Package ‘SubpathwayGMir’

May 20, 2015

Type Package
Title Identify Metabolic Subpathways Mediated by MicroRNAs
Version 1.0
Date 2015-05-20
Author Li Feng, Chunquan Li and Xia Li
Maintainer Li Feng <biofengfeng@sina.com>
Description Routines for identifying metabolic subpathways mediated by microRNAs (miRNAs) through topologically locating miRNAs and genes within reconstructed Kyoto Encyclopedia of Genes and Genomes (KEGG) metabolic pathway graphs embedded by miRNAs. (1) This package can obtain the reconstructed KEGG metabolic pathway graphs with genes and miRNAs as nodes, through converting KEGG metabolic pathways to graphs with genes as nodes and compounds as edges, and then integrating miRNA-target interactions verified by low-throughput experiments from four databases (TarBase, miRecords, mirTarBase and miR2Disease) into converted pathway graphs. (2) This package can locate metabolic subpathways mediated by miRNAs by topologically analyzing the "le- nient distance" of miRNAs and genes within reconstructed KEGG metabolic pathway graphs. (3) This package can identify significantly enriched miRNA-mediated metabolic subpathways based on located subpathways by hypergenomic test. (4) This package can support six species for metabolic subpathway identification, such as caenorhabditis elegans, drosophila melanogaster, danio rerio, homo sapiens, mus musculus and rattus norvegi- cus, and user only need to update interested organism-specific environment variables.
Depends R (>= 3.0.2), XML, igraph
Collate fdr.est.R getBackground.R getEdgeLabel.R getEdgeLty.R
getInteGraphList.R getK2riData.R getLayout.R getLocSubGraph.R
mytriangle.R
LazyData Yes
License GPL (>= 2)
biocViews Statistics, Annotation, SubPathways, Graphs, MicroRNAs
NeedsCompilation no
Repository CRAN
Date/Publication 2015-05-20 14:15:38
The SubpathwayGMir package

Description

SubpathwayGMir is an R package for identifying metabolic subpathways mediated by microRNAs (miRNAs).

Introduction

SubpathwayGMir is an R package for identifying miRNA-mediated metabolic subpathways by topologically analyzing miRNAs and genes within reconstructed Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway graphs, which integrated miRNA-target interactions verified by low-throughput experiments.

Author(s)

Li Feng, Chunquan Li and Xia Li

See Also

getInteGraphList, getLocSubGraph, identifyGraph, initializeK2ri, updateOrgEnvir
getBackground

Get the background of molecules

Description
getBackground attempts to get the background of user-specified molecules.

Usage
getBackground(type = "gene_miRNA")

Arguments
type A character string. Should be one of "gene", "miRNA" or "gene_miRNA".

Details
The default background is obtained from the environment variable. For human, the reference gene background is all human genes in KEGG pathways. The reference miRNA background is collected from miRBase database.

Value
A character vector.

Author(s)
Li Feng, Chunquan Li and Xia Li

See Also
identifyGraph

Examples
## Not run:
## get all background of genes
bgGene <- getBackground(type="gene")

## get all background of miRNAs
bgMir <- getBackground(type="miRNA")

## get all background of miRNAs and genes
bgGMir <- getBackground(type="gene_miRNA")

## End(Not run)
getIntGraphList

Get the reconstructed metabolic pathway graphs

Description
Get the reconstructed KEGG metabolic pathway graphs embedded by miRNAs through integrating experimentally verified miRNA-target interactions.

Usage
getIntGraphList(graphList, relations)

Arguments
- graphList: A graphList. There nodes must be represented by genes.
- relations: A data frame. It contains two columns, the first is miRNA names and the second is its target names.

Details
The argument "relations" represents user-interested miRNA-target interactions, which can be returned from the getK2riData.

Value
A graph list.

Author(s)
Li Feng, Chunquan Li and Xia Li

See Also
plotGraph, getLocSubGraph, getK2riData

Examples
## Not run:
### Integrate miRNAs into KEGG pathway graphs ###

## get hsa-specified miRNA-target interactions ##
exPMir2Tar <- getK2riData(K2riData="expMir2Tar")
row1 <- which(exPMir2Tar[,["LowTHEXps"]]=="YES")
row2 <- which(exPMir2Tar[,["Species"]]=="hsa")
relations <- unique(exPMir2Tar[intersect(row1,row2),c(2:3)])

## get direct KEGG metabolic pathway graphs ##
graphList <- getK2riData(K2riData="MetabolicGEGEMGraph")
getK2riData

# get reconstructed pathway graph list #
InteGraphList <- getInteGraphList(graphList, relations)

# visualize the reconstructed pathways #
plotGraph(InteGraphList[[1]], layout=layout.random)

## get undirect KEGG metabolic pathway graphs ##
graphList <- getK2riData(K2riData="MetabolicGEGEUEMGraph")

## end (Not run) ##

---

GetK2riData

Get the environment data

Description

Get variables in current environment.

Usage

GetK2riData(K2riData)

Arguments

K2riData A character string. It must be one of them, including "expMir2Tar", "miRNA2Org", "BGMiRNA", "BGGene", "gene2symbol", "gene2path", "MetabolicGEGEUEMGraph" and "MetabolicGEGEEMGraph".

Details

The parameter K2riData is "expMir2Tar", which represents to obtain all miRNA-target interactions verified by experiments.
The parameter K2riData is "miRNA2Org", which represents to obtain miRNA-organism data.
The parameter K2riData is "BGMiRNA", which represents to obtain miRNA background data.
The parameter K2riData is "BGGene", which represents to obtain gene backgound data.
The parameter K2riData is "gene2symbol", which represents to obtain gene-symbol data.
The parameter K2riData is "gene2path", which represents to obtain gene-pathway data.
The parameter K2riData is "MetabolicGEGEUEMGraph", which represents to obtain undirect KEGG metabolic pathway graphs with genes as nodes.
The parameter K2riData is "MetabolicGEGEEMGraph", which represents to obtain direct KEGG metabolic pathway graphs with genes as nodes.
### getLocSubGraph

**Get the located metabolic subpathways**

**Description**

Locate metabolic subpathways mediated by miRNAs.

**Usage**

```r
getLocSubGraph(moleculeList, graphList, type="gene_miRNA", n=1, s=10, method = "shortestPaths")
```

**Arguments**

- **moleculeList**  
  A character vector. Such as differentially expressed miRNAs and/or genes under disease phenotypes.

- **graphList**  
  A graph list. There nodes must be represented by genes.

- **type**  
  A character string. Should be one of "gene", "miRNA" or "gene_miRNA".

- **n**  
  An integer. The maximum acceptable quantities of non-signature node at the shortest path between each two differential molecules.

- **s**  
  An integer. The minimum acceptable quantities of nodes in located subpathways.

- **method**  
  A character string. In which the shortest path algorithms will be used. See the function `get.shortest.paths`.

**Examples**

```r
## not run:
# obtain all miRNA-target interactions #
expMir2Tar <- GetK2riData(K2riData="expMir2Tar")
expMir2Tar[1:6,]

# obtain miRNA background #
BGmiRNA <- GetK2riData(K2riData="BGmiRNA")
BGmiRNA[1:10]

## endNot run
```

**Author(s)**

Li Feng, Chunquan Li and Xia Li

**See Also**

`updateOrgEnvir`
getLocSubGraph

Details

We apply lenient distance similarity method to locate metabolic subpathways mediated by miRNAs. We first map user interested differentially expressed miRNAs and/or genes to pathways as signatures. For a given pathway, we compute the shortest path between any two signatures. In shortest path, if the number of non-signature nodes between two signatures is no more than n, then these two signature nodes and other nodes at the shortest path are added into the same node set. We extract the corresponding subgraph in the pathway graph according to each node set. We finally define these subgraphs with node number $\geq s$ as the subpathway regions of the pathway. The argument $n$ is maximum number of permitted non-signature nodes at the shortest path between signature nodes. The default parameter $n=1$. The argument $s$ is used to filter subpathways in which the number of nodes are less than the parameter $s$. The default parameter $s=10$. The argument method determines which shortest path algorithms will be used. We set the default value as "get.shortest.paths".

Value

A list of graphs.

Author(s)

Li Feng, Chunquan Li and Xia Li

See Also

identifyGraph, get.shortest.paths

Examples

## not run:
### Integrate miRNAs to KEGG pathway graphs ###

### get hsa-specific miRNA-target interactions ###

```r
expMir2Tar <- GetK2riData(K2riData="expMir2Tar")
row1 <- which(expMir2Tar[["LowTHExps"]]=="YES")
row2 <- which(expMir2Tar[["Species"]]=="hsa")
relations <- unique(expMir2Tar[intersect(row1, row2), c(2:3)])
```

# get user-interested miRNAs and genes sets.
moleculeList <- c(getBackground(type="gene")[1:1000],
                 getBackground(type="miRNA")[1:2000])

### get direct KEGG metabolic pathway graphs ###

graphList <- GetK2riData(K2riData="MetabolicGEDEEGGraph")
# get reconstructed pathway graph list.
InteGraphList <- getInteGraphList(graphList, relations)
# get locate subpathways.
subGraphList <- getLocSubGraph(moleculeList, InteGraphList,
type="gene_miRNA", n=1, s=10)
# visualize the located subpathways.
plotGraph(subGraphList[[1]], layout=layout.random)
```
Annotate and identify subpathways

Description
Annotate user-interested molecules to pathways and identify significantly enriched subpathways.

Usage

```r
identifyGraph(moleculeList, graphList, type="gene_miRNA", background=getBackground(type), order="pvalue", decreasing=FALSE)
```

Arguments

- `moleculeList`: A character vector. Such as differentially expressed miRNAs and/or genes under disease phenotypes.
- `graphList`: A graph list. There nodes must be represented by genes or miRNAs and genes.
- `type`: A character string. Should be one of "gene", "miRNA" or "gene_miRNA".
- `background`: A character vector of molecules.
- `order`: A character string. Should be one of "pvalue", "fdr".
- `decreasing`: A logical. Should the sort be ordered by increasing or decreasing?

Details
The function can support the annotation and identification of metabolic subpathways based on genes, miRNAs or gene_miRNAs sets. The argument `moleculeList` supports three kinds of molecular sets: "genes", "miRNAs" or "gene_miRNAs".

The argument `type` represent the type of input molecules, including one of "genes", "miRNAs" or "gene_miRNA".

Detailed background information is provided in the function `getBackground`.

When many correlated subpathways are considered, the parameter `order` is used to order the pathways on the basis of "pvalue" or "fdr".

The parameter `decreasing` is set TRUE that represent the order would be performed by decreasing.
Value

A list. It includes: 'pathwayId', 'pathwayName', 'annMoleculeList', 'annMoleculeNumber', 'annBgMoleculeList', 'annBgNumber', 'MoleculeNumber', 'bgNumber', 'pvalue', and 'fdr', corresponding to pathway identifier, pathway name, the submitted molecules annotated to a pathway, the number of submitted molecules annotated to a pathway, the background molecules annotated to a pathway, the number of background molecules annotated to a pathway, the number of submitted molecules, the number of background molecules, p-value of the hypergeometric test, and Benjamini-Hochberg fdr values.

The background molecules annotated to a pathway are equal to all molecules in the pathway. For example, if the submitted molecules are human genes, the background molecules annotated to a pathway are equal to all human genes in the pathway.

The number of background molecules is the number of all molecules. For example, if the submitted molecules are human genes, the number of background molecules is equal to all human genes.

To visualize and save the results, the list can be converted to the data.frame by the function `printGraph`.

Note that `moleculeList` must be a 'character' vector. The genes must be represented by NCBI gene ids, and miRNAs must be represented by mature miRNA name in miRBase.

Author(s)

Li Feng, Chunquan Li and Xia Li

See Also

`printGraph`, `getBackground`, `GetK2riData`

Examples

```r
## Not run:
### annotate and identify subpathways ###
## get hsa-specific mirna-target interactions ##
expMir2Tar <- GetK2riData(K2riData="expMir2Tar")
row1 <- which(expMir2Tar[["LowTHExps"]]=="YES")
row2 <- which(expMir2Tar[["Species"]]=="hsa")
relations <- unique(expMir2Tar[intersect(row1,row2),c(2:3)])

# get user-interested miRNAs and genes sets
moleculeList <- c(getBackground(type="gene")[1:1000],
                  getBackground(type="miRNA")[1:2000])

## get direct KEGG metabolic pathway graphs ##
graphList <- GetK2riData(K2riData="MetabolicGEGEEMGraph")
# get reconstructed pathway graph list.
InteGraphList <- getIntegraphList(graphList, relations)
# get locate subpathways.
subGraphList <- getLocSubGraph(moleculeList,InteGraphList,
```

initializeK2ri

Initialize environment variable k2ri

Description

Initialize environment variable k2ri.

Usage

initializeK2ri()

Details

We can use the initializeK2ri to initialize the environment variable k2ri in current environment. The environment variable k2ri contains many informations. We can use the function ls to see all variables and use ls(k2ri) to see all informations in current environment, which include BGGene, BGmiRNA, expMir2Tar, gene2path, gene2symbol, mirNA2Org, MetabolicGEGUEMGraph, and MetabolicGEGEEUMGraph. We can use the function get to obtain one of them.

Author(s)

Li Feng, Chunquan Li and Xia Li
plotGraph

Examples

## not run:
# initialize environment k2ri.
initializeK2ri()

# see whether k2ri is exist in R or not.
ls()

# see all environment variable contained in k2ri.
ls(k2ri)

## End(Not run)

---

plotGraph  Visualize a pathway graph

Description

Visualize a pathway graph.

Usage

```r
plotGraph(graph, margin=0, vertex.label.cex=0.6, vertex.label.font=1,
    vertex.size=8, vertex.size2=6, edge.arrow.size=0.2,
    edge.arrow.width=3, vertex.label=V(graph)$graphics_name,
    vertex.shape=V(graph)$graphics_type, layout=getLayout(graph),
    vertex.label.color="black", vertex.color=V(graph)$graphicsbgcolor,
    vertex.frame.color="dimgray", edge.color="dimgray",
    edge.label=getEdgeLabel(graph), edge.label.cex=0.6,
    edge.label.color="dimgray", edge.lty=getEdgeLty(graph),
    axes=FALSE, xlab="", ylab="", sub=NULL, main=NULL,...)
```

Arguments

- **graph**: The igraph object of a pathway graph.
- **margin**: A numeric. The value is usually between -0.5 and 0.5, which is able to zoom in or out a pathway graph. The default is 0.
- **vertex.label.cex**: A numeric vector of node label size.
- **vertex.label.font**: A numeric vector of label font.
- **vertex.size**: A numeric vector of Node size. See `plot.igraph`
- **vertex.size2**: A numeric vector of Node size.
- **edge.arrow.size**: Edge arrow size. The default is 0.2.
edge.arrow.width
Edge arrow width. The default is 3.

vertex.label
A vector of node label. The default is graphics_name.

vertex.shape
A vector of node shape. The default is graphics_type.

layout
A matrix of x-y coordinates with two dims. Determine the placement of the nodes for drawing a graph. The default is the KEGG node coordinates that are originally obtained from the KGML file.

vertex.label.color
A vector of node label colors. The default is black.

vertex.color
A vector of node colors. The default is the KEGG node color.

vertex.frame.color
A vector of node frame color. The default is dimgray.

edge.color
A vector of edge color. The default is dimgray.

edge.label
A vector of edge label.

edge.label.cex
Edge label size.

edge.label.color
A vector of edge label color. The default is dimgray.

edge.lty
A vector of line type for the edges.

axes
A logical. whether to plot axes. The default is FALSE.

xlab
A character string. The label of the horizontal axis. The default is the empty string.

ylab
A character string. The label of the vertical axis. The default is the empty string.

sub
A character string of subtitle.

main
A character string of main title.

...
The arguments passed to or from methods. See plot.

Details
The function plotGraph is able to display a pathway graph.
The argument layout is used to determine the placement of the nodes for drawing a graph. There are mainly two preprocessed methods to determine the placement of the nodes for drawing a pathway graph: the KEGG pathway layout and layout provided in the function plot.igraph of the igraph package. The default layout is the KEGG layout, for which the coordinates of nodes in KEGG is used to determine the placement of the nodes for drawing a graph. Therefore, the returned figure by the function may be very similar to the KEGG pathway graph when information in the pathway graph is complete relatively. The layouts provided in igraph include layout.reingold.tilford, layout.random, layout.circle, layout.sphere, ... .

Author(s)
Li Feng, Chunquan Li and Xia Li

See Also
plot, layout.random
Examples

```r
## Not run:  
### get metabolic pathway graphs ###  
g <- getK2riData(K2riData="MetabolicGEEEMGraph")
# visualize the graph
plotGraph(g[[1]], layout=layout.random)

## End(Not run)
```

printGraph  

Print the results of graph annotation and identification

Description

Print the results of graph annotation and identification.

Usage

```r
printGraph(ann, detail=FALSE)
```

Arguments

- `ann`: A list. The value was returned from the function `identifyGraph`.
- `detail`: A logical. If true, gene lists from the function `identifyGraph` are converted into strings, which are used to display and write results with genes.

Value

A data.frame. Columns include `pathwayId`, `pathwayName`, `annMoleculeRatio`, `annBgRatio`, `pvalue`, `fdr`, `annMoleculeList`, `annBgMoleculeList`. Detailed information is provided in `identifyGraph`.

Author(s)

Li Feng, Chunquan Li and Xia Li

See Also

- `identifyGraph`
## Not run:

```r
# get hsa-specific miRNA-target interactions
expMir2Tar <- getK2riData(K2riData="expMir2Tar")
row1 <- which(expMir2Tar["LowTHEXps"] == "YES")
row2 <- which(expMir2Tar["Species"] == "hsa")
relations <- unique(expMir2Tar[intersect(row1, row2), c(2:3)])

# get direct KEGG metabolic pathway graphs
graphList <- getK2riData(K2riData="MetabolicGEGEMGraph")

# get reconstructed pathway graph list.
InteGraphList <- getIntegraphList(graphList, relations)

# get user-interested miRNAs and genes sets.
moleculeList <- c(getBackground(type="gene")[1:1000],
geBackground(type="miRNA")[1:2000])

# get locate subpathways.
subGraphList <- getLocSubGraph(moleculeList, InteGraphList,
                              type="gene_miRNA", n=1, s=10)

# annotate and identify subpathways.
ann <- identifyGraph(moleculeList, subGraphList, type="gene_miRNA")

# convert ann to a data frame.
result <- printGraph(ann, detail=TRUE)

# save the result.
write.table(head(result), "result.txt", sep=\"\t\", col.names=TRUE, row.names=FALSE)
```

## End(Not run)

---

**updateOrgEnvir**

*Update the organism-specific environment variables*

**Description**

Update the organism-specific environment variables that user interested.

**Usage**

```r
updateOrgEnvir(org = "hsa", path = "http://rest.kegg.jp", verbose = TRUE)
```
**updateOrgEnvir**

**Arguments**

- **org**  
  A character string. It supports six species and must be the abbreviation of a genome name, such as cel (caenorhabditis elegans), dme (drosophila melanogaster), dre (danio rerio), hsa (homo sapiens), mmu (mus musculus) and rno (rattus norvegicus).

- **path**  
  A character string. The reference path for downloading organism-specific data.

- **verbose**  
  A logical. If TRUE, the additional diagnostics are printed.

**Details**

This package supports to identify metabolic subpathways among six organisms. We only need to update the organism-specific environment variables before subpathway identification. The six organisms contain cel (caenorhabditis elegans), dme (drosophila melanogaster), dre (danio rerio), hsa (homo sapiens), mmu (mus musculus) and rno (rattus norvegicus). The default value of the argument **org** is "hsa" (human).

**Author(s)**

Li Feng, Chunquan Li and Xia Li

**Examples**

```r
## Not run:

## update organism and the type of gene identifiers ##

updateOrgEnvir("mmu")

# show the current environment variables
ls(k2ri)

# show the background of miRNAs
k2ri$BGMiRNA[1:3]

## End(Not run)
```
Index

*Topic file
  getBackground, 3
  getInteGraphList, 4
  GetK2riData, 5
  getLocSubGraph, 6
  identifyGraph, 8
  initializeK2ri, 10
  plotGraph, 11
  printGraph, 13
  updateOrgEnvir, 14

get, 10
get.shortest.paths, 6, 7
getBackground, 3, 8, 9
getInteGraphList, 2, 4
GetK2riData, 4, 5, 9
getLocSubGraph, 2, 4, 6

identifyGraph, 2, 3, 7, 8, 13
initializeK2ri, 2, 10, 10

layout, 12
layout.random, 12
ls, 10

plot, 12
plot.igraph, 11, 12
plotGraph, 4, 11
printGraph, 9, 13

SubpathwayGMir
  (SubpathwayGMir-package), 2
SubpathwayGMir-package, 2

updateOrgEnvir, 2, 6, 14