Package 'rprismtools'

December 15, 2022

Title Useful functions META-PRISM project not related to data

Version 1.5.0

Maintainer 'Yoann Pradat' <yoann.pradat@centralesupelec.fr>

Description Meta PRISM project is an institutional project from the French University Hospital Gustave Roussy. This project aims at describing the molecular landscape of metastatic cancers from all solid tumor types.

License BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

URL https://github.com/gustaveroussy/MetaPRISM/tree/master/functions/rprismtools

Depends R (>= 3.0)

Imports DT,
ComplexHeatmap,
RColorBrewer,
dplyr,
graphics,
htmltools,
htmlwidgets,
magrittr,
plotly,
rlang,
scales,
stringr,
tidyselect,
tidyr,
utils

Suggests testthat

R topics documented:

compute_coordinates_burden_plot 2
compute_groups_order_burden_plot 3
cumulative_scatter_plots 3

draw_burden_plot 4
draw_facetted_heatmap_barplots_1 6
draw_facetted_heatmap_barplots_2 8
draw_numbered_heatmap 9
draw_upset_plot 10
get_label_colors 11
get_tables_for_facetted_heatmap_barplots 11
make_upset_m 12
progress 13
rect_plot_colors 13
render_table 14
rprismtools 15
select_in_plot_evt 15
select_vals_in_dfs 16

Index **17**

`compute_coordinates_burden_plot` *Compute the coordinates of each point in the burden plot.*

Description

Compute the coordinates of each point in the burden plot.

Usage

```
compute_coordinates_burden_plot(
  df,
  groups_order,
  col_burden,
  col_groups,
  offset = 0.1
)
```

Arguments

`df`	A `data.frame` containing the columns `col_burden` and `col_groups`.
`groups_order`	A character vector containing ordered group names.
`col_burden`	Name of the column holding burden values.
`col_groups`	Name of the column holding groups values.
`offset`	(optional) The j^th scatter plot is drawn between j+offset and j+1-offset.

See Also

`compute_groups_order_burden_plot()`

compute_groups_order_burden_plot

Compute the groups order for the burden plot.

Description

Compute the groups order for the burden plot.

Usage

```
compute_groups_order_burden_plot(
  df,
  col_burden,
  col_groups,
  groups_keep = NULL
)
```

Arguments

df	A data.frame containing the columns col_burden and col_groups.
col_burden	Name of the column holding burden values.
col_groups	Name of the column holding groups values.
groups	(optional) Character vector containing names of groups to be retained.

Value

a character vector containing ordered group names.

cumulative_scatter_plots

Draw multiple scatter plots showing cumulative distributions

Description

Scatter plots are organized by groups and stacks. The groups define the numbers of subplots while stacks define the number of points clouds in each subplot.

Usage

```
cumulative_scatter_plots(
  dfs,
  groups,
  stacks,
  offset,
  ytitle,
  ytitlefont = list(size = 14),
  yname = "Burden",
  yrange = c(10^-3, 10^4),
```

```
  ylabelsfont = list(size = 12),
  min_median_size = 5,
  colors_bkgd = c("#f3f3f3", "#fffff"),
  markers_scatter = list(list(size = 5, color = "black")),
  lines_median = list(list(color = "red")),
  lwd_axes = 2,
  col_id = "Subject_Id"
)
```

Arguments

dfs	A 2-level named list of dataframes. 1st level of names are stacks, 2nd level of names are groups.
groups	A character vector.
stacks	A character vector.
offset	For group i, the scatter plot is drawn between i+offset and i+1-offset.
ytitle	Title y-axis.
ytitlefont	Font title y-axis.
yname	Name of the y variable. Used for the hovering template.
yrange	Range y-axis.
ylabelsfont	Font of y-axis labels.
min_median_size	Minimum number of points required in the points cloud to draw the median.
colors_bkgd	Vector of colors for the background. If less than the number of grounds, values are recycled using rep function.
markers_scatter	List of markers characteristics.
lines_median	List of median line characteristics.
lwd_axes	Linewidth of line delimiting y anx x axes.
col_id	Name of column id for hovering.

Author(s)

Yoann Pradat

---

draw_burden_plot *Draw multiple scatter plots showing cumulative distributions*

Description

Scatter plots are organized by groups and stacks. The groups define the numbers of subplots while stacks define the number of points clouds in each subplot.

Usage

```
draw_burden_plot(
  dfs,
  groups,
  stacks,
  offset,
  stacks2colors,
  marker_size = 4,
  marker_opacity = 1,
  median_width = 2,
  yname = "Burden",
  yrange = c(10^-3, 10^4),
  min_median_size = 5,
  ytitle = "Burden",
  margin = list(t = 250),
  lwd_axes = 2,
  ytitlefont = list(size = 18),
  xlabelsfont = list(size = 20),
  ylabelsfont = list(size = 20),
  legendfont = list(size = 18),
  col_id = "Subject_Id"
)
```

Arguments

`dfs`	A 2-level named list of dataframes. 1st level of names are stacks, 2nd level of names are groups.
`groups`	A character vector.
`stacks`	A character vector.
`offset`	For group i, the scatter plot is drawn between i+offset and i+1-offset.
`stacks2colors`	A named list with names being values in `stacks` and values being color names.
`marker_size`	Size of the scatter markers.
`median_width`	Width of the median line.
`yname`	Name of the y variable. Used for the hovering template.
`yrange`	Range y-axis.
`min_median_size`	Minimum number of points required in the points cloud to draw the median.
`ytitle`	Title y-axis.
`margin`	Named list passed to the `margin` parameter of `plotly::layout`.
`lwd_axes`	Linewidth of line delimiting y anx x axes.
`ytitlefont`	Font title y-axis.
`xlabelsfont`	Font of the x-axis labels.
`ylabelsfont`	Font of y-axis labels.
`legendfont`	Font of the legend.
`col_id`	Name of column id for hovering.

Author(s)

Yoann Pradat

`draw_facetted_heatmap_barplots_1` *Draw a facetted figure with template 1.*

Description

The main plot in the figure is the central heatmap. This heatmap serves to show the distribution of counts or percentages of an event in a fixed number of variables and fixed number of samples.

The figure may additionally contain a

- a left double inverted barplot it is used to display -log10(pvalues) distribution from a per-variable test (e.g Fisher stratified by the samples).
- a middle right barplot it is used to display total variable counts.
- an extreme right barplot it is used to display additional information about a variable in the form of a stacked barplot. It is useful to show the breakdown of a total variable count into types.

Usage

```
draw_facetted_heatmap_barplots_1(
  dfs,
  col_var,
  names2plots,
  names2colors,
  colors_palette_heatmap = "Reds",
  colors_limits_heatmap = NULL,
  col_pval = "p.value",
  width_one = 100,
  height_one = 10,
  alpha_left = 0.1,
  alpha_heatmap = 0.1,
  col_stack = NULL,
  width_edges_heatmap = 0.05,
  fonts = list(z_heatmap = list(size = 8), x_tick_heatmap = list(size = 10),
   y_tick_heatmap = list(size = 6), x_tick_row_bar = list(size = 8), x_title_row_bar =
     list(size = 10), cohort_size = list(size = 10), legend = list(size = 9)),
  bargap = 0.1,
  stacks2colors = NULL,
  showlegend = TRUE,
  add_colors_names = TRUE,
  x_tick_heatmap_side = "bottom",
  x_title_row_bar = "",
  add_cohorts_sizes = F
)
```

Arguments

`dfs`	a named list of dataframes containing the values indicated in the list `names2plots`.
`col_var`	A character vector defining the field used as variables.

`names2plots` a list with names among 'left_pvals', 'heatmap', 'heatmap_hover', 'heatmap_pvals', 'middle_right_bar', 'extreme_right_bar'. The corresponding values must be in the list of names of `dfs`.

`names2colors` a list with names that must match values of the vectors names2plots$left_pvals and names2plots$heatmap_pvals.

`colors_palette_heatmap`
(optional) The color palette to be used for the heatmap. You may use "Reds", "Blues" or a list of colors that matches the length of color_limits_heatmap.

`colors_limits_heatmap`
(optional) The limit values for deciding on cell colors in the heatmap.

`width_one` (optional) The width of the plot is obtained as `width_one` multiplied by the number of columns in `z`.

`height_one` (optional) The height of the plot is obtained as `height_one` multiplied by the number of rows in `z`.

`alpha_left` (optional) Threshold for drawing the bar for a significant variable pvalue.

`alpha_heatmap` (optional) Threshold for highlighting edges of a significant cell pvalue.

`col_stack` (optional) A character vector defining the field used for the extreme right stack barplot. Used only if 'extreme_right_bar' in the names of `names2plots`.

`width_edges_heatmap`
(optional) Width of the line defining the edges of the heatmap.

`fonts` (optional) List of fonts. Required names are 'x_tick_heatmap', 'y_tick_heatmap', 'x_tick_rowbar' and 'legend'. Each font is a list with font parameters, e.g, 'size'.

`bargap` (optional) Spacing between bars of barplot.

`stacks2colors` (optional) A list of colors with names corresponding to values for the extreme right stack barplot. Used only if 'extreme_right_bar' in the names of `names2plots`.

`showlegend` (optional) Set to FALSE to hide the legend.

`add_colors_names`
(optional) Set to FALSE to not use colors in x-axis labels.

`x_tick_heatmap_side`
(optional) Choose 'bottom' or 'top'.

`x_title_row_bar`
(optional) Title for the x-axis of the side bar plot.

`add_cohorts_sizes`
(optional) Set to TRUE to show cohort sizes above or below x tick labels.

Value

a list of plotly figure objects.

Author(s)

Yoann Pradat

`draw_facetted_heatmap_barplots_2` *Draw a facetted figure with template 2.*

Description

Draw a facetted figure having a heatmap in its center. This heatmap serves to show the distribution of counts or percentages of events (rows) per groups of observations (columns). The figure may additionally contain barplots positioned above or below the heatmap.

Usage

```
draw_facetted_heatmap_barplots_2(
  dfs,
  col_var,
  names2plots,
  names2colors,
  linewidth = 2,
  width_one = 100,
  height_one = 10,
  alpha_left = 0.1,
  alpha_heatmap = 0.1,
  width_edges_heatmap = 0.05,
  legend_titles = c("ESCAT levels", "# of actionable<br>alterations"),
  legend_x = -0.25,
  legend_x_to_y_ratio = 0.2,
  legend_y_gap = 0.06,
  fonts = list(z_heatmap = list(size = 8), x_tick_heatmap = list(size = 10),
   y_tick_heatmap = list(size = 10), x_tick_row_bar = list(size = 10), y_title_row_bar =
    list(size = 12), cohort_size = list(size = 10), legend = list(size = 12),
    legend_label = list(size = 12), legend_title = list(size = 16)),
  add_colors_names = TRUE
)
```

Arguments

`dfs`	a named list of dataframes containing the values indicated in the list `names2plots`.
`col_var`	A character vector defining the field used as variables.
`names2plots`	a list with names among 'heatmap', 'heatmap_color', 'heatmap_hover', 'left_pvals', 'heatmap_pvals', 'barplot_top', 'barplot_bot'. The corresponding values must be in the list of names of `dfs`.
`names2colors`	a list with names that must match values of the vectors names2plots$left_pvals and names2plots$heatmap_pvals.
`linewidth`	(optional) Linewidth of external borders (axes).
`width_one`	(optional) The width of the plot is obtained as `width_one` multiplied by the number of columns in `z`.
`height_one`	(optional) The height of the plot is obtained as `height_one` multiplied by the number of rows in `z`.
`alpha_left`	(optional) Threshold for drawing the bar for a significant variable pvalue.

alpha_heatmap (optional) Threshold for highlighting edges of a significant cell pvalue.

width_edges_heatmap
(optional) Width of the line defining the edges of the heatmap.

legend_titles (optional) Titles for legends left of barplots.

legend_x (optional) Position on the x-axis of the legend next to the heatmap barplots.

legend_x_to_y_ratio
(optional) Ratio for controlling the aspect of legend items.

legend_y_gap (optional) For controlling vertical gap between legend items.

fonts (optional) List of fonts. Required names are 'x_tick_heatmap', 'y_tick_heatmap', 'x_tick_rowbar' and 'legend'. Each font is a list with font parameters, e.g, 'size'.

add_colors_names
(optional) Set to FALSE to not use colors in x-axis labels.

showlegend (optional) Set to FALSE to hide the legend.

Value

a list of plotly figure objects.

Author(s)

Yoann Pradat

---

draw_numbered_heatmap *Draw a simple heatmap with numbers*

Description

Draw a simple heatmap with numbers

Usage

```
draw_numbered_heatmap(
  df_z,
  z_name,
  colors_limits = NULL,
  colors_palette = "Reds",
  width_one = 50,
  height_one = 15,
  black_white_cutoff = 0.5,
  font = list(size = 8)
)
```

Arguments

df_z a data.frame with numeric values.

z_name Name of the values for the hovering template. 'middle_right_bar', 'extreme_right_bar'. The corresponding values must be in the list of names of dfs.

colors_limits (optional) A numeric vector, used to associate to each unique value in z an interval.

colors_palette	(optional) A name ("Reds" or "Blues") or a character vector of color names/codes.
width_one	(optional) The width of the plot is obtained as width_one multiplied by the number of columns in z.
height_one	(optional) The height of the plot is obtained as height_one multiplied by the number of rows in z.
black_white_cutoff	(optional) Cutoff for deciding on text font color.
font	(optional) Text font parameters.

Value

a plotly figure object.

Author(s)

Yoann Pradat

`draw_upset_plot` *Draw Upset plot with predefined them*

Description

Draw an upset plot using the function `ComplexHeatmap::Upset`. It takes as input an object produced by `ComplexHeatmap::make_comb_mat` and draws the plot with a predefined theme.

Usage

```
draw_upset_plot(
  m,
  row_annot_fontsize = 8,
  pt_size = grid::unit(3, "mm"),
  lwd = 2,
  height_top_annot = 3,
  width_set_size = 3,
  margin_row_text = 4,
  ...
)
```

Arguments

m	object produced by ComplexHeatmap::make_comb_mat
row_annot_fontsize	(optional) fontsize of rows annotations
pt_size	(optional) size for dots representing combination sets
lwd	(optional) line width for the combination sets
height_top_annot	(optional) height in inches
width_set_size	(optional) height in inches
margin_row_text	(optional) width in inches of the margin surrounding set names
...	Extra parameters passed to ComplexHeatmap::UpSet.

Author(s)

Yoann Pradat

`get_label_colors` *Associate colors to labels*

Description

Takes as input a vector of labels (may be non unique) and returns a vector of the same size with one color for each unique label in the labels. The set of unique labels may be specified. It is useful if not all labels are present in the vector of labels.

Usage

```
get_label_colors(labels, pal = "Dark2", labels_unique = NULL, alpha = 1)
```

Arguments

`labels`	vector.
`pal`	(optional) name of a palette of RColorBrewer.
`labels_unique`	(optional) If not NULL, used to defined colors for labels that are not in the labels vector. Useful when you wish to harmonize colors between plots that do not have all the labels each.
`alpha`	(optional) double in [0,1].

Value

a character vector of colors of same size as labels.

Author(s)

Yoann Pradat

`get_tables_for_facetted_heatmap_barplots`
Prepare tables for the facetted heatmap barplots

Description

Return a list of dataframes with events counts and percentages.

Usage

```
get_tables_for_facetted_heatmap_barplots(
  df_evt_count,
  df_tt_count,
  tt_keep,
  col_evt,
  col_tt,
  col_evt_classes = NULL
)
```

Arguments

df_evt_count	The event counts aggreagted by tumor type.
df_tt_count	Table of tumor type counts. It must contain the columns Count and col_tt.
tt_keep	List of tumor types included in the plot.
col_evt	Name of the column containing the events.
col_tt	Name of the column containing tumor types.
col_evt_classes	(optional) Name of the column containing subclasses of events.

Value

a list of tables.

---

make_upset_m *Build a* ComplexHeatmap::Heatmap-class *object from a dataframe*

Description

Apply the ComplexHeatmap::make_comb_mat on a named list prepared from a dataframe.

Usage

```
make_upset_m(df, field_set, field_identifier)
```

Arguments

df	dataframe with one field for identifiers and one field to group identifiers
field_set	Name of the field identifying groups.
field_identifier	Name of the field identifier entries.

Value

m a ComplexHeatmap::Heatmap-class object

Author(s)

Yoann Pradat

progress *A progress bar function to run within a* for *loop*

Description

This function should be run within a for loop, it produces a progress bar to indicate how many iterations have passed

Usage

```
progress(i, n)
```

Arguments

i	A numeric value indicating the current iteration of the containing for loop
n	A numeric value indicating the total number of iterations of the containing for loop

Author(s)

Joseph Crispell see repo https://github.com/JosephCrispell/basicPlotteR

Examples

```
# Set the number of iterations of the for loop
n <- 1000

for(i in 1:n){

  # Sleep for a tenth of a second
  Sys.sleep(0.01)

  # Update the progress bar
  progress(i, n)
}
```

rect_plot_colors *Display a list of colors*

Description

From a named list of colors, draw a set of rectangles showing the color and the name inside the rectangle.

Usage

```
rect_plot_colors(line = NULL, col = NULL, colors, cex = 1)
```

Arguments

line	How many lines in the facetted plot.
col	How many columns in the facetted plot.
colors	Named list of colors.
cex	Font size.

Author(s)

Yoann Pradat

---

render_table *Render a table using DT*

Description

Render a table using DT

Usage

```
render_table(
  df,
  caption,
  full = F,
  nrows = 5,
  extensions = c("Buttons", "Responsive"),
  buttons = c("copy", "csv", "excel")
)
```

Arguments

df	data.frame to render
caption	the table caption
full	(optional) if T, render the full table
nrows	(optional) number of rows to render. Ignored if full=T.
extensions	(optional) passed to parameter 'extensions' of DT::datatable.
buttons	(optional) passed to parameter 'options' of DT::datatable. Supported values are "copy", "csv" and "excel".

Author(s)

Yoann Pradat

`rprismtools` *rprismtools: A package with util functions.*

Description

'rprismtools' provides small functions useful for making plots, for rendering html widgets or for performing common tasks.

Author(s)

Yoann Pradat

`select_in_plot_evt` *Get the union of values that meet threshold requirements.*

Description

Get the union of values that meet threshold requirements.

Usage

```
select_in_plot_evt(
  dfs_plot,
  min_counts_evt,
  col_evt,
  max_evt = NULL,
  max_evt_cohort = NULL
)
```

Arguments

`dfs_plot`	List of dataframes.
`min_counts_evt`	List of minimum counts for each cohort.
`col_evt`	Name of the column from which to extract a entries.
`max_evt`	An integer giving the maximum number of events to be considered.
`max_evt_cohort`	A name identifying the table in `dfs_plot` to be used for selecting the events to be kept in case there are more than `max_evt` selected after filtering by `min_counts_evt`.

Value

a vector.

`select_vals_in_dfs` *Subset a list of dataframes on a column.*

Description

Subset a list of dataframes on a column.

Usage

```
select_vals_in_dfs(dfs, col, vals)
```

Arguments

`dfs`	List of dataframes.
`col`	The name of the column.
`vals`	Vals that should exist in the column. If values are missing, they are added as extra rows.

Value

a data.frame

Index

∗ **bar**
progress, 13
∗ **for**
progress, 13
∗ **progress**
progress, 13

compute_coordinates_burden_plot, 2
compute_groups_order_burden_plot, 3
compute_groups_order_burden_plot(), *2*
cumulative_scatter_plots, 3

draw_burden_plot, 4
draw_facetted_heatmap_barplots_1, 6
draw_facetted_heatmap_barplots_2, 8
draw_numbered_heatmap, 9
draw_upset_plot, 10

get_label_colors, 11
get_tables_for_facetted_heatmap_barplots, 11

make_upset_m, 12

progress, 13

rect_plot_colors, 13
render_table, 14
rprismtools, 15

select_in_plot_evt, 15
select_vals_in_dfs, 16