Package 'PAFway'

February 5, 2020

Type Package

Title Pairwise Association of Functional Annotations

Version 0.1.3

Description Finds pairs of functional annotations or gene ontology (GO) terms that are enriched within a directed network (such as a gene regulatory network). This works with or without edge weights and includes visualizations (both as a network where the functions are nodes and as a heatmap). PAFway is an acronym for Pairwise Associations of Functional annotations in biological networks and pathWAYs.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports igraph, ggplot2, GGally, network, sna, scales, stats, grDevices

RoxygenNote 7.0.2

Suggests knitr, gplots, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Daphne Ezer [aut, cre], Mahair Mahjoub [aut]

Maintainer Daphne Ezer <daphne.ezer@york.ac.uk>

Repository CRAN

Date/Publication 2020-02-05 16:40:02 UTC

R topics documented:

`draw_heatmap` *Draw network of enriched functional annotation pairs as a heatmap*

Description

Draw network of enriched functional annotation pairs as a heatmap

Usage

```
draw_heatmap(
  graph,
  adjMethod = NULL,
  xlab = "downstream",
  ylab = "upstream",
  colPal = NULL
)
```

Arguments

`graph`	The output of either the pafway or pafway_edge_weight functions
`adjMethod`	The method for correcting for multiple hypotheses. This can be any method that is acceptable to the p.adjust function in the stats package: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" or "none". If this is NULL, then no adjustment will be made.
`xlab`	The label for the x-axis of the heatmap
`ylab`	The label for the y-axis of the heatmap
`colPal`	The color palette of the heatmap

Value

A matrix that has the same number of rows and columns as length(GOtypes). This will contain p-values.

Examples

```
nodes=paste("node", c(1:300))
set.seed(123)
randomGO=c("A", "B", "C", "D", "E", "F", "G", "H", "I",
"J", "K", "L", "M", "N")[sample(c(1:14), 300, replace=TRUE)]
names(randomGO)=nodes
edgesRandom=sapply(c(1:1000), function(i){
   nodes[sample(300, 2)]
})
getBinomPvalueRandom1=pafway(randomGO, t(edgesRandom), unique(randomGO))
draw_heatmap(getBinomPvalueRandom1)
colPal1=c(colorRampPalette(c("red3", "lightpink", "white", "white"))(20),
```

```
colorRampPalette(c("white", "white", "lightgreen", "darkgreen"))(20))
draw_heatmap(getBinomPvalueRandom1, adjMethod="bonferroni", xlab="Downstream",
ylab="Upstream", colPal=colPal1)
```

---

draw_network *Draw network of enriched functional annotation pairs*

Description

Draw network of enriched functional annotation pairs

Usage

```
draw_network(graph, pval = 0.05, adjMethod = NULL, seed = 123)
```

Arguments

graph	The output of either the pafway or pafway_edge_weight functions
pval	The threshold of p-value at which to draw an arrow
adjMethod	The method for correcting for multiple hypotheses. This can be any method that is acceptable to the p.adjust function in the stats package: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" or "none". If this is NULL, then no adjustment will be made.
seed	The random seed that will be used.

Value

A matrix that has the same number of rows and columns as length(GOtypes). This will contain p-values.

Examples

```
a=matrix(c(0.1, 0.003, 0.005, 0.004, 0.5, 0.7, 0.001, 0.002, 0.003), nrow=3)
colnames(a)=c('A', 'B', 'C')
rownames(a)=c('A', 'B', 'C')
draw_network(a)
```

pafway *Find pairwise-associations between annotations in a network without edge weights.*

Description

Find pairwise-associations between annotations in a network without edge weights.

Usage

```
pafway(GO, edges, GOtypes, exact = TRUE, adjustByEdgeCount = FALSE)
```

Arguments

GO	A vector of Strings, equal to the length of all the nodes. The names of the vector should be the names of the nodes. The values should either be the functional annotations, or a concatenation of the functional annotations, separated by a "_" symbol.
edges	A matrix of Strings, with at least two columns. Each row will represent an edge, linking the node in the first column to the node in the second column. Please make sure the node names are the same as those in "GO"
GOtypes	This is a vector that contains the functional annotations or GO terms that are of interest
exact	A boolean. If it is true, it will look for an exact match between the term in GOtypes and the vector GO. Otherwise, it will look for substrings.
adjustByEdgeCount	A boolean. If true, then the probability of observing a functional annotation will be calculated in terms of the number of nodes, but if it is false, it is calculated in terms of the number of edges that contain that node.

Value

A matrix that has the same number of rows and columns as length(GOtypes). This will contain p-values.

Examples

```
nodes=paste("node", c(1:10))
set.seed(123)
randomGO=c("A", "B", "C")[sample(c(1:3), 10, replace=TRUE)]
names(randomGO)=nodes
edgesRandom=sapply(c(1:100), function(i){
   nodes[sample(10, 2)]
})
pafway(randomGO, t(edgesRandom), unique(randomGO))
```

pafway_edge_weights *Find pairwise-associations between annotations in a network with edge weights.*

Description

Find pairwise-associations between annotations in a network with edge weights.

Usage

```
pafway_edge_weights(
  GO,
  edges,
  GOtypes,
  exact = TRUE,
  adjustByEdgeCount = FALSE,
  step = 0.001,
  thresholdZero = 1e-04
)
```

Arguments

GO	A vector of Strings, equal to the length of all the nodes. The names of the vector should be the names of the nodes. The values should either be the functional annotations, or a concatenation of the functional annotations, separated by a "_" symbol.
edges	A matrix of Strings, with at least three columns. Each row will represent an edge, linking the node in the first column to the node in the second column, and the third column will contain an edge weight. Please make sure the node names are the same as those in "GO"
GOtypes	This is a vector that contains the functional annotations or GO terms that are of interest
exact	A boolean. If it is true, it will look for an exact match between the term in GOtypes and the vector GO. Otherwise, it will look for substrings.
adjustByEdgeCount	A boolean. If true, then the probability of observing a functional annotation will be calculated in terms of the number of nodes, but if it is false, it is calculated in terms of the number of edges that contain that node.
step	FFT is used to speed up the calculation. In the first step, a certain number of values are evenly sampled from the function, across its range. This value will determine the distance between sampled points.
thresholdZero	In order to decrease the space and time requirements, values in the probability distributions that are below a certain threshold are set to be exactly zero. This is the threshold.

Value

A matrix that has the same number of rows and columns as length(GOtypes). This will contain p-values.

Examples

```
nodes=paste("node", c(1:10))
set.seed(123)
randomGO=c("A", "B", "C")[sample(c(1:3), 10, replace=TRUE)]
names(randomGO)=nodes
edgesRandom=sapply(c(1:20), function(i){
   nodes[sample(10, 2)]
})
getBinomPvalueRandom1=pafway_edge_weights(randomGO, cbind(t(edgesRandom),
rnorm(length(edgesRandom[1,]), 1, 0.001)), unique(randomGO))
```

---

pafway_meta *Compare the frequency of observed edges of a specific type and the expected frequency of these edges, given the presence of a different edge type*

Description

Compare the frequency of observed edges of a specific type and the expected frequency of these edges, given the presence of a different edge type

Usage

```
pafway_meta(GO, edges, GOtypes)
```

Arguments

GO	A vector of Strings, equal to the length of all the nodes. The names of the vector should be the names of the nodes. The values should either be the functional annotations, or a concatenation of the functional annotations, separated by a "_" symbol.
edges	A matrix of Strings, with at least two columns. Each row will represent an edge, linking the node in the first column to the node in the second column. Please make sure the node names are the same as those in "GO"
GOtypes	This is a vector that contains the functional annotations or GO terms that are of interest

Value

A matrix that has the same number of rows and columns as length(GOtypes)*length(GOtypes). The value at position [i,j] will contain ratio between the observed and expected frequency of edges of type i, based on the frequency of edges of type j.

Examples

```
GO=c(rep('A,C', 5), rep('A', 5), rep('C', 5), rep('B,D', 5), rep('B', 5), rep('D', 5))
names(GO)=paste("node", 1:length(GO))
edges=cbind(names(GO)[1:(length(GO)/2)], names(GO)[(length(GO)/2+1):length(GO)])
GOtypes=c('A', "B", "C", "D")
pafway_meta(GO, edges, GOtypes)
```

Index