Package ‘FlexDotPlot’

October 12, 2022

Title Flexible Dot Plot

Version 0.2.2

Description Flexible Dotplot and Pacman plot for multimodal data.

Depends R (>= 3.6.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports ggplot2, grDevices, stats, FactoMineR, scales, reshape2, ggdendro, grImport2, grid, gridExtra, sisal, dendextend, shiny, shinydashboard, shinyWidgets, ggforce, colourpicker, htmltools, bsplus, DT, magrittr

Suggests knitr, rmarkdown, markdown,

VignetteBuilder knitr

URL `https://github.com/Simon-Leonard/FlexDotPlot`

BugReports `https://github.com/Simon-Leonard/FlexDotPlot/issues`

NeedsCompilation no

Author Simon Leonard [aut, cre] (<`https://orcid.org/0000-0002-6437-5800`>)

Maintainer Simon Leonard <`simon.leonard@univ-rennes1.fr`>

Repository CRAN

Date/Publication 2022-04-04 11:50:09 UTC

R topics documented:

`CBMC8K_example_data` *CBMC 8K example dataset*

Description

A dataset containing expression informations of 12 genes in 11 celltypes. Original dataset from seurat-data package.

Usage

```
data(CBMC8K_example_data)
```

Format

A data frame with 132 rows and 12 variables

Details

- features.plot. Gene symbol
- id. Cell type
- RNA.pct.exp. RNA level : Percentage of cells expressing the gene in the cell type
- RNA.avg.exp. RNA level : Average expression of the gene in the cell type
- RNA.avg.exp.log2p1. RNA.avg.exp with log2(x+1) tranformation
- RNA.avg.exp.scaled. RNA level : Scaled average expression
- ADT.pct.exp.sup0. ADT level : Percentage of cells expressing the gene in the cell type (with expression >0)
- ADT.pct.exp.sup.cutoff. ADT level : Percentage of cells expressing the gene in the cell type (with expression > background)
- ADT.avg.exp. ADT level : Average expression of the gene in the cell type
- ADT.avg.exp.log2p1. ADT.avg.exp with log2(x+1) tranformation
- ADT.avg.exp.scaled. ADT level : Scaled average expression
- canonical_marker. If the gene is a canonical marker of the cell type (yes or no)

References

`https://github.com/satijalab/seurat-data`

`CellphoneDB_example_data` *CellphoneDB example dataset*

Description

A dataset containing CellPhoneDB results of 11 gene pairs in 10 combinations of 2 cell types. Raw data obtained in doi: 10.1016/j.ccell.2021.02.015. Script used to generate this dataset from the raw data is available in the FlexDotPlot_cellphoneDB_dataset vignette.

Usage

```
data(CellphoneDB_example_data)
```

Format

A data frame with 55 rows and 11 variables

Details

- pair. Gene pair
- clusters. Cell type pair
- pvalue. pvalue from CellPhoneDB
- mean. Log2 mean (gene pair expressions) from CellPhoneDB
- mean1. Average expression of the first gene in the first cell type
- mean2. Average expression of the second gene in the second cell type
- pct1. Percentage of cells from the first cell type expressing the first gene
- pct2. Percentage of cells from the second cell type expressing the second gene
- log2mean1. mean1 with log2 transformation
- log2mean2. mean2 with log2 transformation
- log10pval. pvalue with -log10 transformation

References

doi: 10.1016/j.ccell.2021.02.015

`dot_plot` *Dot-plot - Pacman-plot function*

Description

Create dotplots to represent two discrete factors (x & y) described by several other factors. Each combination of the two discrete factors (x & y) can be described with : 1 continuous factor (setting shape size), 3 continuous or discrete factors (setting shape type, shape color and text on shape).

Usage

```
dot_plot(
  data.to.plot,
  size_var = NA,
  col_var = NA,
  text_var = NA,
  shape_var = 16,
  size_legend = "",
  col_legend = "",
  shape_legend = "",
  cols.use = "default",
  text.size = NA,
  text.vjust = 0,
  shape_use = "default",
  shape.scale = 12,
  scale.by = "radius",
  scale.min = NA,
  scale.max = NA,
  plot.legend = TRUE,
  do.return = FALSE,
  x.lab.pos = c("both", "top", "bottom", "none"),
  y.lab.pos = c("left", "right", "both", "none"),
  x.lab.size.factor = 1,
  y.lab.size.factor = 1,
  vertical_coloring = NA,
  horizontal_coloring = NA,
  size.breaks.number = 4,
  color.breaks.number = 5,
  shape.breaks.number = 5,
  size.breaks.values = NA,
  color.breaks.values = NA,
  shape.breaks.values = NA,
  display_max_sizes = TRUE,
  transpose = FALSE,
  dend_x_var = NULL,
  dend_y_var = NULL,
  dist_method = c("euclidean", "maximum", "manhattan", "canberra", "binary",
```

```
    "minkowski"),
  hclust_method = c("ward.D", "single", "complete", "average", "mcquitty", "median",
    "centroid", "ward.D2"),
   do.plot = TRUE
)
```

Arguments

`data.to.plot`	Input data. Can be a list or a data.frame. If data.frame : Column 1 = x axis (Factor); Col2= y axis (Factor). If list : x and y axis are fixed by row and col names of list elements.
`size_var`	If numeric : Column/List index which control shape sizes. This column/element has to be numeric. Can also be a column/element name or a vector of the same size than the input dataset. Set to NA if you don't want to control shape size.
`col_var`	If numeric : Column/List index which control shape colors. Can also be a column/element name or a vector of the same size than the input dataset. Set to NA if you don't want to control shape color.
`text_var`	If numeric : Column/List index which control text to add on shapes. Can also be a column/element name or a vector of the same size than the input dataset. Set to NA if you don't want to add text.
`shape_var`	If numeric = Similar to pch : square=15; circle=16; triangle=17. Can also be a column/element name or a vector of the same size than the input dataset.
`size_legend`	Custom name of shape legend.
`col_legend`	Custom name of shape color.
`shape_legend`	Name of the shape legend if shape_var is a vector.
`cols.use`	1 color or a vector containing multiple colors to color shapes. If coloring is continuous, default colors are taken from a "lightgrey" to "blue" gradient. If coloring is discrete, default colors are taken from the default ggplot2 palette.
`text.size`	Size of text to display on the shapes.
`text.vjust`	Vertical justification of text to display on the shapes. Default value = 0, which mean no justification. Recommended value is between -0.5 and 0.5.
`shape_use`	Shapes to uses (only when shape is controled by a discrete factor). Default shapes : \u25A0 \u25CF \u25C6 \u2BC8 \u2BC7 \u2BC6 \u2BC5 \u25D8 \u25D9 \u2726 \u2605 \u2736 \u2737.
`shape.scale`	Scale the size of the shapes, similar to cex.
`scale.by`	Scale the size by size or radius.
`scale.min`	Set lower limit for scaling, use NA for default values.
`scale.max`	Set upper limit for scaling, use NA for default values.
`plot.legend`	Plot the legends ?
`do.return`	Return ggplot2 object ?
`x.lab.pos`	Where to display x axis labels. This must be one of "bottom","top","both" or "none".

`y.lab.pos`	Where to display y axis labels. This must be one of "left","right","both"or "none".
`x.lab.size.factor`	Factor resizing x-axis labels (default=1)
`y.lab.size.factor`	Factor resizing y-axis labels (default=1)
`vertical_coloring`	Which color use to color the plot vertically ? (colors are repeated untill the end of the plot). Setting vertical and horizontal coloring at the same time is not recommended !
`horizontal_coloring`	Which color use to color the plot horizontally ? (colors are repeated untill the end of the plot). Setting vertical and horizontal coloring at the same time is not recommended !
`size.breaks.number`	Number of shapes with different size to display in the legend. Not used if size.breaks.values is not NA.
`color.breaks.number`	Number of labels for the color gradient legend. Not used if color.breaks.values is not NA.
`shape.breaks.number`	Number of shapes to display in the legend. Used when shape is controled by a continuous factor only. Not used if shape.breaks.values is not NA.
`size.breaks.values`	Vector containing numerical labels for the size legend.
`color.breaks.values`	Vector containing numerical labels for continuous color legend.
`shape.breaks.values`	Vector containing numerical labels for continuous shape legend.
`display_max_sizes`	Boolean : Display max shape size behind each shape ? (Default=TRUE)
`transpose`	Reverse x axis and y axis ?
`dend_x_var`	A vector containing Column/List indexes or Column/List names to compute the x axis dendrogramm.
`dend_y_var`	A vector containing Column/List indexes or Column/List names to compute the y axis dendrogramm.
`dist_method`	The distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".
`hclust_method`	The agglomeration method to be used. This must be one of "single", "complete", "average", "mcquitty", "ward.D", "ward.D2", "centroid" or "median".
`do.plot`	Print the plot ? (default=TRUE)

Value

Print the plot (if do.plot=TRUE) and return a list containing input data, executed command, resulting dot plot and computed dendrograms (if do.return=TRUE)

Author(s)

Simon Leonard - simon.leonard@univ-rennes1.fr

Examples

```
library(FlexDotPlot)
data(CBMC8K_example_data)
dotplot = dot_plot(data.to.plot=CBMC8K_example_data, size_var="RNA.avg.exp.scaled",
col_var="ADT.avg.exp.scaled", text_var="ADT.pct.exp.sup.cutoff",
shape_var="canonical_marker", shape_use = c("\u25CF","\u2737"),x.lab.pos="bottom",
y.lab.pos="left", cols.use=c("lightgrey","orange","red", "darkred"),
size_legend="RNA", col_legend="ADT", shape_legend="Canonical marker ?",
shape.scale =12, text.size=3, plot.legend = TRUE,
size.breaks.number=4, color.breaks.number=4, shape.breaks.number=5,
dend_x_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
dend_y_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
dist_method="euclidean",hclust_method="ward.D", do.return = TRUE)
```

`PBMC3K_example_data` *PBMC 3K example dataset*

Description

A dataset containing expression informations of 39 genes in 8 celltypes. Original dataset from seurat-data package.

Usage

```
data(PBMC3K_example_data)
```

Format

A data frame with 312 rows and 13 variables

Details

- genes.plot. Gene symbol
- id. Cell type
- pct.exp. Percentage of cells expressing the gene in the cell type
- pct.exp2. pct.exp with 2 decimal places
- pct.exp100. pct.exp.2 x 100
- avg.exp. Average expression of the gene in the cell type
- avg.exp.scale. Scaled average expression
- abs_avg_exp_scale Absolute value of avg.exp.scale
- avg_logFC. log fold-chage of the average expression of the gene between the cell type and the others

- avg_logFC2. avg_logFC with 2 decimal places
- p_val_adj. Adjusted p-value based on bonferroni correction
- p_val_adj2. p_val_adj with 2 decimal places
- pval_symb. Adjusted p-value classification ("<1e-100" or "<1e-50" or "<1e-10" or "<0.01" or ">0.01")

References

https://github.com/satijalab/seurat-data

`rotate_dot_plot_dendrogram`

Interactively rotate dendrograms from dot_plot outputs

Description

Take a output from dot_plot function and allow interactive dendrogram rotation with dendextend package

Usage

```
rotate_dot_plot_dendrogram(dot_plot_output, axis_to_rotate = c("x", "y"))
```

Arguments

`dot_plot_output`	Output from `dot_plot` function function
`axis_to_rotate`	Dendrogram to rotate "x" or "y"

Value

Print and return rotated dot plot

Author(s)

Simon Leonard - simon.leonard@univ-rennes1.fr

Examples

```
# Perform dot_plot
if(interactive()){
library(FlexDotPlot)
data(CBMC8K_example_data)

# Run dot_plot
dotplot_output = dot_plot(data.to.plot=CBMC8K_example_data, size_var="RNA.avg.exp.scaled",
dend_x_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
dend_y_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
```

```
dist_method="euclidean",hclust_method="ward.D", do.return = TRUE)

# The following command has to be run when the user
#is running example("rotate_dot_plot_dendrogram") only.
dotplot_output$command=call("dot_plot", data.to.plot=as.name("CBMC8K_example_data"),
size_var="RNA.avg.exp.scaled",
dend_x_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
dend_y_var=c("RNA.avg.exp.scaled","ADT.avg.exp.scaled"),
dist_method="euclidean",hclust_method="ward.D", do.return = TRUE)

# y  dendrogram rotation
r1=rotate_dot_plot_dendrogram(dotplot_output, axis_to_rotate = "y")
# add x dendrogram rotation to previous result
#r2=rotate_dot_plot_dendrogram(r1, axis_to_rotate = "x")
}
```

`Shiny_dot_plot` *Shiny dotplot*

Description

Shiny application to perform dot-plot pacman-plot

Usage

```
Shiny_dot_plot()
```

Author(s)

Simon Leonard - simon.leonard@univ-rennes1.fr

Examples

```
if(interactive()) Shiny_dot_plot()
```

Index