Package: GOplot (via r-universe)

September 10, 2024

Type Package

Title Visualization of Functional Analysis Data

Version 1.0.2

Date 2016-03-30

URL `https://github.com/wencke/wencke.github.io`

BugReports `https://github.com/wencke/wencke.github.io/issues`

Description Implementation of multilayered visualizations for enhanced graphical representation of functional analysis data. It combines and integrates omics data derived from expression and functional annotation enrichment analyses. Its plotting functions have been developed with an hierarchical structure in mind: starting from a general overview to identify the most enriched categories (modified bar plot, bubble plot) to a more detailed one displaying different types of relevant information for the molecules in a given set of categories (circle plot, chord plot, cluster plot, Venn diagram, heatmap).

Depends ggplot2 (>= 2.0.0), ggdendro (>= 0.1-17), gridExtra (>= 2.0.0), RColorBrewer (>= 1.1.2), R (>= 3.2.3)

License GPL-2

Suggests knitr, rmarkdown

VignetteBuilder knitr

LazyData TRUE

RoxygenNote 5.0.1

Repository https://wencke.r-universe.dev

RemoteUrl https://github.com/wencke/wencke.github.io

RemoteRef HEAD

RemoteSha 91963feeaf4847f1ccff049511fb230eac153cc2

Contents

`chord_dat` *Creates a binary matrix.*

Description

The function creates a matrix which represents the binary relation (1= is related to, 0= is not related to) between selected genes (row) and processes (column). The resulting matrix can be visualized with the `GOChord` function.

Usage

```
chord_dat(data, genes, process)
```

Arguments

`data`	A data frame with at least two coloumns: GO ID|term and genes. Each row contains exactly one GO ID|term and one gene. A column containing logFC values is optional and might be used if `genes` is missing.
`genes`	A character vector of selected genes OR data frame with coloumns for gene ID and logFC.
`process`	A character vector of selected processes

Details

If more than one logFC value for each gene is at disposal, only one should be used to create the binary matrix. The other values have to be added manually later.

Value

A binary matrix

See Also

GOChord

Examples

```
## Not run:
# Load the included dataset
data(EC)

# Building the circ object
circ <- circle_dat(EC$david, EC$genelist)

# Building the binary matrix
chord <- chord_dat(circ, EC$genes, EC$process)


## End(Not run)
```

`circle_dat` *Creates a plotting object.*

Description

The function takes the results from a functional analysis (for example DAVID) and combines it with a list of selected genes and their logFC. The resulting data frame can be used as an input for various ploting functions.

Usage

```
circle_dat(terms, genes)
```

Arguments

`terms`	A data frame with columns for 'category', 'ID', 'term', adjusted p-value ('adj_pval') and 'genes'
`genes`	A data frame with columns for 'ID', 'logFC'

Details

Since most of the gene- annotation enrichment analysis are based on the gene ontology database the package was build with this structure in mind, but is not restricted to it. Gene ontology is structured as an acyclic graph and it provides terms covering different areas. These terms are grouped into three independent `categories`: BP (biological process), CC (cellular component) or MF (molecular function).

The "ID" and "term" columns of the `terms` data frame refer to the ID and term description, whereas the ID is optional.

The "ID" column of the `genes` data frame can contain any unique identifier. Nevertheless, the identifier has to be the same as in "genes" from `terms`.

Examples

```
## Not run:
#Load the included dataset
data(EC)

#Building the circ object
circ<-circular_dat(EC$david, EC$genelist)

## End(Not run)
```

`EC` *Transcriptomic information of endothelial cells.*

Description

The data set contains the transcriptomic information of endothelial cells from two steady state tissues (brain and heart). More detailed information can be found in the paper by Nolan et al. 2013. The data was normalized and a statistical analysis was performed to determine differentially expressed genes. DAVID functional annotation tool was used to perform a gene- annotation enrichment analysis of the set of differentially expressed genes (adjusted p-value < 0.05).

Usage

```
data(EC)
```

Format

A list containing 5 items

Source

`http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE47067`

`GOBar` *Z-score coloured barplot.*

Description

Z-score coloured barplot of terms ordered alternatively by z-score or the negative logarithm of the adjusted p-value

Usage

```
GOBar(data, display, order.by.zscore = T, title, zsc.col, label.size)
```

Arguments

`data`	A data frame containing at least the term ID and/or term, the adjusted p-value and the z-score. A possible input can be generated with the `circle_dat` function
`display`	A character vector indicating whether a single plot ('single') or a facet plot with panels for each category should be drawn (default='single')
`order.by.zscore`	Defines the order of the bars. If TRUE the bars are ordered according to the z-scores of the processes. Otherwise the bars are ordered by the negative logarithm of the adjusted p-value
`title`	The title of the plot
`zsc.col`	Character vector to define the colour scale for the z-score of the form c(high, midpoint,low)
`label.size`	Defines the font size of x-axis text labels.

Details

If `display` is used to facet the plot the width of the panels will be proportional to the length of the x scale.

Examples

```
## Not run:
#Load the included dataset
data(EC)

#Building the circ object
circ<-circular_dat(EC$david, EC$genelist)

#Creating the bar plot
GOBar(circ)

#Faceting the plot
GOBar(circ, display='multiple')

## End(Not run)
```

`GOBubble` *Bubble plot.*

Description

The function creates a bubble plot of the input `data`. The input `data` can be created with the help of the `circle_dat` function.

Usage

```
GOBubble(data, display, title, colour, labels, ID = T, table.legend = T,
  table.col = T, bg.col = F)
```

Arguments

`data`	A data frame with coloumns for category, GO ID, term, adjusted p-value, z-score, count(num of genes)
`display`	A character vector. Indicates whether it should be a single plot ('single') or a facet plot with panels for each category (default='single')
`title`	The title (on top) of the plot
`colour`	A character vector which defines the colour of the bubbles for each category
`labels`	Sets a threshold for the displayed labels. The threshold refers to the -log(adjusted p-value) (default=5)
`ID`	If TRUE then labels are IDs else terms
`table.legend`	Defines whether a table of GO ID and GO term should be displayed on the right side of the plot or not (default = TRUE)
`table.col`	If TRUE then the table entries are coloured according to their category, if FALSE then entries are black
`bg.col`	Should only be used in case of a facet plot. If TRUE then the panel backgrounds are coloured according to the displayed category

Details

The x- axis of the plot represents the z-score. The negative logarithm of the adjusted p-value (corresponding to the significance of the term) is displayed on the y-axis. The area of the plotted circles is proportional to the number of genes assigned to the term. Each circle is coloured according to its category and labeled alternatively with the ID or term name.If static is set to FALSE the mouse hover effect will be enabled.

Examples

```
## Not run:
#Load the included dataset
data(EC)

#Building the circ object
circ <- circular_dat(EC$david, EC$genelist)

#Creating the bubble plot colouring the table entries according to the category
GOBubble(circ, table.col = T)

#Creating the bubble plot displaying the term instead of the ID and without the table
GOBubble(circ, ID = F, table.legend = F)

#Faceting the plot
GOBubble(circ, display = 'multiple')

## End(Not run)
```

`GOChord` *Displays the relationship between genes and terms.*

Description

The GOChord function generates a circularly composited overview of selected/specific genes and their assigned processes or terms. More generally, it joins genes and processes via ribbons in an intersection-like graph. The input can be generated with the `chord_dat` function.

Usage

```
GOChord(data, title, space, gene.order, gene.size, gene.space, nlfc = 1,
  lfc.col, lfc.min, lfc.max, ribbon.col, border.size, process.label, limit)
```

Arguments

`data`	The matrix represents the binary relation (1= is related to, 0= is not related to) between a set of genes (rows) and processes (columns); a column for the logFC of the genes is optional
`title`	The title (on top) of the plot
`space`	The space between the chord segments of the plot
`gene.order`	A character vector defining the order of the displayed gene labels
`gene.size`	The size of the gene labels
`gene.space`	The space between the gene labels and the segement of the logFC
`nlfc`	Defines the number of logFC columns (default=1)
`lfc.col`	The fill color for the logFC specified in the following form: c(color for low values, color for the mid point, color for the high values)
`lfc.min`	Specifies the minimium value of the logFC scale (default = -3)
`lfc.max`	Specifies the maximum value of the logFC scale (default = 3)
`ribbon.col`	The background color of the ribbons
`border.size`	Defines the size of the ribbon borders
`process.label`	The size of the legend entries
`limit`	A vector with two cutoff values (default= c(0,0)). The first value defines the minimum number of terms a gene has to be assigned to. The second the minimum number of genes assigned to a selected term.

Details

The `gene.order` argument has three possible options: "logFC", "alphabetical", "none", which are quite self- explanatory.

Maybe the most important argument of the function is `nlfc`.If your `data` does not contain a column of logFC values you have to set `nlfc = 0`. Differential expression analysis can be performed for multiple conditions and/or batches. Therefore, the data frame might contain more than one logFC value per gene. To adjust to this situation the `nlfc` argument is used as well. It is a numeric value and it defines the number of logFC columns of your `data`. The default is "1" assuming that most of the time only one contrast is considered.

To represent the data more useful it might be necessary to reduce the dimension of `data`. This can be achieved with `limit`. The first value of the vector defines the threshold for the minimum number of terms a gene has to be assigned to in order to be represented in the plot. Most of the time it is more meaningful to represent genes with various functions. A value of 3 excludes all genes with less than three term assignments. Whereas the second value of the parameter restricts the number of terms according to the number of assigned genes. All terms with a count smaller or equal to the threshold are excluded.

See Also

`chord_dat`

Examples

```
## Not run:
# Load the included dataset
data(EC)

# Generating the binary matrix
chord<-chord_dat(circ,EC$genes,EC$process)

# Creating the chord plot
GOChord(chord)

# Excluding process with less than 5 assigned genes
GOChord(chord, limit = c(0,5))

# Creating the chord plot genes ordered by logFC and a different logFC color scale
GOChord(chord,space=0.02,gene.order='logFC',lfc.col=c('red','black','cyan'))

## End(Not run)
```

`GOCircle` *Circular visualization of the results of a functional analysis.*

Description

The circular plot combines gene expression and gene- annotation enrichment data. A subset of terms is displayed like the `GOBar` plot in combination with a scatterplot of the gene expression data. The whole plot is drawn on a specific coordinate system to achieve the circular layout.The segments are labeled with the term ID.

Usage

```
GOCircle(data, title, nsub, rad1, rad2, table.legend = T, zsc.col, lfc.col,
  label.size, label.fontface)
```

Arguments

`data`	A special data frame which should be the result of `circle_dat`
`title`	The title of the plot
`nsub`	A numeric or character vector. If it's numeric then the number defines how many processes are displayed (starting from the first row of `data`). If it's a character string of processes then these processes are displayed
`rad1`	The radius of the inner circle (default=2)
`rad2`	The radius of the outer circle (default=3)
`table.legend`	Shall a table be displayd or not? (default=TRUE)
`zsc.col`	Character vector to define the colour scale for the z-score of the form c(high, midpoint,low)
`lfc.col`	A character vector specifying the colour for up- and down-regulated genes
`label.size`	Size of the segment labels (default=5)
`label.fontface`	Font style of the segment labels (default='bold')

Details

The outer circle shows a scatter plot for each term of the logFC of the assigned genes. The colours can be changed with the argument `lfc.col`.

The `nsub` argument needs a bit more explanation to be used wisely. First of all, it can be a numeric or a character vector. If it is a character vector then it contains the IDs or term descriptions of the displayed processes.If `nsub` is a numeric vector then the number defines how many terms are displayed. It starts with the first row of the input data frame.

See Also

`circle_dat`, `GOBar`

Examples

```
## Not run:
# Load the included dataset
data(EC)

# Building the circ object
circ <- circle_dat(EC$david, EC$genelist)

# Creating the circular plot
GOCircle(circ)

# Creating the circular plot with a different colour scale for the logFC
GOCircle(circ, lfc.col = c('purple', 'orange'))

# Creating the circular plot with a different colour scale for the z-score
GOCircle(circ, zsc.col = c('yellow', 'black', 'cyan'))

# Creating the circular plot with different font style
```

```
GOCircle(circ, label.size = 5, label.fontface = 'italic')

## End(Not run)
```

`GOCluster` *Circular dendrogram.*

Description

GOCluster generates a circular dendrogram of the `data` clustering using by default euclidean distance and average linkage.The inner ring displays the color coded logFC while the outside one encodes the assigned terms to each gene.

Usage

```
GOCluster(chord, process, metric, clust, clust.by, nlfc, lfc.col, lfc.min,
  lfc.max, lfc.space, lfc.width, term.col, term.space, term.width)
```

Arguments

`process`	A character vector of selected processes (ID or term description)
`metric`	A character vector specifying the distance measure to be used (default='euclidean'), see `dist`
`clust`	A character vector specifying the agglomeration method to be used (default='average'), see `hclust`
`clust.by`	A character vector specifying if the clustering should be done for gene expression pattern or functional categories. By default the clustering is done based on the functional categories.
`nlfc`	If TRUE `data` contains multiple logFC columns (default= FALSE)
`lfc.col`	Character vector to define the color scale for the logFC of the form c(high, midpoint,low)
`lfc.min`	Specifies the minimium value of the logFC scale (default = -3)
`lfc.max`	Specifies the maximum value of the logFC scale (default = 3)
`lfc.space`	The space between the leafs of the dendrogram and the ring for the logFC
`lfc.width`	The width of the logFC ring
`term.col`	A character vector specifying the colors of the term bands
`term.space`	The space between the logFC ring and the term ring
`term.width`	The width of the term ring
`data`	A data frame which should be the result of `circle_dat` in case the data contains only one logFC column. Otherwise `data` is a data frame whereas the first column contains the genes, the second the term and the following columns the logFCs of the different contrasts.

Details

The inner ring can be split into smaller rings to display multiply logFC values resulting from various comparisons.

Examples

```
## Not run:
#Load the included dataset
data(EC)

#Generating the circ object
circ<-circular_dat(EC$david, EC$genelist)

#Creating the cluster plot
GOCluster(circ, EC$process)

#Cluster the data according to gene expression and assigning a different color scale for the logFC
GOCluster(circ,EC$process,clust.by='logFC',lfc.col=c('darkgoldenrod1','black','cyan1'))

## End(Not run)
```

GOHeat *Displays heatmap of the relationship between genes and terms.*

Description

The GOHeat function generates a heatmap of the relationship between genes and terms. Biological processes are displayed in rows and genes in columns. In addition genes are clustered to highlight groups of genes with similar annotated functions. The input can be generated with the `chord_dat` function.

Usage

```
GOHeat(data, nlfc, fill.col)
```

Arguments

`data`	The matrix represents the binary relation (1= is related to, 0= is not related to) between a set of genes (rows) and processes (columns)
`nlfc`	Defines the number of logFC columns (default = 0)
`fill.col`	Defines the color scale break points

Details

The heatmap has in general two modes which depend on the `nlfc` argument. If `nlfc = 0`, so no logFC values are available, the coloring encodes for the overall number of processes the respective gene is assigned to. In case of `nlfc = 1` the color corresponds to the logFC of the gene.

Examples

```
## Not run:
# Load the included dataset
data(EC)

# Generate the circ object
circ <- circle_dat(EC$david, EC$genelist)

# Generate the chord object
chord <- chord_dat(circ, EC$genes, EC$process)

# Create the plot with user-defined colors
GOHeat(chord, nlfc = 1, fill.col = c('red', 'yellow', 'green'))

## End(Not run)
```

`GOVenn` *Venn diagram of differentially expressed genes.*

Description

The function compares lists of differentially expressed genes and illustrates possible relations.Additionally it represents the variety of gene expression patterns within the intersection in small pie charts with three segements. Clockwise are shown the number of commonly up- regulated, commonly down-regulated and contra- regulated genes.

Usage

```
GOVenn(data1, data2, data3, title, label, lfc.col, circle.col, plot = T)
```

Arguments

`data1`	A data frame consisting of two columns: ID, logFC
`data2`	A data frame consisting of two columns: ID, logFC
`data3`	A data frame consisting of two columns: ID, logFC
`title`	The title of the plot
`label`	A character vector to define the legend keys
`lfc.col`	A character vector determining the background colors of the pie segments representing up- and down- regulated genes
`circle.col`	A character vector to assign clockwise colors for the circles
`plot`	If TRUE only the venn diagram is plotted. Otherwise the function returns a list with two items: the actual plot and a list containing the overlap entries (default= TRUE)

Details

The `plot` argument can be used to adjust the amount of information that is returned by calling the function. If you are only interested in the actual plot of the venn diagram, `plot` should be set to TRUE. Sometimes you also want to know the elements of the intersections. In this case `plot` should be set to FALSE and the function call will return a list of two items. The first item, that can be accessed by $plot, contains the plotting information. Additionally, a list ($table) will be returned containing the elements of the various overlaps.

Examples

```
## Not run:
#Load the included dataset
data(EC)

#Generating the circ object
circ<-circular_dat(EC$david, EC$genelist)

#Selecting terms of interest
l1<-subset(circ,term=='heart development',c(genes,logFC))
l2<-subset(circ,term=='plasma membrane',c(genes,logFC))
l3<-subset(circ,term=='tissue morphogenesis',c(genes,logFC))

GOVenn(l1,l2,l3, label=c('heart development','plasma membrane','tissue morphogenesis'))

## End(Not run)
```

`reduce_overlap` *Eliminates redundant terms.*

Description

The function eliminates all terms with a gene overlap >= set threshold (`overlap`) The reduced dataset can be used to improve the readability of plots such as `GOBubble` and `GOBar`

Usage

```
reduce_overlap(data, overlap)
```

Arguments

`data`	A data frame created with `circle_dat`.
`overlap`	Skalar indicating the threshold for gene overlap (default = 0.75).

Details

The function is currently very slow.

Examples

```
## Not run:
# Load the included dataset
data(EC)

# Building the circ object
circ <- circle_dat(EC$david, EC$genelist)

# Eliminate redundant terms
reduced_circ <- reduce_overlap(circ)

# Plot reduced data
GOBubble(reduced_circ)


## End(Not run)
```

Index