Package ‘easylabel’

October 13, 2022

Type Package

Title Interactive Scatter Plot and Volcano Plot Labels

Version 0.2.4

Maintainer Myles Lewis <`myles.lewis@qmul.ac.uk`>

Description Interactive labelling of scatter plots, volcano plots and Manhattan plots using a 'shiny' and 'plotly' interface. Users can hover over points to see where specific points are located and click points on/off to easily label them. Labels can be dragged around the plot to place them optimally. Plots can be exported directly to PDF for publication.

Language en-gb

License MIT + file LICENSE

Encoding UTF-8

LazyData true

biocViews

Suggests AnnotationDbi, knitr, org.Hs.eg.db, qvalue, rmarkdown

Imports plotly (>= 4.10.0), shiny, shinycssloaders, shinybusy, RColorBrewer, DT, gtools, splus2R

RoxygenNote 7.1.2

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Author Myles Lewis [aut, cre] (<`https://orcid.org/0000-0001-9365-5345`>), Katriona Goldmann [aut] (<`https://orcid.org/0000-0002-9073-6323`>), Cankut Cubuk [ctb] (<`https://orcid.org/0000-0003-4646-0849`>)

Repository CRAN

Date/Publication 2021-11-15 17:30:02 UTC

R topics documented:

`easylabel` *Interactive scatter plot labels*

Description

Interactive labelling of scatter plots using shiny/plotly interface.

Usage

```
easylabel(
  data,
  x,
  y,
  labs = NULL,
  startLabels = NULL,
  cex.text = 0.72,
  col = NULL,
  colScheme = NULL,
  alpha = 1,
  shape = NULL,
  shapeScheme = 21,
  size = 8,
  sizeRange = c(4, 80),
  xlab = x,
  ylab = y,
  xlim = NULL,
  ylim = NULL,
  xticks = NULL,
  yticks = NULL,
  showOutliers = TRUE,
  outlier_shape = 5,
  outline_col = "white",
  outline_lwd = 0.5,
  plotly_filter = NULL,
  width = 800,
  height = 600,
  showgrid = FALSE,
```

```
    zeroline = TRUE,
    hline = NULL,
    vline = NULL,
    mgp = c(1.8, 0.5, 0),
    Ltitle = "",
    Rtitle = "",
    LRtitle_side = 1,
    labelDir = "radial",
    labCentre = NULL,
    lineLength = 75,
    text_col = "black",
    line_col = "black",
    rectangles = FALSE,
    rect_col = "white",
    border_col = "black",
    padding = 3,
    border_radius = 5,
    showLegend = TRUE,
    legendxy = c(1.02, 1),
    filename = NULL,
    panel.last = NULL,
    fullGeneNames = FALSE,
    AnnotationDb = NULL,
    custom_annotation = NULL,
    output_shiny = TRUE,
    ...
  )
```

Arguments

`data`	Dataset (data.frame or data.table) to use for plot.
`x`	Specifies column of x coordinates in `data`.
`y`	Specifies column of y coordinates in `data`.
`labs`	Specifies the column in `data` with label names for points. Label names do not have to be unique. If `NULL` defaults to `rownames(data)`.
`startLabels`	Vector of initial labels. With a character vector, labels are identified in the column specified by `labs`. With a numeric vector, points to be labelled are referred to by row number.
`cex.text`	Font size for labels. Default 0.72 to match plotly font size. See `text()`.
`col`	Specifies which column in `data` affects point colour. Must be categorical. If it is not a factor, it will be coerced to a factor.
`colScheme`	A single colour or a vector of colours for points.
`alpha`	Alpha value for transparency of points.
`shape`	Specifies which column in `data` controls point shapes. If not a factor, will be coerced to a factor.
`shapeScheme`	A single symbol for points or a vector of symbols. See `pch` in `points()`.

`size`	Either a single value for size of points (default 8), or specifies which column in `data` affects point size for bubble charts.
`sizeRange`	Range of size of points for bubble charts.
`xlab`	x axis title. Accepts expressions when exporting base graphics. Set `cex.lab` to alter the font size of the axis titles (default 1). Set `cex.axis` to alter the font size of the axis numbering (default 1).
`ylab`	y axis title. Accepts expressions when exporting base graphics.
`xlim`	The x limits (x1, x2) of the plot.
`ylim`	The y limits of the plot.
`xticks`	List of custom x axis ticks and labels specified as a list of two named vectors `at = ...` and `labels = ...`. Another method is to use `xaxp` as a vector of the form `c(x1, x2, n)` giving the coordinates of the extreme tick marks and the number of intervals between tick-marks.
`yticks`	List of custom y axis ticks and labels specified as a list of two named vectors `at = ...` and `labels = ...`. Another method is to use `yaxp` as a vector of the form `c(y1, y2, n)` giving the coordinates of the extreme tick marks and the number of intervals between tick-marks.
`showOutliers`	Logical whether to show outliers on the margins of the plot.
`outlier_shape`	Symbol for outliers.
`outline_col`	Colour of symbol outlines. Set to `NA` for no outlines.
`outline_lwd`	Line width of symbol outlines.
`plotly_filter`	Refers to a column of logical values in `data` used to filter rows to reduce the number of points shown by plotly. We recommend using this for datasets with >100,000 rows. When saving to pdf, the full original dataset is still plotted. This is useful for plots with millions of points such as Manhattan plots where a subset of points to be labelled is already known.
`width`	Width of the plot in pixels. Saving to pdf scales 100 pixels to 1 inch.
`height`	Height of the plot in pixels.
`showgrid`	Either logical whether to show gridlines, or a character value where "x" means showing x axis gridlines and "y" means showing y axis gridlines.
`zeroline`	Logical whether to show lines at x = 0 and y = 0.
`hline`	Adds horizontal lines at values of y.
`vline`	Adds vertical lines at values of x.
`mgp`	The margin line for the axis title, axis labels and axis line. See `par()`.
`Ltitle`	A character or expression (see plotmath) value specifying text for left side title. Size of font can be changed using `cex.lab`.
`Rtitle`	A character or expression value specifying text for right side title. Size of font can be changed using `cex.lab`.
`LRtitle_side`	On which side of the plot for `Ltitle` and `Rtitle` (1 = bottom, 3 = top). See `mtext()`.

`labelDir`	Initial label line directions. Options include 'radial' (default) for radial lines around the centre of the plot, 'origin' for radial lines around the origin, 'horiz' for horizontal and 'vert' for vertical, 'xellipse' and 'yellipse' for near-horizontal and near-vertical lines arranged in an elliptical way around the centre, 'rect' for rectilinear lines (a mix of horizontal and vertical), 'x' for diagonal lines, 'oct' for lines in 8 directions around the centre.
`labCentre`	Coordinates in x/y units of the central point towards which radial labels converge. Defaults to the centre of the plot.
`lineLength`	Initial length of label lines in pixels.
`text_col`	Colour of label text. If set to `"match"` label text will match the colour of each point.
`line_col`	Colour of label lines. If set to `"match"` label line will match the colour of each point.
`rectangles`	Logical whether to show rectangles around labels (not supported by plotly).
`rect_col`	Colour for filling rectangles (not supported by plotly). If set to `"match"` rectangle fill colour will match the colour of each point.
`border_col`	Colour of rectangle borders (not supported by plotly). Use `border_col = NA` to omit borders. If set to `"match"` rectangle border colour will match the colour of each point.
`padding`	Amount of padding in pixels around label text.
`border_radius`	Amount of roundedness in pixels to apply to label rectangles (not supported by plotly).
`showLegend`	Logical whether to show or hide the legend.
`legendxy`	Vector of coordinates for the position of the legend. Coordinates are in plotly paper reference with `c(0, 0)` being the bottom left corner and `c(1, 1)` being the top right corner of the plot window. Plotly has unusual behaviour in that the x coordinate always aligns the left side of the legend. However, the y coordinate aligns the top, middle or bottom of the legend dependent on whether it is in the top, middle or bottom 1/3 of the plot window. So `c(1, 0)` positions the legend in the bottom right corner outside the right margin of the plot, while `c(1, 0.5)` centre aligns the legend around the centre of y axis.
`filename`	Filename for saving plots to pdf in a browser. Rstudio opens its own pdf file.
`panel.last`	An expression to be evaluated after plotting has taken place but before the axes, title and box are added. This can be useful for adding extra titles, legends or trend lines. Currently only works when saving plots using base graphics and does not work with plotly. See plot.default
`fullGeneNames`	Logical whether to expand gene symbols using Bioconductor AnnotationDbi package. With multiple matches, returns first value only. See `AnnotationDbi::mapIds()`.
`AnnotationDb`	Annotation database to use when expanding gene symbols. Defaults to human gene database `AnnotationDb = org.Hs.eg.db`.
`custom_annotation`	List of annotations to be added via `plotly::layout()`.
`output_shiny`	Logical whether to output a shiny app. If `FALSE` a plotly figure will be returned.

`...`	Further graphical parameters passed to `plot()` when saving via base graphics. The most useful for most users are likely to be `cex.lab` which alters axis title font size (default 1, see `par()`), `cex.axis` which alters axis numbering font size (default 1), and `panel.last` which allows additional plotting functions to be called after the main plot has been plotted but before the labels and label lines are drawn, which will allow the addition of trend lines, extra titles or legends for example (see `plot.default()`).

Details

Instructions:

- Hover over and click on/off genes which you want to label.
- When you have selected all your chosen genes, then drag gene names to move label positions.
- Click the save button to export a PDF in base graphics.
- The Table tab shows a table view of the dataset to help with annotation.

To export an SVG from plotly:

- Switch to SVG when finalised (only do this at last moment as otherwise editing is very slow).
- Press camera button in modebar to save image as SVG.

Value

By default no return value. If `output_shiny = FALSE` or the shiny button ’Export plotly & exit’ is pressed, a plotly figure is returned.

See Also

`easyVolcano()`, `easyMAplot()`

Examples

```
# Simple example using mtcars dataset
data(mtcars)
# Launch easylabel Shiny app: only run this example in interactive R sessions
if (interactive()) {
easylabel(mtcars, x = 'mpg', y = 'wt', col = 'cyl')
}
```

`easyManhattan` *Interactive Manhattan plot labels*

Description

Interactive labelling of Manhattan plots using 'shiny' and 'plotly' interface.

Usage

```
easyManhattan(
  data,
  chrom = "chrom",
  pos = "pos",
  p = "p",
  labs = "rsid",
  startLabels = NULL,
  pcutoff = 5e-08,
  chromGap = NULL,
  chromCols = c("royalblue", "skyblue"),
  sigCol = "red",
  alpha = 0.7,
  labelDir = "horiz",
  xlab = "Chromosome position",
  ylab = expression("-log"[10] ~ "P"),
  outline_col = NA,
  shapeScheme = 16,
  size = 6,
  width = 1000,
  lineLength = 60,
  npoints = 1e+06,
  nplotly = 1e+05,
  npeaks = NULL,
  span = min(c(nrow(data), npoints), na.rm = TRUE)/200,
  transpose = FALSE,
  filename = NULL,
  ...
)
```

Arguments

`data`	The dataset (data.frame or data.table) for the plot.
`chrom`	The column of chomosome values in `data`.
`pos`	The column of SNP positions in `data`.
`p`	The column of `p` values in `data`.
`labs`	The column of labels in `data`.

`startLabels`	Vector of initial labels. With a character vector, labels are identified in the column specified by `labs`. With a numeric vector, points to be labelled are referred to by row number.
`pcutoff`	Cut-off for p value significance. Defaults to 5E-08.
`chromGap`	Size of gap between chromosomes along the x axis in base pairs. If `NULL` this is automatically calculated dependent on the size of the genome. Default is around 3E07 for a human genome, and smaller for smaller genomes.
`chromCols`	A vector of colours for points by chromosome. Colours are recycled dependent on the length of the vector.
`sigCol`	Colour for statistically significant points. Ignored if set to `NA`.
`alpha`	Transparency for points.
`labelDir`	Option for label lines. See `easylabel()`.
`xlab`	x axis title. Accepts expressions.
`ylab`	y axis title. Accepts expressions.
`outline_col`	Colour of symbol outlines. Passed to `easylabel()`.
`shapeScheme`	A single symbol for points or a vector of symbols. Passed to `easylabel()`.
`size`	Specifies point size. Passed to `easylabel()`.
`width`	Width of the plot in pixels. Saving to pdf scales 100 pixels to 1 inch.
`lineLength`	Initial length of label lines in pixels.
`npoints`	Maximum number of points to plot when saving to pdf. Defaults to plot the top 1 million points by p value. Setting a value of `NA` will plot all points.
`nplotly`	Maximum number of points to display via plotly. We recommend the default setting of 100,000 points (or fewer).
`npeaks`	Number of peaks to label initially.
`span`	a peak is defined as the most significant SNP within a window of width span SNPs centred at that SNP. Large numbers take significantly longer.
`transpose`	Logical whether to transpose the plot.
`filename`	Filename for saving to pdf.
`...`	Other arguments passed to `easylabel()`.

Value

By default no return value. If `output_shiny = FALSE` or the shiny button 'Export plotly & exit' is pressed, a plotly figure is returned. See `easylabel()`.

See Also

`easylabel()` `easyVolcano()`

`easyMAplot` *Interactive MA plot labels*

Description

Interactive labelling of MA plots using shiny/plotly interface.

Usage

```
easyMAplot(
  data,
  x = NULL,
  y = NULL,
  padj = NULL,
  fdrcutoff = 0.05,
  colScheme = c("darkgrey", "blue", "red"),
  hline = 0,
  labelDir = "yellipse",
  xlab = expression("log"[2] ~ " mean expression"),
  ylab = expression("log"[2] ~ " fold change"),
  filename = NULL,
  showCounts = TRUE,
  useQ = FALSE,
  ...
)
```

Arguments

`data`	The dataset for the plot. Automatically attempts to recognises DESeq2 and limma objects.
`x`	Name of the column containing mean expression. For DESeq2 and limma objects this is automatically set.
`y`	Name of the column containing log fold change. For DESeq2 and limma objects this is automatically set.
`padj`	Name of the column containing adjusted p values (optional). For DESeq2 and limma objects this is automatically set. If `y` is specified and `padj` is left blank or equal to `y`, nominal unadjusted p values are used for cut-off for significance.
`fdrcutoff`	Cut-off for FDR significance. Defaults to FDR < 0.05. Can be vector with multiple cut-offs. To use nominal P values instead of adjusted p values, set `y` but leave `padj` blank.
`colScheme`	Colour colScheme. Length must match either length(fdrcutoff) + 1 to allow for non-significant genes, or match length(fdrcutoff) * 2 + 1 to accommodates asymmetric colour colSchemes for positive & negative fold change. (see examples).
`hline`	Vector of horizontal lines (default is y = 0).

`labelDir`	Option for label lines. See `easylabel()`.
`xlab`	x axis title. Accepts expressions.
`ylab`	y axis title. Accepts expressions.
`filename`	Filename for saving to pdf.
`showCounts`	Logical whether to show legend with number of differentially expressed genes.
`useQ`	Logical whether to convert nominal P values to q values. Requires the qvalue Bioconductor package.
`...`	Other arguments passed to `easylabel()`.

Value

By default no return value. If `output_shiny` = `FALSE` or the shiny button 'Export plotly & exit' is pressed, a plotly figure is returned. See `easylabel()`.

See Also

`easylabel()` `easyVolcano()`

---

easyVolcano *Interactive volcano plot labels*

Description

Interactive labelling of volcano plots using shiny/plotly interface.

Usage

```
easyVolcano(
  data,
  x = NULL,
  y = NULL,
  padj = y,
  fdrcutoff = 0.05,
  fccut = NULL,
  colScheme = c("darkgrey", "blue", "red"),
  xlab = expression("log"[2] ~ " fold change"),
  ylab = expression("-log"[10] ~ " P"),
  filename = NULL,
  showCounts = TRUE,
  useQ = FALSE,
  ...
)
```

Arguments

`data`	The dataset for the plot. Automatically attempts to recognises DESeq2 and limma objects.
`x`	Name of the column containing log fold change. For DESeq2 and limma objects this is automatically set.
`y`	Name of the column containing p values. For DESeq2 and limma objects this is automatically set.
`padj`	Name of the column containing adjusted p values (optional). If `y` is specified and `padj` is left blank or equal to `y`, nominal unadjusted p values are used for cut-off for significance instead of adjusted p values.
`fdrcutoff`	Cut-off for FDR significance. Defaults to FDR < 0.05. If `y` is specified manually and `padj` is left blank then this refers to the cut-off for significant points using nominal unadjusted p values.
`fccut`	Optional vector of log fold change cut-offs.
`colScheme`	Colour scheme. If no fold change cut-off is set, 2 colours need to be specified. With a single fold change cut-off, 3 or 5 colours are required, depending on whether the colours are symmetrical about x = 0. Accommodates asymmetric colour schemes with multiple fold change cut-offs (see examples).
`xlab`	x axis title. Accepts expressions.
`ylab`	y axis title. Accepts expressions.
`filename`	Filename for saving to pdf.
`showCounts`	Logical whether to show legend with number of differentially expressed genes.
`useQ`	Logical whether to convert nominal P values to q values. Requires the qvalue Bioconductor package.
`...`	Other arguments passed to `easylabel()`.

Value

By default no return value. If `output_shiny = FALSE` or the shiny button 'Export plotly & exit' is pressed, a plotly figure is returned. See `easylabel()`.

See Also

`easylabel()` `easyMAplot()`

`volc1` *Example volcano data for vignette*

Description

Example DESeq2 volcano data for vignette

Usage

`volc1`

Format

Data frame with 6 rows and 6 variables

`volc2` *Example volcano data for vignette*

Description

Example limma volcano data for vignette

Usage

`volc2`

Format

Data frame with 6 rows and 6 variables

Index