# Package 'ToPASeq'

April 15, 2017

**Type** Package

**Title** Package for Topology-based Pathway Analysis of RNASeq data

**Version** 1.8.0

**Date** 2015-12-02

**Author** Ivana Ihnatova, Eva Budinska

**Maintainer** Ivana Ihnatova <ihnatova@iba.muni.cz>

**Description** Implementation of seven methods for topology-based pathway analysis of both RNASeq and microarray data: SPIA, DEGraph, TopologyGSA, TAPPA, PRS, PWEA and a visualization tool for a single pathway.

**Depends** graphite (>= 1.16), gRbase, graph, locfit, Rgraphviz

**Imports** R.utils, methods, Biobase, parallel, edgeR, DESeq2, SummarizedExperiment, RBGL, DESeq, fields, limma, TeachingDemos, KEGGgraph, qpgraph, clipper, AnnotationDbi, doParallel

**Suggests** RUnit, BiocGenerics, gageData, DEGraph, plotrix, org.Hs.eg.db

**LinkingTo** Rcpp

**LazyData** yes

**License** AGPL-3

**biocViews** Software, GeneExpression, NetworkEnrichment, GraphAndNetwork, RNASeq, Visualization, Microarray, Pathways, DifferentialExpression,

**NeedsCompilation** yes

## R topics documented:

---

ToPASeq-package *Package for topology-based pathway analysis of microarray and RNASeq data*

---

## Description

The package implementats several methods for topology-based pathway analysis of microarray data. The methods present in here are: SPIA, TopologyGSA, DEGraph, Clipper, PWEA, TAPPA, TBS. SPIA, PWEA and TBS were also adapted for RNASeq data.

## Details

| | |
|---|---|
| Package: | ToPASeq |
| Type: | Package |
| Version: | 1.0 |
| Date: | 2014-03-04 |
| License: | AGPL-3 |

## Author(s)

Ivana Ihnatova

Maintainer: Ivana Ihnatova <ihnatova@iba.muni.cz>

## Examples

```
## Not run:
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-biocarta[1:10]
  SPIA(exprLoi2008, classLoi2008,pathways , type="MA", logFC.th=-1, IDs="entrez")
  DEGraph(exprLoi2008, classLoi2008, pathways, type="MA")
  TAPPA(exprLoi2008, classLoi2008, pathways, type="MA")
  TopologyGSA(exprLoi2008, classLoi2008, pathways, type="MA", nperm=200)
  Clipper( exprLoi2008, classLoi2008+1, pathways,type="MA", test="mean")
```

```
  PWEA(exprLoi2008, classLoi2008, pathways,  type="MA", nperm=100)
  TBS( exprLoi2008, classLoi2008, pathways, type="MA",  logFC.th=-1, nperm=100)
}
if (require(gageData)) {

 data(hnrnp.cnts)
 group<-c(rep("sample",4), rep("control",4))
 SPIA( hnrnp.cnts, group, biocarta[1:10], type="RNASeq",  logFC.th=-1, IDs="entrez", test="limma")
 DEGraph(hnrnp.cnts, group, biocarta[1:10], type="RNASeq", norm.method="TMM")
 TAPPA( hnrnp.cnts, group, biocarta[1:10], type="RNASeq", norm.method="TMM")
 TopologyGSA(hnrnp.cnts, group, biocarta[1:10], type="RNASeq",nperm=200, norm.method="TMM")
 Clipper(hnrnp.cnts, group,biocarta[1:10], type="RNASeq",  norm.method="TMM")
 PWEA(hnrnp.cnts, group, biocarta[1:10],  type="RNASeq", test="limma", nperm=100)
 TBS(hnrnp.cnts, group, biocarta[1:10], type="RNASeq", logFC.th=-1, nperm=100, test="limma")
 }

## End(Not run)
```

---

`AdjacencyMatrix2Pathway`

*Function to coerce an adjacency matrix to a Pathway*

---

## Description

The function coerces an adjacency matrix to a Pathway. Two types of matrices are allowed. The first one, where `1` denotes an edge between two nodes and `0` otherwise. This matrix is coerced into a simply pathway were type of all edges is set to `"process"`. The second type of adjacency matrix contains: `1` for an activation, `-1` for an inhibition and `0` otherwise (=no edge between two nodes). In this case, activations are set to `"process(activation)"` and inhibition to `"process(inhibition)"`. The symetricity of the matrix is used to decide between directed and undirected graph. Symmetric matrix is expected for undirected graph and only the lower triangle of the matrix is used to extract the edges of the graph.

## Usage

```
AdjacencyMatrix2Pathway(adjmat, name = "pathway", ident = "unknown", database = "unknown", species
```

## Arguments

| | |
|---|---|
| `adjmat` | An adjacency matrix describing the pathway topology |
| `name` | A character, name of the pathway. Defaults to `"pathway"` |
| `ident` | A character, type of the identificators, e.g `"gene symbol"` |
| `database` | A character, the name of the database the topology comes from |
| `species` | A character, the species to which the topology belong |
| `date` | A date, the date the topology was created |

## Value

An object of class `Pathway`, id is the same as title - name of the pathway

### Author(s)

Ivana Ihnatova

### Examples

```
genes<-paste("gene", 1:10, sep="")
adjmat<-matrix(sample(c(0,0,0,0,1), 100, TRUE),10,10, dimnames=list(genes,genes))
p<-AdjacencyMatrix2Pathway(adjmat)
head(edges(p))

adjmat<-matrix(sample(c(0,0,0,0,1,-1), 100, TRUE),10,10, dimnames=list(genes,genes))
p<-AdjacencyMatrix2Pathway(adjmat)
head(edges(p))
```

---

## clipper *Function to use clipper method on microarray or RNA-Seq data*

---

### Description

clipper is a method for topological gene set analysis. It implements a two-step empirical approach based on the exploitation of graph decomposition into a junction tree to reconstruct the most relevant signal path. In the first step clipper selects significant pathways according to statistical tests on the means and the concentration matrices of the graphs derived from pathway topologies. Then, it "clips" the whole pathway identifying the signal paths having the greatest association with a specific phenotype.

### Usage

```
clipper(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL,  method="
  both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none",
```

### Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the input data, "MA" for microarray and "RNASeq" for RNA-Seq |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg This analysis is needed only for the visualization. |
| method | Character, "mean" or "var", the kind of test to perform on the cliques |

| | |
|---|---|
| testCliques | Logical, if TRUE then the test is applied also on the cliques of the each pathway. It is a very time consuming calculation, especially for many or big pathways |
| nperm | Number of permutations |
| alphaV | Numeric, the threshold for variance test. The calculation of mean test depends on the result of variance test. |
| b | number of permutations for mean analysis |
| permute | always performs permutations in the concentration matrix test. If FALSE, the test is made using the asymptotic distribution of the log-likelihood ratio. This option should be use only if samples size is >=40 per class |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Value

A list,

| | |
|---|---|
| res | A list. First slot is a data frame containing p-values and q-values of mean and variance tests on pathways. The second slot is a list containing data.frames of the most affected paths in each pathway. The columns of the data frames contain: 1 - Index of the starting clique 2 - Index of the ending clique 3 - Index of the clique where the maximum value is reached 4 - length of the path 5 - maximum score of the path 6 - average score along the path 7 - percentage of path activation 8 - impact of the path on the entire pathway 9 - clique involved and significant 10 - clique forming the path 11 - genes forming the significant cliques 12 - genes forming the path |
| topo.sig | if testCliques=TRUE, a list where each slot contains the pvalues and a list of cliques in one pathway. NULL otherwise |
| degtest | A data.frame of gene-level differential expression statistics |

## Note

If there are NA's only in columns 3 to 7, then a junction tree could not be formed.

## Author(s)

Ivana Ihnatova

## References

Martini P, Sales G, Massa MS, Chiogna M, Romualdi C. Along signal paths: an empirical gene set approach exploiting pathway topology. Nucleic Acids Res. 2013 Jan 7;41(1):e19. doi: 10.1093/nar/gks866. Epub 2012 Sep 21. PubMed PMID: 23002139; PubMed Central PMCID: PMC3592432.

## See Also

preparePathways

## Examples

```
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
 pathways<-pathways("hsapiens","kegg")[1]
  clipper( exprLoi2008, classLoi2008, pathways,type="MA", convertTo="none")
}

## Not run:
if (require(gageData)) {

 data(hnrnp.cnts)
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 group<-c(rep("sample",4), rep("control",4))
  pathways<-pathways("hsapiens","kegg")[1:3]
 clipper(hnrnp.cnts, group,pathways, type="RNASeq",  norm.method="TMM", convertTo="none")
 }

## End(Not run)
```

---

`collectWeightsPRS` *Function to calculate gene-level weights for topology-based pathway analysis*

---

## Description

The functions calculate gene-level weigths defined in various topology-based pathway analysis methods (PRS, SPIA, PWEA). In PRS, it is the number of downstream differentially expressed genes. TIF, the statistic defined in PWEA, is related to the ratio of correlation and distance of genes. SPIA defines the so called net pertubation factors.

## Usage

```
collectWeightsPRS(de, all, pathways)
collectWeightsSPIA(de, all, pathways)
prepareTIF(pathways, exprs, alpha)
```

## Arguments

| | |
|---|---|
| `de` | Named numeric vector, the log fold-changes of the differentially expressed genes |
| `all` | Character vector of all genes meassured in the experiment |
| `pathways` | A list of pathways, each pathway is an object of class `Pathway` transformed via `preparePathways()` for the particular method |
| `exprs` | A numeric matrix, gene expression data matrix, rows refer to genes, columns to samples |
| `alpha` | Numeric, a threshold to control the magnitude. In TIF calculation, the effect of a gene on a few nearby and tightly correlated genes can be washed out if the gene influences many other genes weakly. The threshold supresses this washing-out |

## Value

A list, each slot is a vector of gene-level weights for one pathway

## Author(s)

Ivana Ihnatova

## Examples

```
pathways<-pathways("hsapiens","kegg")[1:3]
de<-setNames(rnorm(30),sample(nodes(pathways[[1]]),30))
all<-nodes(pathways[[1]])

path<-preparePathways(pathways[1:3], method="SPIA", genes=all, both.direction=TRUE, convertTo="none")
collectWeightsSPIA(de, all, path)
```

---

# convertIdentifiersByVector

*Function to convert identifiers in pathways by user specified vector*

---

## Description

The function converts identifiers of nodes in a pathway. It uses the user specified named vector for the conversion.

## Usage

```
convertIdentifiersByVector(pathway, conv.table, id.type="unknown")
```

## Arguments

| | |
|---|---|
| pathway | An object of class Pathway |
| conv.table | A named vector in which names correspond to the identifiers present in the pathway and values are the new identifiers to which conversion happens |
| id.type | A character, the type of the identifiers provided e.g "TAIR" for TAIR numbers. This is for informative purposes only. |

## Value

A Pathway in which identifiers have been converted

## Author(s)

Ivana Ihnatova

## See Also

convertIdentifiers

## Examples

```
g<-kegg[["Asthma"]]
conv<-setNames(paste("gene", 1:length(nodes(g)), sep=""), nodes(g))
gc<-convertIdentifiersByVector(g, conv, "dummy")
nodes(gc)
edges(gc)
```

---

DEGraph *Function to use DEGraph method on microarray or RNA-Seq data*

---

## Description

DEGraph implements recent hypothesis testing methods which directly assess whether a particular gene network is differentially expressed between two conditions. In employs Graph Laplacian, Fourier transformation and multivariate T2-statistic

## Usage

```
DEGraph(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL, overall=
  both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none"
```

## Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the data, "MA" for microarray and "RNASeq" for RNA-Seq |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg This analysis is needed only for the visualization. |
| overall | Character, how should the overall p-value for a pathway be calculated. The possible values are: "mean", "min", "biggest". "biggest" returns the p-value of the biggest connected component. |
| useInteractionSigns | Logical, should types of interaction be included in the analysis? |
| EdgeAttrs | A list containing two data.frames. See makeDefaultEdgeData() for the details. The interactions are assigned signs according to the beta column of the second data.frame. The procedure is similar to the SPIA method |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Value

A list:

| | |
|---|---|
| res | Results from analysis of individual pathways. The first column refers to the overall p-value for a pathway. Then groups of four columns follows. One group refers to one connected component and contains a pair of p-values (without and with Fourier transformation), graph and number of Fourier componets used in the test. The number of groups is equal to the highest number of components in analysed pathways. Components are sorted in the decreasing order of their nodes number. |
| topo.sig | NULL, present for the compatibility with outputs from other methods |
| degtest | A data.frame of gene-level statistics of all genes in the dataset |

## Author(s)

Ivana Ihnatova

## References

L. Jacob, P. Neuvial, and S. Dudoit. Gains in power from structured two-sample tests of means on graphs. Technical Report arXiv:q-bio/1009.5173v1, arXiv, 2010.

## See Also

preparePathways

## Examples

```
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-pathways("hsapiens","biocarta")[1:10]
    DEGraph(exprLoi2008, classLoi2008, pathways, type="MA")
}
## Not run:
if (require(gageData)) {

 data(hnrnp.cnts)
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 group<-c(rep("sample",4), rep("control",4))
 pathways<-pathways("hsapiens","biocarta")[1:10]
 #pathways<-lapply(pathways, function(p) as(p,"pathway"))
 DEGraph(hnrnp.cnts, group, pathways, type="RNASeq", norm.method="TMM")
}

## End(Not run)
```

`estimateCF` *Function to estimate multi-subunit protein complexes and gene families in a pathway*

## Description

Function estimates the multi-subunit protein complexes and gene families in a pathway. A protein complex consists of proteins connected by undirected binding interaction. A gene family is a set of nodes with same outgoing and/or incomig edges.

## Usage

```
estimateCF(graph)
```

## Arguments

| | |
|---|---|
| `graph` | An object of class `Pathway` |

## Value

| | |
|---|---|
| `complexes` | A list of estimated protein complexes' |
| `famillies` | A list of estimated gene famillies |

The function attempts to assign a representative name to each gene family. The representative name is a common part of the names of individual genes. This approach, however, may lead to ambiguities or missings. Then a general name in a form of `family1`, `family2`, etc. All the complexes are named analogously as `complex1`, `complex2`.

## Author(s)

Ivana Ihnatova

## See Also

`reduceGraph`

## Examples

```
path<-pathways("hsapiens","kegg")[[1]]
estimateCF(path)
```

graphNEL2Pathway *Function to coerce a graphNEL to a Pathway*

## Description

The function coerces a graphNEL to a Pathway. It attempts to recover the edge types from "edgeType" attribute of edgeData. The result contains only the edge types present in the graph. If the edgeData do not contain this attribute, then "process(indirect effect)" is used in order to preserve directionality.

## Usage

```
graphNEL2Pathway(graph, name = "pathway", ident = "unknown", database = "unknown", species = "unkn
```

## Arguments

| | |
|---|---|
| graph | A graphNEL object to be coerced. |
| name | A character, name of the pathway. Defaults to "pathway" |
| ident | A character, type of the identificators, e.g "gene symbol" |
| database | A character, the name of the database the topology comes from |
| species | A character, the species to which the topology belong |
| date | A date, the date the topology was created |

## Value

A coerced Pathway

## Note

When this function is applied on x as reversed operation to pathwayGraph then the order of the edges may differ as well as the directionality of "process(indirect)" edges as they are set as undirected by graphNEL2Pathway.

## Author(s)

Ivana Ihnatova

## Examples

```
pathway<-pathways("hsapiens","kegg")[[1]]
pathway<-pathwayGraph(pathway)
pathway
graphNEL2Pathway(pathway)

set.seed(123)
rg <- randomEGraph(LETTERS[1:20], edges = 30)
p<-graphNEL2Pathway(rg)
p
head(edges(p))
```

KEGG2Pathway *Function to parse KEGG KGML file into a Pathway*

## Description

The function parses a KGML file from KEGG into a Pathway.

## Usage

```
KEGG2Pathway(file, expandGenes = TRUE, expandCom = TRUE, nongene = c("keep", "propagate", "discard
```

## Arguments

| | |
|---|---|
| file | Character, the name of the file to be parsed. Download manually or in bulk from KEGG |
| expandGenes | Logical, should multi-gene nodes be expanded into separate nodes? |
| expandCom | Logical, should undirected binding interactions be added between nodes from one group (usually multi-subunit protein complex, which is turned into a clique) |
| nongene | Character, how should be the non-gene nodes parsed? If "discard" they are removed from the pathway. If "propagate", they are removed but the interactions are preserved (e.g. if gene A interacts with compound c and compound c interacts with gene B, then the interaction between A and B is preserved. Otherwise, they are kept in the pathway topology |
| ident | Character, the type of the node identifiers. |
| database | Character, the name of the database |
| species | Character, the three-letter code for the species-specific pathways. If NULL then, the first 3 letters from the file are used. |

## Value

A Pathway

## Author(s)

Ivana Ihnatova

makeDefautEdgeData *Creates auxiliary data needed for SPIA method*

## Description

This function creates a list containing auxiliary data needed in SPIA method for conversion between edge types and dividing interaction into three categories: positive, negative and neutral

## Usage

```
makeDefaultEdgeData
```

### Details

The first slot called graphite2SPIA contains a mapping table between edge types in topologies from graphite and edge types which are used in the implementation of SPIA in SPIA package. All of the edge types present in the topologies must be also covered by this table otherwise the method could not be applied.

The second slot called beta divides the 25 interaction types into three categories: positive (beta=1), negative (beta=-1 and neutral (beta=0) in the sense of gene regulation. Only user familiar with all the details of SPIA should change this.

### Value

A list of two data frames explained in the *Details* The format is: List of 2 $ graphite2SPIA: chr [1:26, 1:2
..- attr(*, "dimnames")=List of 2 .. ..$ : NULL .. ..$ : chr [1:2] "type" "spiaType"
$ beta :'data.frame': 25 obs. of 2 variables: ..$ rel : chr [1:25] "activation" "compound"
..$ beta: num [1:25] 1 0 0 1 -1 1 0 -1 -1 0 ...

### Source

The data are manualy cerated from the unexported objects from graphite package version 1.10.1.

### Examples

```
str(makeDefaultEdgeData())
```

---

## Pathway-method *Class* "Pathway"

---

### Description

This class represents a biological pathway. changeInteraction and changeDirection are a new generic function designed for Pathway class

### Methods

**edges** signature(object = "Pathway"): retrieves the data.frame describing the pathway edges.

**nodes** signature(object = "Pathway"): retrieves the vector enumerating the identifiers of the pathway nodes.

The methods below perfom basic topological analysis of a pathway. They were defined as generic in graph for graph class. They were implemented for Pathway in this package

**degree** signature(object = "Pathway", Nodes = "character") Returns the number of incoming or outgoing edges for nodes in Nodes

**degree** signature(object = "Pathway", Nodes = "missing") Returns the number of incoming or outgoing edges for all nodes in object

**numNoEdges** signature(objGraph = "Pathway") Returns the number of nodes without any edge

**mostEdges** signature(objGrapg = "Pathway") Returns the nodes with most edges

**acc** `signature(object = "Pathway", index = "character")` Returns the set of nodes accessible from nodes in `index`. The undirected edges are considered as bidirected (directed in both directions)

**connComp** `signature(object = "Pathway")` Returns the connected components present in a pathway. They are returned as list where each slot refers to one component and contains the relevant nodes. The undirected edges are considered as bidirected (directed in both directions)

**edges** `signature(object = "Pathway", which = "character")` Returns the edges relevant to node(s) in `which`

**isAdjacent** `signature(object = "Pathway", from = "character", to = "character")` Returns whether nodes in `from` and `to` are adjacent (there is an edge starting in `from` and ending in `to`

**isConnected** `signature(object = "Pathway")` Returns `TRUE` if a pathway contains only one connected component

**isDirected** `signature(object = "Pathway")` Returns `TRUE` if all edges in a pathway are directed

**edgemode** `signature(object = "Pathway")` Returns the type of edges in a pathway: `directed`, `undirected` or both

**numEdges** `signature(object = "Pathway")` Returns the number of edges in a pathway

**numNodes** `siganture(object = "Pathway")` Returns the number of nodes in a pathway

**edgeNames** `signature(object = "Pathway")` Returns the names of the edges in a following format: starting node ~ ending node

All of the methods below return an object of class `Pathway` with modified topology.

**intersection** `signature(x = "Pathway", y = "Pathway")` compute the intersection of the two supplied graphs. They must have identical nodes.

**join** `signature(x = "Pathway", y = "Pathway")` returns the joining of the two graphs. It is similar to `intersection` but does not require the identical nodes

**union** `signature(x = "Pathway", y = "Pathway")` compute the union of the two supplied graphs. They must have identical nodes.

**subGraph** `signature(snodes = "character", graph = "Pathway")` Given a set of nodes and a pathway this function creates and returns subgraph with only the supplied nodes and any edges between them

**clearNode** `signature(node = "character", object = "Pathway")` Clears all edges incoming and outgoing edges from `node`

**removeEdge** `signature(from = "character", to = "character", graph = "Pathway")` removes all directed edges starting in `from` and ending in `to` and undirected edges between `from` and `to`

**removeNode** `signature(node = "character", object = "Pathway")` removes node(s) `node` from a pathway `object`

**nodes<-** `signature(x = "Pathway", value = "character")` sets node labels of pathway `object` to `value`

**convertIdentifiers** `signature(x = "Pathway", to = "character")` converts the node identifiers/labels in a pathway. `to` is the name of one of the columns provided by an Annotation package (e.g."`SYMBOL`"

preparePathways *Function to prepare pathways for topology-based pathway analysis*

## Description

Functions transforms pathways from graphite package (stored as Pathway-class) into formats required in the particular topology-based method implemented in this package. It also converts identifiers in the pathways and filters pathways according to several criteria.

## Usage

```
preparePathways(pathways, method, both.directions, genes, maxNodes = 150, minEdges = 0, commonTh =
```

## Arguments

| | |
|---|---|
| pathways | A list of pathways, individual pathways are objcets of class Pathway stored in PathwayList |
| method | A character, the pathways will be transformed according to the needs of the particular method. Possible values are: "TAPPA", "PRS", "PWEA", "TopologyGSA", "clipper", "DE |
| both.directions | Logical, indicates how should be the undirected edges directed. If TRUE, an undirected edge is substitued with two directed edges with opposite directions (e.g. A-B becomes A->B and B->A). If FALSE, then an undirected edge is substitued with one directed edge which preserves the order of nodes (e.g. A-B becomes A->B). |
| genes | Character vector, vector of gene identifiers in the expression data |
| maxNodes | Numeric, maximal number of nodes. Pathways with more nodes are filtered out. |
| minEdges | Numeric, minimal number of edges. Pathways with less edges are filtered out. |
| commonTh | Numeric, threshold for number of nodes present in the data. Pathways with less node-identifiers matching to genes are filtered out. |
| filterSPIA | Logical, if TRUE applies filter defined in the SPIA method (relates to the calculation of inversion matrix). |
| convertTo | Character. If "none" no conversion is performed. Otherwise, the function converts node-identifiers in pathways as in graphite. It uses annotation package for the mapping. |
| convertBy | Named character vector, names of the elemenet must match the node-identifiers and the values are the new identifiers to be replaced. This is a more general option designed for pathways outside graphite. |
| EdgeAttrs | A list of two tables required for the filter from SPIA method. See makeDefaultEdgeData for the details. |

## Value

A list of the transformed pathways

## Author(s)

Ivana Ihnatova

## See Also

makeDefaultEdgeData

## Examples

```
#Creating dummy set of genes
set.seed(123)
pathways<-pathways("hsapiens","kegg")[1:3]

genes<-unname(unlist(lapply(pathways[1:3], nodes)))
genes<-sample(genes, length(genes)*0.9)

#Applying the function
paths<-preparePathways(pathways[1:3], "TAPPA", TRUE, genes, maxNodes=65, convertTo="none")
paths
```

---

PRS *Function to use PRS method on microarray or RNA-Seq data*

---

## Description

A function runs PRS method on a gene expression data matrix or count matrix and vector dividing samples into two groups and a set of pathways from graphite package. The PRS method (please see Reference for the details) was adapted to graphite's graphs where each node is represented only by one gene.

## Usage

```
PRS(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL, p.th=0.05, lo
 both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none",
```

## Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the data, "MA" for microarray, "RNASeq" for RNA-Seq, DEtable data.frame from differential expression analysis, or DEGlist a list of: log fold-changes of differentially expressed genes and names of the all genes analyses |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none". Ignored for type: "MA","DEtable", "DElist" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg Ignored for type: "MA","DEtable", "DElist" |

| | |
|---|---|
| p.th | Numeric, threshold for p-values of tests for differential expression of genes. Use 1 if you don't want any threshold to be applied |
| logFC.th | Numeric, threshold for log fold-change of a gene to identify the gene as differentially expressed. Use negative if you don't want any threshold to be applied |
| nperm | Numeric, number of permutations |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Value

A list,

| | |
|---|---|
| res | A data frame with normalized score, p-value and FDR-adjusted p-value for each pathway |
| topo.sig | A list with log fold-changes and number of downstream differentially expressed nodes for nodes of individual pathways |
| degtest | A named vector of statistics from testing the differential expression of genes |

## Author(s)

Ivana Ihnatova

## References

Maysson Al-Haj Ibrahim, Sabah Jassim, Michael Anthony Cawthorne, and Kenneth Langlands. A Topology-Based Score for Pathway Enrichment, Journal of Computational Biology. May 2012, 19(5): 563-573

## See Also

preparePathways

## Examples

```
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-pathways("hsapiens","biocarta")[1:10]
  PRS( exprLoi2008, classLoi2008, pathways, type="MA",  logFC.th=-1, nperm=100)
}
## Not run:
if (require(gageData)) {

 data(hnrnp.cnts)
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 group<-c(rep("sample",4), rep("control",4))
 pathways<-pathways("hsapiens","biocarta")[1:10]
 PRS(hnrnp.cnts, group, pathways, type="RNASeq", logFC.th=-1, nperm=100, test="vstlimma")
}

## End(Not run)
```

PWEA *Function to use PWEA method on microarray or RNA-Seq data*

## Description

The function runs PWEA method (please see References for the details) on gene expression data matrix, vector specifing to which group a sample belongs and a list of pathway graphs. Briefly, it is a weighted GSEA-like method. The weightes are based on the distance and Pearson's correlation between genes in a pathway.

## Usage

```
PWEA(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL, tif=NULL, al
  both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none",
```

## Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples. Or a list of two data.frames: observed and random (after group permutations) of statistics of differential expression of genes |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the data, "MA" for microarray, "RNASeq" for RNA-Seq or "DEtable" for a list of observed and random gene-level statistics |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg |
| tif | A list of Topology Influence Factor's. One slot refers to one pathway. Use prepareTIF() to create it. It is required only if type=="DEtable" |
| alpha | Numeric, a theshold value used during TIF calculation |
| nperm | Numeric, number of permutations. Used only if x %in% c("MA", "RNASeq") |
| ncores | Numeric, number of cores. Used only if x %in% c("MA", "RNASeq"). The permutations are calculated in parallel way |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Value

A list

| | |
|---|---|
| res | A data frame, rows refer to pathways. It contains: Enrichment score for a pathway, p-value and p-value adjusted for multiple hypothesis testing by Benjamini-Hochberg's FDR method. NA's if less than 2 nodes are present in the data |

| | |
|---|---|
| topo.sig | A list, topology influence factors for the genes in individual pathways. NULL if less than 2 nodes are present in the data |
| degtest | A named vector of statistics from testing the differential expression |

## Author(s)

Ivana Ihnatova

## References

Hung, JH., Whitfield, T. W., Yang, TH., Hu, Z., Weng, Z., DeLisi, Ch. (2010) Identification of functional modules that correlate with phenotypic difference: the influence of network topology, Genome Biology, 11:R23

## See Also

preparePathways, prepareTIF

## Examples

```
## Not run:
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-pathways("hsapiens","biocarta")[1:10]
  PWEA(exprLoi2008, classLoi2008, pathways,  type="MA", nperm=100)
}


if (require(gageData)) {
 data(hnrnp.cnts)
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 group<-c(rep("sample",4), rep("control",4))
 pathways<-pathways("hsapiens","biocarta")[1:10]
 PWEA(hnrnp.cnts, group, pathways,  type="RNASeq", test="vstlimma", nperm=100)
}

## End(Not run)
```

---

# reduceGraph *Function to reduce the pathway graph*

---

## Description

Function simplifies a pathway graph topology. It merges a user specified nodes into a one. The specified set of nodes must be either a gene family or a protein complex. By a gene family we mean a set of genes with same outgoing or incoming edges. On the other hand, a protein complex is a set of nodes with only undirected binding edges between them and the number of edges is equal to the complex size.

## Usage

```
reduceGraph(graph, reduction)
```

### Arguments

| | |
|---|---|
| graph | An object of class Pathway, a pathway to be reduced |
| reduction | A named list of reductions to be maded. |

### Value

A Pathway

### Author(s)

Ivana Ihnatova

### Examples

```
pathways<-pathways("hsapiens","kegg")["Prolactin signaling pathway"]
pathways<-convertIdentifiers(pathways[[1]], "SYMBOL")


#gr<-as(pathways,"pathway")
red<-list(RAS=c("NRAS","KRAS","HRAS"), SHC=c("SHC1", "SHC4","SHC2","SHC3"))
reduced<-reduceGraph(pathways, red)
reduced
par(mfrow=c(1,2))

nA<-list(fillcolor=c(NRAS="red", KRAS="red", HRAS="red", SHC1="green", SHC4="green", SHC2="green", SHC3="gre
plot(as(pathways,"graphNEL"), nodeAttrs=nA, attrs=list(node=list(fontsize=30, height=40)), main="Before")

plot(as(reduced,"graphNEL"),
 nodeAttrs=list(fillcolor=c(RAS="red", SHC="green")), attrs=list(node=list(fontsize=30, height=40)), main="A

#this throws an error, "RELA", "FOS","NFKB1" is not correct set of genes
## Not run:
pathways<-pathways("hsapiens","kegg")["Prolactin signaling pathway"]
pathways<-convertIdentifiers(pathways[[1]], "SYMBOL")


gr<-convertIdentifiers(kegg[["Prolactin signaling pathway"]],"SYMBOL")
red<-list(RAS=c("NRAS","KRAS","HRAS"), SHC=c("RELA", "FOS","NFKB1"))
reduced<-reduceGraph(pathways, red)

## End(Not run)
```

---

## res *Function to extract parts of object*

---

### Description

Function extracts part of an object named "res", "topo.sig", "degtable"

### Usage

```
res(object)
topo.sig(object)
degtable(object)
```

### Arguments

| | |
|---|---|
| object | Object of defined class. Methods for topResult are available in this package |

### Value

Extracted parts of an object. Data type varies between parts and the origin of the object

### Author(s)

Ivana Ihnatova

---

## SPIA *Function to use SPIA method on microarray or RNA-Seq data*

---

### Description

The function runs SPIA method on microarray or RNA-Seq data. The implementatio includes the identification of differentially expressed genes and transformation of pathways' topologies to an appropriate form. The SPIA method combines two independent p-values. One p-value comes from overrepresentation analysis and the other is so called pertubation factor.

### Usage

```
SPIA(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL, p.th=0.05,
 both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none",
```

### Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the data, "MA" for microarray, "RNASeq" for RNA-Seq, DEtable data.frame from differential expression analysis, or DEGlist a list of: log fold-changes of differentially expressed genes and names of the all genes analyses |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none". Ignored for type: "MA","DEtable", "DElist" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg Ignored for type: "MA","DEtable", "DElist" |
| p.th | Numeric, threshold for p-values of tests for differential expression of genes. Use 1 if you don't want any threshold to be applied |
| logFC.th | Numeric, threshold for log fold-change of a gene to identify the gene as differentially expressed. Use negative if you don't want any threshold to be applied |

`nperm` Numeric, number of permutations

`combine` Character, the method to combine p-values. Defaults to `"fisher"` for Fisher's method. The other possible value is `"norminv"` for the normal inversion method.

`both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy` Arguments for the `preparePathways()`

## Value

A list:

`res` A matrix with columns as descibed below: pSize - Pathway size, number of genes, NDE - Number of differentially expressed genes, pNDE - P-value of the overrepresentation part of the method, tA - The observed total preturbation accumulation in the pathway, pPERT - P-value of the pertubation part of the method, p - Combined p-value (overrepresentation and pertubation), pFdr - False discovery rate adjusted p, pFWER - FWER adjusted p, Status - If a pathway was identified as Acivated or Inhibited

`topo.sig` A list of accumulated pertubation factors and log fold-changes for genes in individual pathways

`degtest` A numeric vector of gene-level differential expression statistics of all genes in the dataset

## Author(s)

Ivana Ihnatova

## References

Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, Kim JS, Kim CJ, Kusanovic JP, Romero R. A novel signaling pathway impact analysis. Bioinformatics. 2009 Jan 1;25(1):75-82.

Adi L. Tarca, Sorin Draghici, Purvesh Khatri, et. al, A Signaling Pathway Impact Analysis for Microarray Experiments, 2008, Bioinformatics, 2009, 25(1):75-82.

Draghici, S., Khatri, P., Tarca, A.L., Amin, K., Done, A., Voichita, C., Georgescu, C., Romero, R.: A systems biology approach for pathway level analysis. Genome Research, 17, 2007.

## See Also

`preparePathways`

## Examples

```
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
 pathways<-pathways("hsapiens","biocarta")[1:10]
  SPIA(exprLoi2008, classLoi2008,pathways, type="MA", logFC.th=-1)
}
## Not run:
if (require(gageData)) {

 data(hnrnp.cnts)
  hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 group<-c(rep("sample",4), rep("control",4))
```

```
 pathways<-pathways("hsapiens","biocarta")[1:10]
SPIA( hnrnp.cnts, group, pathways, type="RNASeq",  logFC.th=-1, IDs="entrez", test="vstlimma")
}

## End(Not run)
```

---

TAPPA *Function to use TAPPA method on microarray or RNA-Seq data*

---

## Description

The functions analyses the differential expression of pathways via TAPPA method. Expression is compared between two groups of samples by Mann-Whitney test. P-values are later adjusted for multiple hypothesis testing by Benjamini-Hochberg's FDR method.

## Usage

```
TAPPA(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL, test=t.tes
  maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none", convertBy=NULL)
```

## Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the input data, "MA" for microarray and "RNASeq" for RNA-Seq |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg This analysis is needed only for the visualization. |
| test | Function implementing a statistical test comparing PCI scores between groups. It is employed as test(PCI~group)$p.value, where PCI is a numeric vector of the same length as group |
| normalize | Logical, should data be normalized? |
| verbose | Logical, if TRUE names of the pathways are printed as they are analysed |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Value

A list,

| | |
|---|---|
| `res` | A data frame, rows refer to pathways. Columns contain: number of valid PCI-scores, median, min and max of the PCI scores for each group of samples, p-value of the `test` (`p.val`) and adjusted p-value (`p.adj`). If less than two nodes are present in the data, the function puts `NA`'s in all columns. |
| `topo.sig` | `NULL`, it is preserved for the compatibility with other methods implemented in this package |
| `degtest` | A numeric vector of gene-level differential expression statistics |

## Author(s)

Ivana Ihnatova

## References

Gao, S. and Wang, X. (2007) TAPPA: topological analysis of pathway phenotype association. Bioinformatics, 23, pages 3100-3102

## See Also

`preparePathways`

## Examples

```
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-pathways("hsapiens","biocarta")[1:10]
  TAPPA(exprLoi2008, classLoi2008, pathways, type="MA")
}

## Not run:
if (require(gageData)) {

 data(hnrnp.cnts)
 group<-c(rep("sample",4), rep("control",4))
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
pathways<-pathways("hsapiens","biocarta")[1:10]
 TAPPA( hnrnp.cnts, group, pathways, type="RNASeq", norm.method="TMM")
}

## End(Not run)
```

---

# ToPASeq-deprecated *Deprecated functions in package 'ToPASeq'*

---

## Description

These functions are provided for compatibility with older versions of 'ToPASeq' only, and will be defunct at the next release.

## Details

The following functions are deprecated and will be made defunct; use the replacement indicated below:

- AdjacencyMatrix2pathway: AdjacencyMatrix2Pathway
- graphNEL2pathway: graphNEL2Pathway
- KEGG2pathway: KEGG2Pathway

---

TopologyGSA *Function to use TopologyGSA method on microarray or RNA-Seq data*

---

## Description

TopologyGSA method uses graphical models to test the differential expression of a pathway. It also highlights pathway componenets involved in the deregulation.

## Usage

```
TopologyGSA(x, group, pathways, type, preparePaths=TRUE, norm.method=NULL, test.method=NULL , meth
  both.directions=TRUE, maxNodes=150, minEdges=0, commonTh=2, filterSPIA=FALSE, convertTo="none"
```

## Arguments

| | |
|---|---|
| x | An ExpressionSet object or a gene expression data matrix or count matrix, rows refer to genes, columns to samples |
| group | Name or number of the phenoData column or a character vector or factor that contains required class assigments |
| pathways | A list of pathways in a form from graphite package or created by preparePathways() |
| type | Type of the input data, "MA" for microarray and "RNASeq" for RNA-Seq |
| preparePaths | Logical, by default the pathways are transformed with preparePathways(). Use FALSE, if you have done this transformation separately |
| norm.method | Character, the method to normalize RNAseq data. If NULL then TMM-normalization is performed. Possible values are: "TMM", "DESeq2", "rLog", "none" |
| test.method | Character, the method for differentiall expression analysis of RNAseq data. If NULL then "voomlimma" is used. Possible values are: "DESeq2", "voomlimma", "vstlimma", "edg This analysis is needed only for the visualization. |
| method | Either "var" and "mean". Determine the type of test used by topologyGSA. |
| alpha | Numeric, threshold for statistical significance of variance test. It influences the method for the mean test |
| testCliques | Logical, if TRUE, then the test is also performed on individual cliques. It can be very computationally complex. |
| ... | Other arguments to be passed to the method. See details for better explanation |
| both.directions, maxNodes, minEdges, commonTh, filterSPIA, convertTo, convertBy | Arguments for the preparePathways() |

## Details

The method requires a Directed Acyclic Graph (DAG). Therefore if a pathway contain also undirected or bidirected edges and error is thrown.

The user can further specify for the mean test:

1. **perms** number of permutations of the test,
2. **paired**logical flag. If TRUE Hotelling test for paired samples is calculated and the test on the variances is not performed

Or for the variance test:

1. **variance**logical flag. If TRUE the estimates of the covariance matrices are included in the result.
2. **s1**First group covariance matrix estimation.
3. **s2**Second group covariance matrix estimation.

## Value

A list

| | |
|---|---|
| `res` | a list with one entry for each successfully analyzed pathway |
| `topo.sig` | if `testCliques=TRUE`, a list where each slot contains the pvalues and a list of cliques in one pathway. `NULL` otherwise |
| `degtest` | A numeric vector of gene-level differential expression statistics |

## Author(s)

Ivana Ihnatova

## References

Massa MS, Chiogna M, Romualdi C. Gene set analysis exploiting the topology of a pathway. BMC System Biol. 2010 Sep 1;4:121.

## Examples

```
## Not run:
if (require(DEGraph)) {
  data("Loi2008_DEGraphVignette")
  pathways<-pathways("hsapiens","biocarta")[1:10]


 TopologyGSA(exprLoi2008, classLoi2008, pathways, type="MA", method="mean", alpha=0.05, perms=200)
   TopologyGSA(exprLoi2008, classLoi2008, pathways, type="MA", method="mean", alpha=0.05, perms=200, testCli
  }

if (require(gageData)) {

 data(hnrnp.cnts)
 group<-c(rep("sample",4), rep("control",4))
 hnrnp.cnts<-hnrnp.cnts[rowSums(hnrnp.cnts)>0,]
 pathways<-pathways("hsapiens","biocarta")[1:10]
 TopologyGSA(hnrnp.cnts, group,pathways, type="RNASeq",method="mean", alpha=0.05,
   perms=200, norm.method="TMM")
```

```
}

## End(Not run)
```

# Index