-SNE plot (Fig. 6).

17. The command `FeaturePlot` can be an effective way to visualize individual marker genes in different clusters (Fig. 8). From the heatmap, *IL7R*, *CD14*, *CST7*, *MS4A1*, and *AIF1* are the top marker genes for cluster 0, 1, 2, 3, 4.

```
FeaturePlot(object = scRNA, features.plot =
  c("IL7R", "CD14","CST3","MS4A1","AIF1"), cols.use
  = c("grey", "blue"), reduction.use = "tsne",
  no.legend = F,nCol = 3)
```

18. Generate a violin plot as another approach to visualize the distribution of expression levels of individual genes in each cluster (Fig. 9).

```
VlnPlot(object = scRNA, features.plot = c("IL7R",
  "CD14","CST7","MS4A1","AIF1"), x.lab.rot = F)
```

19. In many cases, the cell sample is expected to include cell types that are known to express specific marker genes, based on the literature or on specialized databases classifying cell types by gene expression: for example, $CD8^+$ T cells expressing

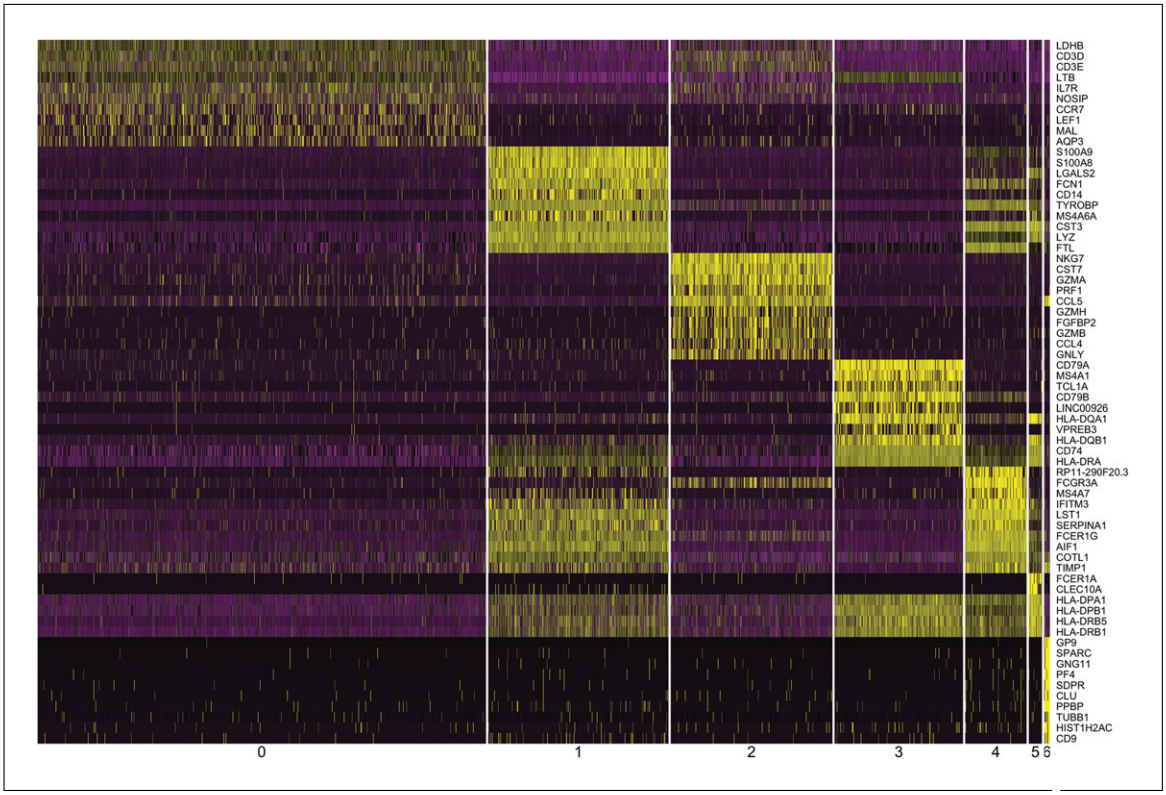

Figure 7 Heatmap of expression levels across all cell clusters of top marker genes in each cluster

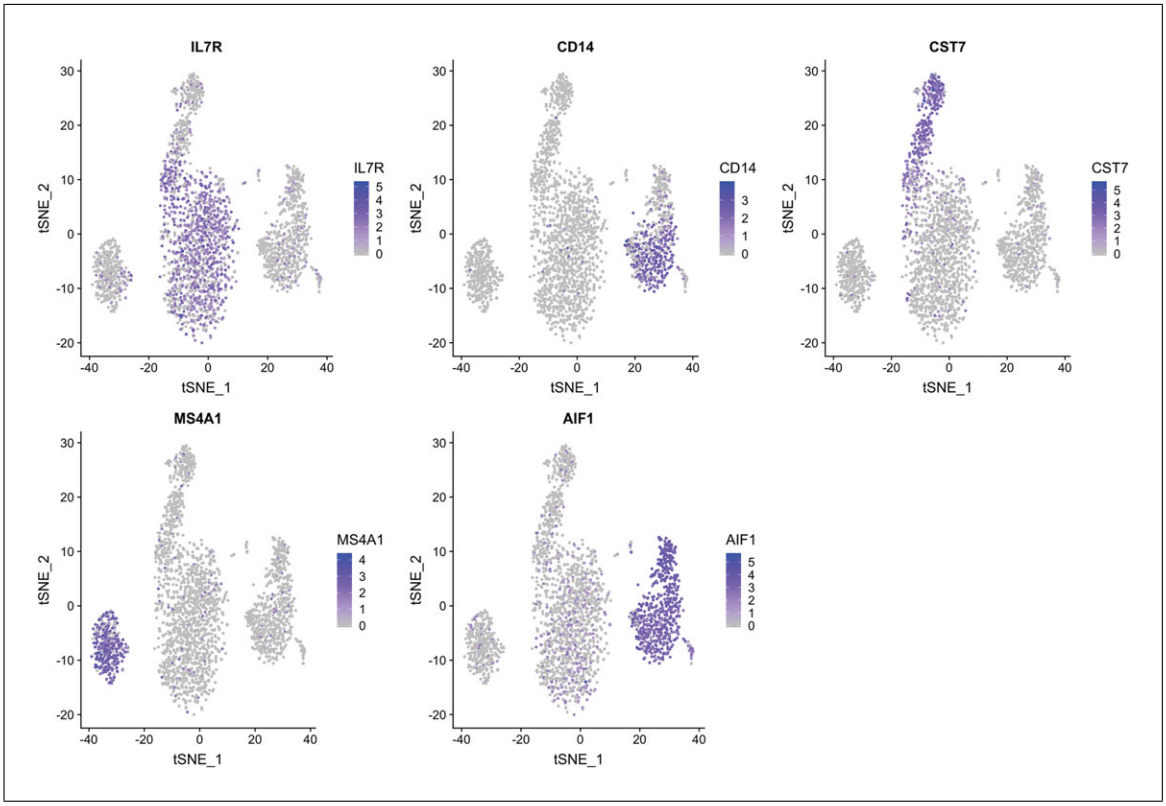

Figure 8 tSNE plots color-coded by expression of selected marker genes.

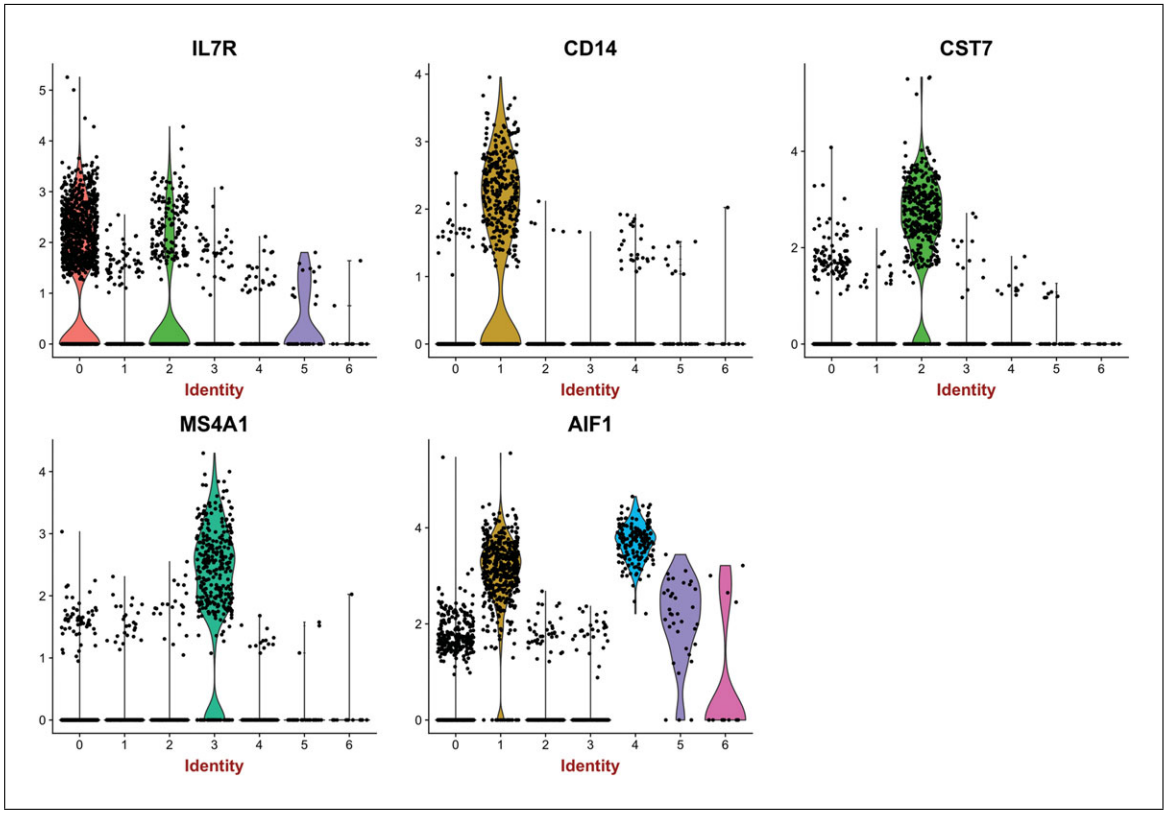

Figure 9 Violin plots showing the distributions of expression levels of selected marker genes in each cluster.

Table 1 Canonical Marker Gene(s) for Each Cell Type

Cluster ID	Marker(s)	Cell type
0	*IL7R*	$CD4^+$ T cells
1	*CD14*, *LYZ*	$CD14^+$ monocytes
2	*CD8A*	CD8 T cells
3	*MS4A1*	B cells
4	*FCGR3A*, *MS4A7*	$FCGR3A^+$ monocytes
5	*FCER1A*, *CST3*	Dendritic cells
6	*PPBP*	Megakaryocytes

the *CD8A* gene as a marker. In such cases, it may be appropriate to investigate the expression of these known marker genes among the cell clusters identified by total expression patterns in the previous steps. In the current example, the sample contains various types of blood cells that are well known to express marker genes (Table 1). These characterized cell type markers appear among the gene markers of individual clusters determined in the previous steps, which provides the opportunity to assign these clusters to known biological cell types (Fig. 7). After manual inspection of cluster-specific gene markers identified in steps 15 to 16 and identification of known cell type markers, re-label and re-plot the tSNE clusters with the following commands. As a result, the tSNE clusters will be labeled by the names of biological cell types (Fig. 10) defined in the second command.

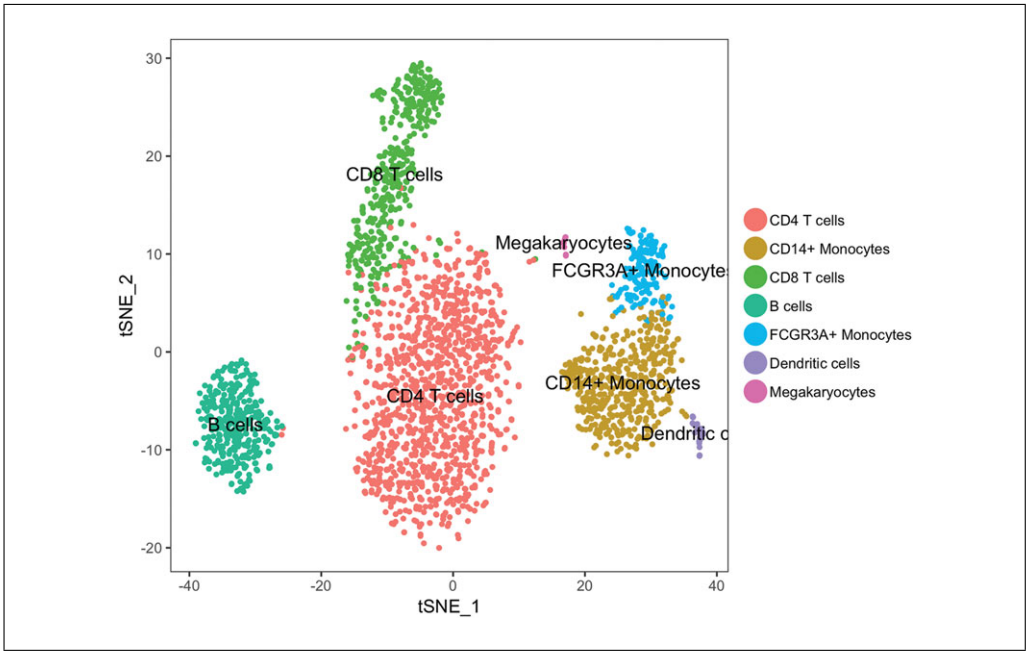

Figure 10 Single-cell clusters in tSNE marked with known cell type by canonical marker genes.

```
current.cluster.ids <- c(0, 1, 2, 3, 4, 5, 6)
new.cluster.ids <- c("CD4 T cells", "CD14+
  Monocytes","CD8 T cells","B cells", "FCGR3A+
  Monocytes", "Dendritic cells", "Megakaryocytes")
scRNA@ident <- plyr::mapvalues(x = scRNA@ident,
  from = current.cluster.ids, to = new.cluster.ids)
TSNEPlot(object = scRNA, do.label = TRUE)
```

ANALYSIS, VISUALIZATION, AND INFERENCE OF SINGLE-CELL TRAJECTORIES USING MONOCLE

BASIC PROTOCOL 2

Another example of a widely used method for scRNA-seq analysis is Monocle (Qiu et al., 2017), whose goals and functionality are in general similar to those of SEURAT. A key distinction of Monocle's approach is that instead of grouping similar cells into a few clusters, which may correspond, for example, to distinct cell types, Monocle aims to infer continuous trajectories connecting individual cells, which may correspond to a snapshot of a continuous biological process, for example the gradual process of cell differentiation. Here we describe the basics of the Monocle workflow using the same example dataset as in Basic Protocol 1 and compare the results produced by SEURAT and Monocle.

Necessary Resources

Software

R: *https://www.r-project.org*

Hardware

Computer with Unix, Linux, or Mac OS X operating systems with ≥1 Gb RAM

1. This step is the same as in Basic Protocol 1: process raw FASTQ files with scRNA-seq reads into a single-cell gene expression table with rows corresponding to genes and columns corresponding to individual single cells.

The specifics of this processing depend on the choice of the experimental scRNA-seq platform. For the purpose of learning how to use this protocol, the user can download an example of a gene expression table generated from 2700 PBMCs using a standard 10X Genomics workflow: sample "3k PBMCs from a Healthy Donor" at https://support.10xgenomics.com/single-cell-gene-expression/datasets. The table can also be downloaded from https://github.com/MolBioBioinformatics/scRNA. Each step of the protocol can then be carried out using the data in this table so that users can verify that they are carrying out the protocol correctly by determining whether the results they obtain match the outputs shown in each of the figures.

2. Load the Monocle package in R and import the previous dataset (the single-cell gene-expression table) from SEURAT to Monocle.

```
library(monocle)
scRNA_monocle <- importCDS(scRNA)
```

3. Use the following commands in Monocle to carry out a workflow similar to SEURAT: estimate library size for normalization; calculate the dispersion of expression, i.e., the extent of variation across cells, for each gene; and filter out genes with low dispersion, i.e., genes that are ubiquitously expressed with little variation from cell to cell (and thus do not have a strong contribution to the distinction between cells).

```
scRNA_monocle <- estimateSizeFactors(scRNA_monocle)
scRNA_monocle <- estimateDispersions(scRNA_monocle)
disp_table <- dispersionTable(scRNA_monocle)
ordering_genes <- subset(disp_table, mean_expression
  >= 0.1)
scRNA_monocle <- setOrderingFilter(scRNA_monocle,
  ordering_genes)
```

4. Monocle learns the "trajectory" of cells in two steps. First, it reduces the dimensionality of the data, connects most similar cells as steps in a hypothetical process, and constructs a sequence of transcriptionally similar cells using the MST algorithm. Use the following commands for this:

```
scRNA_monocle <- reduceDimension(scRNA_monocle)
scRNA_monocle <- orderCells(scRNA_monocle)
```

5. Use command `plot_cell_trajectory` to visualize the inferred trajectories (Fig. 11):

```
plot_cell_trajectory(scRNA_monocle)
```

6. As a more advanced version of the same command, color cells by the expression of a few genes of interest (Fig. 12) — in this case, all cells are colored based on the expression of marker genes identified in step 16 of Basic Protocol 1:

```
plot_cell_trajectory(scRNA_monocle,markers=c
  (c("IL7R",
  "CD14","CST7","MS4A1","AIF1")),use_color_gradient =
  TRUE)
```

The Monocle trajectories organize cells differently from the PCA or t-SNE described in steps 10 and 11 of Basic Protocol 1. The cells are positioned in a linear order along a few directions, as opposed to being grouped generally into sets of similar cells. This different assumption about the cell population produces different results. For example, the marker genes for distinct cell groups in PCA and t-SNE plots (Figs. 3, 5, and 8) are now present in parts of trajectories and their branches (Fig. 12), and some are present in multiple trajectories.

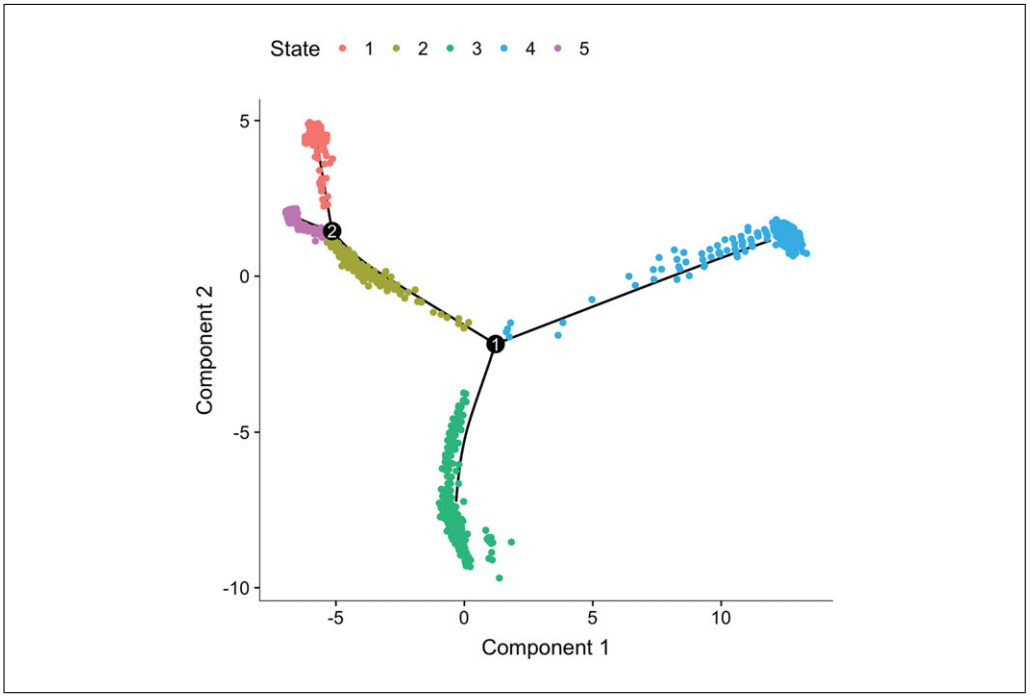

Figure 11 Single-cell trajectory and states inferred by Monocle.

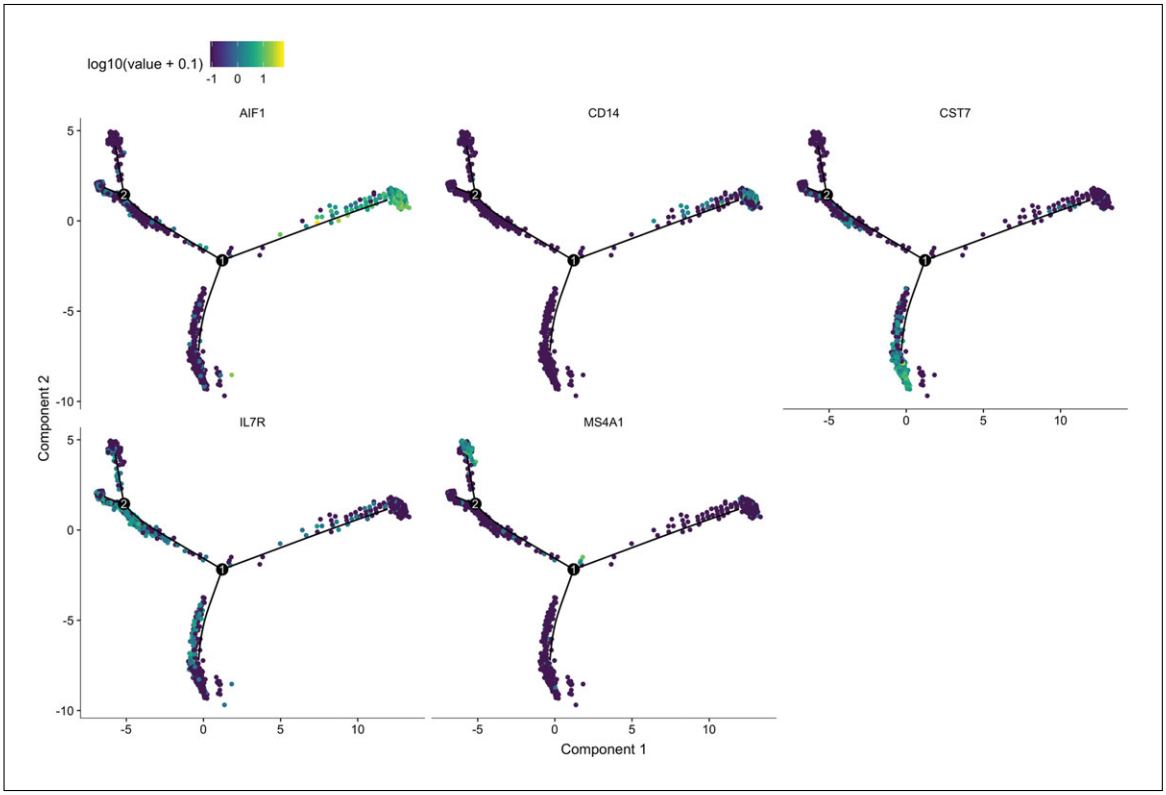

Figure 12 Levels of expression of individual genes color-coded over single-cell trajectories and states that were inferred by Monocle.

Figure 13 shows the mapping of cell from Monocle trajectories (states) back to the t-SNE plot produced by Seurat. The Monocle-defined state 1 is enriched with cells that highly express MS4A1, corresponding to the SEURAT-defined cluster 3 (B cells; see also Figs. 8 and 9). Similarly, Monocle-defined states 2, 3, and 4 correspond to

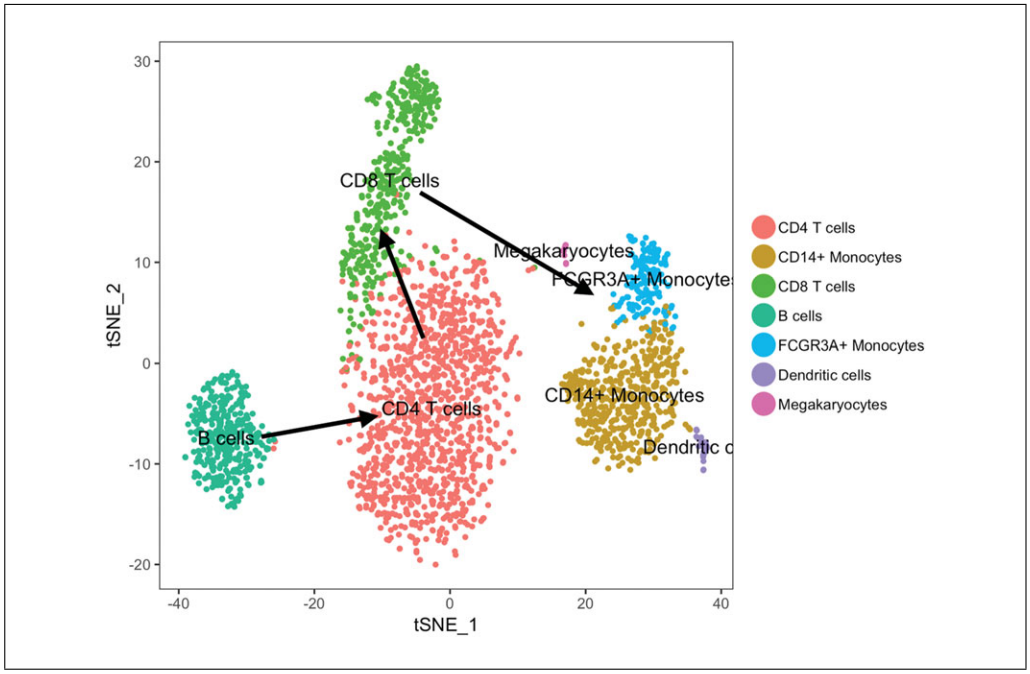

Figure 13 tSNE single-cell clusters, marked by the pseudotemporal trajectory between clusters inferred by Monocle.

SEURAT-defined cluster 0 ($CD4^+$ T cells), cluster 2 ($CD8^+$ T cells), and clusters 1 and 4 ($CD14^+$ monocytes and $FCGR3A^+$ monocytes), respectively (Fig. 13).

7. Treating gene expression as a function of pseudotime, find genes whose expression changes are significantly associated with pseudotime:

```
diff_test_res <- differentialGeneTest
  (scRNA_subset,fullModelFormulaStr =
  "~sm.ns(Pseudotime)")
```

In the Monocle framework, the movement along the sequence of connected cells corresponds to the movement between stages of a hypothetical process (e.g., cell differentiation) in pseudotime. Pseudotime is the measure of advancement along a trajectory; it corresponds to the directionality of the biological process in real time, although the real timing between stages of a trajectory cannot be precisely determined from these data. Monocle assigns a pseudotime value to each cell as a step along a trajectory.

8. These pseudotime-dependent genes may be markers of progression along an underlying biological process: e.g., differentiation markers. Cluster and visualize the expression patterns of all pseudotime-dependent genes in all cells (Fig. 14):

```
sig_gene_names <- row.names(subset(diff_test_res,
  qval < 1e-3))
plot_pseudotime_heatmap(scRNA_monocle[sig_gene_
  names,],
num_clusters = 7,
cores = 1,
show_rownames = T)
```

In the resulting heatmap (Fig. 14), each row represents a gene, whereas each column represents cell progression in pseudotime, with the expression of the gene at a given pseudotime point indicated by the color. The genes are split into clusters based on the similarity of their expression patterns. For example, genes in cluster 1 (top cluster, marked by the green bar at left) are highly expressed at early pseudotime — i.e., at the beginning

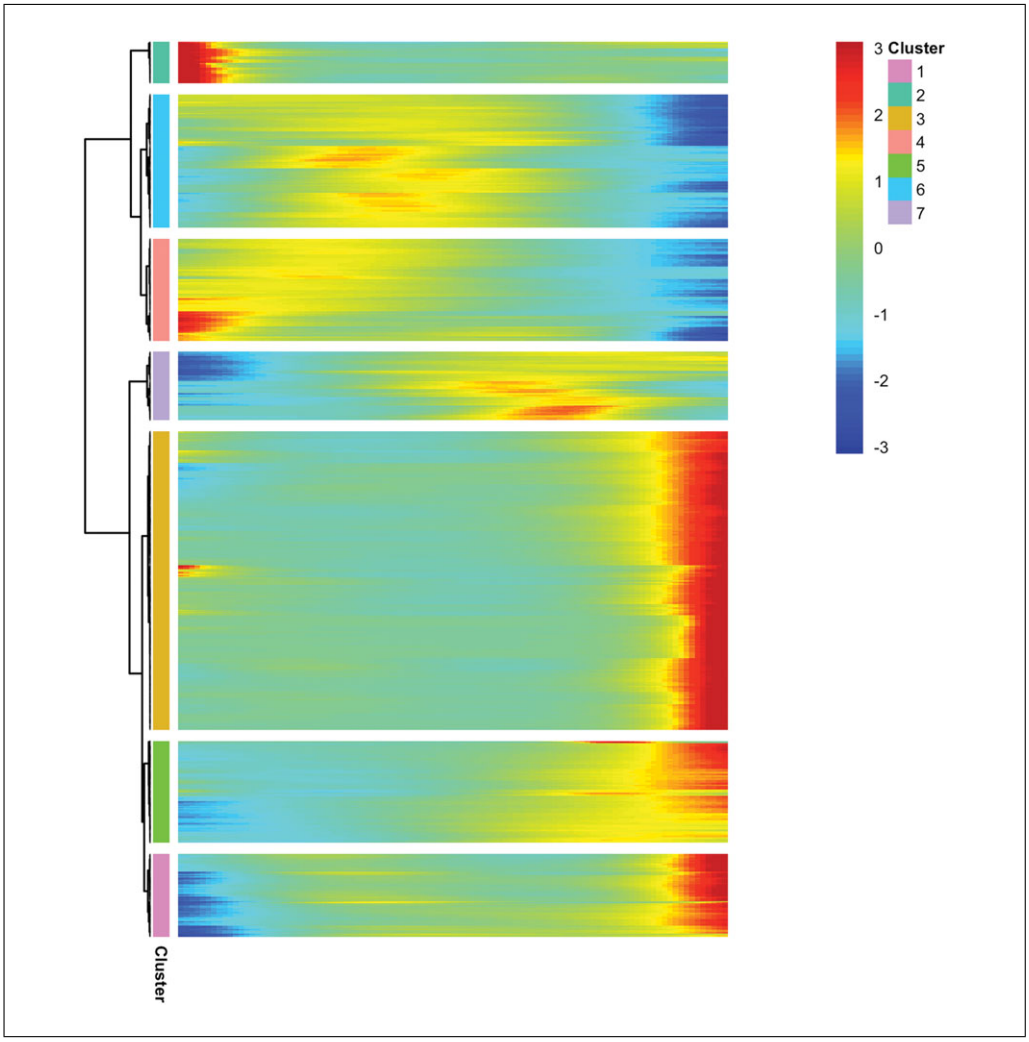

Figure 14 Heatmap of gene expression shown by color across all cell clusters (*y* axis), plotted against pseudotime (*x* axis) inferred by Monocle within each cluster.

of the trajectories — and are downregulated later, whereas genes in cluster 3 (the largest cluster, marked by the yellow bar at left) start being highly expressed only at a later pseudotime (Fig. 14).

COMMENTARY

Background Information

Single-cell RNA-seq (scRNA-seq) has the capacity to reveal the intercellular heterogeneity within a specific tissue at an unprecedented resolution. Recent studies using scRNA-seq can infer cell lineages (Treutlein et al., 2014), identify functional subpopulations of cells (Trapnell et al., 2014), and highlight cell-specific biological characteristics (Tang et al., 2010). Despite the rapid development of scRNA-seq technologies, major analytical challenges remain due to the unique nature of scRNA-seq experiments. To ensure that the scRNA-seq data can be properly analyzed, it is crucial to apply robust computational methods that are tailored to single-cell datasets, with a clear understanding of the strengths and limitations of each method.

Calling statistically significant differences in the levels of gene expression between groups of samples is of great significance in RNA-seq analysis. Simple statistical methods such as the *t*-test or Wilcoxon rank-sum test are used in scRNA-seq workflows such as SINCERA (Guo, Wang, Potter, Whitsett, & Xu, 2015). EdgeR and CuffDiff, which were developed for bulk RNA-seq, can also be applied to scRNA-seq data (Schurch et al., 2016). To account for unique dropout events in scRNA-seq, SCDE, and PAGODA were developed specifically for single-cell differential expression (Kharchenko et al., 2014), assuming a

low-mean Poisson distribution for dropout genes. MAST is another scRNA-seq differential expression detection method that uses a two-part generalized linear model (Finak et al., 2015).

In the past few years, the identification of alternative splicing has become a prominent component of single-cell RNA-seq studies, as a variety of studies have revealed heterogeneity in isoform expression in single cells (Marinov et al., 2014; Song et al., 2017). The mixture of isoforms (MISO) model, a statistical model focusing on alternative splicing detection developed for bulk RNA-seq, has also been applied in single-cell isoform research (Shalek et al., 2013). MISO evaluates sequence reads aligned to splice junctions to estimate expression at the exon level for alternatively spliced exons and isoforms. BRIE and Expedition are built on the same premises as MISO and assess exon expression specifically for scRNA-seq datasets. BRIE applies a Bayesian hierarchical model for isoform estimation (Huang & Sanguinetti, 2017), and the Expedition software suite enables systematic analysis of alternative splicing from single-cell RNA-seq data. Finally, some of the more standard RNA-seq tools (e.g., RSEM; Li & Dewey, 2011), which were originally designed to analyze gene expression in larger tissue or cell samples, have recently incorporated special options for single-cell analysis based on the reference sequences of mature transcripts, as opposed to the exon annotation in the genome.

Critical Parameters and Troubleshooting

Quality control for scRNA-seq involves the analysis of overall gene expression patterns and the number of genes or reads detected per cell (Butler et al., 2018, Kumar et al., 2014). Because of the high level of noise in scRNA-seq datasets, it is necessary to filter out low-quality data. Various methods have been developed to filter genes that are expressed in too few samples (Brennecke et al., 2013, Petropoulos et al., 2016) based on a threshold of FPKM > 1 (where FPKM is the number of fragments per kilobase of transcript per million mapped reads). For experiments that quantify gene expression with UMI counting, one can directly set up a molecule number threshold. OEFinder is designed to identify artifactual genes from scRNA-seq data using the Fluidigm C1 platform for cell capture (Leng et al., 2016).

Since PCA is heavily affected by outlier cells, the user may try various levels of filtering stringency in removing outliers and inspect the corresponding PCA plots and clustering results in order to find the optimal stringency as a tradeoff between the number of removed cells and the robust, biologically meaningful separation between cell clusters.

Expression signatures specific to stages of the cell cycle are an important confounding factor that often underlie the apparent separation of cells into clusters, especially among actively dividing cells of the same general type. One approach to mitigate this factor is to specifically exclude genes whose expression is known to be cell cycle dependent.

When the entire dataset consists of multiple biological replicates (repeated runs of the scRNA-seq experiment on replicated cell samples), systematic variations between cell samples may result in batch effects, which pose substantial problems to downstream statistical analysis. For studies that use more traditional read counting of transcripts and do not involve UMIs, the COMBAT method (Johnson, Li, & Rabinovic, 2007) based on empirical Bayes frameworks is one potential approach that can be directly applied in attempt to mitigate batch effects. Another obvious approach is to analyze each replicate sample separately and draw biological conclusions based on the consistency between results for each replicate.

As discussed above, the inference of cell trajectories implemented in Monocle is based on the assumption that the population of cells represents a snapshot of a temporal biological process or processes. In each specific experiment, this assumption has to be carefully evaluated in the context of the individual experiment and the general goals of the analysis. In single-cell studies that do not have a clearly expected timing component, the user should pay special attention to the interpretation of the biological significance of the generated trajectories.

Time Considerations

The timing depends on the number of cells and number of reads per cell. After installation of all required packages, the full analysis of workflow usually takes 1 to 2 days.

Acknowledgements

This work was supported in part by U.S. National Institutes of Health grant P30 DK040561.

Conflicts of Interest

The authors have declared no conflicts of interest for this article.

Literature Cited

Anders, S., Pyl, P. T., & Huber, W. (2015). HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics*, *31*(2), 166–169. doi: 10.1093/bioinformatics/btu638.

Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A. A., Zhang, X., Proserpio, V., ... Heisler, M. G. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, *10*(11), 1093–1095. doi: 10.1038/nmeth.2645.

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., & Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, *36*(5), 411–420. doi: 10.1038/nbt.4096.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., ... Gingeras, T. R. (2013). STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15–21. doi: 10.1093/bioinformatics/bts635.

Dobin, A., & Gingeras, T. R. (2015). Mapping RNA-seq reads with STAR. *Current Protocols in Bioinformatics*, *51*(1), 11.14.1–11.14.19.

Fan, J., Salathia, N., Liu, R., Kaeser, G. E., Yung, Y. C., Herman, J. L., ... Kharchenko, P. V. (2016). Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nature Methods*, *13*(3), 241–244. doi: 10.1038/nmeth.3734.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., ... Linsley, P. S. (2015). MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, *16*(1), 278. doi: 10.1186/s13059-015-0844-5.

Guo, M., Wang, H., Potter, S. S., Whitsett, J. A., & Xu, Y. (2015). SINCERA: A pipeline for single-cell RNA-seq profiling analysis. *PLoS Computational Biology*, *11*(11), e1004575. doi: 10.1371/journal.pcbi.1004575.

Huang, Y., & Sanguinetti, G. (2017). BRIE: Transcriptome-wide splicing quantification in single cells. *Genome Biology*, *18*(1), 123. doi: 10.1186/s13059-017-1248-5.

Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., ... Linnarsson, S. (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, *11*(2), 163–166. doi: 10.1038/nmeth.2772.

Ji, F., & Sadreyev, R. I. (2018). RNA-seq: Basic bioinformatics analysis. *Current Protocols in Molecular Biology*, *124*(1), e68. doi: 10.1002/cpmb.68.

Johnson, W. E., Li, C., & Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, *8*(1), 118–127. doi: 10.1093/biostatistics/kxj037.

Kharchenko, P. V., Silberstein, L., & Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature Methods*, *11*(7), 740–742. doi: 10.1038/nmeth.2967.

Kumar, M. E., Bogard, P. E., Espinoza, F. H., Menke, D. B., Kingsley, D. M., & Krasnow, M. A. (2014). Defining a mesenchymal progenitor niche at single-cell resolution. *Science*, *346*(6211), 1258810. doi: 10.1126/science.1258810.

Leng, N., Choi, J., Chu, L. F., Thomson, J. A., Kendziorski, C., & Stewart, R. (2016). OEFinder: A user interface to identify and visualize ordering effects in single-cell RNA-seq data. *Bioinformatics*, *32*(9), 1408–1410. doi: 10.1093/bioinformatics/btw004.

Li, B., & Dewey, C. N. (2011). RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, *12*(1), 323. doi: 10.1186/1471-2105-12-323.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Marinov, G. K., Williams, B. A., McCue, K., Schroth, G. P., Gertz, J., Myers, R. M., & Wold, B. J. (2014). From single-cell to cell-pool transcriptomes: Stochasticity in gene expression and RNA splicing. *Genome Research*, *24*(3), 496–510. doi: 10.1101/gr.161034.113.

Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, *14*(4), 417–419. doi: 10.1038/nmeth.4197.

Petropoulos, S., Edsgärd, D., Reinius, B., Deng, Q., Panula, S. P., Codeluppi, S., ... Lanner, F. (2016). Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, *165*(4), 1012–1026. doi: 10.1016/j.cell.2016.03.023.

Pierson, E., & Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, *16*(1), 241. doi: 10.1186/s13059-015-0805-z.

Qiu, X., Hill, A., Packer, J., Lin, D., Ma, Y. A., & Trapnell, C. (2017). Single-cell mRNA quantification and differential analysis with Census. *Nature Methods*, *14*(3), 309–315. doi: 10.1038/nmeth.4150.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., & Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, *33*(5), 495–502. doi: 10.1038/nbt.3192.

Schurch, N. J., Schofield, P., Gierliński, M., Cole, C., Sherstnev, A., Singh, V., ... Blaxter, M. (2016). How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA*, *22*(6), 839–851. doi: 10.1261/rna.053959.115.

Shalek, A. K., Satija, R., Adiconis, X., Gertner, R. S., Gaublomme, J. T., Raychowdhury, R.,

... Trombetta, J. J. (2013). Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, *498*(7453), 236–240. doi: 10.1038/nature12172.

Shin, J., Berg, D. A., Zhu, Y., Shin, J. Y., Song, J., Bonaguidi, M. A., ... Song, H. (2015). Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell*, *17*(3), 360–372. doi: 10.1016/j.stem.2015.07.013.

Stegle, O., Teichmann, S. A., & Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, *16*(3), 133–145. doi: 10.1038/nrg3833.

Song, Y., Botvinnik, O. B., Lovci, M. T., Kakaradov, B., Liu, P., Xu, J. L., & Yeo, G. W. (2017). Single-cell alternative splicing analysis with expedition reveals splicing dynamics during neuron differentiation. *Molecular Cell*, *67*(1), 148–161. doi: 10.1016/j.molcel.2017.06.003.

Tang, F., Barbacioru, C., Bao, S., Lee, C., Nordman, E., Wang, X., ... Surani, M. A. (2010). Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell RNA-Seq analysis. *Cell Stem Cell*, *6*(5), 468–478. doi: 10.1016/j.stem.2010.03.015.

Treutlein, B., Brownfield, D. G., Wu, A. R., Neff, N. F., Mantalas, G. L., Espinoza, F. H., ... Quake, S. R. (2014). Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature*, *509*(7500), 371–375. doi: 10.1038/nature13173.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., ... Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, *32*(4), 381. doi: 10.1038/nbt.2859.

Trapnell, C., Pachter, L., & Salzberg, S. L. (2009). TopHat: Discovering splice junctions with RNA-Seq. *Bioinformatics*, *25*(9), 1105–1111. doi: 10.1093/bioinformatics/btp120.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., ... Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, *7*(3), 562–578. doi: 10.1038/nprot.2012.016.

Yan, L., Yang, M., Guo, H., Yang, L., Wu, J., Li, R., ... Huang, J. (2013). Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nature Structural & Molecular Biology*, *20*(9), 1131–1139.