

# Package ‘SubpathwayLNCE’

January 19, 2016

**Type** Package

**Title** Identify Signal Subpathways Competitively Regulated by LncRNAs  
Based on ceRNA Theory

**Version** 1.0

**Date** 2016-1-15

**Author** Xinrui Shi, Chunquan Li and Xia Li

**Maintainer** Xinrui Shi <xinrui103@163.com>

**Description** Identify dysfunctional subpathways competitively regulated by lncRNAs through integrating lncRNA-mRNA expression profile and pathway topologies.

**Depends** R (>= 2.10), igraph, RBGL, utils, BiasedUrn, graph, graphics, stats

**Suggests** XML

**Collate** ExpProcess.r getBackgroundLnc.r getEdgeLabel.r getEdgeLty.r  
getEnvironmentData.r getInteGraphList.r getInteUMGraph.r  
getKGeneFromGene.r getLayout.r getLncGenePairs.r  
getExampleData.r getLocSubGraphLnc.r getNodeLabel.r  
getOneNodePath.r getSymbolFromGene.r identifyLncGraphW.r  
plotAnnGraph.r plotGraphL.r printGraphW.r

**LazyData** Yes

**License** GPL (>= 2)

**biocViews** Statistics, SuPathways, LncRNAs, enrichment analysis, ceRNA

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-01-19 08:17:59

## R topics documented:

envData	2
getBackgroundLnc	2
GetExampleData	3
getInteGraphList	5

getInteUMGraph . . . . .	6
getLncGenePairs . . . . .	7
getLocSubGraphLnc . . . . .	8
identifyLncGraphW . . . . .	9
plotAnnGraph . . . . .	11
plotGraphL . . . . .	12
printGraphW . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

envData	<i>The variables in the environment variable envData of the system</i>
---------	--

---

### Description

The variables in the environment variable envData of the system.

### Format

An environment variable

### Author(s)

Xinrui Shi, Chunquan Li and Xia Li

---

getBackgroundLnc	<i>Get the background of molecules</i>
------------------	--

---

### Description

getBackgroundLnc attempts to get the background of user-specified molecules.

### Usage

```
getBackgroundLnc(type = "gene_lncRNA")
```

### Arguments

type	A character string. Should be one of "gene", "lncRNA" or "gene_lncRNA".
------	---

### 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 lncRNA background is collected from GeneCode database.

**Value**

A character vector.

**Author(s)**

Xinrui Shi, Chunquan Li and Xia Li

**See Also**

[identifyLncGraphW](#)

**Examples**

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

## get all background of lncRNAs
bgLncRNA <- getBackgroundLnc(type="lncRNA")

## get all background of lncRNAs and genes
bgGL <- getBackgroundLnc(type="gene_lncRNA")

## End(Not run)
```

---

GetExampleData

*Get the environment data*

---

**Description**

Get variables in current environment.

**Usage**

```
GetExampleData(exampleData)
```

**Arguments**

`exampleData` A character string. It must be one of them, including "code", "compound", "g2", "gene", "gene2path", "gene2symbol", "GeneExp", "keggGene2gene", "lnc2Name", "lncBackground", "LncExp", "LncGenePairs", "mart", "nocode", "pp", "resultT" and "sub".

**Details**

The parameter `exampleData` is "code", which represents to obtain all human coding gene collected from Genecode.

The parameter `K2riData` is "compound", which represents to obtain all human compound.

The parameter `K2riData` is "g2", which represents to obtain undirect KEGG metabolic pathway graphs with genes as nodes.

The parameter `K2riData` is "gene", which represents to obtain interested gene of example.

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 "GeneExp", which represents to obtain gene expression profile of example.

The parameter `K2riData` is "keggGene2gene", which represents to obtain kegggene-gene data.

The parameter `K2riData` is "lnc2Name", which represents to obtain lncRNA Ensemble-Name data.

The parameter `K2riData` is "lncBackground", which represents to obtain background of lncRNA, gene or lncRNA-gene.

The parameter `K2riData` is "lncExp", which represents to obtain lncRNA expression profile of example.

The parameter `K2riData` is "lncGenePairs", which represents to obtain the competitively regulated lncRNA-mRNA interactions.

The parameter `K2riData` is "mart", which represents to obtain gene ensemble-name-entrez data.

The parameter `K2riData` is "nocode", which represents to obtain all human lncRNA collected from Genecode.

The parameter `K2riData` is "resultT", which represents to obtain result of example.

The parameter `K2riData` is "sub", which represents to obtain subpathway of example.

The parameter `K2riData` is "geneLnc", which represents to obtain user-interested genes and lncRNAs of example.

**Author(s)**

Xinrui Shi, Chunquan Li and Xia Li

**Examples**

```
## Not run:  
  
# obtain lncRNA expression profile #  
LncExp <- GetExampleData(exampleData="LncExp")  
  
## End(Not run)
```

---

getInteGraphList	<i>Get the reconstructed signal pathway graphs</i>
------------------	--

---

### Description

Get the reconstructed KEGG signal pathway graphs embedded by lncRNAs of competitive regulation.

### Usage

```
getInteGraphList(graphList,relations)
```

### Arguments

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

### Details

The argument "relations" represents user-interested lncRNA-target interactions, which can be returned from the GetExampleData.

### Value

A graph list.

### Author(s)

Xinrui Shi, Chunquan Li and Xia Li

### See Also

[plotGraphL](#), [getLocSubGraphLnc](#), [GetExampleData](#)

### Examples

```
## Not run:
### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
g2<-GetExampleData(exampleData="g2")
interUMGraph<-getInteGraphList(g2,LncGenePairs)
#interUMGraph<-getInteUMGraph(LncGenePairs)
### get user-interested lncRNAs and genes sets.
#geneLnc<-GetExampleData(exampleData="geneLnc")
# get locate subpathways.
#sub<-getLocSubGraphLnc(geneLnc,interUMGraph,type="gene_lncRNA",n=1,s=8)
#SubcodeLncResult<-identifyLncGraphW(geneLnc,sub,type="gene_lncRNA",bet=1)
#resultT<-printGraphW(SubcodeLncResult,detail=TRUE)
```

```
## End(Not run)
```

---

getInteUMGraph	<i>Get the reconstructed condition-specific lncRNA-regulated signal pathways</i>
----------------	--

---

### Description

Get the reconstructed condition-specific LncRNA-regulated signal pathways embedded by lncRNAs of competitive regulation through linking to their regulated-mRNAs.

### Usage

```
getInteUMGraph(LncGenePairs)
```

### Arguments

LncGenePairs    A data frame. It contains two columns, the first is lncRNA names and the second is mRNA names.

### Details

The argument "LncGenePairs" represents competitively regulated lncRNA-mRNA interactions, which can be returned from the getLncGenePairs.

### Value

A graph list.

### Author(s)

Xinrui Shi, Chunquan Li and Xia Li

### See Also

[getLncGenePairs](#)

### Examples

```
## Not run:

### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
inteUMGraph<-getInteUMGraph(LncGenePairs)

## End(Not run)
```

---

getLncGenePairs      *Get the co-express lncRNA-mRNA interactions*

---

**Description**

Get the co-express lncRNA-mRNA interactions reached a significant positive threshold. These lncRNAs-mRNAs pairs are competitively regulated.

**Usage**

```
getLncGenePairs(GeneExp, LncExp, a=0.025)
```

**Arguments**

GeneExp	A gene expression profile. The first row of gene expression profile is names of mRNA, and the gene expression profile and the lncRNA expression profile must be matched.
LncExp	A lncRNA expression profile. The first row of lncRNA expression profile is names of lncRNA, and the gene expression profile and the lncRNA expression profile must be matched.
a	A parameter, namely the significant positive threshold.

**Details**

Co-express coefficient for any pair of relations in the candidate lncRNA-mRNA network is calculated basing on matched lncRNA and mRNA expression profiles, those relations had reached a significant positive threshold were retained.

**Value**

A data frame. The data frame has two columns, the first is lncRNA names, the other is mRNA names.

**Author(s)**

Xinrui Shi, Chunquan Li and Xia Li

**Examples**

```
## Not run:  
  
GeneExp<-GetExampleData(exampleData="GeneExp")  
LncExp<-GetExampleData(exampleData="LncExp")  
LncGenePairs<-getLncGenePairs(GeneExp, LncExp, a=0.025)  
  
## End(Not run)
```

---

getLocSubGraphLnc      *Get the located signal subpathways*

---

**Description**

Locate signal subpathways competitively regulated by lncRNAs.

**Usage**

```
getLocSubGraphLnc(moleculeList, graphList, type="gene_lncRNA",  
                  n=1, s=8, method = "shortestPaths")
```

**Arguments**

moleculeList	A character vector. Such as competitively regulated lncRNAs and/or differentially expressed genes under disease phenotypes. Gene should be geneid.
graphList	A graph list. There nodes must be represented by genes.
type	A character string. Should be one of "gene", "lncRNA" or "gene_lncRNA".
n	An integer. The maximum acceptable quantities of non-signature node at the shortest path between each two interested molecules.
s	An integer. The minimum acceptable quantities of nodes in located subpathways.
method	A character string. In which the shorest path algorithms will be used.

**Details**

We apply lenient distance similarity method to locate signal subpathways competitively regulated by miRNAs. We first map user competitively regulated lncRNAs and/or interested 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  $s=8$ . 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)**

Xinrui Shi, Chunquan Li and Xia Li



**See Also**

[getInteGraphList](#), [identifyLncGraphW](#)

**Examples**

```
## Not run:

### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
interUMGraph<-getInteUMGraph(LncGenePairs)
### get user-interested lncRNAs and genes sets.
##geneLnc<-c(getBackground(type="gene")[1:3000],unique(LncGenePairs[1,]))
geneLnc<-GetExampleData(exampleData="geneLnc")
# get locate subpathways.
sub<-getLocSubGraphLnc(geneLnc,interUMGraph,type="gene_lncRNA",n=1,s=8)

## End(Not run)
```

---

identifyLncGraphW	<i>Annotate and identify subpathways</i>
-------------------	--

---

**Description**

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

**Usage**

```
identifyLncGraphW(moleculeList,graphList,type="gene_lncRNA",
                  background=getBackgroundLnc(type),
                  order="pvalue",decreasing=FALSE,bet=1)
```

**Arguments**

moleculeList	A character vector. Such as user-interested lncRNAs and/or genes under disease phenotypes.
graphList	A graph list. There nodes must be represented by genes or lncRNAs and genes.
type	A character string. Should be one of "gene", "lncRNA" or "gene_lncRNA".
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?
bet	A number. A parameter.

## Details

The function can support the annotation and identification of metabolic subpathways based on genes, lncRNAs or gene\_lncRNAs sets. The argument `moleculeList` supports three kinds of molecular sets: "genes", "lncRNAs" or "gene\_lncRNAs".

The argument `type` represent the type of input molecules, including one of "genes", "lncRNAs" or "gene\_lncRNA".

Detailed background information is provided in the function [getBackgroundLnc](#).

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 include: 'pathwayId', 'pathwayName', 'annMoleculeList', 'annMoleculeNumber', 'annBgMoleculeList', 'annBgNumber', 'MoleculeNumber', 'bgNumber', 'annWeight', '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, the weight of lncRNA competitively regulated subpathway 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 [printGraphW](#).

Note that `moleculeList` must be a 'character' vector. The genes must be represented by NCBI gene ids, and lncRNAs must be represented by mature lncRNA Ensemble name.

## Author(s)

Xinrui Shi, Chunquan Li and Xia Li

## See Also

[printGraphW](#), [getBackgroundLnc](#), [GetExampleData](#)

## Examples

```
## Not run:

### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
#LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
#inteUMGraph<-getInteUMGraph(LncGenePairs)
### get user-interested lncRNAs and genes sets.
geneLnc<-GetExampleData(exampleData="geneLnc")
```

```
# get locate subpathways.
#sub<-getLocSubGraphLnc(geneLnc, interUMGraph, type="gene_lncRNA", n=1, s=8)
sub<-GetExampleData(exampleData="sub")
SubcodeLncResult<-identifyLncGraphW(geneLnc, sub, type="gene_lncRNA", bet=1)

## End(Not run)
```

---

plotAnnGraph

*Visualize annotated pathway graphs*


---

## Description

Visualize annotated pathway graphs.

## Usage

```
plotAnnGraph(pathwayId, graphList, ann, gotoKEGG=FALSE, orgSpecific=TRUE,
displayInR=TRUE, match=TRUE, vertex.frame.color="red", ...)
```

## Arguments

pathwayId	A character vector of pathway identifier, e.g., c("path:00010", "path:00020").
graphList	A graph list.
ann	A list, e.g., the return value of the function <a href="#">identifyLncGraphW</a> .
gotoKEGG	A logical. If TRUE, go to KEGG web site. The identified nodes are colored red.
orgSpecific	A logical.
match	A logical.
displayInR	A logical. If TRUE, display pathway graphs on the device.
vertex.frame.color	A character defining the frame color of the nodes which represent the annotated nodes in the pathway.
...	The arguments passed to or from methods. See <a href="#">plotGraphL</a> , <a href="#">plot.igraph</a> and see <a href="#">plot</a> .

## Details

The function can visualize the annotated pathway (or subpathways) graphs. The red nodes in the result graph represent the nodes which include the submitted molecules.

## Author(s)

Xinrui Shi, Chunquan Li, Xia Li

## See Also

[plotGraphL](#), [identifyLncGraphW](#), [printGraphW](#)

## Examples

```
## Not run:
### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
g2<-GetExampleData(exampleData="g2")
interUMGraph<-getInteGraphList(g2,LncGenePairs)
interUMGraph<-getInteUMGraph(LncGenePairs)
### get user-interested lncRNAs and genes sets.
geneLnc<-GetExampleData(exampleData="geneLnc")
# get locate subpathways.
sub<-getLocSubGraphLnc(geneLnc,interUMGraph,type="gene_lncRNA",n=1,s=8)
SubcodeLncResult<-identifyLncGraphW(geneLnc,sub,type="gene_lncRNA",bet=1)
resultT<-printGraphW(SubcodeLncResult,detail=TRUE)
plotAnnGraph("path:04020_1",sub,SubcodeLncResult,gotoKEGG=FALSE)

## End(Not run)
```

---

plotGraphL

*Visualize a pathway graph*

---

## Description

Visualize a pathway graph.

## Usage

```
plotGraphL(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)$graphics_bgcolor,
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.

<code>vertex.size</code>	A numeric vector of Node size. See <a href="#">plot.igraph</a>
<code>vertex.size2</code>	A numeric vector of Node size.
<code>edge.arrow.size</code>	Edge arrow size. The default is 0.2.
<code>edge.arrow.width</code>	Edge arrow width. The default is 3.
<code>vertex.label</code>	A vector of node label. The default is <code>graphics_name</code> .
<code>vertex.shape</code>	A vector of node shape. The default is <code>graphics_type</code> .
<code>layout</code>	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.
<code>vertex.label.color</code>	A vector of node label colors. The default is black.
<code>vertex.color</code>	A vector of node colors. The default is the KEGG node color.
<code>vertex.frame.color</code>	A vector of node frame color. The default is <code>dimgray</code> .
<code>edge.color</code>	A vector of edge color. The default is <code>dimgray</code> .
<code>edge.label</code>	A vector of edge label.
<code>edge.label.cex</code>	Edge label size.
<code>edge.label.color</code>	A vector of edge label color. The default is <code>dimgray</code> .
<code>edge.lty</code>	A vector of line type for the edges.
<code>axes</code>	A logical. whether to plot axes. The default is <code>FALSE</code> .
<code>xlab</code>	A character string. The label of the horizontal axis. The default is the empty string.
<code>ylab</code>	A character string. The label of the vertical axis. The default is the empty string.
<code>sub</code>	A character string of subtitle.
<code>main</code>	A character string of main title.
<code>...</code>	The arguments passed to or from methods. See <a href="#">plot</a> .

## Details

The function `plotGraphL` 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)**

Xinrui Shi, Chunquan Li and Xia Li

**See Also**

[plot](#), [layout.random](#)

**Examples**

```
## Not run:
### get signal pathway graphs ###

g<- GetExampleData(exampleData="g2")

# visualize the graph
plotGraphL(g[[9]],layout=layout.random)

## End(Not run)
```

---

printGraphW

*Print the results of graph annotation and identification*

---

**Description**

Print the results of graph annotation and identification.

**Usage**

```
printGraphW(ann,detail=FALSE)
```

**Arguments**

ann	A list. The value was returned from the function <a href="#">identifyLncGraphW</a> .
detail	A logical. If true, gene lists from the function <a href="#">identifyLncGraphW</a> are converted into strings, which are used to display and write results with genes.

**Value**

A data.frame. Columns include pathwayId, pathwayName, annMoleculeRatio, annBgRatio,annWeight, pvalue, 'fdr', annMoleculeList, annBgMoleculeList. Detailed information is provided in [identifyLncGraphW](#).

**Author(s)**

Xinrui Shi, Chunquan Li and Xia Li

**See Also**

[identifyLncGraphW](#)

### Examples

```
## Not run:
### Integrate lncRNAs of competitive regulation into KEGG pathway graphs ###
  LncGenePairs<-GetExampleData(exampleData="LncGenePairs")
  inteUMGraph<-getInteUMGraph(LncGenePairs)
### get user-interested lncRNAs and genes sets.
  geneLnc<-GetExampleData(exampleData="geneLnc")
# get locate subpathways.
  sub<-getLocSubGraphLnc(geneLnc,inteUMGraph,type="gene_lncRNA",n=1,s=8)
  SubcodeLncResult<-identifyLncGraphW(geneLnc,sub,type="gene_lncRNA",bet=1)
  resultT<-printGraphW(SubcodeLncResult,detail=TRUE)

## End(Not run)
```

# Index

## \*Topic **file**

- envData, [2](#)
- getBackgroundLnc, [2](#)
- GetExampleData, [3](#)
- getInteGraphList, [5](#)
- getInteUMGraph, [6](#)
- getLncGenePairs, [7](#)
- getLocSubGraphLnc, [8](#)
- identifyLncGraphW, [9](#)
- plotAnnGraph, [11](#)
- plotGraphL, [12](#)
- printGraphW, [14](#)

envData, [2](#)

getBackgroundLnc, [2](#), [10](#)

GetExampleData, [3](#), [5](#), [10](#)

getInteGraphList, [5](#), [9](#)

getInteUMGraph, [6](#)

getLncGenePairs, [6](#), [7](#)

getLocSubGraphLnc, [5](#), [8](#)

identifyLncGraphW, [3](#), [9](#), [9](#), [11](#), [14](#)

layout, [13](#)

layout.random, [14](#)

plot, [11](#), [13](#), [14](#)

plot.igraph, [11](#), [13](#)

plotAnnGraph, [11](#)

plotGraphL, [5](#), [11](#), [12](#)

printGraphW, [10](#), [11](#), [14](#)