Package: scFeatureFilter (via r-universe)

November 25, 2026

Type Package

Title A correlation-based method for quality filtering of single-cell RNAseq data

Version 1.34.0

Description An R implementation of the correlation-based method developed in the Joshi laboratory to analyse and filter processed single-cell RNAseq data. It returns a filtered version of the data containing only genes expression values unaffected by systematic noise.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 4.5.0)

Imports dplyr (>= 0.7.3), ggplot2 (>= 2.1.0), magrittr (>= 1.5), rlang (>= 0.1.2), tibble (>= 1.3.4), stats, methods

RoxygenNote 7.3.2

Encoding UTF-8

Suggests testthat, knitr, rmarkdown, BiocStyle, MASS, SingleCellExperiment, SummarizedExperiment, scRNAseq, cowplot

VignetteBuilder knitr

biocViews ImmunoOncology, SingleCell, RNASeq, Preprocessing, GeneExpression

URL https://bioconductor.org/packages/scFeatureFilter/

BugReports https://github.com/gdevailly/scFeatureFilter/issues

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:40:09 UTC

RemoteUrl https://github.com/bioc/scFeatureFilter

RemoteRef RELEASE_3_24

RemoteSha 92a76d2df70b33143db6260be6b7608c95d7b6b2

Contents

bin_scdata . . . 2
calculate_cvs . . . 3
correlate_windows . . . 4
correlations_to_densities . . . 6
define_top_genes . . . 7
determine_bin_cutoff . . . 8
filter_expression_table . . . 9
get_mean_median . . . 10
plot_correlations_distributions . . . 11
plot_mean_variance . . . 12
plot_metric . . . 13
plot_top_window_autocor . . . 14
sc_feature_filter . . . 15
scData_hESC . . . 16

Index **18**

`bin_scdata` *Bin genes by mean expression.*

Description

Divides the genes that were not included in the top window in windows of the same size with decreasing mean expression levels.

Usage

```
bin_scdata(dataset, window_number = NULL, window_size = NULL, verbose = TRUE)
```

Arguments

`dataset`	A list, containing the top window generated by `extract_top_genes` as the first element, and the rest of undivided genes as the second. Usually the output of `define_top_genes`
`window_number`	An integer, indicating the number of bins to be used.
`window_size`	An integer, indicating the number of genes to be included in each window. Ignored if `window_size` is defined.
`verbose`	A boolean. Should the function print a message about window size or the number of windows created?

Details

Two binning methods are available:

- `window_number`: Divides the genes into the number of windows specified.
- `window_size`: Divides the genes into windows of the size specified.

This function adds a bin number column to the data frame.

This function is designed to take the list output by the `extract_top_window` function as an argument, operating only on the second element of it. Once the genes in it have been binned, both elements of the list are bound together in a data frame and returned. The output contains a new column `bin`, which indicates the window number assigned to each gene.

Value

A data frame containing the binned genes.

Examples

```
library(magrittr)
expMat <- matrix(
    c(1, 1, 1,
      1, 2, 3,
      0, 1, 2,
      0, 0, 2),
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 1) %>%
    bin_scdata(window_number = 2)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 1) %>%
    bin_scdata(window_size = 1)
```

---

calculate_cvs *Compute mean expression level, standard deviation and coefficient of variation of each feature.*

Description

Compute mean expression level, standard deviation and coefficient of variation (CV) of each feature (i.e. gene or transcript) in the supplied data. Filter features with high proportion of 0 expression.

Usage

```
calculate_cvs(data, max_zeros = 0.75, sce_assay = NULL)
```

Arguments

data	A data frame, a matrix or a SingleCellExperiment object. If data frame or matrix, it should contain expression values for each gene as rows, and expression values for the cells as columns.
max_zeros	A number between 0 and 1 indicating the maximum proportion of zero expression values allowed per row. Features with a higher proportion of 0 will be discarded.
sce_assay	if data is an SingleCellExperiment object, sce_assay should be one of names(assays(<SingleCellE

Details

Before CV computation, the function removes all rows that have a proportion of zeros above the specified threshold. Genes with many 0s are poorly informative, and would bias the later correlations. Removing them also prevents division by zero when calculating CVs.

The data provided must cell/sample names as column names. Feature name can be given either in the first column or as row names.

In the output, mean, standard deviation and CV are incorporated as new columns in the data frame, named mean, sd and cv.

Value

A data frame, containing the filtered data with additional columns: mean, standard deviation and cv values for each row.

Examples

```
expMat <- matrix(
    c(1, 1, 1,
      1, 2, 3,
      0, 1, 2,
      0, 0, 2),
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)
calculate_cvs(expMat)
calculate_cvs(expMat, max_zeros = 0.5)
```

---

correlate_windows *Calculate correlations against top window.*

Description

Calculates pairwise correlations between all features each window against all features in the reference window.

Usage

```
correlate_windows(dataset, n_random = 3, ...)
```

Arguments

dataset	A data frame containing all the binned genes. Usually the output of bin_scdata.
n_random	Number of top window randomization to serve as a negative control. Default to 3.
...	Additional arguments to be passed to cor. Default method is pearson which is the fastest.

Details

This function:

- correlates each feature in each window to each feature in the top window.
- randomize the top window by shuffling expression value, and correlate each gene in each window to the randomized top window. This negative control is repeated as many time as specified by the n_random parameter.

The input of this function is usually the output of the bin_scdata function.

Value

A tibble containing correlation values.

Examples

```
library(magrittr)
expMat <- matrix(
    c(1, 1, 5,
      1, 2, 3,
      0, 1, 4,
      0, 0, 2),
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 2) %>%
    bin_scdata(window_number = 1) %>%
    correlate_windows
```

`correlations_to_densities` *Transform the correlation table to density distributions of correlation values*

Description

Takes the output of `correlate_windows` and computes density curves of correlation coefficient for each window comparison.

Usage

```
correlations_to_densities(df, n = 64, absolute_cc = TRUE)
```

Arguments

`df`	A data frame, usually the output of `correlate_windows`.
`n`	Resolution of the correlation density curve. Default to 64.
`absolute_cc`	Should the function use the absolute value of correlation coefficients? Default to `TRUE` to simplify plots and avoid annoying, non-symmetrical, near 0, shifts of distributions.

Value

A `tibble` with columns `bin`, `window`, `cor_coef` and `density`.

Examples

```
library(magrittr)
expMat <- matrix(
    c(1, 1, 5,
      1, 2, 3,
      0, 1, 4,
      0, 0, 2),
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 2) %>%
    bin_scdata(window_number = 1) %>%
    correlate_windows %>%
    correlations_to_densities
```

`define_top_genes` *Define the reference window using the most highly expressed features.*

Description

Define the group of features in the dataset that will be considered as reference, the top window, by specifying either a number of features or an expression threshold.

Usage

```
define_top_genes(
  dataset,
  window_size = NULL,
  mean_expression = NULL,
  min_expression = NULL
)
```

Arguments

`dataset`	A data frame, containing features as rows and cells as columns, and where the mean expression value for each gene has been added as a column. Usually the output of `calculate_cvs`.
`window_size`	Number of features in the defined top window. Recommended to 100 features.
`mean_expression`	A number. Genes with a mean expression across cells higher than the value will be selected. Ignored if `window_size` is defined.
`min_expression`	A number. Genes with a minimum expression across all cells higher than the value will be selected. Ignored if `window_size` or `mean_expression` is defined.

Details

There are three selection methods available:

- `window_size`: features are ranked by mean expression across cells, and the top slice of the specified size is selected.
- `mean_expression`: the `mean` column is checked, and all features with mean expression above the threshold indicated are selected.
- `min_expression`: features where all expression values are above the expression threshold indicated are selected.

In general, it is advisable to avoid generating top windows larger than 250 features (100 features is the recommended value), to prevent excessively long computation time as well as to preserve the quality of the analysis, as the top window should only include a subset of reliable values.

Value

A list with two elements, both data frames: the defined top window, and the rest of the genes.

Examples

```
library(magrittr)
expMat <- matrix(
    c(1, 1, 1,
      1, 2, 3,
      0, 1, 2,
      0, 0, 2),
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 2)

calculate_cvs(expMat) %>%
    define_top_genes(mean_expression = 1.5)
```

---

determine_bin_cutoff *Determine a threshold for selecting bins of features based on the metric table*

Description

Takes the output of get_mean_median and decide until which window to keep based on background level and a threshold.

Usage

```
determine_bin_cutoff(
  metric_table,
  threshold = 2,
  selected_metric = c("mean", "median", "score"),
  random_function_summarisation = mean
)
```

Arguments

metric_table	A data frame, usually the output of get_mean_median.
threshold	How many time higher than the background should the last bin be? Default to 2.
selected_metric	Which metric to use (i.e. which column from metric_table to work with). Default to mean.
random_function_summarisation	A function used to aggregate the randomised control across bin. Default to mean.

Details

Background level is estimated by averaging correlation coefficient obtained from the top window randomisations.

Bins (or windows) of features are kept until the mean (or median) correlation coefficient falls under a threshold value `threshold` x `background level`.

Value

A number, the first bin of features to discard.

See Also

`get_mean_median`, `plot_metric`

Examples

```
myData <- tibble::tibble(
    bin = rep(c(1, 2, 3), each = 3),
    window = rep(c("top_window", "shuffled_top_window_1", "shuffled_top_window_2"), 3),
    mean = c(0.8, 0.1, 0.11, 0.14, 0.12, 0.09, 0.10, 0.13, 0.08)
)
determine_bin_cutoff(myData)
```

---

filter_expression_table

Filter binned expression matrix

Description

Takes a binned expression table (the output of `bin_scdata`), a bin number (usually the output of `determine_bin_cutoff`) and returned a filtered expression table or matrix.

Usage

```
filter_expression_table(bined_table, bin_cutoff, as_matrix = FALSE)
```

Arguments

`bined_table`	A `tibble`, usually the output of `bin_scdata`.
`bin_cutoff`	the number of the first bin to be filtered out. Can be the output of `determine_bin_cutoff`).
`as_matrix`	A boolean. Should the return be a `tibble` (`FALSE`, the default) or a `matrix` (`TRUE`).

Value

A `tibble` or a `matrix` depending on the value of `as_matrix`

See Also

bin_scdata, determine_bin_cutoff

Examples

```
myData <- tibble::data_frame(
    bin = rep(c(1, 2, 3), each = 3),
    mean = 9:1,
    sd = runif(9),
    cv = runif(9),
    cell1 = 8:0 + runif(9),
    cell2 = 8:0 + runif(9)
)
filter_expression_table(myData, bin_cutoff = 2)
filter_expression_table(myData, bin_cutoff = 3)
```

---

get_mean_median *Extract mean and median correlation coefficient values*

Description

Takes the output of correlate_windows and extract the mean and the median correlation value for each window comparison.

Usage

```
get_mean_median(df, absolute_cc = TRUE)
```

Arguments

df	A data frame, usually the output of correlate_windows.
absolute_cc	Should the function work of absolute value of correlation coefficients? Default to TRUE to simplify plots and avoid annoying, non-symmetrical, near 0, shifts of distributions.

Value

A data_frame with columns bin, window, mean and median.

Examples

```
library(magrittr)
expMat <- matrix(
    c(1, 1, 5,
      1, 2, 3,
      0, 1, 4,
      0, 0, 2),
```

```
    ncol = 3, byrow = TRUE, dimnames = list(paste("gene", 1:4), paste("cell", 1:3))
)

calculate_cvs(expMat) %>%
    define_top_genes(window_size = 2) %>%
    bin_scdata(window_number = 1) %>%
    correlate_windows(n_random = 2) %>%
    get_mean_median
```

`plot_correlations_distributions` *Produce a density plot of correlation values for each window of feature*

Description

Feature by feature correlation values between every windows and the reference to window of features are visualized as density lines, one facet per comparison. Two density lines are drown in each facets:

- A thin colored line, the correlations between the bin and the reference top bin of features
- A thicker blue line with grey error area, the correlations between the bin and the **randomized** top bin of features. The lines are not shown if `n_random = 0` in `correlate_windows`.

Usage

```
plot_correlations_distributions(
  df,
  metrics = NULL,
  vlines = c("mean", "median"),
  facet_ncol = 4
)
```

Arguments

`df`	A `tibble`, usually the output of `correlations_to_densities`.
`metrics`	Optional. The output of `get_mean_median`. Dashed line will represent mean or median of the correlation coefficient distributions.
`vlines`	A string, either "mean" or "median". Should the dashed line represent the mean or the median of the correlation coefficient distributions? Ignored if `metrics` is `NULL`.
`facet_ncol`	The number of columns to arrange the plots.

Value

A ggplot2 plot.

See Also

correlations_to_densities, get_mean_median

Examples

```
library(magrittr)
myData <- scData_hESC %>%
calculate_cvs %>%
    define_top_genes(window_size = 100) %>%
    bin_scdata(window_size = 1000)

corDistrib <- correlate_windows(myData, n_random = 3)

corDens <- correlations_to_densities(corDistrib)

plot_correlations_distributions(corDens)

metrics <- get_mean_median(corDistrib)

plot_correlations_distributions(corDens, metrics = metrics)
```

---

plot_mean_variance *Produce a mean expression x coefficient of variation scatter plot.*

Description

Use the output of calculate_cvs or bin_scdata and plot a feature mean expression x coefficient of variation scatter plot. Mean expression is represented as `log10(mean + 1)`. Each dot represents a feature. Means and coefficient of variations were obtained across single cells. Optionally, colours each dot according to the defined bins of features. Optionally, adds a density2d geom.

Usage

```
plot_mean_variance(
  df,
  density = TRUE,
  colourByBin = TRUE,
  density_color = "blue",
  ...
)
```

Arguments

df	A `tibble`, usually the output of calculate_cvs or bin_scdata.
density	A boolean. Should a `density2d` geom be added to the plot?
colourByBin	A boolean. Should feature be coloured by bin? Need a `bin` column in `df` (i.e. the output of bin_scdata).

density_color	Colour of the density2d curves.
...	Further arguments are passed to geom_point such as size.

Value

A ggplot2 plot.

See Also

calculate_cvs, bin_scdata

Examples

```
library(magrittr)
scData_hESC %>%
   calculate_cvs %>%
   plot_mean_variance(colourByBin = FALSE)

scData_hESC %>%
   calculate_cvs %>%
   define_top_genes(window_size = 100) %>%
   bin_scdata(window_size = 1000) %>%
   plot_mean_variance
```

---

plot_metric *Produce a bar chart of mean (or median) correlation coefficient per bin of feature.*

Description

Use the output of get_mean_median and produce a bar chart of mean (or median) correlation coefficient per bin of features. Correlations against the randomised top window are shown as dot-and-whiskers, and are used to estimate a background level.

Usage

```
plot_metric(
  metric_table,
  selected_metric = c("mean", "median", "score"),
  show_ctrl = TRUE,
  control_color = "blue",
  show_threshold = TRUE,
  threshold = 2,
  threshold_color = "red",
  line_size = 1,
  annotate_lines = TRUE
)
```

Arguments

metric_table	A tibble, usually the output of get_mean_median.
selected_metric	Which column in metricsTable to use? Default to mean.
show_ctrl	A boolean. Should a dashed line indicate the estimated background level?
control_color	The colour of the background dashed line (default to blue).
show_threshold	A boolean. Should a dashed line indicate the estimated threshold level?
threshold	How many times the background level should be multiplies do determine a threshold? Default to 2. The higher the more stringent.
threshold_color	The colour of the threshold dashed line (default to blue).
line_size	Thickness of the dashed lines.
annotate_lines	A boolean. Should the dashed lines be annotated?

Value

A ggplot2 plot.

See Also

get_mean_median

Examples

```
scData_hESC |>
    calculate_cvs() |>
    define_top_genes(window_size = 100) |>
    bin_scdata(window_size = 1000) |>
    correlate_windows(n_random = 3) |>
    get_mean_median() |>
    plot_metric()
```

---

plot_top_window_autocor

Utility plot to choose a top_window size

Description

Plot mean autocorrelation value of the features of the top window depending on increasing top window size.

Usage

```
plot_top_window_autocor(sc_data, from = 10, to = 400, by = 2, ...)
```

Arguments

sc_data	A tibble, usually the output of link{calculate_cvs}.
from	Minimum size of the top window.
to	Maximum size of the top window.
by	Size of the steps to walk form from to to. See seq.
...	Arguments to be passed to cor, for example method = "spearman"

Value

A ggplot2 plot.

Examples

```
plot_top_window_autocor(calculate_cvs(scData_hESC))
```

---

sc_feature_filter *Filter scRNA-seq expression matrix to keep only highly informative features. Integrated pipeline.*

Description

This pipeline function takes an expression matrix as an input and select the features (genes, transcripts) with an estimated technical noise level lower that biological variation in the data. This is achieved by binning the data and calculating the correlation for each bin with highly expressed (lowest noise) gene set (see the vignette for details on the method).

Usage

```
sc_feature_filter(
  sc_data,
  print_plots = FALSE,
  max_zeros = 0.75,
  threshold = 2,
  top_window_size = 100,
  other_window_size = 1000,
  n_random = 3,
  sce_assay = NULL
)
```

Arguments

sc_data	A data frame, a matrix or a SingleCellExperiment object. If data frame or matrix, it should contain expression values for each gene as rows, and expression values for the cells as columns.

print_plots — A boolean. Should the function produce three plots as a side effect? Plots are the output of plot_mean_variance, plot_correlations_distributions and plot_metric.

max_zeros — A number between 0 and 1. Maximum proportion of cells with 0 expression for a feature to be kept.

threshold — A number higher than 1. The higher the more stringent the feature selection will be. See determine_bin_cutoff.

top_window_size — Size of the reference bin. See define_top_genes

other_window_size — Size of the other bins of feature. See bin_scdata

n_random — Number of control windows generated by shuffling the top bin of features.

sce_assay — if sc_data is an SingleCellExperiment object, sce_assay should be one of names(assays(<SingleCellExperiment>)).

Details

The function can optionally produce three plots of print_plots is TRUE. It is recommended to open a graphical device (i.e. through pdf or png), to call scFeatureFilter,and then to close the device with dev.off.

Value

A matrix or a tibble, depending on the type of sc_data, containing only the top expressed features.

Examples

```
sc_feature_filter(scData_hESC)

# with plots

pdf("diagnostic.pdf")
sc_feature_filter(sc_data, print_plots = TRUE)
dev.off()
```

---

scData_hESC *Expression data from 32 human embryonic stem cells*

Description

Expression of 60,468 Gencode gene (in FPKM) from 32 single cell RNAseq of human embryonic stem cells

Usage

```
scData_hESC
```

Format

A `tibble` with 60,468 rows (genes) and 33 columns (cells):

tracking_id The Gencode human encode gene id

next 33 columns Single embryonic stem cells

Value

A `tibble`.

Source

The example dataset was processed by Mantsoki et al. (2016) https://www.ncbi.nlm.nih.gov/pubmed/26951854, data from human Embryonic Stem cells was generated by Yan et al. (2013)https://www.ncbi.nlm.nih.gov/pubmed/23934149.

Index

∗ **datasets**
scData_hESC, 16

bin_scdata, 2, *5*, *9, 10*, *12, 13*, *16*

calculate_cvs, 3, *12, 13*
cor, *5*
correlate_windows, 4, *6*, *10, 11*
correlations_to_densities, 6, *11, 12*

define_top_genes, 7, *16*
determine_bin_cutoff, 8, *9, 10*, *16*

filter_expression_table, 9

get_mean_median, *8, 9*, 10, *11–14*

plot_correlations_distributions, 11, *16*
plot_mean_variance, 12, *16*
plot_metric, *9*, 13, *16*
plot_top_window_autocor, 14

sc_feature_filter, 15
scData_hESC, 16

tibble, *6*