

# scRNA-seq Cheatsheet

---

The tables below consist of valuable functions or commands that will help you through this module.

Each table represents a different library/tool and its corresponding commands.

Please note that these tables are not intended to tell you all the information you need to know about each command.

The hyperlinks found in each piece of code will take you to the documentation for further information on the usage of each command.

Please be aware that the documentation will generally provide information about the given function's most current version (or a recent version, depending on how often the documentation site is updated).

This will usually (but not always!) match what you have installed on your machine.

If you have a different version of R or other R packages, the documentation may differ from what you have installed.

## Table of Contents

- [Base R](#)
- [Salmon](#) and [alevinQC](#)
- [SingleCellExperiment](#), [S4Vectors](#), [txmimeta](#), and [DropletUtils](#)
- [scraper](#) and [scater](#)
- [purrr](#), [stringr](#), and [tibble](#)
- [SingleR](#)

## Base R

Read the Base [R documentation](#).

| Library/Package | Piece of Code   | What it's called              | What it does  |
|-----------------|---|-------------------------------|---|
| Base R          | <code>rowSums()</code>                                | Row sums                      | Calculates sums for each row  |
| Base R          | <code>colSums()</code>                                | Column sums                   | Calculates sums for each column   |
| Base R          | <code>t()</code>                                      | Transpose                     | Returns the transpose of a matrix or data frame   |
| Base R          | <code>prcomp()</code>                                 | Principal Components Analysis | Executes a principal components analysis on specified matrix or data frame  |
| Base R          | <pre>&lt;-<br/>function(x)<br/>{ &lt;code&gt; }</pre> | Function                      | Creates a function that would take the defined parameters as input and execute the commands within the curly braces |

## Salmon and alevinQC

Read the command-line tool [Salmon documentation](#).

Read the R package [alevinQC documentation](#).

| Software/package      | Piece of Code                 | What it's called | What it does  |
|-----------------------|-------------------------------|------------------|---|
| Salmon                | <code>salmon alevin</code>    | Salmon Alevin    | Runs the Alevin quantification from the command line                            |
| <code>alevinQC</code> | <code>alevinQCReport()</code> | Alevin QC Report | Produces a QC (quality check) report from the <code>salmon alevin</code> output |

## SingleCellExperiment, S4Vectors, tximeta, and DropletUtils

Read the [SingleCellExperiment package documentation \(and e-book\)](#), and a [vignette on its usage](#).

Note that some of the `SingleCellExperiment` functions link to documentation from other packages like `SummarizedExperiment` or `ExperimentSubset`.

In fact, `SingleCellExperiment` objects are based around existing Bioconductor functions in those packages, so the function usage is equivalent!

Read the [tximeta package documentation](#), and a [vignette on its usage](#).

Read the [DropletUtils package documentation](#).

| Library/Package                   | Piece of Code                       | What it's called       | What it does  |
|-----------------------------------|-------------------------------------|------------------------|---|
| <code>SingleCellExperiment</code> | <code>SingleCellExperiment()</code> | Single Cell Experiment | Creates a <code>SingleCellExperiment</code> object  |
| <code>SingleCellExperiment</code> | <code>colData()</code>              | Column Data            | Extracts and stores cell-level metadata that describes features of the <code>SingleCellExperiment</code> object                   |
| <code>SingleCellExperiment</code> | <code>rowData()</code>              | Row Data               | Extracts and stores gene-level metadata that describes features of the <code>SingleCellExperiment</code> object                   |
| <code>SingleCellExperiment</code> | <code>assay()</code>                | Assay                  | Extracts and stores a given assay from a <code>SingleCellExperiment</code> object   |
| <code>SingleCellExperiment</code> | <code>assayNames()</code>           | Assay names            | Returns a vector of the names of all assays in a <code>SingleCellExperiment</code> object   |
| <code>SingleCellExperiment</code> | <code>logcounts()</code>            | Log counts             | Extracts and stores log-transformed single-cell experiment count data as an assay of the <code>SingleCellExperiment</code> object |
| <code>SingleCellExperiment</code> | <code>counts()</code>               | Counts                 | Extracts and stores raw single-cell experiment count data as an assay of the <code>SingleCellExperiment</code> object             |
| <code>SingleCellExperiment</code> | <code>reducedDim()</code>           | Reduced dim            | Extracts or stores a given reduced dimension from a <code>SingleCellExperiment</code> object                                      |
| <code>SingleCellExperiment</code> | <code>reducedDimNames()</code>      | Reduced dim names      | Returns a vector of the names of all reduced dimensions in a <code>SingleCellExperiment</code> object                             |

| Library/Package | Piece of Code                       | What it's called   | What it does   |
|-----------------|-------------------------------------|--|--|
| S4Vectors       | <code>DataFrame()</code>            | Data frame   | Not to be confused with <code>data.frame()</code> from Base R. This is a slightly different data frame-like object needed for storing information in <code>SingleCellExperiment</code> object's <code>colData</code> slot. |
| tximeta         | <code>tximeta()</code>              | Transcript Quantification Import with Automatic Metadata | Load a directory of results produced by Salmon/or alevin output, including the associated metadata   |
| DropletUtils    | <code>read10xCounts()</code>        | Read 10x counts  | Load data from a 10X Genomics experiment into R  |
| DropletUtils    | <code>emptyDrops()</code>           | Empty drops  | Use the overall gene expression patterns in the sample to identify empty droplets  |
| DropletUtils    | <code>emptyDropsCellRanger()</code> | Empty drops Cell Ranger                                  | Use an approach analogous to Cell Ranger's algorithm to identify empty droplets  |

## scrn and scater

Read the [scrn package documentation](#), and a [vignette on its usage](#).

Read the [scater package documentation](#), and a [vignette on its usage](#).

| Library/Package | Piece of Code                    | What it's called                | What it does  |
|-----------------|----------------------------------|---------------------------------|---|
| scrn            | <code>quickCluster()</code>      | Quick Clustering                | Groups similar cells into clusters which are stored in the <code>SingleCellExperiment</code> object and are used for the calculation of size factors by <code>scrn::computeSumFactors</code>  |
| scrn            | <code>computeSumFactors()</code> | Compute Sum Factors             | Returns a numeric vector of computed sum factors for each cell cluster stored in the <code>SingleCellExperiment</code> object. The cluster-based size factors are deconvolved into cell-based size factors that are stored in the <code>SingleCellExperiment</code> object and used by the <code>scrn::normalize</code> function for the normalization of each cell's gene expression profile |
| scrn            | <code>getTopHVGs()</code>        | Get top highly variable genes   | Identify variable genes in a <code>SingleCellExperiment</code> object, based on variance  |
| scrn            | <code>modelGeneVar()</code>      | model per gene variance         | Model the per gene variance of a <code>SingleCellExperiment</code> object   |
| scrn            | <code>findMarkers()</code>       | Find marker genes               | Find candidate marker genes for clusters of cells   |
| scater          | <code>logNormCounts()</code>     | Normalize log counts            | Returns the <code>SingleCellExperiment</code> object with normalized expression values for each cell, using the size factors stored in the object   |
| scater          | <code>addPerCellQC()</code>      | Add per cell quality control    | For a <code>SingleCellExperiment</code> object, calculate and add quality control per cell and store in <code>colData</code>  |
| scater          | <code>addPerFeatureQC()</code>   | Add per feature quality control | For a <code>SingleCellExperiment</code> object, calculate and add quality control per feature (genes usually) and store in <code>rowData</code>   |
| scater          | <code>calculatePCA()</code>      | Calculate PCA                   | Calculates principal components analysis on a <code>SingleCellExperiment</code> object, returning a PCA matrix  |
| scater          | <code>runPCA()</code>            | Run PCA                         | Calculates principal components analysis on a <code>SingleCellExperiment</code> object, returning an SCE object with a PCA reduced dimension  |

| Library/Package | Piece of Code                 | What it's called        | What it does  |
|-----------------|-------------------------------|-------------------------|---|
| scater          | <code>calculateUMAP()</code>  | Calculate UMAP          | Calculates uniform manifold approximate projection on a <code>SingleCellExperiment</code> object, returning a UMAP matrix                               |
| scater          | <code>runUMAP()</code>        | Run UMAP                | Calculates uniform manifold approximate projection on a <code>SingleCellExperiment</code> object, returning an SCE object with a UMAP reduced dimension |
| scater          | <code>calculateTSNE()</code>  | Calculate t-SNE         | Calculates t-stochastic neighbor embedding on a <code>SingleCellExperiment</code> object, returning an SCE object with a TSNE reduced dimension         |
| scater          | <code>runTSNE()</code>        | Calculate UMAP          | Calculates t-stochastic neighbor embedding on a <code>SingleCellExperiment</code> object, returning a t-SNE matrix                                      |
| scater          | <code>plotReducedDim()</code> | Plot reduced dimensions | Plot a given reduced dimension slot from a <code>SingleCellExperiment</code> object by its name   |
| scater          | <code>plotPCA()</code>        | Plot PCA                | Plot the "PCA"-named reduced dimension slot from a <code>SingleCellExperiment</code> object   |
| scater          | <code>plotUMAP()</code>       | Plot UMAP               | Plot the "UMAP"-named reduced dimension slot from a <code>SingleCellExperiment</code> object  |

## purrr, stringr, and tibble

Read the [purrr](#) package documentation.

Read the [stringr](#) package documentation.

Read the [tibble](#) package documentation.

| Library/Package | Piece of Code             | What it's called | What it does  |
|-----------------|---------------------------|------------------|---|
| purrr           | <code>map()</code>        | map              | Apply a function across each element of list; return a list       |
| purrr           | <code>map_df()</code>     | map df           | Apply a function across each element of list; return a data frame |
| purrr           | <code>imap()</code>       | imap             | Apply a function across each element of list and its index/names  |
| stringr         | <code>str_remove()</code> | String remove    | Remove matched string patterns                                    |
| tibble          | <code>as_tibble()</code>  | As tibble        | Coerce <code>data.frame</code> or matrix to a tibble              |

Note that `purrr::map()` functions can take advantage of R's new (as of version 4.1.0) [anonymous function syntax](#):

```
# One-line syntax:
\(x) # function code goes here #

# Multi-line syntax:
\(x) {
  # function code goes      #
  # inside the curly braces #
}

# Example: Use an anonymous function with `purrr::map()`
# to get the colData's rownames for each SCE in `list_of_sce_objects`
purrr::map(
  list_of_sce_objects,
  \(x) rownames(colData(x))
)
```

## SingleR

Read the [SingleR](#) package documentation, and an [e-book on its usage](#).

| Library/Package | Piece of Code               | What it's called             | What it does   |
|-----------------|-----------------------------|------------------------------|--|
| SingleR         | <code>trainSingleR()</code> | Train the SingleR classifier | Build a <code>SingleR</code> classifier model object from an annotated reference dataset |

| Library/Package | Piece of Code                  | What it's called            | What it does   |
|-----------------|--------------------------------|-----------------------------|--|
| SingleR         | <code>classifySingleR()</code> | Classify cells with SingleR | Use a <code>SingleR</code> model object to assign cell types to the cells in an <code>SCE</code> object  |
| SingleR         | <code>SingleR()</code>         | Annotate scRNA-seq data     | Combines <code>trainSingleR()</code> and <code>classifySingleR()</code> to assign cell types to an <code>SCE</code> object from an annotated reference dataset |