Package 'SetRank'

May 11, 2016

Type Package

Title Advanced Gene Set Enrichment Analysis

Version 1.1.0

Date 2016-05-10

Author Cedric Simillion

Maintainer Cedric Simillion <cedric.simillion@dkf.unibe.ch>

Description Implements an algorithm to conduct advanced
gene set enrichment analysis on the results of genomics experiments.

License GPL-3

Imports XML, data.table, igraph

RoxygenNote 5.0.1

Suggests R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Repository CRAN

Date/Publication 2016-05-11 12:06:23

R topics documented:

buildSetCollection 2
createCytoscapeVizMap 4
createIDConverter 5
createPathwayTable 5
cytoscapeExport 6
exportGeneNets 7
exportMultipleResults 8
exportSingleResult 9
fixGraphML 10
generateContinuousMapping 10
generatePMapping 11
membershipTable 11

propertyFunctions 12
setRankAnalysis 12
styleXML 13

Index **14**

`buildSetCollection` *Create a gene set collection*

Description

Builds an object containing the collection of all gene sets to be used by the `setRankAnalysis` function.

Usage

```
buildSetCollection(..., referenceSet = NULL, maxSetSize = 500)
```

Arguments

`referenceSet`	Optional but very strongly, recommended. A vector of geneIDs specifying the background gene set against which to test for over-representation of genesets. The default is to use all genes present in the supplied gene annotation tables. However, many experiments are intrinsically biased for certain pathways e.g. because they only contain samples from a specific tissue. Supplying a suitable reference set will remove this bias. See the vignette for more details.
`maxSetSize`	The maximum number of genes in a gene set. Any gene sets with more genes will not be considered during the analysis.
`...`	One or more data frame objects containing the annotation of genes with pathway identifiers and descriptions. The idea is to provide one data frame per pathway database. Several gene set databases are provided in the organism-specific GeneSets packages. Alternatively, you can specify your own annotation tables. See the *Details* section for more information.

Value

A gene set collection which is a list object containing the following fields:

- maxSetSizeThe maximum set size applied when constructing the collection.
- referenceSetA vector listing all gene IDS that are part of the reference.
- setsA list of vectors. The list names are the pathway IDs as supplied in the `termID` column of the annotation frame(s) supplied.. Each vector contains all geneIDs of the gene set and has three attributes set: `ID`, `name`, and `db` which correspond respectively to the `termID`, `termName`, and `dbName` fields of the annotation frame.
- gThe size of the reference set.
- bigSetsA list of pathway IDs of gene sets with sizes bigger than the specified maximum set size.

- intersection.p.cutoffThe p-value cutoff used to determine which intersections of pairs of gene sets (see *Details*) are significant.
- intersectionsA data frame listing all significant intersections together with the p-value.

Execution time

This function typically takes some time to execute as it pre-calculates all significant intersections between pairs of gene sets in the collection. An intersection between two gene sets is considered significant if it contains more elements than expected by chance, given the sizes of both sets. Computation time can be sped up dramatically by running this function on multiple CPU-cores. To do so, simply set the `mc.cores` option to the desired number of cores to use, like so: `options("mc.cores=4")` Performing this calculation beforehand allows to re-use the same setCollection object for different analysis. It is therefore recommended to separate the creation of the setCollection object and the actual analysis in different scripts. Once the collection is created, it can be stored on disk using the `save` command. The analysis script can then load the collection using the `load` command.

Creation of custom annotation tables

geneID The gene identifier. Can be any type of identifier, but one must make sure that all annotation frames passed to `buildSetCollection` use the same identifier. As the packages created by the GeneSets package use Entrez Gene identifiers, it is best to use these in your own annotation frames as well. Also, make sure the identifiers as passed as character and not as integer values.

termID Pathway identifier. Make sure each pathway identifier is unique across all pathway databases used. You can do this by prefixing the IDs with a namespace identifier like `"REACTOME:"`.

termName Name of the pathway. A string describing the pathway, e.g. "negative regulation of sterol metabolism"

description Pathway description. A longer description of the pathway. This field can be a full paragraph describing what this pathway does.

dbName A short string given the name of the pathway database used for the annotation. E.g. "KEGG".

Author(s)

Cedric Simillion

Examples

```
options(mc.cores=1)
referenceSet = sprintf("gene_%02d", 1:50)
geneSets = lapply(1:9, function(i) sample(referenceSet[((i-1)*5):((i+1)*5)], 5))
annotationTable = data.frame(termID=sprintf("set_%02d", rep(1:9, each=5)),
        geneID=unlist(geneSets),
        termName = sprintf("dummy gene set %d", rep(1:9, each=5)),
        dbName = "dummyDB",
        description = "A dummy gene set DB for testing purposes")
collection = buildSetCollection(annotationTable, referenceSet=referenceSet)
```

createCytoscapeVizMap *Create a Cytoscape visual style for one or more SetRank networks.*

Description

Generates a VizMap XML file that can be imported, together with the network in Cytoscape, a tool for network analysis and visualization (urlhttp://www.cytoscape.org).

Usage

```
createCytoscapeVizMap(network = NULL, networkList = NULL,
  outputFile = "setrank.xml")
```

Arguments

network	A single network to generate a style for.
networkList	A list of networks to generate a single style for.
outputFile	The path for the ouputfile.

Details

You must either pass a single network, through the network argument, or a list of networks, through the networkList argument, to the function. When using the latter option, the function will generate a single style applicable to all networks. This style allows to easily spot differences between different networks generated from the same dataset.

Specifying both arguments will results in an error. Networks should of course be generated by setRankAnalysis function.

Value

None. The function writes out a Cytoscape VizMap XML file as a side effect.

Author(s)

Cedric Simillion

createIDConverter *Create a function to convert gene or protein IDs*

Description

Creates a function based on an AnnotationDb package. This package accepts a vector of input IDs and returns a vector of output IDs. If an input ID cannot be mapped to an output ID, to output vector will be one element shorter. This behaviour can be changed by setting the additional na.rm argument to FALSE. Likewise, if an input ID maps to multiple output IDs, the output vector will contain all of the latter. If you really need the output vector to have the same length as the input vector, you can set the drop.ambiguous argument to TRUE

Usage

```
createIDConverter(annotationPackageName, from, to)
```

Arguments

annotationPackageName	The name of the AnnotationDb package that will be used to create the conversion function. The package will be loaded automically if necessary.
from	The ID type to convert from. This should be one of the available keytypes in the AnnotationDb package. Use the keytypes function to find out which keytypes can be used.
to	The ID type to convert to. This should be one of the available columns in the AnnotationDb package. Use the cols function to find out which column names can be used.

Value

A function which takes a vector of input IDs as single argument and returns another vector with the converted IDs.

Author(s)

Cedric Simillion

createPathwayTable *Creates a table of all significant pathways in different conditions.*

Description

Creates a table of all significant pathways in different conditions.

Usage

```
createPathwayTable(networkList, setCollection)
```

Arguments

networkList	A list of SetRank networks created using the same set collection.
setCollection	The set collection used to perform the SetRank analysis.

Value

A data frame with column names being the names of the networks in the networkList argument and the rownames gene set IDs. The cells contain the adjusted p-values of each gene set in each network. When a gene set is not present in a network, the p-value will be set to 1. Two additional columns called "description" and "score" are added. The former is simply the description of the gene set. The latter is a score which attempts to reflect the importance of a gene set across the difference networks. The higher the score, the more important the network. This score is a combination of the number of networks where the gene set is observed and the geometric mean of the p-values of that set in these networks.

Author(s)

Cedric Simillion

---

cytoscapeExport *Prepare networks for visualization in Cytoscape*

Description

Exports a list of networks in GraphML format and creates a Cytoscape a VizMap XML file with a visualisation style applicable to all networks.

Usage

```
cytoscapeExport(networkList, outputDir)
```

Arguments

networkList	A named list of networks generated by the setRankAnalysis function.
outputDir	The path where the files should be written to. The last element in the path will be created when it doesn't exist.

Details

Each network will be written to a separate file in the specified output directory. The names of the supplied list of networks will be used as filenames with the .net.xml extension added. The VizMap file will be called setrank.xml and will also be written to the output directory.

Value

None. The function writes out files as a side effect.

Author(s)

Cedric Simillion

`exportGeneNets` *Create gene interaction networks for all significant gene sets.*

Description

Creates for every gene set present in one or more gene set networks a gene interaction network. This network shows all known or predicted protein-protein interactions between all genes in the gene set. Each network is written out to a file in GraphML format, with the extension .net.xml. Additionally, a file called gene_net_styles.viz.xml is created as well. This file contains for each gene set network a Cytoscape visualisation style which can be used to overlay the original expression data on top of the gene set-specific interaction network for. See the vignette for more details.

Usage

```
exportGeneNets(topTables, networks, collection, string, outDir,
  geneSetIDs = NULL, fields = c(geneID = "ENTREZID", symbol = "SYMBOL",
  logFC = "logFC", p = "adj.P.Val"))
```

Arguments

`topTables`	A named list object containing one or more data frames. The names should be the same as those of `networks` argument. Each data frame should contain the expression analysis data used as input for the corresponding network in the `networks` arguments. They should at least contain four columns listing the NCBI Entrez Gene identifier, the gene symbol, the observed log-fold change and the (adjusted) p-value of each gene. See the description of the `fields` argument for details.
`networks`	A named list object containing one or more gene set networks as returned by the `setRankAnalysis` function, based on the data in the `topTables` argument.
`collection`	The gene set collection object used to create the gene set networks in the `networks` argument.
`string`	An igraph object containing a species-specific protein-protein interaction network from which to retrieve. You can use the SetRankTools set of scripts to generate this object for your species of interest.
`outDir`	The directory where to write the output files. If this directory doesn't exist, it will be created.
`geneSetIDs`	The list of gene set identifiers for which to create gene interaction networks. When omitted, the union of all gene sets found in the `networks` argument will be used.

fields	A named vector of strings specifying the column names of the data frames in the topTables argument.

Author(s)

Cedric Simillion

---

exportMultipleResults *Export multiple SetRank networks and accompanying tables.*

Description

Given a list of SetRank networks, writes out the following files for each network:

1. SetRank network in GML format called <n>.gml where <n> is the network name.
2. A TAB-delimited file listing the signficant pathways in the network, called <n>_pathways.txt with <n> the network name.
3. A TAB-delimited file listing which significant genes belong to which pathway, called <n>_membership.txt with <n> again the network name.

The network names will be taken from the names of the input list. Additionally, two global files will be created as well:

1. A Cytoscape VizMap visualisation file, calledsetrank.xml.
2. A TAB-delimited file listing which pathways are found in which networks, called "pathways.txt".

Usage

```
exportMultipleResults(networkList, selectedGenesList, collection,
  IDConverter = NULL, outputPath = "./")
```

Arguments

networkList	A named list of SetRank networks.
selectedGenesList	A named list with the same names as the networkList argument. Each list should be a vector with the set of significant genes as Entrez Gene IDs used to construct the SetRank network with the same name.
collection	The set collection used for the SetRank analysis.
IDConverter	Optional. By default, Entrez Gene IDs will be displayed in the output tables. This argument can be used to convert these into more human-friendly gene symbols. When supplied, should be a function that takes a vector of Entrez Gene IDs as single argument and returns the values of the corresponding gene symbols or whatever identifier you wish to have displayed in the output tables.
outputPath	The name of the directory where the results should be written. If the last element of the path doesn't exist, a directory will be created.

Value

None. Files are written out as a side effect.

`exportSingleResult` *Export a SetRank network and accompanying tables.*

Description

Given a single SetRank analysis result, writes out the following files:

1. SetRank network in GML format called `<n>.gml` where `<n>` is the specified network name.
2. A Cytoscape VizMap visualisation file, called `setrank.xml`.
3. A TAB-delimited file listing the signficant pathways in the network, called `<n>_pathways.txt` with `<n>` again the network name.
4. A TAB-delimited file listing which significant genes belong to which pathway, called `<n>_membership.txt` with `<n>` again the network name.

Usage

```
exportSingleResult(network, selectedGenes, collection, networkName,
  IDConverter = NULL, outputPath = "./")
```

Arguments

`network`	A SetRank network.
`selectedGenes`	A vector with the set of significant genes as Entrez Gene IDs. This should be the same set passed to the `setRankAnalysis` function.
`collection`	The set collection used for the SetRank analysis.
`networkName`	A name used to name the different output files.
`IDConverter`	Optional. By default, Entrez Gene IDs will be displayed in the output tables. This argument can be used to convert these into more human-friendly gene symbols. When supplied, should be a function that takes a vector of Entrez Gene IDs as single argument and returns the values of the corresponding gene symbols or whatever identifier you wish to have displayed in the output tables.
`outputPath`	The name of the directory where the results should be written. If the last element of the path doesn't exist, a directory will be created.

Value

None. Files are written out as a side effect. genes = sprintf("gene_ geneSets = lapply(1:9, function(i) sample(genes[((i-1)*10):((i+1)*10)], 10)) annotationTable = data.frame(termID=sprintf("set_ geneID=unlist(geneSets), termName = sprintf("dummy gene set dbName = "dummyyDB", description = "A dummy gene set DB for testing purposes") collection = buildSetCollection(annotationTable, referenceSet=genes) network = setRankAnalysis(genes, collection, TRUE) exportSingleResult(network, genes, collection, "example", function(x) x, "example_dir")

Author(s)

Cedric Simillion

`fixGraphML` *Fix the igraph graphML export*

Description

This function only exists to fix some issues with the graphML output of the igraph module.

Usage

```
fixGraphML(fileName)
```

Arguments

`fileName`	the name of the file to fix.

Author(s)

Cedric Simillion

`generateContinuousMapping` *Generates a continuous VizMap property mapping.*

Description

Used by the `createCytoscapeVizMap` function. For internal use only.

Usage

```
generateContinuousMapping(attributeName, visualValues, attributeValues,
  attributeType = "float", lesserValue = NA, greaterValue = NA)
```

Arguments

`attributeName`	The node or edge attribute for which to create a visual property mapping.
`visualValues`	The values the visual property should have a key points.
`attributeValues`	The key point attribute values
`attributeType`	The type of the attribute. Must be either "`float`" or "`integer`".
`lesserValue`	Optional. The value for the visual property for attribute values below the lowest key point specified in `attributeValues`.
`greaterValue`	Optional. The value for the visual property for attribute values above the highest key point specified in `attributeValues`.

Value

An xmlNode object representing the VizMap property mapping.

`generatePMapping` *Generates a visual property mapping for p-value attributes*

Description

Used by the `createCytoscapeVizMap` function. For internal use only.

Usage

```
generatePMapping(attributeTables, graphAttribute)
```

Arguments

`attributeTables`	A list of two elements, "node" and "edge" containing the concatenated node and edge attributes for all networks supplied to `createCytoscapeVizMap`.
`graphAttribute`	The name of the node or edge attribute for which to create the mapping

Value

An xmlNode object representing the VizMap property mapping.

`membershipTable` *Create a gene set membership table.*

Description

Creates a table showing which significant genes belong to which significant gene sets. This table allows to investigate the results of a SetRank analysis in more detail.

Usage

```
membershipTable(selectedGenes, collection, network)
```

Arguments

`selectedGenes`	The set of significant genes used during the SetRank analysis.
`collection`	The setCollection used during the SetRank analysis.
`network`	A network generated by the `setRankAnalysis` function.

Author(s)

Cedric Simillion

References

A matrix of boolean values. The rownames are the geneIDs, the column names are the gene set IDs. A TRUE value in a cell indicates that the gene of the row is present in the gene set of the column; FALSE means otherwise.

---

propertyFunctions *Attribute handlers for generating the VizMap XML*

Description

A list of functions. Each name corresponds to a node or edge visual property. The function tells the createCytoscapeVizMap function what type of attribute mapping to generate.

---

setRankAnalysis *Advanced gene set enrichment analysis.*

Description

Performs advanced gene set enrichment analysis on a set of genes.

Usage

```
setRankAnalysis(geneIDs, setCollection, use.ranks = TRUE, setPCutoff = 0.01,
  fdrCutoff = 0.05, delete = TRUE)
```

Arguments

geneIDs	A vector containing the set of gene IDs to test for gene set enrichment. This is typically the list of significant genes returned by the analysis of an omics dataset.
setCollection	A gene set collection object, generated with the buildSetCollection function.
use.ranks	Logical value indicating if the geneIDs vector is in ranked order or not. When TRUE, a ranked analysis will be performed.
setPCutoff	The p-value cutoff to be used to consider a gene set significant. Recommended value: 0.01
fdrCutoff	The cutoff to be applied on the corrected p-value after false-positive sets have been removed.
delete	A flag indicating if non-significant gene sets should be deleted, which is the behaviour that you want. It is best to ignore this argument, it was only added for debugging purposes.

Value

An igraph object. Use the igraph `get.data.frame` function to get a data frame with all the significant gene sets.

Author(s)

Cedric Simillion

Examples

```
options(mc.cores=1)
reference = sprintf("gene_%03d", 1:50)
geneSets = lapply(1:9, function(i) sample(reference[((i-1)*5):((i+1)*5)], 5))
annotationTable = data.frame(termID=sprintf("set_%02d", rep(1:9, each=5)),
        geneID=unlist(geneSets),
        termName = sprintf("dummy gene setet %d", rep(1:9, each=5)),
        dbName = "dummyDB",
        description = "A dummy gene set DB for testing purposes")
collection = buildSetCollection(annotationTable, referenceSet=reference)
genes = reference[sample(c(TRUE, FALSE), 50, TRUE)]
network = setRankAnalysis(genes, collection, TRUE)
```

`styleXML` *VizMap XML template*

Description

data

Details

Datastructure containg the template for the VizMap XML file. For internal use only

Index

buildSetCollection, 2, *12*

createCytoscapeVizMap, 4, *12*
createIDConverter, 5
createPathwayTable, 5
cytoscapeExport, 6

exportGeneNets, 7
exportMultipleResults, 8
exportSingleResult, 9

fixGraphML, 10

generateContinuousMapping, 10
generatePMapping, 11
get.data.frame, *13*

membershipTable, 11

propertyFunctions, 12

setRankAnalysis, *2*, *4*, *6, 7*, *9*, *11*, 12
styleXML, 13