Package 'PathQuant'

May 3, 2016

Type Package

Title Pathway Analysis with distance between genes and metablites using KEGG maps.

Version 0.1.1

Date 2015-07-22

Author Sarah Cherkaoui <cherkaos@gmail.com>
Sandra Therrien-Laperriere <sandra.therrien.laperriere@gmail.com>

Maintainer Sandra Therrien-Laperriere <sandra.therrien.laperriere@gmail.com>

Description Modeling of KEGG metabolism pathway maps to calculate shortest distance between gene-metabolite pairs to asses their biological relevance with tables, statistical and graphical outputs.

License GNU-3

Imports igraph, XML, RUnit, ggplot2, data.table, knitr,rmarkdown,scales,stringr, RCurl

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown

LazyData TRUE

RoxygenNote 5.0.1

R topics documented:

distributionGene *Distance distribution plots for single gene*

Description

Function ploting the distribution of distances between a gene and all measured metabolite, highlithing the distance of its associated metabolites.

Usage

```
distributionGene(pathwayId, association, metabolite, gene)
```

Arguments

pathwayId	KEGG Id of selected pathway.
association	Dataframe with 2 columns, where each line reprensents an associations. First column are the genes and the sencond column as the metabolites. Only use KEGG Ids.
metabolite	Dataframe of 1 column, representing all the measured metabolites. Only use KEGG Ids.
gene.	Gene selected Only use KEGG.

Details

The plot is depicted as frequency bars, which represent the number of metabolites at a given distance for the selected gene. Frequency bars are shown in grey for metabolites that are not associated with the selected gene and in red if there is at least one metabolite associated with this gene.

If a gene or a metabolite is present on multiple edges or nodes, then the shortest distance is selected.

Examples

```
distributionGene(metabolismOverviewMapKEGGId,
shinAndAlDF, completeMetaboDF, "hsa:1373")
```

getDistanceAll *Function calculating shortest distance between every gene in a gene-metabolite pairs and all metabolite.*

Description

Function calculating shortest distance between every gene in a gene-metabolite pairs of your association parameter and every metabolite (in a pair or not) on a graph model of KEGG map selected, where nodes are metabolites and reactions are edges.

Usage

```
getDistanceAll(pathwayId, gene, metabolite)
```

Arguments

pathwayId	KEGG Id of selected pathway.
gene	Dataframe of 1 column, representing all genes reported. Only use KEGG Ids.
metabolite	Dataframe of 1 column, representing all the measured metabolites. Only use KEGG Ids.

Details

If a gene or a metabolite is present on multiple edges or nodes, then shortest distance are calculated for every combinaison possible and the shortest distance is selected.

Output : dataframe with metabolite in columns and gene in rows with shortest distance values.

Examples

```
getDistanceAll(metabolismOverviewMapKEGGId,completeGeneDF,
          completeMetaboDF)
```

---

getDistanceAsso *Function calculating shortest distance between each gene-metabolite pairs.*

Description

Function calculating shortest distance between each gene-metabolite associations on your selected KEGG pathway.

Usage

```
getDistanceAsso(pathwayId, association, ordered = FALSE)
```

Arguments

pathwayId	KEGG Id of selected pathway.
association	Dataframe with 2 columns, where each line reprensents an associations. First column are the genes and the sencond column as the metabolites. Only use KEGG Ids.
ordered	[option] ascendent ordering of distance

Details

If a gene or a metabolite is present on multiple edges or nodes, then the shortest distance is selected. Output : dataframe with the following columns : geneCommonName, geneKEGGId, isGeneInMap, metaboliteCommonName, metaboliteKEGGId, isMetaboliteInMap, distance

Examples

```
getDistanceAsso("hsa01100",shinAndAlDF)
```

`heatmapAsso` *Function that output a heatmap to visualize distance calculated between every gene-metabolite associations in input.*

Description

Function calculting shortest distance between every genes and metabolites in a gene-metabolite pairs of your association parameter on a graph model of KEGG map selected, where nodes are metabolites and reactions are edges.

Usage

```
heatmapAsso(pathwayId, association)
```

Arguments

`pathwayId`	KEGG Id of selected pathway.
`association`	Dataframe with 2 columns, where each line reprensents an associations. First column are genes and the sencond column are metabolites. Only use KEGG Ids.

Details

If a gene or a metabolite is present on multiple edges or nodes, then the shortest distance is selected.

Output : Heatmap of distances calculated between associated genes-metabolites. Columns represent genes and rows represent metabolites. The calculated distance is shown in each cell with the corresponding color code (from red - closest; to yellow - farthest).

Examples

```
heatmapAsso("hsa01100", shinAndAlDF)
```

`permutationTest` *Statistical permutation test to asses the relevance of associated gene-metabolite pairs vs. randomly selected pairs*

Description

Permutation test to evauluate if gene-metabolite associations are significantly closer than randomly selected gene-metabolite pairs.

Usage

```
permutationTest(pathwayId, association, gene, metabolite, permutation,
  output = c("medians", "pvalue", "histogram"))
```

Arguments

pathwayId	KEGG Id of selected pathway.
association	Dataframe with 2 columns, where each line reprensents an associations. First column are genes and the sencond column are metabolites. Only use KEGG Ids.
gene	Dataframe of 1 column, representing all genes reported. Only use KEGG Ids.
metabolite	Dataframe of 1 column, representing all the measured metabolites. Only use KEGG Ids.
permutation	Number desired permutations
output	'medians' of all permutations, 'pvalue' of the permutation test, 'histogram' representing the distribution of all permutations' median

Details

If a gene or a metabolite is present on multiple edges or nodes, then the shortest distance is selected.

Examples

```
permutationFunction("hsa01100", shinAndAlDF,
            completeGeneDF,completeMetaboDF, 1000, "histogram")
```

Index