

Package ‘SCpubr’

January 18, 2023

Type Package

Title Generate Publication Ready Visualizations of Single Cell
Transcriptomics Data

Version 1.1.2

Description A system that provides a streamlined way of generating publication ready plots for known Single-Cell transcriptomics data in a “publication ready” format. This is, the goal is to automatically generate plots with the highest quality possible, that can be used right away or with minimal modifications for a research article.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 4.0.0)

RoxygenNote 7.2.1

URL <https://github.com/enblacar/SCpubr/>,
<https://enblacar.github.io/SCpubr-book/>

BugReports <https://github.com/enblacar/SCpubr/issues/>

Suggests assertthat, circlize, colorspace, covr, ComplexHeatmap, dplyr, ggbeeswarm, ggdist, ggExtra, ggplot2 (>= 3.4.0), ggplotify, ggrastr, ggrepel, ggridges, ggsignif, graphics, magrittr, Nebulosa, patchwork, pheatmap, plyr, rlang, scales, scattermore, Seurat, tibble, tidyr, forcats, infercnv, knitr, Matrix, purrr, qpdf, rgdal, rjags, rmarkdown, stringr, svglite, ggalluvial, testthat (>= 3.0.0), viridis, UCell, AUCell, clusterProfiler, enrichplot, ggnewscale, AnnotationDbi, org.Hs.eg.db

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Software, SingleCell, Visualization

NeedsCompilation no

Author Enrique Blanco-Carmona [cre, aut]
 (<<https://orcid.org/0000-0002-1208-1691>>)

Maintainer Enrique Blanco-Carmona <scpubr@gmail.com>

Repository CRAN

Date/Publication 2023-01-18 12:20:02 UTC

R topics documented:

do_AlluvialPlot	3
do_AzimuthAnalysisPlot	6
do_BarPlot	9
do_BeeSwarmPlot	11
do_BoxPlot	15
do_CellularStatesPlot	19
do_ChordDiagramPlot	24
do_ColorPalette	26
do_CopyNumberVariantPlot	29
do_CorrelationPlot	32
do_DimPlot	34
do_DotPlot	39
do_EnrichmentHeatmap	43
do_ExpressionHeatmap	48
do_FeaturePlot	50
do_FunctionalAnnotationPlot	55
do_GeyserPlot	59
do_GroupedGOTermPlot	63
do_GroupwiseDEPlot	65
do_NebulosaPlot	67
do_PathwayActivityPlot	70
do_RidgePlot	74
do_TermEnrichmentPlot	78
do_TFActivityPlot	81
do_ViolinPlot	84
do_VolcanoPlot	87
human_chr_locations	89
state_dependencies	90

Index

91

do_AlluvialPlot	<i>Generate Alluvial plots.</i>
-----------------	---------------------------------

Description

This function is based on the **ggalluvial** package. It allows you to generate alluvial plots from a given Seurat object.

Usage

```
do_AlluvialPlot(  
  sample,  
  first_group,  
  last_group,  
  middle_groups = NULL,  
  colors.use = NULL,  
  plot.title = NULL,  
  plot.subtitle = NULL,  
  plot.caption = NULL,  
  font.size = 14,  
  font.type = "sans",  
  xlab = NULL,  
  ylab = "Number of cells",  
  repel = FALSE,  
  fill.by = last_group,  
  use_labels = FALSE,  
  stratum.color = "black",  
  stratum.fill = "white",  
  stratum.width = 1/3,  
  stratum.fill.conditional = FALSE,  
  use_geom_flow = FALSE,  
  alluvium.color = "white",  
  flow.color = "white",  
  flip = FALSE,  
  label.color = "black",  
  curve_type = "sigmoid",  
  use_viridis = FALSE,  
  viridis_color_map = "G",  
  viridis_direction = -1,  
  plot.grid = FALSE,  
  grid.color = "grey75",  
  grid.type = "dashed",  
  na.value = "white",  
  legend.position = "right",  
  legend.title = NULL  
)
```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
first_group	character Categorical metadata variable. First group of nodes of the alluvial plot.
last_group	character Categorical metadata variable. Last group of nodes of the alluvial plot.
middle_groups	character Categorical metadata variable. Vector of groups of nodes of the alluvial plot.
colors.use	character Named list of colors corresponding to the unique values in fill.by (which defaults to last_group).
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
xlab, ylab	character Titles for the X and Y axis.
repel	logical Whether to repel the text labels.
fill.by	character One of first_group, middle_groups (one of the values, if multiple mid_groups) or last_group. These values will be used to color the alluvium/flow.
use_labels	logical Whether to use labels instead of text for the stratum.
stratum.color, alluvium.color, flow.color	character Color for the border of the alluvium (and flow) and stratum.
stratum.fill	character Color to fill the stratum.
stratum.width	logical Width of the stratum.
stratum.fill.conditional	logical Whether to fill the stratum with the same colors as the alluvium/flow.
use_geom_flow	logical Whether to use geom_flow instead of geom_alluvium . Visual results might differ.
flip	logical Whether to invert the axis of the displayed plot.
label.color	character Color for the text labels.
curve_type	character Type of curve used in geom_alluvium . One of: <ul style="list-style-type: none"> • linear. • cubic. • quintic. • sine. • arctangent. • sigmoid. • xspline.

use_viridis **logical** | Whether to use viridis color scales.
 viridis_color_map **character** | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).
 viridis_direction **numeric** | Either 1 or -1. Controls how the gradient of viridis scale is formed.
 plot.grid **logical** | Whether to plot grid lines.
 grid.color **character** | Color of the grid in the panels.
 grid.type **character** | One of the possible linetype options:

- blank.
- solid.
- dashed.
- dotted.
- dotdash.
- longdash.
- twodash.

 na.value **character** | Color value for NA.
 legend.position **character** | Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

 legend.title **character** | Title for the legend.

Value

A ggplot2 object.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_AlluvialPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Generate a more fine-grained clustering.
  sample$annotation <- ifelse(sample$seurat_clusters %in% c("0", "3"), "A", "B")

  # Compute basic sankey plot.
  p <- SCpubr::do_AlluvialPlot(sample = sample,

```

```

        first_group = "annotation",
        last_group = "seurat_clusters")

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_AzimuthAnalysisPlot

Generate a report from a Seurat object that has been mapped to a reference using Azimuth.

Description

This functions takes as input a Seurat object that has undergone reference mapping using Azimuth and generates a set of visualizations and a combined output from the results of such mapping. The user can also provide the reference object used for the mapping (if available) to produce a more complete output.

Usage

```

do_AzimuthAnalysisPlot(
  sample,
  annotation.labels,
  annotation.scoring,
  mapping.scoring = "mapping.score",
  annotation.cutoff = 0.75,
  mapping.cutoff = 0,
  group.by = NULL,
  ref.obj = NULL,
  ref.reduction = "ref.umap",
  raster = FALSE,
  pt.size = if (isTRUE(raster)) {
    4
  } else {
    1
  },
  raster.dpi = 2048,
  border.size = if (isTRUE(raster)) {
    1.25
  } else {
    1.5
  },
  border.color = "black",
  na.value = "grey75",

```

```

font.size = 14,
font.type = "sans",
colors.use = NULL,
label = TRUE,
legend.position = "bottom",
viridis_color_map = "G",
viridis_direction = 1
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
annotation.labels	character Metadata column that stores the inferred annotation from Azimuth.
annotation.scoring	character Metadata column that stores the annotation scoring from Azimuth.
mapping.scoring	character Metadata column that stores the mapping scoring from Azimuth.
annotation.cutoff	numeric Value from 0 to 1 to use as cutoff to assign the labels to the object. This is used in conjunction with <code>mapping.cutoff</code> .
mapping.cutoff	numeric Value from 0 to 1 to use as cutoff to assign the labels to the object. This is used in conjunction with <code>annotation.cutoff</code> .
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
ref.obj	Seurat Seurat object used for reference mapping. Providing this object will add an extra plot with the UMAP of the reference and add its silhouette to the UMAP in which the original cells are showed in the context of the UMAP embedding of the reference object.
ref.reduction	character Name of the reduction embedding used to plot the UMAP in the reference object.
raster	logical Whether to raster the resulting plot. This is recommendable if plotting a lot of cells.
pt.size	numeric Size of the dots.
raster.dpi	numeric Pixel resolution for rasterized plots. Defaults to 1024. Only activates on Seurat versions higher or equal than 4.1.0.
border.size	numeric Width of the border of the cells.
border.color	character Color to use for the border of the cells.
na.value	character Color value for NA.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • <code>mono</code>: Mono spaced font. • <code>serif</code>: Serif font family.

	<ul style="list-style-type: none"> • <code>sans</code>: Default font family.
<code>colors.use</code>	<code>named_vector</code> Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of <code>group.by</code> . If <code>group.by</code> is not provided, defaults to the unique values of <code>Idents</code> . If not provided, a color scale will be set by default.
<code>label</code>	<code>logical</code> Whether to plot the cluster labels in the UMAP. The cluster labels will have the same color as the cluster colors.
<code>legend.position</code>	<code>character</code> Position of the legend in the plot. One of: <ul style="list-style-type: none"> • <code>top</code>: Top of the figure. • <code>bottom</code>: Bottom of the figure. • <code>left</code>: Left of the figure. • <code>right</code>: Right of the figure. • <code>none</code>: No legend is displayed.
<code>viridis_color_map</code>	<code>character</code> A capital letter from A to H or the scale name as in <code>scale_fill_viridis</code> .
<code>viridis_direction</code>	<code>numeric</code> Either 1 or -1. Controls how the gradient of viridis scale is formed.

Value

A list containing multiple plots.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_AzimuthAnalysisPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object that has undergone Azimuth mapping.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Generate an Azimuth report.
  out <- SCpubr::do_AzimuthAnalysisPlot(sample = sample,
                                       annotation.labels = "annotation",
                                       annotation.scoring = "annotation.score",
                                       font.size = 18)

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}
```

do_BarPlot	<i>Create Bar Plots.</i>
------------	--------------------------

Description

Create Bar Plots.

Usage

```
do_BarPlot(  
  sample,  
  group.by,  
  order = TRUE,  
  split.by = NULL,  
  position = "stack",  
  font.size = 14,  
  font.type = "sans",  
  legend.position = "bottom",  
  legend.title = NULL,  
  legend.ncol = NULL,  
  legend.nrow = NULL,  
  legend.byrow = FALSE,  
  rotate_x_axis_labels = 45,  
  xlab = NULL,  
  ylab = NULL,  
  colors.use = NULL,  
  flip = FALSE,  
  plot.title = NULL,  
  plot.subtitle = NULL,  
  plot.caption = NULL,  
  plot.grid = TRUE,  
  grid.color = "grey75",  
  grid.type = "dashed"  
)
```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
group.by	character Metadata column to compute the counts of. Has to be either a character or factor column.
order	logical Whether to order the results in descending order of counts.
split.by	character Metadata column to split the values of group.by by. If not used, defaults to the active ids.
position	character Position function from ggplot2 . One of: <ul style="list-style-type: none">• <code>stack</code>: Set the bars side by side, displaying the total number of counts. Uses position_stack.

	<ul style="list-style-type: none"> • <code>fill</code>: Set the bars on top of each other, displaying the proportion of counts from the total that each group represents. Uses <code>position_fill</code>.
<code>font.size</code>	<code>numeric</code> Overall font size of the plot. All plot elements will have a size relationship with this font size.
<code>font.type</code>	<code>character</code> Base font family for the plot. One of: <ul style="list-style-type: none"> • <code>mono</code>: Mono spaced font. • <code>serif</code>: Serif font family. • <code>sans</code>: Default font family.
<code>legend.position</code>	<code>character</code> Position of the legend in the plot. One of: <ul style="list-style-type: none"> • <code>top</code>: Top of the figure. • <code>bottom</code>: Bottom of the figure. • <code>left</code>: Left of the figure. • <code>right</code>: Right of the figure. • <code>none</code>: No legend is displayed.
<code>legend.title</code>	<code>character</code> Title for the legend.
<code>legend.ncol</code>	<code>numeric</code> Number of columns in the legend.
<code>legend.nrow</code>	<code>numeric</code> Number of rows in the legend.
<code>legend.byrow</code>	<code>logical</code> Whether the legend is filled by row or not.
<code>rotate_x_axis_labels</code>	<code>numeric</code> Degree to rotate the X labels. One of: 0, 45, 90.
<code>xlab, ylab</code>	<code>character</code> Titles for the X and Y axis.
<code>colors.use</code>	<code>named_vector</code> Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of <code>group.by</code> . If <code>group.by</code> is not provided, defaults to the unique values of <code>idents</code> . If not provided, a color scale will be set by default.
<code>flip</code>	<code>logical</code> Whether to invert the axis of the displayed plot.
<code>plot.title, plot.subtitle, plot.caption</code>	<code>character</code> Title, subtitle or caption to use in the plot.
<code>plot.grid</code>	<code>logical</code> Whether to plot grid lines.
<code>grid.color</code>	<code>character</code> Color of the grid in the panels.
<code>grid.type</code>	<code>character</code> One of the possible linetype options: <ul style="list-style-type: none"> • <code>blank</code>. • <code>solid</code>. • <code>dashed</code>. • <code>dotted</code>. • <code>dotdash</code>. • <code>longdash</code>. • <code>twodash</code>.

Value

A `ggplot2` object containing a Bar plot.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_BarPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic bar plot, horizontal.
p1 <- SCpubr::do_BarPlot(sample = sample,
                        group.by = "seurat_clusters",
                        legend.position = "none",
                        plot.title = "Number of cells per cluster")

  # Split by a second variable.
sample$modified_orig.ident <- sample(x = c("Sample_A", "Sample_B", "Sample_C"),
                                   size = ncol(sample),
                                   replace = TRUE,
                                   prob = c(0.2, 0.7, 0.1))

  p <- SCpubr::do_BarPlot(sample,
                        group.by = "seurat_clusters",
                        split.by = "modified_orig.ident",
                        plot.title = "Number of cells per cluster in each sample",
                        position = "stack")

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`".)
}

```

do_BeeSwarmPlot

*BeeSwarm plot.***Description**

BeeSwarm plot.

Usage

```

do_BeeSwarmPlot(
  sample,
  feature_to_rank,
  group.by = NULL,

```

```

assay = NULL,
reduction = NULL,
slot = NULL,
continuous_feature = FALSE,
colors.use = NULL,
legend.title = NULL,
legend.type = "colorbar",
legend.position = if (isTRUE(continuous_feature)) {
  "bottom"
} else {

  "none"
},
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.length = 20,
legend.width = 1,
legend.framecolor = "grey50",
legend.tickcolor = "white",
legend.ncol = NULL,
legend.icon.size = 4,
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
xlab = NULL,
ylab = NULL,
font.size = 14,
font.type = "sans",
remove_x_axis = FALSE,
remove_y_axis = FALSE,
flip = FALSE,
viridis_color_map = "G",
viridis_direction = 1,
verbose = TRUE,
raster = FALSE,
raster.dpi = 300,
plot_cell_borders = TRUE,
border.size = 1.5,
border.color = "black",
pt.size = 2,
min.cutoff = NULL,
max.cutoff = NULL
)

```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).
feature_to_rank [character](#) | Feature for which the cells are going to be ranked. Ideal case is that

	this feature is stored as a metadata column.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
assay	character Assay to use. Defaults to the current assay.
reduction	character Reduction to use. Can be the canonical ones such as "umap", "pca", or any custom ones, such as "diffusion". If you are unsure about which reductions you have, use <code>Seurat::Reductions(sample)</code> . Defaults to "umap" if present or to the last computed reduction if the argument is not provided.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
continuous_feature	logical Is the feature to rank and color for continuous? I.e: an enrichment score.
colors.use	named_vector Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of group.by. If group.by is not provided, defaults to the unique values of Idents . If not provided, a color scale will be set by default.
legend.title	character Title for the legend.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none"> • normal: Default legend displayed by ggplot2. • colorbar: Redefined colorbar legend, using guide_colorbar. • colorsteps: Redefined legend with colors going by range, in steps, using guide_colorsteps.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
legend.framewidth, legend.tickwidth	numeric Width of the lines of the box in the legend.
legend.length, legend.width	numeric Length and width of the legend. Will adjust automatically depending on legend side.
legend.framecolor	character Color of the lines of the box in the legend.
legend.tickcolor	character Color of the ticks of the box in the legend.
legend.ncol	numeric Number of columns in the legend.
legend.icon.size	numeric Size of the icons in legend.

`plot.title`, `plot.subtitle`, `plot.caption`
 character | Title, subtitle or caption to use in the plot.

`xlab`, `ylab` **character** | Titles for the X and Y axis.

`font.size` **numeric** | Overall font size of the plot. All plot elements will have a size relationship with this font size.

`font.type` **character** | Base font family for the plot. One of:

- `mono`: Mono spaced font.
- `serif`: Serif font family.
- `sans`: Default font family.

`remove_x_axis`, `remove_y_axis`
 logical | Remove X axis labels and ticks from the plot.

`flip` **logical** | Whether to invert the axis of the displayed plot.

`viridis_color_map`
 character | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

`viridis_direction`
 numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

`verbose` **logical** | Whether to show extra comments, warnings, etc.

`raster` **logical** | Whether to raster the resulting plot. This is recommendable if plotting a lot of cells.

`raster.dpi` **numeric** | Pixel resolution for rasterized plots. Defaults to 1024. Only activates on Seurat versions higher or equal than 4.1.0.

`plot_cell_borders`
 logical | Whether to plot border around cells.

`border.size` **numeric** | Width of the border of the cells.

`border.color` **character** | Color to use for the border of the cells.

`pt.size` **numeric** | Size of the dots.

`min.cutoff`, `max.cutoff`
 numeric | Set the min/max ends of the color scale. Any cell/group with a value lower than `min.cutoff` will turn into `min.cutoff` and any cell with a value higher than `max.cutoff` will turn into `max.cutoff`. In `FeaturePlots`, provide as many values as features. Use NAs to skip a feature.

Value

A `ggplot2` object containing a Bee Swarm plot.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_BeeSwarmPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

```

```
# Define your Seurat object.
sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

# Basic Bee Swarm plot - categorical coloring.
# This will color based on the unique values of seurat_clusters.
p <- SCpubr::do_BeeSwarmPlot(sample = sample,
                             feature_to_rank = "PC_1",
                             group.by = "seurat_clusters",
                             continuous_feature = FALSE)

# Basic Bee Swarm plot - continuous coloring.
# This will color based on the PC_1 values.
p <- SCpubr::do_BeeSwarmPlot(sample = sample,
                             feature_to_rank = "PC_1",
                             group.by = "seurat_clusters",
                             continuous_feature = TRUE)
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}
```

do_BoxPlot

Generate Box Plots.

Description

Generate Box Plots.

Usage

```
do_BoxPlot(
  sample,
  feature,
  group.by = NULL,
  split.by = NULL,
  assay = NULL,
  slot = "data",
  font.size = 14,
  font.type = "sans",
  rotate_x_axis_labels = 45,
  colors.use = NULL,
  na.value = "grey75",
  plot.title = NULL,
  plot.subtitle = NULL,
  plot.caption = NULL,
  xlab = NULL,
  ylab = NULL,
```

```

legend.title = if (is.null(split.by)) {
  if (is.null(group.by)) {

    "Groups"
  }
  else {
    group.by
  }
} else {
  split.by
},
legend.title.position = "top",
legend.position = if (is.null(split.by)) {
  "none"
} else {
  "bottom"
},
boxplot.line.color = "black",
outlier.color = "black",
outlier.alpha = 0.5,
boxplot.linewidth = 1,
boxplot.width = NULL,
plot.grid = TRUE,
grid.color = "grey75",
grid.type = "dashed",
flip = FALSE,
order = FALSE,
use_silhouette = FALSE,
use_test = FALSE,
comparisons = NULL,
test = "wilcox.test",
map_signif_level = TRUE
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
feature	character Feature to represent.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.

font.type	character Base font family for the plot. One of: <ul style="list-style-type: none">• mono: Mono spaced font.• serif: Serif font family.• sans: Default font family.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
colors.use	named_vector Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of group.by. If group.by is not provided, defaults to the unique values of Idents . If not provided, a color scale will be set by default.
na.value	character Color value for NA.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
xlab, ylab	character Titles for the X and Y axis.
legend.title	character Title for the legend.
legend.title.position	character Position for the title of the legend. One of: <ul style="list-style-type: none">• top: Top of the legend.• bottom: Bottom of the legend.• left: Left of the legend.• right: Right of the legend.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none">• top: Top of the figure.• bottom: Bottom of the figure.• left: Left of the figure.• right: Right of the figure.• none: No legend is displayed.
boxplot.line.color	character Color of the borders of the boxplots if use_silhouette is FALSE.
outlier.color	character Color of the outlier dots.
outlier.alpha	numeric Alpha applied to the outliers.
boxplot.linewidth	numeric Width of the lines in the boxplots. Also controls the lines of the tests applied if use_test is set to true.
boxplot.width	numeric Width of the boxplots.
plot.grid	logical Whether to plot grid lines.
grid.color	character Color of the grid in the panels.
grid.type	character One of the possible linetype options: <ul style="list-style-type: none">• blank.• solid.

	<ul style="list-style-type: none"> • dashed. • dotted. • dotdash. • longdash. • twodash.
flip	logical Whether to invert the axis of the displayed plot.
order	logical Whether to order the boxplots by average values. Can not be used alongside split.by.
use_silhouette	logical Whether to color the borders of the boxplots instead of the inside area.
use_test	logical Whether to apply a statistical test to a given pair of elements. Can not be used alongside split.by.
comparisons	A list of length-2 vectors. The entries in the vector are either the names of 2 values on the x-axis or the 2 integers that correspond to the index of the columns of interest.
test	the name of the statistical test that is applied to the values of the 2 columns (e.g. t.test, wilcox.test etc.). If you implement a custom test make sure that it returns a list that has an entry called p.value.
map_signif_level	Boolean value, if the p-value are directly written as annotation or asterisks are used instead. Alternatively one can provide a named numeric vector to create custom mappings from p-values to annotation: For example: c("***"=0.001, "**"=0.01, "*"=0.05). Alternatively, one can provide a function that takes a numeric argument (the p-value) and returns a string.

Value

A ggplot2 object.

Examples

```
# Check Suggests.
value <- SCpubr::check_suggests(function_name = "do_BoxPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic box plot.
  p <- SCpubr::do_BoxPlot(sample = sample,
                          feature = "nCount_RNA")
  p

  # Use silhouette style.
  p <- SCpubr::do_BoxPlot(sample = sample,
                          feature = "nCount_RNA",
```

```

                                use_silhouette = TRUE)
p

# Order by mean values.
p <- SCpubr::do_BoxPlot(sample = sample,
                        feature = "nCount_RNA",
                        order = TRUE)
p

# Apply second grouping.
sample$orig.ident <- ifelse(sample$seurat_clusters %in% c("0", "1", "2", "3"), "A", "B")
p <- SCpubr::do_BoxPlot(sample = sample,
                        feature = "nCount_RNA",
                        split.by = "orig.ident")
p

# Apply statistical tests.
p <- SCpubr::do_BoxPlot(sample = sample,
                        feature = "nCount_RNA",
                        group.by = "orig.ident",
                        use_test = TRUE,
                        comparisons = list(c("A", "B")))
p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}

```

do_CellularStatesPlot *Cellular States plot.*

Description

This plot aims to show the relationships between distinct enrichment scores. If 3 variables are provided, the relationship is between the Y axis and the dual X axis. If 4 variables are provided, each corner of the plot represents how enriched the cells are in that given list. How to interpret this? In a 3-variable plot, the Y axis just means one variable. The higher the cells are in the Y axis the more enriched they are in that given variable. The X axis is a dual parameter one. Cells falling into each extreme of the axis are highly enriched for either x1 or x2, while cells falling in between are not enriched for any of the two. In a 4-variable plot, each corner shows the enrichment for one of the 4 given features. Cells will tend to locate in either of the four corners, but there will be cases of cells locating mid-way between two given corners (enriched in both features) or in the middle of the plot (not enriched for any).

Usage

```
do_CellularStatesPlot(
```

```

sample,
input_gene_list,
x1,
y1,
x2 = NULL,
y2 = NULL,
group.by = NULL,
colors.use = NULL,
legend.position = "bottom",
legend.icon.size = 4,
legend.ncol = NULL,
legend.nrow = NULL,
legend.byrow = FALSE,
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
font.size = 14,
font.type = "sans",
xlab = NULL,
ylab = NULL,
axis.ticks = TRUE,
axis.text = TRUE,
verbose = FALSE,
enforce_symmetry = FALSE,
plot_marginal_distributions = FALSE,
marginal.type = "density",
marginal.size = 5,
marginal.group = TRUE,
plot_cell_borders = TRUE,
plot_enrichment_scores = FALSE,
border.size = 2,
border.color = "black",
pt.size = 2,
raster = FALSE,
raster.dpi = 1024,
plot_features = FALSE,
features = NULL,
viridis_color_map = "G",
viridis_direction = 1,
nbin = 24,
ctrl = 100
)

```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).
input_gene_list [named_list](#) | Named list of lists of genes to be used as input.

x1	character A name of a list from <code>input_gene_list</code> . First feature in the X axis. Will go on the right side of the X axis if y2 is not provided and top-right quadrant if provided.
y1	character A name of a list from <code>input_gene_list</code> . First feature on the Y axis. Will become the Y axis if y2 is not provided and bottom-right quadrant if provided.
x2	character A name of a list from <code>input_gene_list</code> . Second feature on the X axis. Will go on the left side of the X axis if y2 is not provided and top-left quadrant if provided.
y2	character A name of a list from <code>input_gene_list</code> . Second feature on the Y axis. Will become the bottom-left quadrant if provided.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
colors.use	named_vector Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of <code>group.by</code> . If <code>group.by</code> is not provided, defaults to the unique values of <code>Idents</code> . If not provided, a color scale will be set by default.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
legend.icon.size	numeric Size of the icons in legend.
legend.ncol	numeric Number of columns in the legend.
legend.nrow	numeric Number of rows in the legend.
legend.byrow	logical Whether the legend is filled by row or not.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
xlab, ylab	character Titles for the X and Y axis.
axis.ticks	logical Whether to show axis ticks.
axis.text	logical Whether to show axis text.
verbose	logical Whether to show extra comments, warnings, etc.

enforce_symmetry	logical Whether to enforce the plot to follow a symmetry (3 variables, the X axis has 0 as center, 4 variables, all axis have the same range and the plot is squared).
plot_marginal_distributions	logical Whether to plot marginal distributions on the figure or not.
marginal.type	character One of: <ul style="list-style-type: none"> • density: Compute density plots on the margins. • histogram: Compute histograms on the margins. • boxplot: Compute boxplot on the margins. • violin: Compute violin plots on the margins. • densigram: Compute densigram plots on the margins.
marginal.size	numeric Size ratio between the main and marginal plots. A value of 5 means that the main plot is 5 times bigger than the marginal plots.
marginal.group	logical Whether to group the marginal distribution by group.by or current identities.
plot_cell_borders	logical Whether to plot border around cells.
plot_enrichment_scores	logical Whether to report enrichment scores for the input lists as plots.
border.size	numeric Width of the border of the cells.
border.color	character Color to use for the border of the cells.
pt.size	numeric Size of the dots.
raster	logical Whether to raster the resulting plot. This is recommendable if plotting a lot of cells.
raster.dpi	numeric Pixel resolution for rasterized plots. Defaults to 1024. Only activates on Seurat versions higher or equal than 4.1.0.
plot_features	logical Whether to also report any other feature onto the primary plot.
features	character Additional features to plot.
viridis_color_map	character A capital letter from A to H or the scale name as in scale_fill_viridis .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
nbin	numeric Number of bins to use in AddModuleScore .
ctrl	numeric Number of genes in the control set to use in AddModuleScore .

Details

This plots are based on the following publications:

- Neftel, C. *et al.* An Integrative Model of Cellular States, Plasticity, and Genetics for Glioblastoma. *Cell* 178, 835-849.e21 (2019). doi:[10.1016/j.cell.2019.06.024](https://doi.org/10.1016/j.cell.2019.06.024)
- Tirosh, I., Venteicher, A., Hebert, C. *et al.* Single-cell RNA-seq supports a developmental hierarchy in human oligodendrogloma. *Nature* 539, 309–313 (2016). doi:[10.1038/nature20123](https://doi.org/10.1038/nature20123)

Value

A ggplot2 object containing a butterfly plot.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_CellularStatesPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Define some gene sets to query. It has to be a named list.
  gene_set <- list("A" = rownames(sample)[1:10],
                  "B" = rownames(sample)[11:20],
                  "C" = rownames(sample)[21:30],
                  "D" = rownames(sample)[31:40])

  # Using two variables: A scatter plot X vs Y.
  p <- SCpubr::do_CellularStatesPlot(sample = sample,
                                     input_gene_list = gene_set,
                                     x1 = "A",
                                     y1 = "B",
                                     nbin = 1,
                                     ctrl = 10)

  p

  # Using three variables. Figure from: https://www.nature.com/articles/nature20123.
  p <- SCpubr::do_CellularStatesPlot(sample = sample,
                                     input_gene_list = gene_set,
                                     x1 = "A",
                                     y1 = "B",
                                     x2 = "C",
                                     nbin = 1,
                                     ctrl = 10)

  p

  # Using four variables. Figure from: https://pubmed.ncbi.nlm.nih.gov/31327527/
  p <- SCpubr::do_CellularStatesPlot(sample = sample,
                                     input_gene_list = gene_set,
                                     x1 = "A",
                                     y1 = "C",
                                     x2 = "B",
                                     y2 = "D",
                                     nbin = 1,
                                     ctrl = 10)

  p
```

```

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`".)
}

```

do_ChordDiagramPlot *Generate a Chord diagram.*

Description

Generate a Chord diagram.

Usage

```

do_ChordDiagramPlot(
  sample = NULL,
  from = NULL,
  to = NULL,
  colors.from = NULL,
  colors.to = NULL,
  big.gap = 10,
  small.gap = 1,
  link.border.color = NA,
  link.border.width = 1,
  highlight_group = NULL,
  alpha.highlight = 25,
  link.sort = NULL,
  link.decreasing = TRUE,
  z_index = FALSE,
  self.link = 1,
  symmetric = FALSE,
  directional = 1,
  direction.type = c("diffHeight", "arrows"),
  link.arr.type = "big.arrow",
  scale = FALSE,
  alignment = "default",
  annotationTrack = c("grid", "axis"),
  padding_labels = 4,
  ...
)

```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).

from, to [character](#) | Categorical metadata variable to be used as origin and end points of the interactions.

colors.from, colors.to	<code>named_vector</code> Named vector of colors corresponding to the unique values of "from" and "to".
big.gap	<code>numeric</code> Space between the groups in "from" and "to".
small.gap	<code>numeric</code> Space within the groups.
link.border.color	<code>character</code> Color for the border of the links. NA = no color.
link.border.width	<code>numeric</code> Width of the border line of the links.
highlight_group	<code>character</code> A value from from that will be used to highlight only the links coming from it.
alpha.highlight	<code>numeric</code> A value between 00 (double digits) and 99 to depict the alpha of the highlighted links. No transparency needs "FF"
link.sort	pass to <code>chordDiagramFromMatrix</code> or <code>chordDiagramFromDataFrame</code>
link.decreasing	pass to <code>chordDiagramFromMatrix</code> or <code>chordDiagramFromDataFrame</code>
z_index	<code>logical</code> Whether to bring the bigger links to the top.
self.link	<code>numeric</code> Behavior of the links. One of: <ul style="list-style-type: none"> • 1: Prevents self linking. • 2: Allows self linking.
symmetric	pass to <code>chordDiagramFromMatrix</code>
directional	<code>numeric</code> Set the direction of the links. One of: <ul style="list-style-type: none"> • 0: Non-directional data. • 1: Links go from "from" to "to". • -1: Links go from "to" to "from". • 2: Links go in both directions.
direction.type	<code>character</code> How to display the directions. One of: <ul style="list-style-type: none"> • diffHeight: Sets a line at the origin of the group showing to how many groups and in which proportion this group is linked to. • arrows: Sets the connection as arrows. • both: Sets up both behaviors. Use as: <code>c("diffHeight", "arrows")</code>.
link.arr.type	<code>character</code> Sets the appearance of the arrows. One of: <ul style="list-style-type: none"> • triangle: Arrow with a triangle tip at the end displayed on top of the link. • big.arrow: The link itself ends in a triangle shape.
scale	<code>logical</code> Whether to put all nodes the same width.
alignment	<code>character</code> How to align the diagram. One of: <ul style="list-style-type: none"> • default: Allows <code>circulize</code> to set up the plot as it sees fit. • horizontal: Sets the break between "from" and "to" groups on the horizontal axis.

- `vertical`: Sets the break between "from" and "to" groups on the vertical axis.

`annotationTrack` pass to `chordDiagramFromMatrix` or `chordDiagramFromDataFrame`

`padding_labels` `numeric` | Number of extra padding (white spaces) of the labels so that they do not overlap with the scales.

... For internal use only.

Value

A circlize plot.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_ChordDiagramPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic chord diagram.
  sample$assignment <- ifelse(sample$seurat_clusters %in% c("0", "4", "7"), "A", "B")
  sample$assignment[sample$seurat_clusters %in% c("1", "2")] <- "C"
  sample$assignment[sample$seurat_clusters %in% c("10", "5")] <- "D"
  sample$assignment[sample$seurat_clusters %in% c("8", "9")] <- "E"

  p <- SCpubr::do_ChordDiagramPlot(sample = sample,
                                   from = "seurat_clusters",
                                   to = "assignment")

  p
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}
```

Description

This function is an adaptation of colortools package. As the package was removed from CRAN on 23-06-2022, this utility function came to existence in order to cover the gap. It is, on its basis, an adaptation of the package into a single function. Original code, developed by Gaston Sanchez, can be found in: <https://github.com/gastonstat/colortools>

Usage

```
do_ColorPalette(  
  colors.use,  
  n = 12,  
  opposite = FALSE,  
  adjacent = FALSE,  
  triadic = FALSE,  
  split_complementary = FALSE,  
  tetradic = FALSE,  
  square = FALSE,  
  complete_output = FALSE,  
  plot = FALSE,  
  font.size = 14,  
  font.type = "sans"  
)
```

Arguments

colors.use	character One color upon which generate the color scale. Can be a name or a HEX code.
n	numeric Number of colors to include in the color wheel. Use it when all other options are FALSE, otherwise, it becomes 12.
opposite	logical Return the opposing color to the one provided.
adjacent	logical Return the adjacent colors to the one provided.
triadic	logical Return the triadic combination of colors to the one provided.
split_complementary	logical Return the split complementary combination of colors to the one provided.
tetradic	logical Return the tetradic combination of colors to the one provided.
square	logical Return the square combination of colors to the one provided.
complete_output	logical Runs all the previous options and returns all the outputs as a list that contains all color vectors, all plots and a combined plot with everything.
plot	logical Whether to also return a plot displaying the values instead of a vector with the color.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of:

- mono: Mono spaced font.
- serif: Serif font family.
- sans: Default font family.

Value

A character vector with the desired color scale.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_ColorPalette", passive = TRUE)

if (isTRUE(value)){
  # Generate a color wheel based on a single value.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue")
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                plot = TRUE)

  # Generate a pair of opposite colors based on a given one.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                     opposite = TRUE)
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                opposite = TRUE,
                                plot = TRUE)

  # Generate a trio of adjacent colors based on a given one.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                     adjacent = TRUE)
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                adjacent = TRUE,
                                plot = TRUE)

  # Generate a trio of triadic colors based on a given one.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                     triadic = TRUE)
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                triadic = TRUE,
                                plot = TRUE)

  # Generate a trio of split complementary colors based on a given one.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                     split_complementary = TRUE)
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                split_complementary = TRUE,
                                plot = TRUE)

  # Generate a group of tetradic colors based on a given one.
  colors <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                     tetradic = TRUE)
  p <- SCpubr:::do_ColorPalette(colors.use = "steelblue",
                                tetradic = TRUE,
```

```

        plot = TRUE)

# Generate a group of square colors based on a given one.
colors <- SCpubr::do_ColorPalette(colors.use = "steelblue",
                                square = TRUE)
p <- SCpubr::do_ColorPalette(colors.use = "steelblue",
                             square = TRUE,
                             plot = TRUE)

# Retrieve the output of all options.
out <- SCpubr::do_ColorPalette(colors.use = "steelblue",
                              complete_output = TRUE)

## Retrieve the colors.
colors <- out$colors
## Retrieve the plots.
plots <- out$plots
## Retrieve a combined plot with all the options.
p <- out$combined_plot

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_CopyNumberVariantPlot

Display CNV scores from inferCNV as Feature Plots.

Description

Display CNV scores from inferCNV as Feature Plots.

Usage

```

do_CopyNumberVariantPlot(
  sample,
  infercnv_object,
  chromosome_locations,
  group.by = NULL,
  using_metacells = FALSE,
  metacell_mapping = NULL,
  chromosome_focus = NULL,
  legend.type = "colorbar",
  legend.position = "bottom",
  legend.length = 20,
  legend.width = 1,
  legend.framewidth = 0.5,
  legend.tickwidth = 0.5,
  legend.framecolor = "grey50",

```

```

legend.tickcolor = "white",
font.size = 14,
pt.size = 1,
font.type = "sans",
border.size = 2,
border.color = "black",
rotate_x_axis_labels = 45,
plot_cell_borders = TRUE,
enforce_symmetry = TRUE,
legend.title = NULL,
na.value = "grey75",
viridis_color_map = "G",
viridis_direction = 1,
verbose = FALSE,
min.cutoff = NULL,
max.cutoff = NULL
)

```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).

infercnv_object [infercnv](#) | Output inferCNV object run on the same Seurat object.

chromosome_locations [tibble](#) | Tibble containing the chromosome regions to use. Can be obtained using `utils::data("human_chr_locations", package = "SCpubr")`.

group.by [character](#) | Metadata variable to group the output by. Has to be a character of factor column.

using_metacells [logical](#) | Whether inferCNV was run using metacells or not.

metacell_mapping [named_vector](#) | Vector or cell - metacell mapping.

chromosome_focus [character](#) | Region stating which chromosome to plot. Eg: 1p, 19q. NULL will plot all regions.

legend.type [character](#) | Type of legend to display. One of:

- `normal`: Default legend displayed by **ggplot2**.
- `colorbar`: Redefined colorbar legend, using [guide_colorbar](#).
- `colorsteps`: Redefined legend with colors going by range, in steps, using [guide_colorsteps](#).

legend.position [character](#) | Position of the legend in the plot. One of:

- `top`: Top of the figure.
- `bottom`: Bottom of the figure.
- `left`: Left of the figure.
- `right`: Right of the figure.

- none: No legend is displayed.

legend.length, legend.width
numeric | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framewidth, legend.tickwidth
numeric | Width of the lines of the box in the legend.

legend.framecolor
character | Color of the lines of the box in the legend.

legend.tickcolor
character | Color of the ticks of the box in the legend.

font.size
numeric | Overall font size of the plot. All plot elements will have a size relationship with this font size.

pt.size
numeric | Size of the dots.

font.type
character | Base font family for the plot. One of:

- mono: Mono spaced font.
- serif: Serif font family.
- sans: Default font family.

border.size
numeric | Width of the border of the cells.

border.color
character | Color to use for the border of the cells.

rotate_x_axis_labels
numeric | Degree to rotate the X labels. One of: 0, 45, 90.

plot_cell_borders
logical | Whether to plot border around cells.

enforce_symmetry
logical | Return a symmetrical plot axes-wise or continuous color scale-wise, when applicable.

legend.title
character | Title for the legend.

na.value
character | Color value for NA.

viridis_color_map
character | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

viridis_direction
numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

verbose
logical | Whether to show extra comments, warnings, etc.

min.cutoff, max.cutoff
numeric | Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

Value

A list containing Feature Plots for different chromosome regions and corresponding dot plots by groups..

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_CopyNumberVariantPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # This function expects that you have run inferCNV on your
  # own and you have access to the output object.

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds",
                                package = "SCpubr"))

  # Define your inferCNV object.
  infercnv_object <- readRDS(system.file("extdata/infercnv_object_example.rds",
                                         package = "SCpubr"))

  # Get human chromosome locations.
  chromosome_locations = SCpubr::human_chr_locations

  # Compute for a single chromosome.
  out <- SCpubr::do_CopyNumberVariantPlot(sample = sample,
                                          infercnv_object = infercnv_object,
                                          using_metacells = FALSE,
                                          chromosome_locations = chromosome_locations,
                                          chromosome_focus = "1")

  # Retrieve the UMAP for 1p region.
  out$`1p_umap`
  # Retrieve the dot plot for 1p region.
  out$`1p_dotplot`

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_CorrelationPlot *Create correlation matrix heatmaps.*

Description

Create correlation matrix heatmaps.

Usage

```
do_CorrelationPlot(
```



```

sample,
mode = "hvg",
assay = NULL,
group.by = NULL,
column_title = "",
row_title = "",
cluster_cols = TRUE,
cluster_rows = TRUE,
legend.title = "Pearson coef.",
row_names_rot = 0,
column_names_rot = 0,
viridis_color_map = "G",
viridis_direction = 1,
cell_size = 8,
na.value = "grey75",
legend.position = "bottom",
legend.length = 75,
legend.width = 5,
legend.framecolor = "black"
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
mode	character Different types of correlation matrices can be computed. Right now, the only possible value is "hvg", standing for Highly Variable Genes. The sample is subset for the HVG and the data is re-scaled. Scale data is used for the correlation.
assay	character Assay to use. Defaults to the current assay.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
column_title	character Title for the columns of the heatmaps. Only works with single heatmaps.
row_title	character Title for the rows of the heatmaps. Only works with single heatmaps.
cluster_cols	logical Cluster the columns or rows of the heatmaps.
cluster_rows	logical Cluster the rows or rows of the heatmaps.
legend.title	character Title for the legend.
row_names_rot	numeric Degree in which to rotate the row labels.
column_names_rot	numeric Degree in which to rotate the column labels.
viridis_color_map	character A capital letter from A to H or the scale name as in scale_fill_viridis .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
cell_size	numeric Size of each cell in the heatmap.

na.value **character** | Color value for NA.
 legend.position **character** | Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

legend.length, legend.width **numeric** | Length and width of the legend. Will adjust automatically depending on legend side.
 legend.framecolor **character** | Color of the lines of the box in the legend.

Value

A ComplexHeatmap object.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_CorrelationPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Default values.
  p <- SCpubr::do_CorrelationPlot(sample = sample)
  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.`")
}
```

do_DimPlot

Wrapper for [DimPlot](#).

Description

Wrapper for [DimPlot](#).

Usage

```
do_DimPlot(  
  sample,  
  reduction = NULL,  
  group.by = NULL,  
  split.by = NULL,  
  colors.use = NULL,  
  shuffle = TRUE,  
  order = NULL,  
  pt.size = 1,  
  label = FALSE,  
  label.color = "white",  
  label.size = 4,  
  label.box = TRUE,  
  repel = FALSE,  
  cells.highlight = NULL,  
  idents.highlight = NULL,  
  idents.keep = NULL,  
  sizes.highlight = 1,  
  ncol = NULL,  
  plot.title = NULL,  
  plot.subtitle = NULL,  
  plot.caption = NULL,  
  legend.title = NULL,  
  legend.position = "bottom",  
  legend.title.position = "top",  
  legend.ncol = NULL,  
  legend.nrow = NULL,  
  legend.icon.size = 4,  
  legend.byrow = FALSE,  
  raster = FALSE,  
  raster.dpi = 1024,  
  dims = c(1, 2),  
  font.size = 14,  
  font.type = "sans",  
  na.value = "grey75",  
  plot_cell_borders = TRUE,  
  border.size = 2,  
  border.color = "black",  
  plot_marginal_distributions = FALSE,  
  marginal.type = "density",  
  marginal.size = 5,  
  marginal.group = TRUE,  
  plot.axes = FALSE,  
  plot_density_contour = FALSE,  
  contour.position = "bottom",  
  contour.color = "grey90",  
  contour.lineend = "butt",
```

```

    contour.linejoin = "round",
    contour_expand_axes = 0.25
  )

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
reduction	character Reduction to use. Can be the canonical ones such as "umap", "pca", or any custom ones, such as "diffusion". If you are unsure about which reductions you have, use <code>Seurat::Reductions(sample)</code> . Defaults to "umap" if present or to the last computed reduction if the argument is not provided.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
colors.use	named_vector Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of group.by. If group.by is not provided, defaults to the unique values of Idents . If not provided, a color scale will be set by default.
shuffle	logical Whether to shuffle the cells or not, so that they are not plotted cluster-wise. Recommended.
order	character Vector of identities to be plotted. Either one with all identities or just some, which will be plotted last.
pt.size	numeric Size of the dots.
label	logical Whether to plot the cluster labels in the UMAP. The cluster labels will have the same color as the cluster colors.
label.color	character Color of the labels in the plot.
label.size	numeric Size of the labels in the plot.
label.box	logical Whether to plot the plot labels as geom_text (FALSE) or geom_label (TRUE).
repel	logical Whether to repel the text labels.
cells.highlight, idents.highlight	character Vector of cells/identities to focus into. The identities have to match those in <code>Seurat::Idents(sample)</code> . The rest of the cells will be grayed out. Both parameters can be used at the same time.
idents.keep	character Vector of identities to keep. This will effectively set the rest of the cells that do not match the identities provided to NA, therefore coloring them according to na.value parameter.
sizes.highlight	numeric Point size of highlighted cells using cells.highlight parameter.
ncol	numeric Number of columns used in the arrangement of the output plot using "split.by" parameter.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.

`legend.title` **character** | Title for the legend.
`legend.position` **character** | Position of the legend in the plot. One of:

- `top`: Top of the figure.
- `bottom`: Bottom of the figure.
- `left`: Left of the figure.
- `right`: Right of the figure.
- `none`: No legend is displayed.

`legend.title.position` **character** | Position for the title of the legend. One of:

- `top`: Top of the legend.
- `bottom`: Bottom of the legend.
- `left`: Left of the legend.
- `right`: Right of the legend.

`legend.ncol` **numeric** | Number of columns in the legend.
`legend.nrow` **numeric** | Number of rows in the legend.
`legend.icon.size` **numeric** | Size of the icons in legend.
`legend.byrow` **logical** | Whether the legend is filled by row or not.
`raster` **logical** | Whether to raster the resulting plot. This is recommendable if plotting a lot of cells.
`raster.dpi` **numeric** | Pixel resolution for rasterized plots. Defaults to 1024. Only activates on Seurat versions higher or equal than 4.1.0.
`dims` **numeric** | Vector of 2 numerics indicating the dimensions to plot out of the selected reduction. Defaults to `c(1, 2)` if not specified.
`font.size` **numeric** | Overall font size of the plot. All plot elements will have a size relationship with this font size.
`font.type` **character** | Base font family for the plot. One of:

- `mono`: Mono spaced font.
- `serif`: Serif font family.
- `sans`: Default font family.

`na.value` **character** | Color value for NA.
`plot_cell_borders` **logical** | Whether to plot border around cells.
`border.size` **numeric** | Width of the border of the cells.
`border.color` **character** | Color to use for the border of the cells.
`plot_marginal_distributions` **logical** | Whether to plot marginal distributions on the figure or not.
`marginal.type` **character** | One of:

- `density`: Compute density plots on the margins.
- `histogram`: Compute histograms on the margins.

- `boxplot`: Compute boxplot on the margins.
- `violin`: Compute violin plots on the margins.
- `densigram`: Compute densigram plots on the margins.

`marginal.size` **numeric** | Size ratio between the main and marginal plots. A value of 5 means that the main plot is 5 times bigger than the marginal plots.

`marginal.group` **logical** | Whether to group the marginal distribution by `group.by` or current identities.

`plot.axes` **logical** | Whether to plot axes or not.

`plot_density_contour` **logical** | Whether to plot density contours in the UMAP.

`contour.position` **character** | Whether to plot density contours on top or at the bottom of the visualization layers, thus overlapping the clusters/cells or not.

`contour.color` **character** | Color of the density lines.

`contour.lineend` **character** | Line end style (round, butt, square).

`contour.linejoin` **character** | Line join style (round, mitre, bevel).

`contour_expand_axes` **numeric** | To make the contours fit the plot, the limits of the X and Y axis are expanding a given percentage from the min and max values for each axis. This controls such percentage.

Value

A `ggplot2` object containing a `DimPlot`.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_DimPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic DimPlot.
  p <- SCpubr::do_DimPlot(sample = sample)

  # Restrict the amount of identities displayed.
  p <- SCpubr::do_DimPlot(sample = sample,
                          idents.keep = c("1", "3", "5"))

  # Group by another variable rather than `Seurat::Idents(sample)`
```

```

p <- SCpubr::do_DimPlot(sample = sample,
                        group.by = "seurat_clusters")

# Split the output in as many plots as unique identities.
p <- SCpubr::do_DimPlot(sample = sample,
                        split.by = "seurat_clusters")

# Highlight given identities
p <- SCpubr::do_DimPlot(sample,
                        idents.highlight = c("1", "3"))

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_DotPlot

This function is a wrapper for [DotPlot](#). It provides most of its functionalities while adding extra. You can

Description

This function is a wrapper for [DotPlot](#). It provides most of its functionalities while adding extra. You can

Usage

```

do_DotPlot(
  sample,
  features,
  assay = NULL,
  group.by = NULL,
  split.by = NULL,
  legend.type = "colorbar",
  legend.position = "bottom",
  legend.framewidth = 0.5,
  legend.tickwidth = 0.5,
  legend.length = 20,
  legend.width = 1,
  legend.framecolor = "grey50",
  legend.tickcolor = "white",
  dot.scale = 6,
  colors.use = c("#1BFFFF25", "#2E3192"),
  plot.title = NULL,

```

```

plot.subtitle = NULL,
plot.caption = NULL,
xlab = NULL,
ylab = NULL,
font.size = 14,
font.type = "sans",
cluster.idents = FALSE,
flip = FALSE,
rotate_x_axis_labels = 45,
scale.by = "size",
use_viridis = FALSE,
viridis_color_map = "G",
viridis_direction = -1,
na.value = "grey75",
dot_border = TRUE,
plot.grid = TRUE,
grid.color = "grey75",
grid.type = "dashed"
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
features	character Features to represent.
assay	character Assay to use. Defaults to the current assay.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none"> • normal: Default legend displayed by ggplot2. • colorbar: Redefined colorbar legend, using guide_colorbar. • colorsteps: Redefined legend with colors going by range, in steps, using guide_colorsteps.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
legend.framewidth, legend.tickwidth	numeric Width of the lines of the box in the legend.
legend.length, legend.width	numeric Length and width of the legend. Will adjust automatically depending on legend side.

legend.framecolor	character Color of the lines of the box in the legend.
legend.tickcolor	character Color of the ticks of the box in the legend.
dot.scale	numeric Scale the size of the dots.
colors.use	character Two colors if split.by is not set, which will define a gradient. As many numbers as unique values in split.by, if set, which each own will define its own gradient. Defaults to predefined color scales if not provided.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
xlab, ylab	character Titles for the X and Y axis.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none">• mono: Mono spaced font.• serif: Serif font family.• sans: Default font family.
cluster.idents	logical Whether to cluster the identities based on the expression of the features.
flip	logical Whether to invert the axis of the displayed plot.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
scale.by	character How to scale the size of the dots. One of: <ul style="list-style-type: none">• radius: use radius aesthetic.• size: use size aesthetic.
use_viridis	logical Whether to use viridis color scales.
viridis_color_map	character A capital letter from A to H or the scale name as in scale_fill_viridis .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
na.value	character Color value for NA.
dot_border	logical Whether to plot a border around dots.
plot.grid	logical Whether to plot grid lines.
grid.color	character Color of the grid in the panels.
grid.type	character One of the possible linetype options: <ul style="list-style-type: none">• blank.• solid.• dashed.• dotted.• dotdash.• longdash.• twodash.

Value

A ggplot2 object containing a Dot Plot.

Examples

```
# Check Suggests.
value <- SCpubr::check_suggests(function_name = "do_DotPlot", passive = TRUE)

if (isTRUE(value)){
  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic Dot plot.
  p <- SCpubr::do_DotPlot(sample = sample,
                          features = "EPC1")

  # Querying multiple features.
  genes <- rownames(sample)[1:14]
  p <- SCpubr::do_DotPlot(sample = sample,
                          features = genes)

  # Inverting the axes.
  p <- SCpubr::do_DotPlot(sample = sample,
                          features = genes,
                          cluster.idents = TRUE,
                          plot.title = "Clustered",
                          flip = TRUE)

  # Modifying default colors.
  # Two colors to generate a gradient.
  p <- SCpubr::do_DotPlot(sample = sample,
                          features = genes,
                          colors.use = c("#001219", "#e9d8a6"))

  # Querying multiple features as a named list - splitting by each item in list.
  # Genes have to be unique.
  genes <- list("Naive CD4+ T" = rownames(sample)[1:2],
               "EPC1+ Mono" = rownames(sample)[3:4],
               "Memory CD4+" = rownames(sample)[5],
               "B" = rownames(sample)[6],
               "CD8+ T" = rownames(sample)[7],
               "FCGR3A+ Mono" = rownames(sample)[8:9],
               "NK" = rownames(sample)[10:11],
               "DC" = rownames(sample)[12:13],
               "Platelet" = rownames(sample)[14])

  p <- SCpubr::do_DotPlot(sample = sample,
                          features = genes)

  # Clustering the identities.
  p <- SCpubr::do_DotPlot(sample = sample,
```

```
        features = genes,
        cluster.idents = TRUE,
        plot.title = "Clustered")
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}
```

do_EnrichmentHeatmap *Create enrichment scores heatmaps.*

Description

This function computes the enrichment scores for the cells using [AddModuleScore](#) and then aggregates the scores by the metadata variables provided by the user and displays it as a heatmap, computed by [Heatmap](#).

Usage

```
do_EnrichmentHeatmap(
  sample,
  input_gene_list,
  assay = NULL,
  slot = NULL,
  reduction = NULL,
  group.by = NULL,
  verbose = FALSE,
  flip = FALSE,
  cluster_cols = FALSE,
  cluster_rows = FALSE,
  row_names_rot = 0,
  column_names_rot = 45,
  cell_size = 8,
  na.value = "grey75",
  legend.position = "bottom",
  use_viridis = TRUE,
  viridis_color_map = "G",
  viridis_direction = 1,
  heatmap.legend.length = 75,
  heatmap.legend.width = 5,
  heatmap.legend.framecolor = "black",
  legend.width = 1,
  legend.length = 20,
  legend.framewidth = 1.5,
  legend.tickwidth = 1.5,
  legend.framecolor = "grey50",
  legend.tickcolor = "white",
```

```

legend.type = "colorbar",
font.size = 14,
font.type = "sans",
rotate_x_axis_labels = 45,
enforce_symmetry = FALSE,
heatmap_gap = 0.5,
row_names_side = "right",
row_title_side = "left",
row_title_rot = 90,
column_title = NULL,
row_title = NULL,
nbin = 24,
ctrl = 100,
flavor = "Seurat",
legend.title = if (flavor != "AUCell") {
  "Enrichment"
} else {
  "AUC"
},
ncores = 1,
storeRanks = TRUE,
min.cutoff = NULL,
max.cutoff = NULL,
plot_FeaturePlots = FALSE,
plot_GeyserPlots = FALSE,
plot_BeeSwarmPlots = FALSE,
plot_BoxPlots = FALSE,
plot_ViolinPlots = FALSE,
pt.size = 1,
plot_cell_borders = TRUE,
border.size = 2,
geyser_order_by_mean = TRUE,
geyser_scale_type = "continuous",
boxplot_order_by_mean = TRUE,
violin_plot_boxplot = TRUE,
violin_boxplot_width = 0.2,
return_object = FALSE,
return_matrix = FALSE
)

```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).
input_gene_list [named_list](#) | Named list of lists of genes to be used as input.
assay [character](#) | Assay to use. Defaults to the current assay.
slot [character](#) | Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".

reduction	character Reduction to use. Can be the canonical ones such as "umap", "pca", or any custom ones, such as "diffusion". If you are unsure about which reductions you have, use <code>Seurat::Reductions(sample)</code> . Defaults to "umap" if present or to the last computed reduction if the argument is not provided.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
verbose	logical Whether to show extra comments, warnings, etc.
flip	logical Whether to invert the axis of the displayed plot.
cluster_cols	logical Cluster the columns or rows of the heatmaps.
cluster_rows	logical Cluster the rows or rows of the heatmaps.
row_names_rot	numeric Degree in which to rotate the row labels.
column_names_rot	numeric Degree in which to rotate the column labels.
cell_size	numeric Size of each cell in the heatmap.
na.value	character Color value for NA.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
use_viridis	logical Whether to use viridis color scales.
viridis_color_map	character A capital letter from A to H or the scale name as in <code>scale_fill_viridis</code> .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
heatmap.legend.length, heatmap.legend.width	numeric Width and length of the legend in the heatmap.
heatmap.legend.framecolor	character Color of the edges and ticks of the legend in the heatmap.
legend.length, legend.width	numeric Length and width of the legend. Will adjust automatically depending on legend side.
legend.framewidth, legend.tickwidth	numeric Width of the lines of the box in the legend.
legend.framecolor	character Color of the lines of the box in the legend.
legend.tickcolor	character Color of the ticks of the box in the legend.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none"> • normal: Default legend displayed by ggplot2.

	<ul style="list-style-type: none"> • <code>colorbar</code>: Redefined colorbar legend, using <code>guide_colorbar</code>. • <code>colorsteps</code>: Redefined legend with colors going by range, in steps, using <code>guide_colorsteps</code>.
<code>font.size</code>	<code>numeric</code> Overall font size of the plot. All plot elements will have a size relationship with this font size.
<code>font.type</code>	<code>character</code> Base font family for the plot. One of: <ul style="list-style-type: none"> • <code>mono</code>: Mono spaced font. • <code>serif</code>: Serif font family. • <code>sans</code>: Default font family.
<code>rotate_x_axis_labels</code>	<code>numeric</code> Degree to rotate the X labels. One of: 0, 45, 90.
<code>enforce_symmetry</code>	<code>logical</code> Whether the geyser and feature plot has a symmetrical color scale.
<code>heatmap_gap</code>	<code>numeric</code> Gap in cm between heatmaps.
<code>row_names_side</code>	<code>character</code> Side to put the row names. Either left or right.
<code>row_title_side</code>	<code>character</code> Side to put the row titles Either left or right.
<code>row_title_rot</code>	<code>numeric</code> Degree in which to rotate the row titles.
<code>column_title</code>	<code>character</code> Title for the columns of the heatmaps. Only works with single heatmaps.
<code>row_title</code>	<code>character</code> Title for the rows of the heatmaps. Only works with single heatmaps.
<code>nbin</code>	<code>numeric</code> Number of bins to use in <code>AddModuleScore</code> .
<code>ctrl</code>	<code>numeric</code> Number of genes in the control set to use in <code>AddModuleScore</code> .
<code>flavor</code>	<code>character</code> One of: <code>Seurat</code> , <code>UCell</code> . Compute the enrichment scores using <code>AddModuleScore</code> or <code>AddModuleScore_UCell</code> .
<code>legend.title</code>	<code>character</code> Title for the legend.
<code>ncores</code>	<code>numeric</code> Number of cores used to run UCell scoring.
<code>storeRanks</code>	<code>logical</code> Whether to store the ranks for faster UCell scoring computations. Might require large amounts of RAM.
<code>min.cutoff, max.cutoff</code>	<code>numeric</code> Set the min/max ends of the color scale. Any cell/group with a value lower than <code>min.cutoff</code> will turn into <code>min.cutoff</code> and any cell with a value higher than <code>max.cutoff</code> will turn into <code>max.cutoff</code> . In <code>FeaturePlots</code> , provide as many values as features. Use NAs to skip a feature.
<code>plot_FeaturePlots, plot_GeyserPlots, plot_BeeSwarmPlots, plot_BoxPlots, plot_ViolinPlots</code>	<code>logical</code> Compute extra visualizations for each of the gene lists.
<code>pt.size</code>	<code>numeric</code> Size of the dots.
<code>plot_cell_borders</code>	<code>logical</code> Whether to plot border around cells.
<code>border.size</code>	<code>numeric</code> Width of the border of the cells.
<code>geyser_order_by_mean, boxplot_order_by_mean</code>	<code>logical</code> Whether to order the X axis by the mean of the values.

geyser_scale_type **character** | Type of scale to use. Either "continuous" or "categorical."
 violin_plot_boxplot **logical** | Whether to plot the boxplots inside the violin plots.
 violin_boxplot_width **numeric** | Width of the boxplots in the violin plots.
 return_object **logical** | Return the Seurat object with the enrichment scores stored.
 return_matrix **logical** | Return the enrichment matrix used for the heatmaps for each value in group.by.

Value

A ComplexHeatmap object.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_EnrichmentHeatmap", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Genes have to be unique.
  genes <- list("Naive CD4+ T" = rownames(sample)[1:2],
               "EPC1+ Mono" = rownames(sample)[3:4],
               "Memory CD4+" = rownames(sample)[5],
               "B" = rownames(sample)[6],
               "CD8+ T" = rownames(sample)[7],
               "FCGR3A+ Mono" = rownames(sample)[8:9],
               "NK" = rownames(sample)[10:11],
               "DC" = rownames(sample)[12:13],
               "Platelet" = rownames(sample)[14])

  # Default parameters.
  p <- SCpubr::do_EnrichmentHeatmap(sample = sample,
                                   input_gene_list = genes,
                                   nbin = 1,
                                   ctrl = 10)

  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`".)
}
```

do_ExpressionHeatmap *Create heatmaps of averaged expression by groups.*

Description

This function generates a heatmap with averaged expression values by the unique groups of the metadata variables provided by the user.

Usage

```
do_ExpressionHeatmap(  
  sample,  
  features,  
  group.by = NULL,  
  assay = NULL,  
  slot = "data",  
  flip = FALSE,  
  column_title = NULL,  
  row_title = NULL,  
  cluster_cols = FALSE,  
  cluster_rows = FALSE,  
  legend.title = "Avg. Expression",  
  row_names_rot = 0,  
  column_names_rot = 45,  
  cell_size = 8,  
  na.value = "grey75",  
  legend.position = "bottom",  
  use_viridis = TRUE,  
  viridis_color_map = "G",  
  viridis_direction = 1,  
  heatmap.legend.length = 75,  
  heatmap.legend.width = 5,  
  heatmap.legend.framecolor = "black",  
  rotate_x_axis_labels = 45,  
  enforce_symmetry = FALSE,  
  heatmap_gap = 0.5,  
  row_names_side = "right",  
  row_title_side = "left",  
  row_title_rot = 90,  
  min.cutoff = NULL,  
  max.cutoff = NULL  
)
```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).
features [character](#) | Features to represent.

group.by	character Metadata variable to group the output by. Has to be a character of factor column.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
flip	logical Whether to invert the axis of the displayed plot.
column_title	character Title for the columns of the heatmaps. Only works with single heatmaps.
row_title	character Title for the rows of the heatmaps. Only works with single heatmaps.
cluster_cols	logical Cluster the columns or rows of the heatmaps.
cluster_rows	logical Cluster the rows or rows of the heatmaps.
legend.title	character Title for the legend.
row_names_rot	numeric Degree in which to rotate the row labels.
column_names_rot	numeric Degree in which to rotate the column labels.
cell_size	numeric Size of each cell in the heatmap.
na.value	character Color value for NA.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
use_viridis	logical Whether to use viridis color scales.
viridis_color_map	character A capital letter from A to H or the scale name as in scale_fill_viridis .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
heatmap.legend.length, heatmap.legend.width	numeric Width and length of the legend in the heatmap.
heatmap.legend.framecolor	character Color of the edges and ticks of the legend in the heatmap.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
enforce_symmetry	logical Return a symmetrical plot axes-wise or continuous color scale-wise, when applicable.
heatmap_gap	numeric Gap in cm between heatmaps.
row_names_side	character Side to put the row names. Either left or right.
row_title_side	character Side to put the row titles Either left or right.

row_title_rot **numeric** | Degree in which to rotate the row titles.

min.cutoff, max.cutoff **numeric** | Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

Value

A ComplexHeatmap object.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_ExpressionHeatmap", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Define list of genes.
  genes <- rownames(sample)[1:10]

  # Default parameters.
  p <- SCpubr::do_ExpressionHeatmap(sample = sample,
                                    features = genes,
                                    viridis_direction = -1)
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}
```

do_FeaturePlot

Wrapper for FeaturePlot.

Description

Wrapper for [FeaturePlot](#).

Usage

```
do_FeaturePlot(
  sample,
  features,
  assay = NULL,
```

```
reduction = NULL,
slot = NULL,
order = FALSE,
split.by = NULL,
split.by.idents = NULL,
cells.highlight = NULL,
idents.highlight = NULL,
dims = c(1, 2),
enforce_symmetry = FALSE,
pt.size = 1,
font.size = 14,
font.type = "sans",
legend.title = NULL,
legend.type = "colorbar",
legend.position = "bottom",
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.length = 20,
legend.width = 1,
legend.framecolor = "grey50",
legend.tickcolor = "white",
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
individual.titles = NULL,
individual.subtitles = NULL,
individual.captions = NULL,
ncol = NULL,
viridis_color_map = "G",
viridis_direction = 1,
raster = FALSE,
raster.dpi = 1024,
plot_cell_borders = TRUE,
border.size = 2,
border.color = "black",
na.value = "grey75",
verbose = TRUE,
plot.axes = FALSE,
min.cutoff = rep(NA, length(features)),
max.cutoff = rep(NA, length(features)),
plot_density_contour = FALSE,
contour.position = "bottom",
contour.color = "grey90",
contour.lineend = "butt",
contour.linejoin = "round",
contour_expand_axes = 0.25,
label = FALSE,
label.color = "black",
```

```
  label.size = 4
)
```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
features	character Features to represent.
assay	character Assay to use. Defaults to the current assay.
reduction	character Reduction to use. Can be the canonical ones such as "umap", "pca", or any custom ones, such as "diffusion". If you are unsure about which reductions you have, use <code>Seurat::Reductions(sample)</code> . Defaults to "umap" if present or to the last computed reduction if the argument is not provided.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
order	logical Whether to order the cells based on expression.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
split.by.idents	character Vector of identities to plot. The gradient scale will also be subset to only the values of such identities.
cells.highlight, idents.highlight	character Vector of cells/identities to focus into. The identities have to match those in <code>Seurat::Idents(sample)</code> . The rest of the cells will be grayed out. Both parameters can be used at the same time.
dims	numeric Vector of 2 numerics indicating the dimensions to plot out of the selected reduction. Defaults to <code>c(1, 2)</code> if not specified.
enforce_symmetry	logical Return a symmetrical plot axes-wise or continuous color scale-wise, when applicable.
pt.size	numeric Size of the dots.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
legend.title	character Title for the legend.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none"> • normal: Default legend displayed by ggplot2. • colorbar: Redefined colorbar legend, using guide_colorbar. • colorsteps: Redefined legend with colors going by range, in steps, using guide_colorsteps.
legend.position	character Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

legend.framewidth, legend.tickwidth
numeric | Width of the lines of the box in the legend.

legend.length, legend.width
numeric | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framecolor
character | Color of the lines of the box in the legend.

legend.tickcolor
character | Color of the ticks of the box in the legend.

plot.title, plot.subtitle, plot.caption
character | Title, subtitle or caption to use in the plot.

individual.titles, individual.subtitles, individual.captions
character | Titles or subtitles. for each feature if needed. Either NULL or a vector of equal length of features.

ncol
numeric | Number of columns used in the arrangement of the output plot using "split.by" parameter.

viridis_color_map
character | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

viridis_direction
numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

raster
logical | Whether to raster the resulting plot. This is recommendable if plotting a lot of cells.

raster.dpi
numeric | Pixel resolution for rasterized plots. Defaults to 1024. Only activates on Seurat versions higher or equal than 4.1.0.

plot_cell_borders
logical | Whether to plot border around cells.

border.size
numeric | Width of the border of the cells.

border.color
character | Color to use for the border of the cells.

na.value
character | Color value for NA.

verbose
logical | Whether to show extra comments, warnings, etc.

plot.axes
logical | Whether to plot axes or not.

min.cutoff, max.cutoff
numeric | Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

plot_density_contour
logical | Whether to plot density contours in the UMAP.


```
} else if (isFALSE(value)){  
  message("This function can not be used without its suggested packages.")  
  message("Check out which ones are needed using `SCpubr::state_dependencies()`".)  
}
```

do_FunctionalAnnotationPlot

Compute functional annotation plots using GO or KEGG ontologies

Description

Compute functional annotation plots using GO or KEGG ontologies

Usage

```
do_FunctionalAnnotationPlot(  
  genes,  
  org.db,  
  organism = "hsa",  
  database = "GO",  
  GO_ontology = "BP",  
  min.overlap = if (length(genes) <= 4) {  
    1  
  } else {  
    3  
  },  
  p.adjust.cutoff = 0.05,  
  pAdjustMethod = "BH",  
  minGSSize = 10,  
  maxGSSize = 500,  
  cluster_cols = TRUE,  
  cluster_rows = TRUE,  
  cell_size = 8,  
  heatmap_gap = 0.5,  
  font.size = 10,  
  font.type = "sans",  
  rotate_x_axis_labels = 45,  
  xlab = NULL,  
  ylab = NULL,  
  plot.title = NULL,  
  plot.subtitle = NULL,  
  plot.caption = NULL,  
  plot.grid = TRUE,  
  grid.color = "grey75",  
  grid.type = "dashed",  
  flip = TRUE,
```

```

legend.type = "colorbar",
legend.position = "bottom",
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.length = 20,
legend.width = 1,
legend.framecolor = "grey50",
legend.tickcolor = "white",
heatmap.legend.length = 75,
heatmap.legend.width = 5,
heatmap.legend.framecolor = "black",
viridis_color_map = "G",
viridis_direction = -1,
showCategory = 30,
nWords = 4,
nCluster = 5
)

```

Arguments

genes	character Vector of gene symbols to query for functional annotation.
org.db	OrgDB Database object to use for the query.
organism	character Supported KEGG organism.
database	character Database to run the analysis on. One of: <ul style="list-style-type: none"> • GO. • KEGG.
GO_ontology	character GO ontology to use. One of: <ul style="list-style-type: none"> • BP: For Biological Process. • MF: For Molecular Function. • CC: For Cellular Component.
min.overlap	numeric Filter the output result to the terms which are supported by this many genes.
p.adjust.cutoff	numeric Significance cutoff used to filter non-significant terms.
pAdjustMethod	character Method to adjust for multiple testing. One of: <ul style="list-style-type: none"> • holm. • hochberg. • hommel. • bonferroni. • BH. • BY. • fdr. • none.
minGSSize	numeric Minimal size of genes annotated by Ontology term for testing.

maxGSSize	numeric Maximal size of genes annotated for testing.
cluster_cols	logical Cluster the columns or rows of the heatmaps.
cluster_rows	logical Cluster the rows or rows of the heatmaps.
cell_size	numeric Size of each cell in the heatmap.
heatmap_gap	numeric Gap in cm between heatmaps.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none">• mono: Mono spaced font.• serif: Serif font family.• sans: Default font family.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
xlab, ylab	character Titles for the X and Y axis.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
plot.grid	logical Whether to plot grid lines.
grid.color	character Color of the grid in the panels.
grid.type	character One of the possible linetype options: <ul style="list-style-type: none">• blank.• solid.• dashed.• dotted.• dotdash.• longdash.• twodash.
flip	logical Whether to invert the axis of the displayed plot.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none">• normal: Default legend displayed by ggplot2.• colorbar: Redefined colorbar legend, using guide_colorbar.• colorsteps: Redefined legend with colors going by range, in steps, using guide_colorsteps.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none">• top: Top of the figure.• bottom: Bottom of the figure.• left: Left of the figure.• right: Right of the figure.• none: No legend is displayed.
legend.framewidth, legend.tickwidth	numeric Width of the lines of the box in the legend.

`legend.length`, `legend.width`
 numeric | Length and width of the legend. Will adjust automatically depending on legend side.

`legend.framecolor`
 character | Color of the lines of the box in the legend.

`legend.tickcolor`
 character | Color of the ticks of the box in the legend.

`heatmap.legend.length`, `heatmap.legend.width`
 numeric | Width and length of the legend in the heatmap.

`heatmap.legend.framecolor`
 character | Color of the edges and ticks of the legend in the heatmap.

`viridis_color_map`
 character | A capital letter from A to H or the scale name as in `scale_fill_viridis`.

`viridis_direction`
 numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

`showCategory` **numeric** | Number of enriched terms to display in the output tree plot.

`nWords` **numeric** | The number of words in the cluster tags in the tree plot.

`nCluster` **numeric** | The number of clusters to group the resulting terms in the tree plot. Suggested value is above 2, as two can lead to some errors.

Value

A list containing a heatmap of the presence/absence of the genes in the enriched term, as well as a bar plot, dot plot and tree plot of the enriched terms.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_FunctionalAnnotationPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Need to load this library or equivalent.
  suppressMessages(library("org.Hs.eg.db"))

  # Define list of genes to query.
  genes.use <- c("CCR7", "CD14", "LYZ",
                "S100A4", "MS4A1",
                "MS4A7", "GNLY", "NKG7", "FCER1A",
                "CST3", "PPBP")

  # Compute the grouped GO terms.
  out <- SCpubr::do_FunctionalAnnotationPlot(genes = genes.use,
                                           org.db = org.Hs.eg.db)

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
}

```

```
    message("Check out which ones are needed using `SCpubr::state_dependencies()`.")  
  }
```

`do_GeyserPlot`*Generate a Geyser plot.*

Description

A Geyser plot is a custom plot in which we plot continuous values on the Y axis grouped by a categorical value in the X. This is plotted as a dot plot, jittered so that the dots span all the way to the other groups. On top of this, the mean and .66 and .95 of the data is plotted, depicting the overall distribution of the dots. The cells can, then, be colored by a continuous variable (same as Y axis or different) or a categorical one (same as X axis or different).

Usage

```
do_GeyserPlot(  
  sample,  
  features,  
  assay = NULL,  
  slot = "data",  
  group.by = NULL,  
  split.by = NULL,  
  enforce_symmetry = FALSE,  
  scale_type = "continuous",  
  order_by_mean = TRUE,  
  plot_cell_borders = TRUE,  
  jitter = 0.45,  
  pt.size = 1,  
  border.size = 2,  
  border.color = "black",  
  legend.position = "bottom",  
  legend.width = 1,  
  legend.length = 20,  
  legend.framewidth = 0.5,  
  legend.tickwidth = 0.5,  
  legend.framecolor = "grey50",  
  legend.tickcolor = "white",  
  legend.type = "colorbar",  
  font.size = 14,  
  font.type = "sans",  
  rotate_x_axis_labels = 45,  
  viridis_color_map = "G",  
  viridis_direction = 1,  
  colors.use = NULL,  
  na.value = "grey75",
```

```

legend.ncol = NULL,
legend.nrow = NULL,
legend.icon.size = 4,
legend.byrow = FALSE,
legend.title = NULL,
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
xlab = "Groups",
ylab = feature,
flip = FALSE,
min.cutoff = NULL,
max.cutoff = NULL
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
features	character Features to represent.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
enforce_symmetry	logical Return a symmetrical plot axes-wise or continuous color scale-wise, when applicable.
scale_type	character Type of color scale to use. One of: <ul style="list-style-type: none"> • categorical: Use a categorical color scale based on the values of "group.by". • continuous: Use a continuous color scale based on the values of "feature".
order_by_mean	logical Whether to order the groups by the mean of the data (highest to lowest).
plot_cell_borders	logical Whether to plot border around cells.
jitter	numeric Amount of jitter in the plot along the X axis. The lower the value, the more compacted the dots are.
pt.size	numeric Size of the dots.
border.size	numeric Width of the border of the cells.
border.color	character Color to use for the border of the cells.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure.

- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

legend.length, legend.width
numeric | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framewidth, legend.tickwidth
numeric | Width of the lines of the box in the legend.

legend.framecolor
character | Color of the lines of the box in the legend.

legend.tickcolor
character | Color of the ticks of the box in the legend.

legend.type
character | Type of legend to display. One of:

- normal: Default legend displayed by **ggplot2**.
- colorbar: Redefined colorbar legend, using [guide_colorbar](#).
- colorsteps: Redefined legend with colors going by range, in steps, using [guide_colorsteps](#).

font.size
numeric | Overall font size of the plot. All plot elements will have a size relationship with this font size.

font.type
character | Base font family for the plot. One of:

- mono: Mono spaced font.
- serif: Serif font family.
- sans: Default font family.

rotate_x_axis_labels
numeric | Degree to rotate the X labels. One of: 0, 45, 90.

viridis_color_map
character | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

viridis_direction
numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

colors.use
character | Named vector of colors to use. Has to match the unique values of group.by when scale_type is set to categorical.

na.value
character | Color value for NA.

legend.ncol
numeric | Number of columns in the legend.

legend.nrow
numeric | Number of rows in the legend.

legend.icon.size
numeric | Size of the icons in legend.

legend.byrow
logical | Whether the legend is filled by row or not.

legend.title
character | Title for the legend.

plot.title, plot.subtitle, plot.caption
character | Title, subtitle or caption to use in the plot.

xlab, ylab
character | Titles for the X and Y axis.

`flip` **logical** | Whether to invert the axis of the displayed plot.

`min.cutoff, max.cutoff` **numeric** | Set the min/max ends of the color scale. Any cell/group with a value lower than `min.cutoff` will turn into `min.cutoff` and any cell with a value higher than `max.cutoff` will turn into `max.cutoff`. In `FeaturePlots`, provide as many values as features. Use NAs to skip a feature.

Details

Special thanks to Christina Blume for coming up with the name of the plot.

Value

Either a plot of a list of plots, depending on the number of features provided.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_GeyserPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Geyser plot with categorical color scale.
  p <- SCpubr::do_GeyserPlot(sample = sample,
                             features = "nCount_RNA",
                             scale_type = "categorical")

  p

  # Geyser plot with continuous color scale.
  p <- SCpubr::do_GeyserPlot(sample = sample,
                             features = "nCount_RNA",
                             scale_type = "continuous")

  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}
```

do_GroupedGOTermPlot *Compute an overview of the GO terms associated with the input list of genes.*

Description

Compute an overview of the GO terms associated with the input list of genes.

Usage

```
do_GroupedGOTermPlot(  
  genes,  
  org.db,  
  levels.use = NULL,  
  GO_ontology = "BP",  
  min.overlap = NULL,  
  flip = TRUE,  
  legend.position = "right",  
  heatmap_gap = 0.5,  
  cluster_rows = TRUE,  
  cluster_cols = TRUE,  
  cell_size = 8,  
  reverse.levels = TRUE,  
  colors.use = c("grey90", "#29353d"),  
  rotate_x_axis_labels = 45,  
  font.size = 10,  
  verbose = FALSE  
)
```

Arguments

genes	character Vector of gene symbols to query for functional annotation.
org.db	OrgDB Database object to use for the query.
levels.use	numeric Vector of numerics corresponding to the GO ontology levels to plot. If NULL will compute all recursively until there are no results.
GO_ontology	character GO ontology to use. One of: <ul style="list-style-type: none">• BP: For Biological Process.• MF: For Molecular Function.• CC: For Cellular Component.
min.overlap	numeric Filter the output result to the terms which are supported by this many genes.
flip	logical Whether to invert the axis of the displayed plot.
legend.position	character Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

heatmap_gap	numeric	Gap in cm between heatmaps.
cluster_rows	logical	Cluster the rows or rows of the heatmaps.
cluster_cols	logical	Cluster the columns or rows of the heatmaps.
cell_size	numeric	Size of each cell in the heatmap.
reverse.levels	logical	Whether to place the higher levels first when computing the joint heatmap.
colors.use	character	Vector of 2 colors to use in the heatmap. The first will correspond to the empty values and the second one to the genes present in the terms.
rotate_x_axis_labels		
	numeric	Degree to rotate the X labels. One of: 0, 45, 90.
font.size	numeric	Overall font size of the plot. All plot elements will have a size relationship with this font size.
verbose	logical	Whether to show extra comments, warnings, etc.

Value

A list containing all the matrices for the respective GO levels and all the individual and combined heatmaps.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_GroupedGOTermPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Need to load this library or equivalent.
  suppressMessages(library("org.Hs.eg.db"))

  # Define list of genes to query.
  genes.use <- c("CCR7", "CD14", "LYZ",
                "S100A4", "MS4A1",
                "MS4A7", "GNLY", "NKG7", "FCER1A",
                "CST3", "PPBP")

  # Compute the grouped GO terms.
  out <- SCpubr::do_GroupedGOTermPlot(genes = genes.use,
                                     org.db = org.Hs.eg.db)
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
}
```



```

    message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
  }

```

do_GroupwiseDEPlot *Compute a heatmap with the results of a group-wise DE analysis.*

Description

Compute a heatmap with the results of a group-wise DE analysis.

Usage

```

do_GroupwiseDEPlot(
  sample,
  de_genes,
  group.by = NULL,
  viridis_map_pvalues = "B",
  viridis_map_logfc = "D",
  viridis_map_expression = "G",
  heatmap.legend.length = 75,
  heatmap.legend.width = 5,
  heatmap.legend.framecolor = "black",
  top_genes = 5,
  viridis_direction = -1,
  row_title_p_values = "",
  row_title_logfc = "Clusters",
  row_title_expression = if (is.null(group.by)) {
    ""
  } else {
    rep("",
      length(group.by))
  },
  column_title = "DE genes",
  heatmap_gap = 0.5,
  legend_gap = 1,
  assay = NULL,
  slot = "data",
  legend.position = "bottom",
  row_names_side = "right",
  row_title_side = "left",
  row_title_rot = 90,
  column_names_rot = 45,
  cell_size = 6,
  min.cutoff = NULL,
  max.cutoff = NULL
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
de_genes	tibble DE genes matrix resulting of running <code>Seurat::FindAllMarkers()</code> .
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
viridis_map_pvalues, viridis_map_logfc, viridis_map_expression	character Viridis color map for the heatmap of p-values, logFC or expression. One of A, B, C, D, E, F, G, H.
heatmap.legend.length, heatmap.legend.width	numeric Width and length of the legend in the heatmap.
heatmap.legend.framecolor	character Color of the edges and ticks of the legend in the heatmap.
top_genes	numeric Top N differentially expressed (DE) genes by group to retrieve.
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
row_title_p_values	character Row title for the p-value heatmap. Blank by default.
row_title_logfc	character Row title for the logfc heatmap. Clusters by default.
row_title_expression	character Vector of titles of equal length as group.by.
column_title	character Title for the columns of the heatmaps. Only works with single heatmaps.
heatmap_gap	numeric Gap in cm between heatmaps.
legend_gap	numeric Gap in cm between legends.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
row_names_side	character Side to put the row names. Either left or right.
row_title_side	character Side to put the row titles Either left or right.
row_title_rot	numeric Degree in which to rotate the row titles.
column_names_rot	numeric Degree in which to rotate the column labels.
cell_size	numeric Size of each cell in the heatmap.

min.cutoff, max.cutoff

numeric | Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

Value

A heatmap composed of 3 main panels: $-\log_{10}(\text{adjusted p-value})$, $\log_2(\text{FC})$ and mean expression by cluster.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_GroupwiseDEPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Compute DE genes and transform to a tibble.
  de_genes <- readRDS(system.file("extdata/de_genes_example.rds", package = "SCpubr"))

  # Default output.
  p <- SCpubr::do_GroupwiseDEPlot(sample = sample,
                                  de_genes = de_genes)

  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}
```

do_NebulosaPlot

Wrapper for Nebulosa::plot_density in Seurat.

Description

Wrapper for Nebulosa::plot_density in Seurat.

Usage

```
do_NebulosaPlot(
  sample,
```

```

features,
slot = NULL,
dims = c(1, 2),
pt.size = 1,
reduction = NULL,
combine = TRUE,
method = c("ks", "wkde"),
joint = FALSE,
return_only_joint = FALSE,
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
legend.type = "colorbar",
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.length = 20,
legend.width = 1,
legend.framecolor = "grey50",
legend.tickcolor = "white",
font.size = 14,
font.type = "sans",
legend.position = "bottom",
plot_cell_borders = TRUE,
border.size = 2,
border.color = "black",
viridis_color_map = "G",
viridis_direction = 1,
verbose = TRUE,
na.value = "grey75",
plot.axes = FALSE
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
features	character Features to represent.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
dims	numeric Vector of 2 numerics indicating the dimensions to plot out of the selected reduction. Defaults to c(1, 2) if not specified.
pt.size	numeric Size of the dots.
reduction	character Reduction to use. Can be the canonical ones such as "umap", "pca", or any custom ones, such as "diffusion". If you are unsure about which reductions you have, use <code>Seurat::Reductions(sample)</code> . Defaults to "umap" if present or to the last computed reduction if the argument is not provided.
combine	logical Whether to create a single plot out of multiple features.
method	Kernel density estimation method:

- ks: Computes density using the kde function from the ks package.
- wkde: Computes density using a modified version of the kde2d function from the MASS package to allow weights. Bandwidth selection from the ks package is used instead.

joint **logical** | Whether to plot different features as joint density.

return_only_joint **logical** | Whether to only return the joint density panel.

plot.title, plot.subtitle, plot.caption **character** | Title, subtitle or caption to use in the plot.

legend.type **character** | Type of legend to display. One of:

- normal: Default legend displayed by **ggplot2**.
- colorbar: Redefined colorbar legend, using [guide_colorbar](#).
- colorsteps: Redefined legend with colors going by range, in steps, using [guide_colorsteps](#).

legend.framewidth, legend.tickwidth **numeric** | Width of the lines of the box in the legend.

legend.length, legend.width **numeric** | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framecolor **character** | Color of the lines of the box in the legend.

legend.tickcolor **character** | Color of the ticks of the box in the legend.

font.size **numeric** | Overall font size of the plot. All plot elements will have a size relationship with this font size.

font.type **character** | Base font family for the plot. One of:

- mono: Mono spaced font.
- serif: Serif font family.
- sans: Default font family.

legend.position **character** | Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

plot_cell_borders **logical** | Whether to plot border around cells.

border.size **numeric** | Width of the border of the cells.

border.color **character** | Color to use for the border of the cells.

viridis_color_map **character** | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

viridis_direction **numeric** | Either 1 or -1. Controls how the gradient of viridis scale is formed.

verbose **logical** | Whether to show extra comments, warnings, etc.

na.value **character** | Color value for NA.

plot.axes **logical** | Whether to plot axes or not.

Value

A ggplot2 object containing a Nebulosa plot.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_NebulosaPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic Nebulosa plot.
  p <- SCpubr::do_NebulosaPlot(sample = sample,
                               features = "EPC1")

  # Compute joint density.
  p <- SCpubr::do_NebulosaPlot(sample = sample,
                               features = c("EPC1", "TOX2"),
                               joint = TRUE)

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}
```

do_PathwayActivityPlot

Plot Pathway Activities from decoupleR using Progeny prior knowledge.

Description

Plot Pathway Activities from decoupleR using Progeny prior knowledge.

Usage

```
do_PathwayActivityPlot(  
  sample,  
  activities,  
  group.by = NULL,  
  split.by = NULL,  
  plot_FeaturePlots = FALSE,  
  plot_Heatmaps = TRUE,  
  plot_GeyserPlots = FALSE,  
  row_title = NULL,  
  column_title = NULL,  
  flip = FALSE,  
  cluster_cols = TRUE,  
  cluster_rows = TRUE,  
  row_names_rot = 0,  
  column_names_rot = 45,  
  cell_size = 8,  
  pt.size = 1,  
  plot_cell_borders = TRUE,  
  border.size = 2,  
  na.value = "grey75",  
  legend.position = "bottom",  
  heatmap.legend.length = 75,  
  heatmap.legend.width = 5,  
  heatmap.legend.framecolor = "black",  
  legend.width = 1,  
  legend.length = 20,  
  legend.framewidth = 0.5,  
  legend.tickwidth = 0.5,  
  legend.framecolor = "grey50",  
  legend.tickcolor = "white",  
  legend.type = "colorbar",  
  font.size = 14,  
  font.type = "sans",  
  rotate_x_axis_labels = 45,  
  enforce_symmetry = TRUE,  
  geyser_order_by_mean = TRUE,  
  geyser_scale_type = "continuous",  
  viridis_color_map = "G",  
  viridis_direction = 1,  
  min.cutoff = NULL,  
  max.cutoff = NULL  
)
```

Arguments

sample [Seurat](#) | A Seurat object, generated by [CreateSeuratObject](#).
activities [tibble](#) | Result of running decoupleR method with progeny regulon prior knowl-

edge.

group.by **character** | Metadata variable to group the output by. Has to be a character of factor column.

split.by **character** | Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.

plot_FeaturePlots **logical** | Compute output FeaturePlots for each of the top regulons.

plot_Heatmaps **logical** | Compute output heatmap showcasing the average TF activity per regulon and group.by variable.

plot_GeyserPlots **logical** | Compute output GeyserPlots for each of the top regulons and group.by variable.

row_title **character** | Title for the rows of the heatmaps. Only works with single heatmaps.

column_title **character** | Title for the columns of the heatmaps. Only works with single heatmaps.

flip **logical** | Whether to invert the axis of the displayed plot.

cluster_cols **logical** | Cluster the columns or rows of the heatmaps.

cluster_rows **logical** | Cluster the rows or rows of the heatmaps.

row_names_rot **numeric** | Degree in which to rotate the row labels.

column_names_rot **numeric** | Degree in which to rotate the column labels.

cell_size **numeric** | Size of each cell in the heatmap.

pt.size **numeric** | Size of the dots.

plot_cell_borders **logical** | Whether to plot border around cells.

border.size **numeric** | Width of the border of the cells.

na.value **character** | Color value for NA.

legend.position **character** | Position of the legend in the plot. One of:

- top: Top of the figure.
- bottom: Bottom of the figure.
- left: Left of the figure.
- right: Right of the figure.
- none: No legend is displayed.

heatmap.legend.length, heatmap.legend.width **numeric** | Width and length of the legend in the heatmap.

heatmap.legend.framecolor **character** | Color of the edges and ticks of the legend in the heatmap.

legend.length, legend.width **numeric** | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framewidth, legend.tickwidth **numeric** | Width of the lines of the box in the legend.

legend.framecolor	character Color of the lines of the box in the legend.
legend.tickcolor	character Color of the ticks of the box in the legend.
legend.type	character Type of legend to display. One of: <ul style="list-style-type: none"> • normal: Default legend displayed by ggplot2. • colorbar: Redefined colorbar legend, using guide_colorbar. • colorsteps: Redefined legend with colors going by range, in steps, using guide_colorsteps.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
enforce_symmetry	logical Return a symmetrical plot axes-wise or continuous color scale-wise, when applicable.
geyser_order_by_mean	logical Whether to order the X axis by the mean of the values.
geyser_scale_type	character Type of scale to use. Either "continuous" or "categorical."
viridis_color_map	character A capital letter from A to H or the scale name as in scale_fill_viridis .
viridis_direction	numeric Either 1 or -1. Controls how the gradient of viridis scale is formed.
min.cutoff, max.cutoff	numeric Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

Value

A list containing several output plots according to the user's specifications.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_PathwayActivityPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/
}
```

```

# Define your Seurat object.
sample <- readRDS(system.file("extdata/seurat_dataset_example.rds",
                             package = "SCpubr"))

# Define your activities object.
progeny_activities <- readRDS(system.file("extdata/progeny_activities_example.rds",
                                         package = "SCpubr"))

# General heatmap.
out <- SCpubr::do_PathwayActivityPlot(sample = sample,
                                     activities = progeny_activities,
                                     plot_FeaturePlots = TRUE,
                                     plot_GeyserPlots = TRUE)

p <- out$heatmaps$saverage_scores
p

# Retrieve feature plots.
p <- out$feature_plots$EGFR
p

# Retrieve Geyser plots.
p <- out$geyser_plots$EGFR
p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_RidgePlot

Create ridge plots.

Description

This function computes ridge plots based on the **ggridges** package.

Usage

```

do_RidgePlot(
  sample,
  feature,
  group.by = NULL,
  split.by = NULL,
  assay = "SCT",
  slot = "data",
  continuous_scale = FALSE,
  legend.title = NULL,

```

```

legend.ncol = NULL,
legend.nrow = NULL,
legend.byrow = FALSE,
legend.position = NULL,
legend.width = 1,
legend.length = 20,
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.framecolor = "grey50",
legend.tickcolor = "white",
legend.type = "colorbar",
colors.use = NULL,
font.size = 14,
font.type = "sans",
rotate_x_axis_labels = 45,
plot.title = NULL,
plot.subtitle = NULL,
plot.caption = NULL,
xlab = NULL,
ylab = NULL,
compute_quantiles = FALSE,
compute_custom_quantiles = FALSE,
quantiles = c(0.25, 0.5, 0.75),
compute_distribution_tails = FALSE,
prob_tails = 0.025,
color_by_probabilities = FALSE,
viridis_color_map = "G",
viridis_direction = 1,
plot.grid = TRUE,
grid.color = "grey75",
grid.type = "dashed",
flip = FALSE
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
feature	character Feature to represent.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
continuous_scale	logical Whether to color the ridges depending on a categorical or continuous scale.

`legend.title` [character](#) | Title for the legend.
`legend.ncol` [numeric](#) | Number of columns in the legend.
`legend.nrow` [numeric](#) | Number of rows in the legend.
`legend.byrow` [logical](#) | Whether the legend is filled by row or not.
`legend.position` [character](#) | Position of the legend in the plot. One of:

- `top`: Top of the figure.
- `bottom`: Bottom of the figure.
- `left`: Left of the figure.
- `right`: Right of the figure.
- `none`: No legend is displayed.

`legend.length`, `legend.width` [numeric](#) | Length and width of the legend. Will adjust automatically depending on legend side.
`legend.framewidth`, `legend.tickwidth` [numeric](#) | Width of the lines of the box in the legend.
`legend.framecolor` [character](#) | Color of the lines of the box in the legend.
`legend.tickcolor` [character](#) | Color of the ticks of the box in the legend.
`legend.type` [character](#) | Type of legend to display. One of:

- `normal`: Default legend displayed by **ggplot2**.
- `colorbar`: Redefined colorbar legend, using [guide_colorbar](#).
- `colorsteps`: Redefined legend with colors going by range, in steps, using [guide_colorsteps](#).

`colors.use` [character](#) | Named vector of colors to use. Has to match the unique values of `group.by` or `color.by` (if used) when `scale_type` is set to categorical.
`font.size` [numeric](#) | Overall font size of the plot. All plot elements will have a size relationship with this font size.
`font.type` [character](#) | Base font family for the plot. One of:

- `mono`: Mono spaced font.
- `serif`: Serif font family.
- `sans`: Default font family.

`rotate_x_axis_labels` [numeric](#) | Degree to rotate the X labels. One of: 0, 45, 90.
`plot.title`, `plot.subtitle`, `plot.caption` [character](#) | Title, subtitle or caption to use in the plot.
`xlab`, `ylab` [character](#) | Titles for the X and Y axis.
`compute_quantiles` [logical](#) | Whether to compute quantiles of the distribution and color the ridge plots by them.

```

compute_custom_quantiles
    logical | Whether to compute custom quantiles.
quantiles      numeric | Numeric vector of quantiles.
compute_distribution_tails
    logical | Whether to compute distribution tails and color them.
prob_tails     numeric | The accumulated probability that the tails should contain.
color_by_probabilities
    logical | Whether to color the ridges depending on the probability.
viridis_color_map
    character | A capital letter from A to H or the scale name as in scale\_fill\_viridis.
viridis_direction
    numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.
plot.grid     logical | Whether to plot grid lines.
grid.color    character | Color of the grid in the panels.
grid.type     character | One of the possible linetype options:
    • blank.
    • solid.
    • dashed.
    • dotted.
    • dotdash.
    • longdash.
    • twodash.
flip          logical | Whether to invert the axis of the displayed plot.

```

Value

A ggplot2 object.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_RidgePlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Compute the most basic ridge plot.
  p <- SCpubr::do_RidgePlot(sample = sample,
                             feature = "nFeature_RNA")

  p

  # Use continuous color scale.
  p <- SCpubr::do_RidgePlot(sample = sample,

```

```

        feature = "nFeature_RNA",
        continuous_scale = TRUE,
        viridis_direction = 1)
p

# Draw quantiles of the distribution.
p <- SCpubr::do_RidgePlot(sample = sample,
                          feature = "nFeature_RNA",
                          continuous_scale = TRUE,
                          compute_quantiles = TRUE,
                          compute_custom_quantiles = TRUE)
p

# Draw probability tails.
p <- SCpubr::do_RidgePlot(sample = sample,
                          feature = "nFeature_RNA",
                          continuous_scale = TRUE,
                          compute_quantiles = TRUE,
                          compute_distribution_tails = TRUE)
p

# Draw probability tails.
p <- SCpubr::do_RidgePlot(sample = sample,
                          feature = "nFeature_RNA",
                          continuous_scale = TRUE,
                          compute_quantiles = TRUE,
                          color_by_probabilities = TRUE)
p
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies()`.")
}

```

do_TermEnrichmentPlot *Display the enriched terms for a given list of genes.*

Description

Display the enriched terms for a given list of genes.

Usage

```

do_TermEnrichmentPlot(
  enriched_terms,
  nchar_wrap = 20,
  nterms = 10,
  font.size = 14,
  font.type = "sans",
  plot.title = NULL,

```

```

plot.subtitle = NULL,
plot.caption = NULL,
legend.position = "bottom",
legend.type = "colorbar",
colors.use = NULL,
text_labels_size = 4,
legend.length = 30,
legend.width = 1,
legend.framewidth = 0.5,
legend.tickwidth = 0.5,
legend.framecolor = "grey50",
legend.tickcolor = "white"
)

```

Arguments

- `enriched_terms` **list** | List containing the output(s) of running `Enrichr`.
- `nchar_wrap` **numeric** | Number of characters to use as a limit to wrap the term names. The higher this value, the longer the lines would be for each term in the plots. Defaults to 60.
- `nterms` **numeric** | Number of terms to report for each database. Terms are arranged by adjusted p-value and selected from lowest to highest. Defaults to 5.
- `Enrichr`.
 - `FlyEnrichr`.
 - `WormEnrichr`.
 - `YeastEnrichr`.
 - `FishEnrichr`.
- `font.size` **numeric** | Overall font size of the plot. All plot elements will have a size relationship with this font size.
- `font.type` **character** | Base font family for the plot. One of:
- `mono`: Mono spaced font.
 - `serif`: Serif font family.
 - `sans`: Default font family.
- `plot.title`, `plot.subtitle`, `plot.caption` **character** | Title, subtitle or caption to use in the plot.
- `legend.position` **character** | Position of the legend in the plot. One of:
- `top`: Top of the figure.
 - `bottom`: Bottom of the figure.
 - `left`: Left of the figure.
 - `right`: Right of the figure.
 - `none`: No legend is displayed.
- `legend.type` **character** | Type of legend to display. One of:
- `normal`: Default legend displayed by **ggplot2**.

- `colorbar`: Redefined colorbar legend, using `guide_colorbar`.
- `colorsteps`: Redefined legend with colors going by range, in steps, using `guide_colorsteps`.

`colors.use` **character** | Character vector of 2 colors (low and high ends of the color scale) to generate the gradient.

`text_labels_size`
 numeric | Controls how big or small labels are in the plot.

`legend.length`, `legend.width`
 numeric | Length and width of the legend. Will adjust automatically depending on legend side.

`legend.framewidth`, `legend.tickwidth`
 numeric | Width of the lines of the box in the legend.

`legend.framecolor`
 character | Color of the lines of the box in the legend.

`legend.tickcolor`
 character | Color of the ticks of the box in the legend.

Value

A `ggplot2` object with enriched terms.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_TermEnrichmentPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your enriched terms.
  enriched_terms <- readRDS(system.file("extdata/enriched_terms_example.rds", package = "SCpubr"))
  enriched_terms$GO_Cellular_Component_2021 <- NULL
  enriched_terms$Azimuth_Cell_Types_2021 <- NULL

  # Default plot.
  p <- SCpubr::do_TermEnrichmentPlot(enriched_terms = enriched_terms)
  p
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}
```

do_TFActivityPlot *Plot TF Activities from decoupleR using Dorothea prior knowledge.*

Description

Plot TF Activities from decoupleR using Dorothea prior knowledge.

Usage

```
do_TFActivityPlot(  
  sample,  
  activities,  
  n_tfs = 25,  
  group.by = NULL,  
  split.by = NULL,  
  plot_FeaturePlots = FALSE,  
  plot_Heatmaps = TRUE,  
  plot_GeyserPlots = FALSE,  
  row_title = NULL,  
  column_title = NULL,  
  flip = FALSE,  
  cluster_cols = TRUE,  
  cluster_rows = TRUE,  
  row_names_rot = 0,  
  column_names_rot = 45,  
  cell_size = 8,  
  pt.size = 1,  
  plot_cell_borders = TRUE,  
  border.size = 2,  
  na.value = "grey75",  
  legend.position = "bottom",  
  heatmap.legend.length = 75,  
  heatmap.legend.width = 5,  
  heatmap.legend.framecolor = "black",  
  legend.width = 1,  
  legend.length = 20,  
  legend.framewidth = 0.5,  
  legend.tickwidth = 0.5,  
  legend.framecolor = "grey50",  
  legend.tickcolor = "white",  
  legend.type = "colorbar",  
  font.size = 14,  
  font.type = "sans",  
  rotate_x_axis_labels = 45,  
  enforce_symmetry = TRUE,  
  geyser_order_by_mean = TRUE,  
  geyser_scale_type = "continuous",
```

```

viridis_color_map = "G",
viridis_direction = 1,
min.cutoff = NULL,
max.cutoff = NULL
)

```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
activities	tibble Result of running decoupleR method with dorothea regulon prior knowledge.
n_tfs	numeric Number of top regulons to consider for downstream analysis.
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.
plot_FeaturePlots	logical Compute output FeaturePlots for each of the top regulons.
plot_Heatmaps	logical Compute output heatmap showcasing the average TF activity per regulon and group.by variable.
plot_GeyserPlots	logical Compute output GeyserPlots for each of the top regulons and group.by variable.
row_title	character Title for the rows of the heatmaps. Only works with single heatmaps.
column_title	character Title for the columns of the heatmaps. Only works with single heatmaps.
flip	logical Whether to invert the axis of the displayed plot.
cluster_cols	logical Cluster the columns or rows of the heatmaps.
cluster_rows	logical Cluster the rows or rows of the heatmaps.
row_names_rot	numeric Degree in which to rotate the row labels.
column_names_rot	numeric Degree in which to rotate the column labels.
cell_size	numeric Size of each cell in the heatmap.
pt.size	numeric Size of the dots.
plot_cell_borders	logical Whether to plot border around cells.
border.size	numeric Width of the border of the cells.
na.value	character Color value for NA.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure.

- right: Right of the figure.
- none: No legend is displayed.

heatmap.legend.length, heatmap.legend.width
numeric | Width and length of the legend in the heatmap.

heatmap.legend.framecolor
character | Color of the edges and ticks of the legend in the heatmap.

legend.length, legend.width
numeric | Length and width of the legend. Will adjust automatically depending on legend side.

legend.framewidth, legend.tickwidth
numeric | Width of the lines of the box in the legend.

legend.framecolor
character | Color of the lines of the box in the legend.

legend.tickcolor
character | Color of the ticks of the box in the legend.

legend.type **character** | Type of legend to display. One of:

- normal: Default legend displayed by **ggplot2**.
- colorbar: Redefined colorbar legend, using [guide_colorbar](#).
- colorsteps: Redefined legend with colors going by range, in steps, using [guide_colorsteps](#).

font.size **numeric** | Overall font size of the plot. All plot elements will have a size relationship with this font size.

font.type **character** | Base font family for the plot. One of:

- mono: Mono spaced font.
- serif: Serif font family.
- sans: Default font family.

rotate_x_axis_labels
numeric | Degree to rotate the X labels. One of: 0, 45, 90.

enforce_symmetry
logical | Whether the geyser and feature plot has a symmetrical color scale.

geyser_order_by_mean
logical | Whether to order the X axis by the mean of the values.

geyser_scale_type
character | Type of scale to use. Either "continuous" or "categorical".

viridis_color_map
character | A capital letter from A to H or the scale name as in [scale_fill_viridis](#).

viridis_direction
numeric | Either 1 or -1. Controls how the gradient of viridis scale is formed.

min.cutoff, max.cutoff
numeric | Set the min/max ends of the color scale. Any cell/group with a value lower than min.cutoff will turn into min.cutoff and any cell with a value higher than max.cutoff will turn into max.cutoff. In FeaturePlots, provide as many values as features. Use NAs to skip a feature.

Value

A list containing several output plots according to the user's specifications.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_TFActivityPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds",
                                package = "SCpubr"))

  # Define your activities object.
  dorothea_activities <- readRDS(system.file("extdata/dorothea_activities_example.rds",
                                             package = "SCpubr"))

  # General heatmap.
  out <- SCpubr::do_TFActivityPlot(sample = sample,
                                   activities = dorothea_activities,
                                   plot_FeaturePlots = TRUE,
                                   plot_GeyserPlots = TRUE)

  p <- out$heatmaps$aaverage_scores
  p

  # Retrieve feature plots.
  p <- out$feature_plots$ARID2
  p

  # Retrieve Geyser plots.
  p <- out$geyser_plots$ARID2
  p
} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}
```

do_ViolinPlot

Wrapper for [VlnPlot](#).

Description

Wrapper for [VlnPlot](#).

Usage

```
do_ViolinPlot(  
  sample,  
  features,  
  assay = NULL,  
  slot = NULL,  
  group.by = NULL,  
  split.by = NULL,  
  colors.use = NULL,  
  pt.size = 0,  
  line_width = 0.5,  
  y_cut = rep(NA, length(features)),  
  plot_boxplot = TRUE,  
  boxplot_width = 0.2,  
  legend.position = "none",  
  plot.title = NULL,  
  plot.subtitle = NULL,  
  plot.caption = NULL,  
  xlab = rep(NA, length(features)),  
  ylab = rep(NA, length(features)),  
  font.size = 14,  
  font.type = "sans",  
  rotate_x_axis_labels = 45,  
  plot.grid = TRUE,  
  grid.color = "grey75",  
  grid.type = "dashed",  
  flip = FALSE,  
  ncol = NULL,  
  share.y.lims = FALSE,  
  legend.title = NULL,  
  legend.ncol = NULL,  
  legend.nrow = NULL,  
  legend.byrow = FALSE  
)
```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
features	character Features to represent.
assay	character Assay to use. Defaults to the current assay.
slot	character Data slot to use. Only one of: counts, data, scale.data. Defaults to "data".
group.by	character Metadata variable to group the output by. Has to be a character of factor column.
split.by	character Secondary metadata variable to further group (split) the output by. Has to be a character of factor column.

colors.use	named_vector Named vector of valid color representations (either name of HEX codes) with as many named colors as unique values of group.by. If group.by is not provided, defaults to the unique values of Idents . If not provided, a color scale will be set by default.
pt.size	numeric Size of points in the Violin plot.
line_width	numeric Width of the lines drawn in the plot. Defaults to 1.
y_cut	numeric Vector with the values in which the Violins should be cut. Only works for one feature.
plot_boxplot	logical Whether to plot a Box plot inside the violin or not.
boxplot_width	numeric Width of the boxplots. Defaults to 0.2.
legend.position	character Position of the legend in the plot. One of: <ul style="list-style-type: none"> • top: Top of the figure. • bottom: Bottom of the figure. • left: Left of the figure. • right: Right of the figure. • none: No legend is displayed.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
xlab, ylab	character Titles for the X and Y axis.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
rotate_x_axis_labels	numeric Degree to rotate the X labels. One of: 0, 45, 90.
plot.grid	logical Whether to plot grid lines.
grid.color	character Color of the grid in the panels.
grid.type	character One of the possible linetype options: <ul style="list-style-type: none"> • blank. • solid. • dashed. • dotted. • dotdash. • longdash. • twodash.
flip	logical Whether to invert the axis of the displayed plot.
ncol	numeric Number of columns used in the arrangement of the output plot using "split.by" parameter.

share.y.lims **logical** | When querying multiple features, force the Y axis of all of them to be on the same range of values (this being the max and min of all features combined).

legend.title **character** | Title for the legend.

legend.ncol **numeric** | Number of columns in the legend.

legend.nrow **numeric** | Number of rows in the legend.

legend.byrow **logical** | Whether the legend is filled by row or not.

Value

A ggplot2 object containing a Violin Plot.

Examples

```
# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_ViolinPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Basic violin plot.
  p <- SCpubr::do_ViolinPlot(sample = sample,
                             feature = "nCount_RNA")
  p

  # Remove the box plots.
  p <- SCpubr::do_ViolinPlot(sample = sample,
                             feature = "nCount_RNA",
                             plot_boxplot = FALSE)
  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.`")
}
```

do_VolcanoPlot

Compute a Volcano plot out of DE genes.

Description

Compute a Volcano plot out of DE genes.

Usage

```
do_VolcanoPlot(
  sample,
  de_genes,
  pval_cutoff = 0.05,
  FC_cutoff = 2,
  pt.size = 2,
  border.size = 1.5,
  border.color = "black",
  font.size = 14,
  font.type = "sans",
  plot.title = NULL,
  plot.subtitle = NULL,
  plot.caption = NULL,
  plot_lines = TRUE,
  line_color = "grey75",
  line_size = 0.5,
  add_gene_tags = TRUE,
  order_tags_by = "both",
  n_genes = 5,
  use_labels = FALSE,
  colors.use = "steelblue"
)
```

Arguments

sample	Seurat A Seurat object, generated by CreateSeuratObject .
de_genes	tibble Output of <code>Seurat::FindMarkers()</code> .
pval_cutoff	numeric Cutoff for the p-value.
FC_cutoff	numeric Cutoff for the avg_log2FC.
pt.size	numeric Size of the dots.
border.size	numeric Width of the border of the cells.
border.color	character Color to use for the border of the cells.
font.size	numeric Overall font size of the plot. All plot elements will have a size relationship with this font size.
font.type	character Base font family for the plot. One of: <ul style="list-style-type: none"> • mono: Mono spaced font. • serif: Serif font family. • sans: Default font family.
plot.title, plot.subtitle, plot.caption	character Title, subtitle or caption to use in the plot.
plot_lines	logical Whether to plot the division lines.
line_color	character Color for the lines.
line_size	numeric Size of the lines in the plot.

add_gene_tags **logical** | Whether to plot the top genes.
 order_tags_by **character** | Either "both", "pvalue" or "logfc".
 n_genes **numeric** | Number of top genes in each side to plot.
 use_labels **logical** | Whether to use labels instead of text for the tags.
 colors.use **character** | Color to generate a tetradic color scale with.

Value

A volcano plot as a ggplot2 object.

Examples

```

# Check Suggests.
value <- SCpubr:::check_suggests(function_name = "do_VolcanoPlot", passive = TRUE)

if (isTRUE(value)){
  # Consult the full documentation in https://enblacar.github.io/SCpubr-book/

  # Define your Seurat object.
  sample <- readRDS(system.file("extdata/seurat_dataset_example.rds", package = "SCpubr"))

  # Retrieve DE genes.
  de_genes <- readRDS(system.file("extdata/de_genes_example.rds", package = "SCpubr"))

  # Generate a volcano plot.
  p <- SCpubr::do_VolcanoPlot(sample = sample,
                              de_genes = de_genes)

  p

} else if (isFALSE(value)){
  message("This function can not be used without its suggested packages.")
  message("Check out which ones are needed using `SCpubr::state_dependencies`.")
}

```

human_chr_locations *Chromosome arm locations for human genome GRCh38.*

Description

A tibble containing the chromosome, arm and start and end coordinates.

Usage

```
data(human_chr_locations)
```

Format

A tibble with 48 rows and 4 columns:

chr Chromosome.

arm Chromosome arm.

start Start coordinates.

end End coordinates.

state_dependencies *State SCpubr current function dependencies.*

Description

State SCpubr current function dependencies.

Usage

```
state_dependencies(function_name = NULL, return_dependencies = FALSE)
```

Arguments

function_name **character** | Name of an exported function from SCpubr. If NULL, return all functions.

return_dependencies

logical | Whether to have the dependencies as an output object instead of a printed message.

Value

None

Examples

```
# See all dependencies.
SCpubr::state_dependencies()

# See the dependencies for a single package.
SCpubr::state_dependencies(function_name = "do_DimPlot")
```

Index

- * **datasets**
 - human_chr_locations, 89
- AddModuleScore, 22, 43, 46
- AddModuleScore_UCell, 46
- character, 4, 5, 7–10, 12–14, 16, 17, 21, 22, 24, 25, 27, 30, 31, 33, 34, 36–38, 40, 41, 44–49, 52–54, 56–58, 60, 61, 63, 64, 66, 68–70, 72, 73, 75–77, 79, 80, 82, 83, 85–90
- chordDiagramFromDataFrame, 25, 26
- chordDiagramFromMatrix, 25, 26
- CreateSeuratObject, 4, 7, 9, 12, 16, 20, 24, 30, 33, 36, 40, 44, 48, 52, 60, 66, 68, 71, 75, 82, 85, 88
- DimPlot, 34
- do_AlluvialPlot, 3
- do_AzimuthAnalysisPlot, 6
- do_BarPlot, 9
- do_BeeSwarmPlot, 11
- do_BoxPlot, 15
- do_CellularStatesPlot, 19
- do_ChordDiagramPlot, 24
- do_ColorPalette, 26
- do_CopyNumberVariantPlot, 29
- do_CorrelationPlot, 32
- do_DimPlot, 34
- do_DotPlot, 39
- do_EnrichmentHeatmap, 43
- do_ExpressionHeatmap, 48
- do_FeaturePlot, 50
- do_FunctionalAnnotationPlot, 55
- do_GeyserPlot, 59
- do_GroupedGOTermPlot, 63
- do_GroupwiseDEPlot, 65
- do_NebulosaPlot, 67
- do_PathwayActivityPlot, 70
- do_RidgePlot, 74
- do_TermEnrichmentPlot, 78
- do_TFActivityPlot, 81
- do_ViolinPlot, 84
- do_VolcanoPlot, 87
- DotPlot, 39
- FeaturePlot, 50
- geom_alluvium, 4
- geom_flow, 4
- geom_label, 36
- geom_text, 36
- guide_colorbar, 13, 30, 40, 46, 52, 57, 61, 69, 73, 76, 80, 83
- guide_colorsteps, 13, 30, 40, 46, 52, 57, 61, 69, 73, 76, 80, 83
- Heatmap, 43
- human_chr_locations, 89
- Idents, 8, 10, 13, 17, 21, 36, 86
- infercnv, 30
- list, 79
- logical, 4, 5, 7–10, 13, 14, 17, 18, 21, 22, 25, 27, 30, 31, 33, 36–38, 41, 45–47, 49, 52–54, 57, 60–64, 68–70, 72, 73, 75–77, 82, 83, 86–90
- named_list, 20, 44
- named_vector, 8, 10, 13, 17, 21, 25, 30, 36, 86
- numeric, 4, 5, 7, 8, 10, 13, 14, 16, 17, 21, 22, 25–27, 31, 33, 34, 36–38, 40, 41, 45–47, 49, 50, 52–54, 56–58, 60–64, 66–70, 72, 73, 76, 77, 79, 80, 82, 83, 86–89
- position_fill, 10
- position_stack, 9
- scale_fill_viridis, 5, 8, 14, 22, 31, 33, 41, 45, 49, 53, 58, 61, 69, 73, 77, 83

Seurat, [4](#), [7](#), [9](#), [12](#), [16](#), [20](#), [24](#), [30](#), [33](#), [36](#), [40](#),
[44](#), [48](#), [52](#), [60](#), [66](#), [68](#), [71](#), [75](#), [82](#), [85](#),
[88](#)

state_dependencies, [90](#)

tibble, [30](#), [66](#), [71](#), [82](#), [88](#)

VlnPlot, [84](#)