# Package ‘graphite’

May 29, 2024

**Version** 1.50.0  
**Date** 2024-04-29  
**Title** GRAPH Interaction from pathway Topological Environment  
**Description** Graph objects from pathway topology derived from KEGG, Panther, PathBank, PharmGKB, Reactome SMPDB and WikiPathways databases.  
**License** AGPL-3  
**URL** [https://github.com/sales-lab/graphite](https://github.com/sales-lab/graphite)  
**BugReports** [https://github.com/sales-lab/graphite/issues](https://github.com/sales-lab/graphite/issues)  

**Depends**  
R (>= 4.2), methods  

**Imports**  
AnnotationDbi, graph (>= 1.67.1), httr, rappdirs, stats, utils, graphics, rlang, purrr  

**Suggests**  
checkmate, a4Preproc, ALL, BiocStyle, codetools, hgu133plus2.db, hgu95av2.db, impute, knitr, org.Hs.eg.db, parallel, R.rsp, RCy3, rmarkdown, SPIA (>= 2.2), testthat, topologyGSA (>= 1.4.0)  

**Collate**  

**VignetteBuilder** R.rsp  

**biocViews** Pathways, ThirdPartyClient, GraphAndNetwork, Network, Reactome, KEGG, Metabolomics  

**git_url** [https://git.bioconductor.org/packages/graphite](https://git.bioconductor.org/packages/graphite)  

**git_branch** RELEASE_3_19  

**git_last_commit** 0b9f354  

**git_last_commit_date** 2024-04-30  

**Repository** Bioconductor 3.19  

**Date/Publication** 2024-05-29  

**Author** Gabriele Sales [cre], Enrica Calura [aut], Chiara Romualdi [aut]  

**Maintainer** Gabriele Sales <gabriele.sales@unipd.it>
as.list.PathwayList

Contents

as.list.PathwayList .......................................................... 2
buildPathway ............................................................... 3
convertIdentifiers ......................................................... 4
cytoscapePlot ............................................................. 5
Pathway-class .............................................................. 6
pathwayDatabases .......................................................... 7
pathwayGraph .............................................................. 8
PathwayList-class .......................................................... 8
pathways ................................................................. 9
Pathways-class ............................................................. 10
prepareSPIA ................................................................. 11
runSPIA ................................................................. 12
runTopologyGSA ........................................................... 13

Index

as.list.PathwayList  Conversion of PathwayLists into lists.

Description

Converts a PathwayList into a list of Pathways.

Usage

```r
## S3 method for class 'PathwayList'
as.list(x, ...)
```

Arguments

- `x`  a PathwayList object
- `...` extra arguments to as.list

Value

A list of pathways.

Author(s)

Gabriele Sales

See Also

PathwayList

Examples

```r
as.list(pathways("hsapiens", "kegg"))
```
buildPathway

Build a Pathway object.

Description

This function creates a new object of type Pathway given a data frame describing its edges.

Usage

buildPathway(id, title, species, database, proteinEdges,
              metaboliteEdges = NULL, mixedEdges = NULL,
              timestamp = NULL)

Arguments

id the pathway identifier.
title the title of the pathway.
species the species the pathway belongs to.
database the name of the database the pathway derives from.
proteinEdges a data.frame of edges between proteins (or genes).
               Must have the following columns: src_type, src, dest_type, dest, direction and type.
               Direction must be one of the two strings: "directed" or "undirected".
metaboliteEdges interactions between metabolites.
               Can be NULL. Otherwise, it must have the same structure as proteinEdges.
mixedEdges interactions between metabolites and proteins.
               Can be NULL. Otherwise, it must have the same structure as proteinEdges.
timestamp when the pathway was annotated, by default the time buildPathway is called.

Value

A new Pathway instance.

Examples

data.frame(src_type = "ENTREZID", src = "672",
           dest_type = "ENTREZID", dest = "7157",
           direction = "undirected", type = "binding")

# Example with metabolites:
data.frame(src_type = "ENTREZID", src = "672",
           dest_type = "ENTREZID", dest = "7157",
           direction = "undirected", type = "binding")
mixed <- data.frame(src_type = "CHEBI", src="77750",
dest_type = "ENTREZID", dest="7157",
direction="undirected", type="binding")
pathway <- buildPathway("#1", "example", "hsapiens", "database",
edges, mixedEdges = mixed)

convertIdentifiers(  
 Convert the node identifiers of a pathway.

Description

Converts the node identifiers of pathways.

If the option $Ncpus$ is set to a value larger than 1 and the package parallel is installed, the conversion procedure will automatically use multiple cores.

Usage

convertIdentifiers(x, to)

Arguments

x can be a list of pathways or a single pathway
to a string describing the type of the identifier. Can assume the values "entrez", "symbol" or the name of one of the columns provided by an Annotation package (for example, "UNIPROT").

Value

A Pathway object.

See Also

Pathway

Examples

r <- pathways("hsapiens", "reactome")
convertIdentifiers(r$mTORC1-mediated signalling", "symbol")
cytoscapePlot

Plot a pathway graph in Cytoscape

Description

Renders the topology of a pathway as a Cytoscape graph.

Usage

`cytoscapePlot(pathway, ..., cy.ver = 3)`

Arguments

- `pathway`: a `Pathway` object.
- `...`: optional arguments forwarded to `pathwayGraph`.
- `cy.ver`: select a Cytoscape version. Only version 3 is supported in this release.

Details

Requires the `RCy3` package.

Value

An invisible list with two items:

- `graph`: the `graphNEL` object sent to Cytoscape.
- `suid`: the `RCy3` network SUID.

See Also

- `Pathway`
- `pathwayGraph`

Examples

```r
## Not run:
r <- pathways()
cytoscapePlot(convertIdentifiers(reactome$'Unwinding of DNA', "symbol"))

## End(Not run)`
Pathway-class  

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A biological pathway.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Pathway instance actually stores multiple variants of the same biological data.</td>
</tr>
<tr>
<td>This is the list of included variants:</td>
</tr>
<tr>
<td>• proteins: includes only interactions among proteins;</td>
</tr>
<tr>
<td>• metabolites: includes only interactions among metabolites;</td>
</tr>
<tr>
<td>• mixed: includes all available interactions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pathwayId(p)</code>: Returns the native ID of the pathway.</td>
</tr>
<tr>
<td><code>pathwayTitle(p)</code>: Returns the title of the pathway.</td>
</tr>
<tr>
<td><code>pathwayDatabase(p)</code>: Returns the name of the database the pathway was derived from.</td>
</tr>
<tr>
<td><code>pathwaySpecies(p)</code>: Returns the name of the species in which the pathway was annotated.</td>
</tr>
<tr>
<td><code>pathwayTimestamp(p)</code>: Returns the date of pathway data retrieval.</td>
</tr>
<tr>
<td><code>pathwayURL(p)</code>: Returns the URL of the pathway in its original database, if available.</td>
</tr>
<tr>
<td><code>convertIdentifiers(p, to)</code>: Returns a new pathway using a different type of node identifiers.</td>
</tr>
<tr>
<td><code>edges(p, which = c(&quot;proteins&quot;, &quot;metabolites&quot;, &quot;mixed&quot;), stringsAsFactors = TRUE)</code>: Returns a data.frame describing the edges of this pathway.</td>
</tr>
<tr>
<td>The option which selects the desired pathway variant (see section &quot;Variants&quot; above).</td>
</tr>
<tr>
<td>If <code>stringsAsFactors</code> is <code>TRUE</code>, strings are converted to factors.</td>
</tr>
<tr>
<td><code>nodes(p, which = c(&quot;proteins&quot;, &quot;metabolites&quot;, &quot;mixed&quot;))</code>: Returns the names of the nodes belonging to this pathway.</td>
</tr>
<tr>
<td>The option which selects the desired pathway variant (see section &quot;Variants&quot; above).</td>
</tr>
<tr>
<td><code>plot(p)</code>: Shows the pathway topology in Cytoscape.</td>
</tr>
<tr>
<td><code>runClipper(p, expr, classes, method, ...)</code>: Runs a clipper analysis over the pathway.</td>
</tr>
<tr>
<td><code>runTopologyGSA(p, test, exp1, exp2, alpha, ...)</code>: Runs a topologyGSA analysis over the pathway.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabriele Sales</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>See Also</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pathways</code></td>
</tr>
</tbody>
</table>
Examples

```r
reactome <- pathways("hsapiens", "reactome")
pathway <- reactome[[1]]

pathwayTitle(pathway)
pathwaySpecies(pathway)
nodes(pathway)
edges(pathway)
```

---

```r
pathwayDatabases
```

List the available pathway databases.

---

Description

Obtains the list of pathway databases available through graphite.

Usage

```r
pathwayDatabases()
```

Value

Returns a `data.frame` with two columns: `species` and `database`.

Author(s)

Gabriele Sales

See Also

`pathways`

Examples

```r
pathwayDatabases()
```
pathwayGraph  

*Graph representing the topology of a pathway*

**Description**

Builds a graphNEL object representing the topology of a pathway.

**Usage**

`pathwayGraph(pathway, which = "proteins", edge.types = NULL)`

**Arguments**

- `pathway`: a `Pathway` object.
- `which`: the pathway variant you want. See `Pathway` documentation for a list of the supported variants.
- `edge.types`: keep only the edges matching the type names in this vector.

**Value**

A graphNEL object.

**See Also**

`Pathway`  
`graphNEL`

**Examples**

```r
r <- pathways("hsapiens", "reactome")
pathwayGraph(r$"mTORC1-mediated signalling", edge.types="Binding")
```

---

**PathwayList-class**

**Class** "PathwayList"

**Description**

A collection of pathways from a single database.

**Extends**

Class "Pathways", directly.
Methods

\[l[i]\] returns a selection of the pathways contained in the pathway list.
\[l[[i]]\] gives access to one of the pathways contained in the pathway list.
\['l\$'title'\] loads a pathways by its title.
convertIdentifiers(l, to) returns a new list of pathways using a different type of node identifiers.
length(l) returns the number of pathways contained in the list.
names(l) returns the titles of the pathways contained in the list.
prepareSPIA(l, pathwaySetName, print.names=FALSE) prepares the pathways for a SPIA analysis.
runClipper(l, expr, classes, method, maxNodes=150, ...) runs a clipper analysis over all the pathways in the list.
runTopologyGSA(l, test, exp1, exp2, alpha, maxNodes=150, ...) runs a topologyGSA analysis over all the pathways in the list.

Author(s)

Gabriele Sales

See Also

pathways

pathways Retrieve a list of pathways.

Description

Retrieve a list of pathways from a database for a given species.

graphite currently supports the following databases:

- KEGG
- PANTHER
- PathBank
- PharmGKB
- Reactome
- SMPDB
- WikiPathways

Call the pathwayDatabase function for more details.

Usage

pathways(species, database)
Arguments

- **species**: one of the supported species
- **database**: the name of the pathway database

Value

A PathwayList object.

See Also

*PathwayList*, *pathwayDatabases*

Examples

```r
pathways("hsapiens", "reactome")
```

Description

A virtual class acting as a common parent to all other classes representing pathway databases.

Objects from the Class

A virtual Class: No objects may be created from it.

Methods

No methods defined with class "Pathways" in the signature.

Author(s)

Gabriele Sales

See Also

*PathwayList*
prepareSPIA

Description

Prepare pathway dataset needed by runSPIA. See runSPIA and spia for more details.

Usage

prepareSPIA(db, pathwaySetName, print.names = FALSE)

Arguments

db a PathwayList object or a list of Pathways.
pathwaySetName name of the output pathway set.
print.names print pathway names as the conversion advances.

Value

This function has no return value.

References


See Also

runSPIA
spia
PathwayList
**runSPIA**  
*Run SPIA analysis*

**Description**
Run a topological analysis on an expression dataset using SPIA.

**Usage**
```
runSPIA(de, all, pathwaySetName, ...)
```

**Arguments**
- **de**: A named vector containing log2 fold-changes of the differentially expressed genes. The names of this numeric vector are Entrez gene IDs.
- **all**: A vector with the Entrez IDs in the reference set. If the data was obtained from a microarray experiment, this set will contain all genes present on the specific array used for the experiment. This vector should contain all names of the 'de' argument.
- **pathwaySetName**: The name of a pathway set created with `prepareSPIA`.
- **...**: Additional options to pass to `spia`.

**Details**
The `spia` option "organism" is internally used. It is an error use it in the additional options.

**Value**
The same of `spia`, without KEGG links. A data frame containing the ranked pathways and various statistics: pSize is the number of genes on the pathway; NDE is the number of DE genes per pathway; tA is the observed total perturbation accumulation in the pathway; pNDE is the probability to observe at least NDE genes on the pathway using a hypergeometric model; pPERT is the probability to observe a total accumulation more extreme than tA only by chance; pG is the p-value obtained by combining pNDE and pPERT; pGFdr and pGFWER are the False Discovery Rate and respectively Bonferroni adjusted global p-values; and the Status gives the direction in which the pathway is perturbed (activated or inhibited).

**References**
runTopologyGSA

Run a topological analysis on an expression dataset using topologyGSA.

Description

Use graphical models to test the pathway components highlighting those involved in its deregulation.

If the option Ncpus is set to a value larger than 1 and the package parallel is installed, the conversion procedure will automatically use multiple cores.

Usage

runTopologyGSA(x, test, exp1, exp2, alpha, ...)

Arguments

x a PathwayList, a list of Pathways or a single Pathway object.

test Either "var" and "mean". Determine the type of test used by topologyGSA.

exp1 Experiment matrix of the first class, genes in columns.

exp2 Experiment matrix of the second class, genes in columns.

alpha Significance level of the test.

Examples

```r
if (require(SPIA) && require(hgu133plus2.db)) {
  data(colorectalCancer)
  top$ENTREZ <- mapIds(hgu133plus2.db, top$ID, "ENTREZID", "PROBEID", multiVals = "first")
  top <- top[!is.na(top$ENTREZ) & !duplicated(top$ENTREZ),]
  top$ENTREZ <- paste("ENTREZID", top$ENTREZ, sep = ":")
  tg1 <- top[top$adj.P.Val < 0.05,]
  DE_Colorectal = tg1$logFC
  names(DE_Colorectal) <- tg1$ENTREZ
  ALL_Colorectal <- top$ENTREZ
  kegg <- pathways("hsapiens", "kegg")[1:20]
  kegg <- convertIdentifiers(kegg, "ENTREZID")
  prepareSPIA(kegg, "keggEx")
  runSPIA(de = DE_Colorectal, all = ALL_Colorectal, "keggEx")
  unlink("keggExSPIA.RData")
}
```
runTopologyGSA

... Additional parameters forwarded to topologyGSA.
When invoked on a PathwayList, can use the named option "maxNodes" to limit the analysis to those pathways having up to this given number of nodes.

Details
This function produces a warning and returns NULL when the number of genes in common between the expression matrices and the pathway is less than 3.

Value
See documentation of pathway.var.test and pathway.mean.test.

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
if (require(topologyGSA)) {
  data(examples)
  colnames(y1) <- paste("SYMBOL", colnames(y1), sep = ":")
  colnames(y2) <- paste("SYMBOL", colnames(y2), sep = ":")

  k <- pathways("hsapiens", "kegg")
  p <- convertIdentifiers(k["Fc epsilon RI signaling pathway"], "SYMBOL")
  runTopologyGSA(p, "var", y1, y2, 0.05)
}
Index

* analysis
  runSPIA, 12
  runTopologyGSA, 13
* classes
  Pathway-class, 6
  PathwayList-class, 8
  Pathways-class, 10
* spia
  runSPIA, 12
* topologyGSEA
  runTopologyGSA, 13
* topology
  runSPIA, 12
  runTopologyGSA, 13
  [,PathwayList-method
    (PathwayList-class), 8
  ][,PathwayList-method
    (PathwayList-class), 8
  ]$,PathwayList-method
    (PathwayList-class), 8
  as.list.PathwayList, 2
  buildPathway, 3
  convertIdentifiers, 4
  convertIdentifiers,Pathway-method
    (Pathway-class), 6
  convertIdentifiers,PathwayList-method
    (PathwayList-class), 8
  cytoscapePlot, 5
  edges,Pathway-method (Pathway-class), 6
  graphNEL, 5, 8
  length,PathwayList-method
    (PathwayList-class), 8
  names,PathwayList-method
    (PathwayList-class), 8
  nodes,Pathway-method (Pathway-class), 6
  Pathway, 2–5, 8, 11, 13
  Pathway-class, 6
  pathway.mean.test, 14
  pathway.var.test, 14
  pathwayDatabase, 9
  pathwayDatabase (Pathway-class), 6
  pathwayDatabases, 7, 10
  pathwayGraph, 5, 8
  pathwayId (Pathway-class), 6
  PathwayList, 2, 10, 11, 13, 14
  PathwayList-class, 8
  Pathways, 8
  pathways, 6, 7, 9, 9
  Pathways-class, 10
  pathwaySpecies (Pathway-class), 6
  pathwayTimestamp (Pathway-class), 6
  pathwayTitle (Pathway-class), 6
  pathwayURL (Pathway-class), 6
  plot,Pathway, ANY-method
    (Pathway-class), 6
  prepareSPIA, 11, 12
  prepareSPIA, list-method (prepareSPIA), 11
  prepareSPIA, PathwayList-method
    (PathwayList-class), 8
  runClipper,Pathway-method
    (Pathway-class), 6
  runClipper,PathwayList-method
    (PathwayList-class), 8
  runSPIA, 11, 12
  runTopologyGSA, 13
  runTopologyGSA, list-method
    (runTopologyGSA), 13
  runTopologyGSA, Pathway-method
    (Pathway-class), 6
  runTopologyGSA, PathwayList-method
    (PathwayList-class), 8
runTopologyGSAMulti (runTopologyGSA), 13
spia, 11–13