Package: barcodetrackR (via r-universe)

October 31, 2026

Type Package

Title Functions for Analyzing Cellular Barcoding Data

Version 1.20.0

Date 2021-02-18

Description barcodetrackR is an R package developed for the analysis and visualization of clonal tracking data. Data required is samples and tag abundances in matrix form. Usually from cellular barcoding experiments, integration site retrieval analyses, or similar technologies.

URL https://github.com/dunbarlabNIH/barcodetrackR

License file LICENSE

LazyData FALSE

Encoding UTF-8

biocViews Software, Visualization, Sequencing

Imports cowplot, circlize, dplyr, ggplot2, ggdendro, ggridges, graphics, grDevices, magrittr, plyr, proxy, RColorBrewer, rlang, scales, shiny, stats, SummarizedExperiment, S4Vectors, tibble, tidyr, vegan, viridis, utils

RoxygenNote 7.1.1

Suggests BiocStyle, knitr, magick, rmarkdown, testthat

VignetteBuilder knitr

Depends R (>= 4.1)

PackageStatus Deprecated

Config/pak/sysreqs cmake make libicu-dev libuv1-dev zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:55:32 UTC

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

RemoteRef RELEASE_3_23

RemoteSha 0735f99f0deb7c6d3c9bbfb18dd9368c9020b19a

Contents

barcode_binary_heatmap . . . 2
barcode_ggheatmap . . . 3
barcode_ggheatmap_stat . . . 5
barcode_stat_test . . . 7
bias_histogram . . . 9
bias_lineplot . . . 11
bias_ridge_plot . . . 12
build_index_html . . . 14
chord_diagram . . . 14
clonal_contribution . . . 15
clonal_count . . . 17
clonal_diversity . . . 19
cor_plot . . . 20
create_SE . . . 21
dist_plot . . . 22
estimate_barcode_threshold . . . 24
get_top_clones . . . 25
launchApp . . . 26
mds_plot . . . 27
rank_abundance_plot . . . 28
rank_abundance_stat_test . . . 29
scatter_plot . . . 30
stat_hist . . . 31
subset_SE . . . 32
threshold . . . 33
threshold_SE . . . 33
wu_subset . . . 34

Index **36**

`barcode_binary_heatmap` *Barcode Binary Heatmap*

Description

Creates a binary heatmap showing the absence or presence of new clones in samples ordered from L to R in the SummarizedExperiment.

Usage

```
barcode_binary_heatmap(
  your_SE,
  plot_labels = NULL,
  threshold = 0,
  your_title = NULL,
```

```
  label_size = 12,
  return_table = FALSE
)
```

Arguments

your_SE	A Summarized Experiment object.
plot_labels	Vector of x axis labels. Defaults to colnames(your_SE).
threshold	Clones with a proportion below this threshold will be set to 0.
your_title	The title for the plot.
label_size	The size of the column labels.
return_table	Logical. Whether or not to return table of barcode sequences with their presence or absence in each sample indicated as a 1 or 0 resepctively in the value column column.

Value

Displays a binary heat map in the current plot window. Or if return_table is set to TRUE, returns a dataframe indicating the presence or absence of each barcode in each sample.

Examples

```
data(wu_subset)
barcode_binary_heatmap(your_SE = wu_subset[, 1:4])
```

---

barcode_ggheatmap *barcode_ggheatmap*

Description

Creates a heatmap displaying the log abundance of the top 'n' clones from each sample in the Summarized Experiment object, using ggplot2. Clones are on the y-axis and samples are on the x-axis. The ordering and clustering of clones on the y-axis as well as all aesthetics of the plot can be controlled through the arguments described below.

Usage

```
barcode_ggheatmap(
  your_SE,
  plot_labels = NULL,
  n_clones = 10,
  cellnote_assay = "stars",
  your_title = NULL,
  grid = TRUE,
  label_size = 12,
  dendro = FALSE,
```

```
  cellnote_size = 4,
  distance_method = "Euclidean",
  minkowski_power = 2,
  hclust_linkage = "complete",
  row_order = "hierarchical",
  clusters = 0,
  percent_scale = c(0, 2.5e-05, 0.001, 0.01, 0.1, 1),
 color_scale = c("#4575B4", "#4575B4", "lightblue", "#fefeb9", "#D73027", "red4"),
  return_table = FALSE
)
```

Arguments

your_SE	A Summarized Experiment object.
plot_labels	Vector of x axis labels. Defaults to colnames(your_SE).
n_clones	The top 'n' clones to plot.
cellnote_assay	Character. One of "stars", "counts", or "proportions." To have no cellnote, set cellnote_size to 0.
your_title	The title for the plot.
grid	Logical. Include a grid or not in the heatmap.
label_size	The size of the column labels.
dendro	Logical. Whether or not to show row dendrogram when hierarchical clustering.
cellnote_size	The numerical size of the cell note labels. To have no cellnote, set cellnote_size to 0.
distance_method	Character. Use summary(proxy::pr_DB) to see all possible options for distance metrics in clustering.
minkowski_power	The power of the Minkowski distance (if minkowski is the distance method used).
hclust_linkage	Character. One of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
row_order	Character; "hierarchical" to perform hierarchical clustering on the output and order in that manner, "emergence" to organize rows by order of presence in data (from left to right), or a character vector of rows within the summarized experiment to plot.
clusters	How many clusters to cut hierarchical tree into for display when row_order is "hierarchical".
percent_scale	A numeric vector through which to spread the color scale (values inclusive from 0 to 1). Must be same length as color_scale.
color_scale	A character vector which indicates the colors of the color scale. Must be same length as percent_scale.
return_table	Logical. Whether or not to return table of barcode sequences with their log abundance in the 'value' column and cellnote for each sample instead of displaying a plot.

Value

Displays a heatmap in the current plot window. Or if return_table is set to TRUE, returns a dataframe of the barcode sequences, log abundances, and cellnotes for each sample.

Examples

```
data(wu_subset)
barcode_ggheatmap(
    your_SE = wu_subset, n_clones = 10,
    grid = TRUE, label_size = 6
)
```

---

barcode_ggheatmap_stat

Barcode Top Clone Heatmap

Description

Creates a heatmap from the columns of data in the Summarized Experiment object, with the option to label based on statistical analysis. Uses ggplot2.

Usage

```
barcode_ggheatmap_stat(
  your_SE,
  sample_size,
  stat_test = "chi-squared",
  stat_option = "subsequent",
  reference_sample = NULL,
  stat_display = "top",
  show_all_significant = FALSE,
  p_threshold = 0.05,
  p_adjust = "none",
  bc_threshold = 0,
  plot_labels = NULL,
  n_clones = 10,
  cellnote_assay = "stars",
  your_title = NULL,
  grid = TRUE,
  label_size = 12,
  dendro = FALSE,
  cellnote_size = 4,
  distance_method = "Euclidean",
  minkowski_power = 2,
  hclust_linkage = "complete",
  row_order = "hierarchical",
  clusters = 0,
```

```
  percent_scale = c(0, 2.5e-05, 0.001, 0.01, 0.1, 1),
 color_scale = c("#4575B4", "#4575B4", "lightblue", "#fefeb9", "#D73027", "red4"),
  return_table = FALSE
)
```

Arguments

your_SE	A Summarized Experiment object.
sample_size	A numeric vector providing the sample size of each column of the SummarizedExperiment passed to the function. This sample size describes the samples that the barcoding data is meant to approximate.
stat_test	The statistical test to use on the constructed contingency table for each barcoe. Options are "chi-squared" and "fisher."
stat_option	For "subsequent" statistical testing is performed on each column of data compared to the column before it. For "reference," all other columns of data are compared to a reference column.
reference_sample	Provide the column name of the reference column if stat_option is set to "reference." Defaults to the first column in the SummarizedExperiment.
stat_display	Choose which clones to display on the heatmap. IF set to "top," the top n_clones ranked by abundance for each sample will be displayed. If set to "change," the top n_clones with the lowest p-value from statistical testing will be shown for each sample. If set to "increase," the top n_clones (ranked by p-value) which increase in abundance for each sample will be shown. And if set to "decrease," the top n_clones (ranked by lowest p-value) which decrease in abdundance will be shown.
show_all_significant	Logical. If set to TRUE when stat_display = "change," "increase," or "decrease" then the n_clones argument will be overriden and all clones with a statistically singificant change, increase, or decrease in proportion will be shown.
p_threshold	The p_value threshold to use for statistical testing
p_adjust	Character, default = "none". To correct p-values for muiltiple comparisons, set to any of the p value adjustment methods in the p.adjust function in R stats, which includes "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", and "fdr".
bc_threshold	Clones must be above this proportion in at least one sample to be included in statistical testing.
plot_labels	Vector of x axis labels. Defaults to colnames(your_SE).
n_clones	The top 'n' clones to plot.
cellnote_assay	Character. One of "stars", "reads", "proportions" or "p_val"
your_title	The title for the plot.
grid	Logical. Include a grid or not in the heatmap.
label_size	The size of the column labels.
dendro	Logical. Whether or not to show row dendrogram when hierarchical clustering.

`cellnote_size` The numerical size of the cell note labels.

`distance_method`
Character. Use summary(proxy::pr_DB) to see all possible options for distance metrics in clustering.

`minkowski_power`
The power of the Minkowski distance (if minkowski is the distance method used).

`hclust_linkage` Character. One of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

`row_order` Character; "hierarchical" to perform hierarchical clustering on the output and order in that manner, "emergence" to organize rows by order of presence in data (from left to right), or a character vector of rows within the summarized experiment to plot.

`clusters` How many clusters to cut hierarchical tree into for display when row_order is "hierarchical".

`percent_scale` A numeric vector through which to spread the color scale (values inclusive from 0 to 1). Must be same length as color_scale.

`color_scale` A character vector which indicates the colors of the color scale. Must be same length as percent_scale.

`return_table` Logical. Whether or not to return table of barcode sequences with their log abundance in the 'value' column and cellnote (* indicating statistical signficant change, for example) for each sample instead of displaying a plot. Note, for more in-depth statistical analysis, use the ‘"barcode_stat_test‘ function.

Value

Displays a heatmap in the current plot window. Or if return_table is set to TRUE, returns a dataframe of the barcode sequences, log abundances, and cellnote for each sample.

Examples

```
data(wu_subset)
barcode_ggheatmap_stat(
    your_SE = wu_subset[, 1:4], sample_size = rep(5000, 4),
    stat_test = "chi-squared", stat_option = "subsequent",
    p_threshold = 0.05, n_clones = 10,
    cellnote_assay = "stars", bc_threshold = 0.005
)
```

`barcode_stat_test` *Barcode Statistical Test*

Description

Carries out a specific instance of statistical testing relevant to clonal tracking experiments. For longitudinal observations (of barcode abundances) in the provided SE object, use a Chi-squared or Fisher exact test whether each barcode proportion has changed between samples.
Each column in the provided SE will be "tested" against the reference sample. If the ‘stat_option‘ argument is set to its default of "subsequent" then each sample will be compared to the sample before it. If this argument is set to "reference" the reference sample column name must be provided and each column will be tested against that reference sample.

Usage

```
barcode_stat_test(
  your_SE,
  sample_size,
  stat_test = "chi-squared",
  stat_option = "subsequent",
  reference_sample = NULL,
  p_adjust = "none",
  bc_threshold = 0
)
```

Arguments

your_SE	A Summarized Experiment object containing clonal tracking data as created by the barcodetrackR ‘create_SE‘ function.
sample_size	A numeric vector providing the sample size of each column of the SummarizedExperiment passed to the function. This sample size describes the samples that the barcoding data is meant to approximate, for example the number of cells barcodes were extracted from.
stat_test	The statistical test to use on the constructed contingency table for each barcode. Options are "chi-squared" and "fisher." For information, see [chisq.test](https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/chi [fisher.test](https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/fisher.test)
stat_option	For "subsequent" statistical testing is performed on each column of data compared to the column before it. For "reference," all other columns of data are compared to a reference column specified in the ‘reference_sample‘ arguument.
reference_sample	Provide the column name of the reference column if stat_option is set to "reference." Defaults to the first column in the SummarizedExperiment.
p_adjust	Character, default = "none". To correct p-values for muiltiple comparisons, set to any of the p value adjustment methods in the p.adjust function in R stats, which includes "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", and "fdr".
bc_threshold	Clones must be above this proportion in at least one sample to be included in statistical testing. Default is 0. Use this to ignore low-abundance clones which are more likely to be noise or artifact.

Value

Returns a list of 3 dataframes containing the following information for each observation (or barcode) which passed the provided bc_threshold:
[["FC"]], Fold Change of barcode abundance for each sample relative to the previous sample or to the specified reference sample. Please note that for maximal user control over results, the FC dataframe will contain 0 for barcodes where the test sample has an abundance of 0, Inf for barcodes where the reference sample had an abundance of 0 and NaN for a barcode where both the test and reference sample have an abundance of 0;
[["log_FC"]], same as previous but the log Fold Change. Please note that again for maximal user control, the log_FC dataframe will contain NaN values when the FC was Nan, -Inf values when the FC was 0, and Inf values when the FC was Inf;
[["p_val"]], the p-value returned from either the Chi-squared or Fisher exact test indicating whether each barcode changed in proportion between the test sample and the reference sample. Please note that the p value will be NaN if both abundances are 0, otherwise a p-value will be assigned.
Also, note that one column of each resulting dataframe will contain all NAs - in the case where the ‘stat_option‘ argument is set to "subsequent" then this will be the first sample since there is no subsequent sample to compare to. In the case where the ‘stat_option‘ argument is set to "reference" then the reference sample will contain NAs.

Examples

```
data(wu_subset)
barcode_stat_test(
    your_SE = wu_subset[, 1:4], sample_size = rep(5000, 4),
    stat_test = "chi-squared", stat_option = "subsequent",
    bc_threshold = 0.0001
)
```

`bias_histogram` *Bias histogram*

Description

Given a summarized experiment, gives histogram of log biases for 2 cell types. Each stacked bar in the histogram represents a clone binned by log bias defined as the log2 of the percentage abundance in the sample specified in "bias_1" divided by the percentage abundance in "bias_2."

Usage

```
bias_histogram(
  your_SE,
  split_bias_on,
  bias_1,
  bias_2,
  split_bias_over,
  bias_over = NULL,
  remove_unique = FALSE,
```

```
  breaks = c(10, 2, 1, 0.5),
  text_size = 10,
  linesize = 0.4,
  ncols = 1,
  scale_all_y = TRUE,
  return_table = FALSE
)
```

Arguments

your_SE	Your SummarizedExperiment of barcode data and associated metadata
split_bias_on	The column in ‘colData(your_SE)‘ from which ‘bias_1‘ and ‘bias_2‘ will be chosen
bias_1	The first cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the RIGHT side of the histogram.
bias_2	The second cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the LEFT side of the ridge plot
split_bias_over	The column in ‘colData(your_SE)‘ that you wish to observe the bias split for. The output will contain a faceted plot: one facet for each value of ‘split_bias_over‘ comparing the samples matching ‘bias_1‘ and ‘bias_2‘ from the ‘split_bias_on‘ argument.
bias_over	Choice(s) from the column designated in ‘split_bias_over‘ that will be used for plotting. Defaults to all.
remove_unique	If set to true, only clones present in both samples will be considered.
breaks	Numeric. The breaks specified for bins on the x-axis (how biased the clones are towards one factor or the other).
text_size	The size of the text in the plot.
linesize	The linewidth of the stacked bars which represent individual barcodes
ncols	Numeric. Number of columns to plot on using plot_grid from cowplot.
scale_all_y	Logical. Whether or not to plot all plots on the same y axis limits.
return_table	Logical. If set to TRUE, instead of a plot, tbe function will return a list containing a dataframe for each sample-sample log bias combination containing each barcode sequence and its bias between the samples.

Value

Histogram of log bias for two factors faceted over another set of factors. Or, if return_table is set to TRUE, a list of dataframes containing the log bias data for each bias comparison passed to the function.

Examples

```
data(wu_subset)
bias_histogram(
    your_SE = wu_subset, split_bias_on = "celltype",
    bias_1 = "B", bias_2 = "T",
    split_bias_over = "months", ncols = 2
)
```

`bias_lineplot` *Bias line plot*

Description

Given a summarized experiment and a specified factor to compare bias between "split_bias_on", shows the value of that bias plotted against another specified factor "split_bias_over" where each clone is represented by a line shaded by its overall abundance in the two samples being compared.

Usage

```
bias_lineplot(
  your_SE,
  split_bias_on,
  bias_1,
  bias_2,
  split_bias_over,
  bias_over = NULL,
  remove_unique = FALSE,
  text_size = 16,
  keep_numeric = TRUE,
  return_table = FALSE
)
```

Arguments

`your_SE`	SummarizedExperiment of barcode data and associated metadata
`split_bias_on`	The column of metadata corresponding to cell types (or other factor to be compared.)
`bias_1`	The first cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the UPPER side of the line plot
`bias_2`	The second cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the LOWER side of the line plot
`split_bias_over`	The column of metadata to plot by. If numeric, y axis will be in increasing order. If categorical, it will follow order of metadata.

bias_over	Choice(s) from the column designated in 'split_bias_over' that will be used for plotting. Defaults to all.
remove_unique	Logical. If set to true, only clones present in both samples will be considered.
text_size	Numeric. The size of the text in the plot.
keep_numeric	Logical. Whether to keep the numeric spacing within split_bias_over or switch to discrete x scale.
return_table	Logical. If set to TRUE, rather than returnign a plot, the function will return a dataframe containing for each barcode sequence and each point of comparison: the bias value, the added proportion between the two factors at that point (cumul_sum), and the maximum cumul_sum (peak_abundance) of that barcode sequence at any point of comparison.

Value

Bias line plot for two lineages over time. Or if return_table is set to TRUE, a dataframe containing the bias values for each barcode sequence between the two samples at all points of comparison.

Examples

```
data(wu_subset)
bias_lineplot(
    your_SE = wu_subset, split_bias_on = "celltype",
    bias_1 = "B", bias_2 = "T", split_bias_over = "months"
)
```

---

bias_ridge_plot *Bias Ridge plot*

Description

Given a summarized experiment and a specified factor to compare bias between, gives ridge plots which show the density of clones at each value of log bias where log bias is calculated as log((normalized abundance in sample 1 + 1)/(normalized abundance in sample 2 + 1)). If the weighted option is set to TRUE, the density estimator will weight the estimation by the added proportion of the clone between the two samples.

Usage

```
bias_ridge_plot(
  your_SE,
  split_bias_on,
  bias_1,
  bias_2,
  split_bias_over,
  bias_over = NULL,
  remove_unique = FALSE,
  weighted = FALSE,
```

```
  text_size = 16,
  add_dots = FALSE,
  return_table = FALSE
)
```

Arguments

your_SE	Your SummarizedExperiment of barcode data and associated metadata
split_bias_on	The column of metadata corresponding to cell types (or whatever factors you want to compare the bias between).
bias_1	The first cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the RIGHT side of the ridge plot
bias_2	The second cell type (or other factor) to be compared. Must be a possible value of the split_bias_on column of your metadata. Will be on the LEFT side of the ridge plot
split_bias_over	The column of metadata to plot by. If numeric, y axis will be in increasing order. If categorical, it will follow order of metadata.
bias_over	Choice(s) from the column designated in 'split_bias_over' that will be used for plotting. Defaults to all.
remove_unique	If set to true, only clones present in both samples will be considered.
weighted	If true, the density estimation will be weighted by the overall contribution of each barcode to the two samples being compared.
text_size	Numeric. The size of the text in the plot.
add_dots	Logical. Whether or not to add dots underneath the density plots. Dot size is proportion to the added proportion of the clone in the two samples.
return_table	Logical. If true, rather than returning a plot, the function will return a dataframe containing the calculated bias and cumul_sum which contains the added proportion between the two samples, for each barcode sequence across each sample considered.

Value

Bias plot for two lineages over time. Or a dataframe containing the bias value and added proportion of each barcode if return_table is set to TRUE.

Examples

```
data(wu_subset)
bias_ridge_plot(
    your_SE = wu_subset, split_bias_on = "celltype",
    bias_1 = "B", bias_2 = "T", split_bias_over = "months",
    add_dots = TRUE
)
```

build_index_html *Build html*

Description

Build html for vignette to index.html in docs

Usage

```
build_index_html(
  target = "vignettes/Introduction_to_barcodetrackR.Rmd",
  output = "index.html"
)
```

Arguments

target	the vignette to build
output	the target for the vignette output

Value

Writes the vignette to docs/index.html. Only for internal use (get outta here!).

chord_diagram *Barcode Chord Diagram*

Description

Creates a chord diagram showing each cell type (or other factor) as a region around a circle and shared clones between these cell types as links between the regions. The space around the regions which is not connected to a chord indicates clones unique to that sample, not shared with other samples.

Usage

```
chord_diagram(
  your_SE,
  weighted = FALSE,
  plot_label = "SAMPLENAME",
  alpha = 1,
  your_title = NULL,
  text_size = 12,
  return_table = FALSE
)
```

Arguments

your_SE	Summarized Experiment object containing clonal tracking data as created by the barcodetrackR 'create_SE' function.
weighted	Logical. weighted = F which is default will make links based on the number of shared clones between the factors. Weighted = TRUE will make the link width based on the clone's proportion in the samples.
plot_label	Character. Name of colData variable to use as labels for regions. Defaults to SAMPLENAME
alpha	Numeric. Transparency of links. Default = 1 is opaque. 0 is completely transluscent
your_title	Character. The title for the plot.
text_size	Numeric. Size of region labels
return_table	Logical. If set to TRUE, in addition to plotting a chord diagram in the plot window, the function will return a dataframe of the shared clonality used to make the chord diagram. If Weighted is FALSE, the dataframe will contain a row for each set of clones and values of 1 or 0 indicating the samples which share that set of clones, and a freq column which is the number of clones in that set. If weighted is set to TRUE, each row will contain a set of clones and the data will show the proportion that set of clones comprises in each sample. The proportions of 0 indicate which samples do not share that set of clones.

Value

Displays a chord diagram in the current plot window depicting shared clonality between samples (regions) as chords or links between the regions. Or,

Examples

```
data(wu_subset)
chord_diagram(your_SE = wu_subset[, c(4, 8, 12)], plot_label = "celltype")
```

---

clonal_contribution *Clonal contribution plot*

Description

Bar or line plot of percentage contribution of the top clones from a selected sample or all clones across samples matching the specified filter within the SummarizedExperiment object. Usually used for tracking a cell lineage's top clones over time.

Usage

```
clonal_contribution(
  your_SE,
  SAMPLENAME_choice = NULL,
  filter_by,
  filter_selection,
  plot_over,
  plot_over_display_choices = NULL,
  clone_sequences = NULL,
  n_clones = 10,
  graph_type = "bar",
  keep_numeric = TRUE,
  plot_non_selected = TRUE,
  linesize = 0.2,
  text_size = 15,
  your_title = "",
  y_limit = NULL,
  return_table = FALSE
)
```

Arguments

your_SE	A Summarized Experiment object.
SAMPLENAME_choice	The identifying SAMPLENAME from which to obtain the top "n_clones" clones to color. If NULL and clone_sequences is NULL, all clones will be shown as gray.
filter_by	Name of metadata column to filter by e.g. Lineage
filter_selection	The value of the filter column to display e.g. "T" (within Lineage)
plot_over	The column of metadata that you want to be the x-axis of the plot. e.g. Month. For numeric metadata, the x-axis will be ordered in ascending fashion. For categorical metadata, the sample order will be followed.
plot_over_display_choices	Choice(s) from the column designated in plot_over that will be used for plotting. Defaults to all.
clone_sequences	The identifying rownames within your_SE for which to plot. SAMPLENAME_choice should be set to NULL or not specified if clone_sequences is specified.
n_clones	Numeric. Number of top clones from SAMPLENAME_choice that should be assigned a unique color.
graph_type	Choice of "bar" or "line" for how to display the clonal contribution data
keep_numeric	If plot_over is numeric, whether to space the x-axis appropriately according to the numerical values.
plot_non_selected	Plot clones NOT found within the top clones in SAMPLENAME_choice or the specified clones passed to clone_sequences. These clones are colored gray. If

	both SAMPLENAME_choice and clone_sequences are NULL, this argument must be set to TRUE. Otherwise, there will be no data to show.
linesize	Numeric. Thickness of the lines.
text_size	Numeric. Size of text in plot.
your_title	Title string for your plot.
y_limit	Numeric. What the max value of the y scale should be for the "proportions" assay.
return_table	Logical. If set to TRUE, the function will return a dataframe with each sequence that is selected and its percentage contribution to each selected sample rather than a plot.

Value

Displays a stacked area line or bar plot (made by ggplot2) of the samples' top clones. Or, if return_table is set to TRUE, returns a dataframe of the percentage abundances in each sample.

Examples

```
data(wu_subset)
clonal_contribution(
    your_SE = wu_subset, graph_type = "bar",
    SAMPLENAME_choice = "ZJ31_20m_T",
    filter_by = "celltype", filter_selection = "T",
    plot_over = "months", n_clones = 10
)
```

---

clonal_count *Clonal count plot*

Description

A line plot that tracks the total number of clones or the cumulative number of clones from selected samples of the SummarizedExperiment object plotted over a specified variable.

Usage

```
clonal_count(
  your_SE,
  percent_threshold = 0,
  plot_over,
  plot_over_display_choices = NULL,
  keep_numeric = TRUE,
  group_by,
  group_by_choices = NULL,
  cumulative = FALSE,
  point_size = 3,
```

```
  line_size = 2,
  text_size = 12,
  your_title = NULL,
  return_table = FALSE
)
```

Arguments

your_SE	Summarized Experiment object containing clonal tracking data as created by the barcodetrackR 'create_SE' function.
percent_threshold	Numeric. The percent threshold for which to count barcodes as present or not present. Set to 0 by default.
plot_over	The column of metadata that you want to be the x-axis of the plot. e.g. timepoint
plot_over_display_choices	Choice(s) from the column designated in plot_over that will be used for plotting. Defaults to all if left as NULL.
keep_numeric	If plot_over is numeric, whether to space the x-axis appropriately according to the numerical values.
group_by	The column of metadata you want to group by e.g. cell_type.
group_by_choices	Choice(s) from the column designated in group_by that will be used for plotting. Defaults to all if left as NULL.
cumulative	Logical. If TRUE, will plot cumulative counts over the 'plot_over' argument rather than unique counts per sample (the default, which is FALSE).
point_size	Numeric. Size of points.
line_size	Numeric. Size of lines.
text_size	Numeric. Size of text in plot.
your_title	The title for the plot.
return_table	Logical. If set to true, rather than returning a plot, the function will return the clonal count or cumulative count of each sample in a dataframe.

Value

Outputs plot of a diversity measure tracked for groups over a factor. Or if return_table is set to TRUE, a dataframe of the number of clones (or cumulative clones) for each sample.

Examples

```
data(wu_subset)
clonal_count(your_SE = wu_subset, cumulative = FALSE, plot_over = "months", group_by = "celltype")
```

clonal_diversity *Clonal diversity plot*

Description

A line plot that tracks a diversity measure from selected samples of the SummarizedExperiment object plotted over a specified variable.

Usage

```
clonal_diversity(
  your_SE,
  plot_over,
  plot_over_display_choices = NULL,
  keep_numeric = TRUE,
  group_by,
  group_by_choices = NULL,
  index_type = "shannon",
  point_size = 3,
  line_size = 2,
  text_size = 12,
  your_title = NULL,
  return_table = FALSE
)
```

Arguments

your_SE	Summarized Experiment object containing clonal tracking data as created by the barcodetrackR 'create_SE' function.
plot_over	The column of metadata that you want to be the x-axis of the plot. e.g. timepoint
plot_over_display_choices	Choice(s) from the column designated in plot_over that will be used for plotting. Defaults to all if left as NULL.
keep_numeric	If plot_over is numeric, whether to space the x-axis appropriately according to the numerical values.
group_by	The column of metadata you want to group by e.g. cell_type
group_by_choices	Choice(s) from the column designated in group_by that will be used for plotting. Defaults to all if left as NULL.
index_type	Character. One of "shannon", "shannon_count", "simpson", or "invsimpson".
point_size	Numeric. Size of points.
line_size	Numeric. Size of lines.
text_size	Numeric. Size of text in plot.
your_title	Character. The title for the plot.

return_table Logical. IF set to TRUE, rather than returning the plot of clonal diversity, the function will return a dataframe containing the diversity index values for each specified sample.

Value

Outputs plot of a diversity measure tracked for groups over a factor. Or if return_table is set to true, a dataframe will be returned instead.

Examples

```
data(wu_subset)
clonal_diversity(
    your_SE = wu_subset, index_type = "shannon",
    plot_over = "months", group_by = "celltype"
)
```

---

cor_plot *Correlation Plot*

Description

Plots the pairwise correlation between the specified assay of each sample-sample pair in the provided SummarizedExperiment.

Usage

```
cor_plot(
  your_SE,
  assay = "proportions",
  plot_labels = colnames(your_SE),
  method_corr = "pearson",
  your_title = "",
  grid = TRUE,
  label_size = 8,
  plot_type = "color",
  no_negatives = FALSE,
  return_table = FALSE,
  color_scale = "default",
  number_size = 3,
  point_scale = 1
)
```

Arguments

your_SE A Summarized Experiment object.

assay The choice of assay to use for the correlation calculation. Set to "proportions" by default.

plot_labels	Vector of x axis labels. Defaults to colnames(your_SE).
method_corr	Character. One of "pearson", "spearman", or "kendall".
your_title	Character. The title for the plot.
grid	Logical. Include a grid or not in the correlation plot
label_size	Numeric. The size of the column labels.
plot_type	Character. One of "color", "circle", or "number".
no_negatives	Logical. Whether to make negative correlations = 0.
return_table	Logical. Whether or not to return table of p-values, confidence intervals, and R values instead of displaying a plot.
color_scale	Character. Either "default" or an odd-numbered color scale where the lowest value will correspond to -1, the median value to 0, and the highest value to 1.
number_size	Numeric. Size of the text label when plot_type is "number".
point_scale	Numeric. The size of the largest point if the plot_type is "circle"

Value

Plots pairwise correlation plot for the samples in your_SE.

Examples

```
data(wu_subset)
cor_plot(your_SE = wu_subset, plot_type = "color")
# "
```

`create_SE` *create_SE*

Description

Creates a SummarizedExperiment object from a data frame containing clonal tracking counts (‘your_data‘) with rows as observations and columns as samples, and the associated metadata (‘meta_data‘) with rows as samples and columns of information describing those samples.

Usage

```
create_SE(
  your_data = NULL,
  meta_data = NULL,
  threshold = 0,
  threshold_type = "relative",
  log_base = exp(1),
  scale_factor = 1e+06
)
```

Arguments

your_data	A data frame. For clonal tracking data, this will be individual barcodes or lineage tracing elements in rows and samples in columns.
meta_data	A data frame containing all meta-data. Must, at the very least, include a column called "SAMPLENAME" that contains all of the colnames within the data frame passed as ‘your_data‘ and only those colnames.
threshold	Numeric. The minimum threshold abundance for a barcode to be maintained in the SE. If ‘threshold_type‘ is relative, this parameter should be between 0 and 1. If ‘threshold_type‘ is absolute, this parameter should be greater than 1.
threshold_type	Character. One of "relative" or "absolute" relative. If a relative threshold is specified, only those rows which have higher than ‘threshold‘ proportion of reads within at least one sample will be kept as non-zero. If an absolute threshold is specified, only those rows which have an absolute read count higher than ‘threshold‘ in at least one sample will be kept as non-zero.
log_base	A numeric indicating which base to use when logging the normalized data
scale_factor	A numeric indicating what scaling factor to use in normalization. For the default value of 1 million, barcode proportions on a per sample basis will be multiplied by 1 million before log+1 normalization.

Value

Returns a SummarizedExperiment holding your clonal tracking data and the associated metadata.

Examples

```
count_path <- system.file("extdata",
    "/WuC_etal_appdata/sample_data_ZJ31.txt",
    package = "barcodetrackR"
)
wu_dataframe <- read.delim(count_path, row.names = 1)
metadata_path <- system.file("extdata",
    "/WuC_etal_appdata/sample_metadata_ZJ31.txt",
    package = "barcodetrackR"
)
wu_metadata <- read.delim(metadata_path)
wu_SE <- create_SE(
    your_data = wu_dataframe, meta_data = wu_metadata,
    threshold = 0
)
```

---

dist_plot *Pairwise Distance Plot*

Description

Plots the pairwise distances of the specified assay between each sample-sample pair in the provided SummarizedExperiment.

Usage

```
dist_plot(
  your_SE,
  assay = "proportions",
  plot_labels = colnames(your_SE),
  dist_method = "euclidean",
  cluster_tree = FALSE,
  your_title = "",
  grid = TRUE,
  label_size = 10,
  plot_type = "color",
  no_negatives = FALSE,
  return_table = FALSE,
  color_pal = "Blues",
  number_size = 3,
  point_scale = 5,
  minkowski_p = 2
)
```

Arguments

your_SE	A Summarized Experiment object.
assay	The choice of assay to use for the correlation calculation. Set to "proportions" by default.
plot_labels	Vector of x axis labels. Defaults to colnames(your_SE).
dist_method	Character. Distance OR similarity measure from the ‘proxy‘ package. Full list of distance and similarity measures can be found using ‘summary(proxy::pr_DB)‘. Default is "euclidean". Distances will be calculated for distance measures, while similarities will be calculated for similarity measures. Distance OR similarity measure will be calculated using the ‘assay‘ specified.
cluster_tree	Logical. Whether to cluster samples and plot a hierarchical tree calculated from the distance or similarity measure used. Default is FALSE.
your_title	Character. The title for the plot.
grid	Logical. Include a grid or not in the resulting plot.
label_size	Numeric. The size of the column labels.
plot_type	Character. One of "color", "circle", or "number".
no_negatives	Logical. Whether to make negative correlations = 0.
return_table	Logical. Whether or not to return table of p-values, confidence intervals, and R values instead of displaying a plot.
color_pal	Character. One of ’Reds’, ’Purples’, ’Oranges’, ’Greys’, ’Greens’, or ’Blues’ that designates the brewer.pal color scale to use.
number_size	Numeric. size of the text label when plot_type is "number".
point_scale	Numeric. The size of the largest point if the plot_type is "circle".
minkowski_p	Numeric. If ’Minkowski’ is chosen, the ’p’ used to calculate the Minkowski distance.

Value

Plots pairwise correlation plot for the samples in your_SE.

Examples

```
data(wu_subset)
dist_plot(your_SE = wu_subset, plot_type = "color")
# "
```

`estimate_barcode_threshold`

Estimate Barcode Threshold

Description

Estimates an appropriate minimum abundance threshold for reliably detected barcodes in a clonal tracking dataset.

For a specified capture efficiency C, the minimum clone size N that we can expect to detect with confidence level P is calculated from:
'P = 1 - (1 - C)^(N)'

The proportional abundance of a clonal tag of size N is
'N / (T * F)'
where T is the total population size of cells or genomes and F is the frequency or proportion of the total population which is labeled or genetically modified with the clonal tag.

The population size and proportion labeled must be determined experimentally. The capture efficiency should be estimated for a given clonal tracking technique by simulating the barcode retrieval process in silico and finding the capture efficiency which leads to a total # of detected barcodes matching the experimentally determined number. Adair et al '(PMID: 32355868)' performed this analysis for viral integration site analysis and DNA barcode sequencing and determined good estimates for the capture efficiencies of these two technologies to be 0.05 and 0.4 respectively.

Usage

```
estimate_barcode_threshold(
  capture_efficiency = NULL,
  population_size,
  proportion_labeled,
  confidence_level = 0.95,
  verbose = TRUE
)
```

Arguments

`capture_efficiency`
: Numeric. The capture efficiency of the clonal tracking method to detect a given clone. Must be between 0 and 1. See the description for details on how to estimate this value for a given experiment.

`population_size`
: Numeric. The total number of cells/genomes within each sample analyzed in the clonal tracking study. This is an experimentally determined value.

`proportion_labeled`
: Numeric. The proportion of the ‘population_size‘ which is genetically modified or contains a clonal tracking index. This is an experimentally determined value.

`confidence_level`
: Numeric. The confidence level for estimatig the minimum abundance threshold. Must be between 0 and 1. Default is 0.95 for 95 percent confidence that a clone with proportion ‘relative_threshold‘ will be detected. Increasing this parameter closer to one will result in a more stringent abundance threshold and decreasing this parameter will result in a more permissive abundance threshold.

`verbose`
: Logical. Whether to print the calculated threshold.

Value

Returns a single numeric ‘relative_threshold‘ describing the proportional abundance above which clones can be considered reliable given the provided capture efficiency and labeled population size. Pass this value into the function ‘threshold_SE‘ to threshold an existing SummarizedExperiment object or the function ‘create_SE‘ to threshold a SummarizedExperiment object upon creation from dataframes of counts and metadata.

Examples

```
estimate_barcode_threshold(
    capture_efficiency = 0.4,
    population_size = 500000,
    proportion_labeled = 0.3,
    confidence_level = 0.95,
    verbose = TRUE
)
```

---

get_top_clones *get_top_clones (helper function)*

Description

Retrieves the sequence(s) (row-identifier(s)) of the top "n_clones" within the specified sample from a SummarizedExperiment object.

Usage

```
get_top_clones(your_SE, SAMPLENAME_choice, n_clones = 10)
```

Arguments

your_SE	A summarized experiment.
SAMPLENAME_choice	Name of the SAMPLENAME identifier within your_SE from which to retrieve the top clones from.
n_clones	Numeric. Number of top clones from the specified sample that should be retrieved.

Value

The row indices for the top n_clones in the dataset, using the 'ranks' assay.

Examples

```
data(wu_subset)
get_top_clones(wu_subset, "ZJ31_6m_T", n_clones = 10)
```

---

launchApp *Launch Barcode App*

Description

Launches the Shiny Barcode App.

Usage

```
launchApp(x = NULL)
```

Arguments

x	NULL

Value

Page launching the Shiny Barcode App

Examples

```
if (interactive()) launchApp()
```

mds_plot *MDS Plot*

Description

Calculates a simmilarity/dissimlarity index or metrix for each sample-sample pair and reduces the resulting dist matrix into two dimensions

Usage

```
mds_plot(
  your_SE,
  group_by = "SAMPLENAME",
  method_dist = "bray",
  assay = "proportions",
  your_title = NULL,
  point_size = 3,
  text_size = 12,
  return_table = FALSE,
  kmeans_cluster = FALSE,
  k.param = 3,
  draw_ellipses = FALSE
)
```

Arguments

your_SE	Summarized Experiment object containing clonal tracking data as created by the barcodetrackR 'create_SE' function.
group_by	Column of metadata to color samples by. Can also specify "kmeans_cluster" if kmeans_cluster argument is set to TRUE, and then the grouping variables will be the clusterinng result.
method_dist	Dissimilarity index from vegan. One of "manhattan", "euclidean", "canberra", "clark", "bray", "kulczynski", "jaccard", "gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", or "cao".
assay	The assay to calculate the index on
your_title	Character. The title for the plot.
point_size	Numeric. The size of the points.
text_size	Numeric. Size of text in plot.
return_table	Logical. If set to true, the function will return a dataframe containing each samples reduced measure of dissimilarity coordinates.
kmeans_cluster	Logical. If set to true, each sample will be assigned a cluster computed by kmeans on the chosen assay.
k.param	Numeric. If kmeans_cluster is TRUE, provide the number of kmeans clusters to identify.
draw_ellipses	Logical. If kmeans_cluster is TRUE, draw ellipses around the different kmeans clusters.

Value

Plots dissimilarity indices between samples in your_SE. Or if return table is set to TRUE, returns a dataframe of each sample's reduced measures of dissimilarity coordinates.

Examples

```
data(wu_subset)
mds_plot(your_SE = wu_subset, method_dist = "bray", group_by = "celltype")
# "
```

`rank_abundance_plot` *Rank Abundance Plot*

Description

Create a rank abundance plot of the barcodes in the chosen samples provided in 'your_SE'. Use this function to visualize the distribution of barcode abundances within sample(s). Note: If comparing the visualization to the statistical testing results from 'rank_abundance_stat_test' function in barcodetrackR, please set the 'scale_rank' to TRUE. The K-S test is agnostic to number of samples so it is directly comparable to the visualization produced when the barcode ranks are scaled between 0 and 1.

Usage

```
rank_abundance_plot(
  your_SE,
  scale_rank = FALSE,
  point_size = 3,
  your_title = NULL,
  text_size = 12,
  plot_labels = NULL,
  return_table = FALSE
)
```

Arguments

`your_SE`	Summarized Experiment object containing clonal tracking data as created by the barcodetrackR 'create_SE' function.
`scale_rank`	Logical. Whether or not to scale all ranks from 0 to 1 or keep barcode ranks as their actual integer values. When 'scale_rank' is set to FALSE, all samples will not necessarily have the same x maximum.
`point_size`	Numeric. Size of the points for the plot.
`your_title`	Character. Specify a title for the plot.
`text_size`	Numeric. Size of text in plot.
`plot_labels`	Vector of labels for each sample. If not specified, the colnames(your_SE) will be used.

`return_table` Logical. If set to TRUE, rather than a plot, the function will return a dataframe containing for each sample, each barcode in rank order with its abundance in that sample, its scaled rank (0 to 1), and the cumulative sum of abundance for all barcodes with rank <= the rank of that barcode.

Value

Displays a rank-abundance plot (made by ggplot2) of the samples chosen.
Each point represents a single barcode with the x-value describing its rank in abundance with 1 being the most abundant barcode
The y-value representing the cumulative abundance of all barcodes with rank less than or equal to the x-axis value.
If the return_table is set to TRUE, instead of a plot, a datframe with the rank abundance data will be returned.

Examples

```
data(wu_subset)
rank_abundance_plot(your_SE = wu_subset[, 1:4], point_size = 2)
```

---

rank_abundance_stat_test

Rank Abundance Statistical Test

Description

Carries out a specific instance of statistical testing relevant to clonal tracking experiments. For the provided SummarizedExperiment, compare the rank-abundance distribution which is described by the increase in cumulative abundance within that sample as barcode abundances are added, starting with the most abundant barcode. The two-sided Kolmogorov-Smirnov statistical test is carried out comparing each pair of samples using the R function ks.test:https://www.rdocumentation.org/packages/dgof/versions/1.2/topi
Note that this test compares rank-abundance distribution regardless of whether the samples share the same barcodes or lineage tracing elements. The test could be employed on two samples with no barcode sequence overlap, simply to compare whether the rank abundance distribution of barcodes is drawn from the same distribution.

Usage

```
rank_abundance_stat_test(your_SE, statistical_test = "ks")
```

Arguments

`your_SE` Summarized Experiment object containing clonal tracking data as created by the barcodetrackR ‘create_SE‘ function.

`statistical_test`
The statistical test used to compare distributions. For now, the only implemented test is the Kolmogorov-Smirnov test.

Value

Returns a list containing two dataframes
[["D_statistic"]] is a dataframe containing pairwise D-statistics between each pair of samples in your_SE. The D statistic represents the maximal difference between the two rank abundance distributions.
[["p_value]] A dataframe containing the p-value computed by the KS test for each pair of samples. The null hypothesis is that the two rank-abundance profiles come from the same distribution.

Examples

```
data(wu_subset)
rank_abundance_stat_test(your_SE = wu_subset, statistical_test = "ks")
```

---

scatter_plot *Scatter Plot*

Description

Plots a scatter plot of two samples in the Summarized Experiment object

Usage

```
scatter_plot(
  your_SE,
  assay = "proportions",
  plot_labels = colnames(your_SE),
  method_corr = "pearson",
  display_corr = TRUE,
  point_size = 0.5,
  your_title = "",
  text_size = 12
)
```

Arguments

your_SE	A Summarized Experiment object of two samples.
assay	The choice of assay to plot on the scatter plot. Set to "proportions" by default.
plot_labels	The labels for the X and Y axis of the plot
method_corr	Character. One of "pearson", "spearman", or "kendall". Can also use "manhattan" to compute manhattan distance instead.
display_corr	Logical. Whether to display the computer correlation or not.
point_size	Numeric. The size of the points being plotted.
your_title	Logical. The title for the plot.
text_size	Numeric. Size of text in plot.

Value

Displays a scatter plot of the specified assay for the specified samples in your_SE with correlation value optionally displayed.

Examples

```
data(wu_subset)
scatter_plot(your_SE = wu_subset[, c(4, 8)])
# "
```

`stat_hist` *Stat histogram*

Description

Given a summarized experiment, gives a histogram of the acc assay or choice of metadata.

Usage

```
stat_hist(
  your_SE,
  data_choice = "assay stats",
  assay_choice = "counts",
  metadata_stat = NULL,
  group_meta_by = NULL,
  scale_all_y = FALSE,
  y_log_axis = FALSE,
  text_size = 12,
  n_bins = 30,
  n_cols = NULL,
  your_title = NULL
)
```

Arguments

`your_SE`	Your SummarizedExperiment of barcode data and associated metadata.
`data_choice`	Either "assay stats" which allows you to view the distribution of values in the ‘assay_choice‘ assay, or "metadata stats" which allows you to view the distribution of metadata values in your SummarizedExperiment object.
`assay_choice`	When data_choice is set to "assay stats", designates which assay will be used.
`metadata_stat`	When data_choice is set to "metadata stats", The metadata values that will be used.
`group_meta_by`	When data_choice is set to "metadata stats", facet the histogram using this column of metadata. If NULL, no grouping or faceting applied
`scale_all_y`	Logical. Whether or not to plot all plots on the same y axis limits.

y_log_axis	Logical. Whether or not to put y axis on log scale
text_size	Size of text.
n_bins	Number of bins for histograms. Default is 30.
n_cols	Number of columns for faceted histograms. If NULL (default) will automatically choose n_cols for facetting.
your_title	Character. The title for the plot.

Value

Histogram of chosen statistics

Examples

```
data(wu_subset)
stat_hist(
    your_SE = wu_subset[, 1], data_choice = "assay stats",
    assay_choice = "counts"
)
```

---

subset_SE *subset_SE*

Description

Subsets an existing SummarizedExperiment object.

Usage

```
subset_SE(your_SE, ...)
```

Arguments

your_SE	A SummarizedExperiment object.
...	Arguments passed to subset_SE in the form of 'X = keys' where 'X' is a column from SE's colData and 'keys' are entries in the colData to subset.

Value

Returns a subsetted SummarizedExperiment object.

Examples

```
data(wu_subset)
wu_B.5month <- subset_SE(wu_subset, celltype = "B", timepoint = "6.0")
```

threshold *Threshold*

Description

This is a helper which function takes in sequence data in table form, along with a threshold, to each column (e.g. if threshold is set as 0.0005, only rows in which an element is above 0.05 its column will be kept).

Usage

```
threshold(your_data, thresh = 5e-04, thresh_type = "relative")
```

Arguments

your_data	A data frame. Usually individual barcodes in rows and samples in columns.
thresh	Numeric.
thresh_type	Character. One of "relative" or "absolute"

Value

A data frame where all rows (barcodes) that did not have at least one element meet the threshold have been discarded.

Examples

```
data(wu_subset)
threshold(SummarizedExperiment::assay(wu_subset, assay = "counts"),
    thresh = 0.0005
)
```

threshold_SE *Threshold SE*

Description

Removes barcodes from a SummarizedExperiment object which have an abundance lower than the provided relative or absolute threshold. See the function ‘estimate_barcode_threshold‘ to estimate an appropriate threshold for an SE.

Usage

```
threshold_SE(
  your_SE,
  threshold_value,
  threshold_type = "relative",
  verbose = TRUE
)
```

Arguments

your_SE	A Summarized Experiment object.
threshold_value	Numeric. The minimum threshold abundance for a barcode to be maintained in the SE. If ‘threshold_type‘ is relative, this parameter should be between 0 and 1. If ‘threshold_type‘ is absolute, this parameter should be greater than 1.
threshold_type	Character. One of "relative" or "absolute" relative. If a relative threshold is specified, only those rows which have higher than ‘threshold_value‘ proportion of reads within at least one sample will be kept as non-zero. If an absolute threshold is specified, only those rows which have an absolute read count higher than ‘threshold_value‘ in at least one sample will be kept as non-zero.
verbose	Logical. If TRUE, print the total number of barcodes removed from the SE.

Value

Returns a SummarizedExperiment containing only barcodes which passed the supplied threshold in at least one sample. All of the defualt assays are re-calculated after thresholding is applied. Note that since tthe SE is re-instantiated, any custom assays should be recalculated after thresholding.

Examples

```
data(wu_subset)
threshold_SE(
    your_SE = wu_subset, threshold_value = 0.005,
    threshold_type = "relative", verbose = TRUE
)
```

---

wu_subset *Small subset of Wu barcoding dataset*

Description

A SummarizedExperiment object containing a subset of the Wu barcoding dataset. It includes peripheral blood T, B, Gr, NK_56, and NK-16 samples from the first 4 times points of macaque ZJ31.

Usage

```
data(wu_subset)
```

Format

A SummarizedExperiment object with 215 features rows and 20 samples:

assays includes the counts, proportions, ranks, normalized, and logs assays

colData includes the accompanying metadata for the samples

metadata includes the scale_factor used and the log_base used in the log assay ...

Source

system.file("sample_data/WuC_etal/monkey_ZJ31.txt", package = "barcodetrackR") system.file("sample_data/WuC_etal/mo
package = "barcodetrackR") wu_SE <- create_SE(your_data = wu_dataframe, meta_data = wu_metadata,
threshold = 0.005) wu_subset <- wu_SE[,1:20] http://dx.doi.org/10.1126/sciimmunol.aat9781

Index

∗ **datasets**
wu_subset, 34

barcode_binary_heatmap, 2
barcode_ggheatmap, 3
barcode_ggheatmap_stat, 5
barcode_stat_test, 7
bias_histogram, 9
bias_lineplot, 11
bias_ridge_plot, 12
build_index_html, 14

chord_diagram, 14
clonal_contribution, 15
clonal_count, 17
clonal_diversity, 19
cor_plot, 20
create_SE, 21

dist_plot, 22

estimate_barcode_threshold, 24

get_top_clones, 25

launchApp, 26

mds_plot, 27

rank_abundance_plot, 28
rank_abundance_stat_test, 29

scatter_plot, 30
stat_hist, 31
subset_SE, 32

threshold, 33
threshold_SE, 33

wu_subset, 34