Package ‘Prius’

June 2, 2016

Type Package

Title Affected pathway ranking in differential gene expression analysis

Version 0.99.0

Date 2016-04-02

Author Shailesh Patil, Bharath Venkatesh, Randeep Singh

Maintainer Bharath Venkatesh <bharath.venkatesh85@gmail.com>

Depends R (>= 3.2.0)

Suggests iRefR,TCGAbiolinks,knitr

VignetteBuilder knitr

Imports sqldf,entropy,igraph,stringr,GEOquery,Biobase

Description An R package to analyse affected pathways in differential expression experiments. The approach is to disturb each node of a gene-gene interaction network proportional to the observed expression from the experiment and rank the pathways based on how much they are affected. The package currently supports microarray data from Gene Expression Omnibus, PPI data from iRef and Reactome and pathways from Reactome.

biocViews GeneExpression,DifferentialExpression,Genetics,NetworkEnrichment,GraphAndNetwork, Microarray

License MIT + file LICENSE

URL https://github.com/bhatturam/prius

BugReports https://github.com/bhatturam/prius/issues

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

computePageRanks *Compute the pageRanks for the genes.*

Description

This function computes the pageRanks for the genes based on the input personalization vectors constructed from experimental data.

Usage

```
computePageRanks(personalizationVectors, PPIGraph, alpha)
```

Arguments

`personalizationVectors`
A data frame with two columns npv - normal personalization vector and dpv - disease personalization vector. There are as many rows as number of HGNC Symbols in experimentData

`PPIGraph` PPIGraph PPIGraph A list containing igraph_object - the igraph object and vertex_map - a frame containing node attributes currently indegree and outdegree. The output of `createIGraphObject`

`alpha` The damping factor

Value

A data frame with two columns containing the normal page rank and the disease page rank. There are as many rows as number of HGNC Symbols in PPI Graph

Examples

```
## Not run:
gds = getGEO('GDS3837',AnnotGPL = TRUE, getGPL = TRUE,destdir = "/tmp")
gpl = getGEO(Meta(gds)$platform,destdir = "/tmp")
eset = GDS2eSet(gds, GPL=gpl,do.log2=FALSE)
numPairs = dim(pData(eset))[1]/2
expressionDataGEO=importExpressionDataGEO(eset,gpl,defaultProbeSelector,probeCombinerMean)
experimentDataGEO=runTestOnData(expressionDataGEO,1:60,61:120,logPairedTTestFunction)
personalizationVectorsGEO=computePersonalizationVectors(experimentDataGEO,reactomePPIGraph,data.frame(foldChan
pageRanksGEO=computePageRanks(personalizationVectorsGEO,reactomePPIGraph,0.7)

## End(Not run)
```

computePathwayScores *Compute the pathway scores from the pageRanks.*

Description

This function computes the pathway scores for the given pathways and the pageRanks using the passed distanceFunction

Usage

```
computePathwayScores(pageRanks, pathwayMembership, distanceFunction,
  distanceFunctionExtraArgs)
```

Arguments

`pageRanks` A data frame with two columns containing the normal page rank and the disease page rank. There are as many rows as number of HGNC Symbols in PPI Graph. The output from `computePageRanks`

`pathwayMembership` A list containing two elements - pathways- a single column data frame with description string for each unique pathway id memberships - a list of character vectors of gene ids indexed by pathway id, containing the list of hgnc symbols of genes in each pathway. The output of `loadPathwayDataReactome`

`distanceFunction` A function to compare the normal and disease pagerank vectors. See `KLDivergenceDistanceFunction` or `meanAbsoluteDeviationDistanceFunction` foe examples

`distanceFunctionExtraArgs` A list of extra arguments to be passed to the distance function if any.

Value

A data frame containing the deviation score for each pathway.

Examples

```
## Not run:
gds = getGEO('GDS3837',AnnotGPL = TRUE, getGPL = TRUE,destdir =
"/tmp") gpl = getGEO(Meta(gds)$platform,destdir = "/tmp") eset =
GDS2eSet(gds, GPL=gpl,do.log2=FALSE) numPairs = dim(pData(eset))[1]/2
expressionDataGEO=importExpressionDataGEO(eset,gpl,defaultProbeSelector,probeCombinerMean)
experimentDataGEO=runTestOnData(expressionDataGEO,1:60,61:120,logPairedTTestFunction)
personalizationVectorsGEO=computePersonalizationVectors(experimentDataGEO,reactomePPIGraph,data.frame(foldChan
pageRanksGEO=computePageRanks(personalizationVectorsGEO,reactomePPIGraph,0.7)
pathwayScoresGEO=computePathwayScores(pageRanksGEO,reactomePathwayData$pathwayMembership,meanAbsoluteDeviation

## End(Not run)
```

---

computePersonalizationVectors

Compute the personalization vectors based on experimental data based on a score function

---

Description

This function computes the personalization vectors from the experiment data based on an input score function based on the test parameters.

Usage

```
computePersonalizationVectors(experimentalData, PPIGraph, defaults,
  scoreFunction, scoreFunctionExtraArgs)
```

Arguments

`experimentalData`
: experimentalData A data frame with as many rows as HGNC Gene Symbols, with the columns defined by the output of testFunction. The output of `runTestOnData`.

`PPIGraph`
: PPIGraph A list containing igraph_object - the igraph object and vertex_map - a frame containing node attributes currently indegree and outdegree. The output of `createIGraphObject`

`defaults`
: A data frame corresspoding to the default experimental values for genes missing from the experiment. This must be of the same format as the output of the scoreFunction.

`scoreFunction`
: A score function that accepts the experimentalData and the PPI Graph to return the normal and disease personalization vectors. Foe example see `outdegreeNormalizedFCScoreFunctio` or `outdegreeNormalizedFCOneMinusPScoreFunction`

`scoreFunctionExtraArgs`
: A list of extra arguments that are to be passed to the score function if any.

Value

A data frame with two columns npv - normal personalization vector and dpv - disease personalization vector. There are as many rows as number of HGNC Symbols in PPIGraph

Examples

```
## Not run:
gds = getGEO('GDS3837',AnnotGPL = TRUE, getGPL = TRUE,destdir = "/tmp")
gpl = getGEO(Meta(gds)$platform,destdir = "/tmp")
eset = GDS2eSet(gds, GPL=gpl,do.log2=FALSE)
numPairs = dim(pData(eset))[1]/2
expressionDataGEO=importExpressionDataGEO(eset,gpl,defaultProbeSelector,probeCombinerMean)
experimentDataGEO=runTestOnData(expressionDataGEO,1:60,61:120,logPairedTTestFunction)
personalizationVectorsGEO=computePersonalizationVectors(experimentDataGEO,reactomePPIGraph,data.frame(foldChan

## End(Not run)
```

createIGraphObject *Create graph object from gene interaction list*

Description

Convert a data frame containing the gene interaction list into a form usable by the other functions in the library. This is just a wrapper around the igraph graph.edgelist function, which creates and returns an igraph object with a data frame containing the in and outdegrees.

Usage

```
createIGraphObject(geneInteractions)
```

Arguments

`geneInteractions`

- a data frame that contains the gene interaction graph in two columns, the first being the source HGNCSymbol and the second being the target HGNCSymbol of an interaction. See `getGeneInteractionsFromReactomeMITAB`, `getGeneInteractionsFromIRefMIT` for more details

Value

A list containing igraph_object - the igraph object and vertex_map - a frame containing node attributes currently indegree and outdegree

References

http://igraph.org/r/doc/graph_from_edgelist.html

See Also

getGeneInteractionsromReactomeMITAB, getGeneInteractionsFromIRefMITAB

Examples

```
## Not run:
     data_folder=tempdir()
     usmap<-getUniProtToHGNCSymbolMapping(data_folder)
     mitab <- downloadReactomeInteractionsMITAB(data_folder)
     geneInteraction <- getGeneInteractionListFromReactomeMITAB(mitab,usmap)
     ppiGraph<-createIGraphObject(geneInteraction)

## End(Not run)
```

---

downloadReactomeInteractionsMITAB

Download PPI Reactions from Reactome in MITAB format

Description

Downloads current PPI data (protein-to-protein interactions) in MITAB format from reactome, saves it in a local directory and returns a data frame usable by the library. It can reuse downloaded files if the downloaded folder is passed as an argument

Usage

```
downloadReactomeInteractionsMITAB(data_folder, species = "HomoSapiens")
```

Arguments

data_folder	- The folder where the data resides, or where it should be downloaded if it doesnt exist
species	- The species being analysed. Currently only supports the string 'HomoSapiens'

Value

a data frame that results when a mitab file is read

References

http://www.reactome.org/pages/download-data/, http://www.psidev.info/groups/molecular-interactions

Examples

```
## Not run:
     data_folder <- tempdir()
     reactomeMITAB<-downloadReactomeInteractionsMITAB(data_folder)

## End(Not run)
```

---

downloadReactomePathways

Download current Pathway Data from Reactome

Description

Downloads current pathway data (protein-to-pathway mapping) from reactome, saves it in a local directory and returns a data frame usable by the library. It can reuse downloaded files if the downloaded folder is passed as an argument

Usage

```
downloadReactomePathways(data_folder, species = "HomoSapiens")
```

Arguments

data_folder	- The folder where the data resides, or where it should be downloaded if it doesnt exist
species	- The species being analysed. Currently only supports the string 'HomoSapiens'

Value

a data frame containing three columns, uniProtID, reactomeID, reactomeDescription which stand for the protein identifier, pathway identifier and the pathway description

References

http://www.reactome.org/pages/download-data/

Examples

```
## Not run:
     dataFolder <- tempdir()
     pathwayData<-downloadReactomePathways(dataFolder)

## End(Not run)
```

`foldChangeFunction` *Simple fold change test function*

Description

Computes fold change for two sample expression data for a single gene. It is meant to be used with `runTestOnData`.

Usage

```
foldChangeFunction(nData, dData, hgncSymbol, extraArgs)
```

Arguments

`nData`	The scalar normal expression values for a gene
`dData`	The scalar disease expression values for a gene
`hgncSymbol`	The HGNC Symbol of the gene
`extraArgs`	A list of named extra arguments, the only name supported is exponent (assumed to be 1 if not provided for non log transformed data). An exponent not equal to 1 assumes that the expression data has been log transformed with base exponent

Value

A single row data frame with one column foldChange

`getGeneInteractionsFromIRefMITAB` *Get a gene interaction data frame from a iRef ppi mitab data frame*

Description

Get a gene interaction data frame from a iRef ppi mitab data frame

Usage

```
getGeneInteractionsFromIRefMITAB(mitab)
```

Arguments

`mitab`	@param mitab - The mitab data frame containing PPI's e.g Obtained using `http://www.inside-r.org/packages/cran/iRefR/docs/get_irefindex`

Value

a data frame that contains the gene interaction graph in two columns, the first being the source HGNCSymbol and the second being the target HGNCSymbol of an interaction

Examples

```
## Not run:
library(iRefR)
iref_mitab=get_irefindex(tax_id="9606",data_folder="/tmp/",iref_version = "13.0")
irefGeneInteractions = getGeneInteractionListFromIRefMITAB(iref_mitab)

## End(Not run)
```

---

getGeneInteractionsFromReactomeMITAB

Get a gene interaction data frame from a reactome ppi mitab data frame

---

Description

Get a gene interaction data frame from a reactome ppi mitab data frame

Usage

```
getGeneInteractionsFromReactomeMITAB(mitab, usmap)
```

Arguments

`mitab`	- The mitab data frame containing PPI's See `downloadReactomeInteractionsMITAB`
`usmap`	- A mapping file between UniProt ID's and HGNC Symbols See `getUniProtToHGNCSymbolMapping`

Value

a data frame that contains the gene interaction graph in two columns, the first being the source HGNCSymbol and the second being the target HGNCSymbol of an interaction

See Also

`downloadReactomeInteractionsMITAB`

Examples

```
## Not run:
     data_folder=tempdir()
     usmap<-getUniProtToHGNCSymbolMapping(data_folder)
     mitab <- downloadReactomeInteractionsMITAB(data_folder)
     geneInteractions <- getGeneInteractionsFromReactomeMITAB(mitab,usmap)

## End(Not run)
```

getUniProtToHGNCSymbolMapping

Get a mapping table between UniProt ID's and HGNC Gene Symbols from genenames.org

Description

This function downloads data from genenames.org to create a mapping table between Universal Protein Resource (UniProt) Identifiers to HUGO Gene Nomenclature Committee; gene symbols - which are the default ID's used in this package. This is particularly useful in the case of reactome data, in which interactions as well as pathway membership use UniProt ID's. It can reuse downloaded files if the downloaded folder is passed as an argument

Usage

```
getUniProtToHGNCSymbolMapping(data_folder)
```

Arguments

`data_folder`	- The folder where the data resides, or where it should be downloaded if it doesnt exist

References

http://www.genenames.org/cgi-bin/download

Examples

```
## Not run:
data_folder<-tempdir()
usmap<-getUniProtToHGNCSymbolMapping(data_folder)

## End(Not run)
```

importExpressionDataGEO

Import expression data and probe mappings from GEO dataset

Description

This function processes the expression set and gpl files obtained using the GEOQuery package

Usage

```
importExpressionDataGEO(eset, gpl, selectorFunction = probeSelectorATSAT,
  combinerFunction = probeCombinerMean)
```

Arguments

- `eset` An expression set, for more details refer to https://www.bioconductor.org/packages/3.3/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf and http://svitsrv25.epfl.ch/R-doc/library/GEOquery/html/GDS2MA.html
- `gpl` GEO Platform entity, refer http://svitsrv25.epfl.ch/R-doc/library/GEOquery/html/GPL-class.html for more details
- `selectorFunction` a function that accepts a probeName string and returns true if the probe is to be considered. See `probeSelectorATSAT` for an example.
- `combinerFunction` a function that accepts a comma separated string and returns true if the probe is to be considered. See `probeCombinerMean` for an example.

Value

A data frame with columns as sample names, and rows containing the expression values of each as HGNC Symbol.

Examples

```
## Not run:
library(GEOquery)
data_folder = tempdir()
gds = getGEO('GDS3837',AnnotGPL = TRUE, getGPL = TRUE,destdir = data_folder)
gpl = getGEO(Meta(gds)$platform,destdir = data_folder)
eset = GDS2eSet(gds, GPL=gpl,do.log2=FALSE)
gdata=importExpressionDataGEO(eset,gpl)

## End(Not run)
```

`KLDivergenceDistanceFunction`

KL Divergence between two vectors.

Description

Computes the KL divergence between two vectors. This is meant to be used as a distance function for `computePathwayScores`

Usage

```
KLDivergenceDistanceFunction(A, B, extraArgs)
```

Arguments

`A`	A vector of real values
`B`	A vector of real values of the same size as A
`extraArgs`	NA

Value

A data frame containing the KL Divergence between A and B as a single value

---

`loadExperimentData` *Load experiment data*

Description

Load experiment data

Usage

```
loadExperimentData(expdata, hgnc2probe, selectorFunction = probeSelectorATSAT,
  combinerFunction = probeCombinerMean)
```

Arguments

`expdata`	a probeName x sample matrix of expression values expression data
`hgnc2probe`	a data frame with two columns probeName and HGNCSymbol for mapping probeNames to HGNC Symbols
`selectorFunction`	a function that accepts a probeName string and returns true if the probe is to be considered. See `probeSelectorATSAT` for an example.
`combinerFunction`	a function that accepts a comma separated string and returns true if the probe is to be considered. See `probeCombinerMean` for an example.

Value

A data frame with columns as sample names, and rows as HGNC Symbols containing the expression values for analysis

loadPathwayDataReactome

Load pathway data from Reactome

Description

Load pathway data from Reactome

Usage

```
loadPathwayDataReactome(pdata, usmap)
```

Arguments

`pdata`	a data frame containing three columns, uniProtID, reactomeID, reactomeDescription which stand for the protein identifier, pathway identifier and the pathway description. See `downloadReactomePathways`,for more details
`usmap`	A mapping file between UniProt ID's and HGNC Symbols See `getUniProtToHGNCSymbolMapping`

Value

A list containing two elements - pathways- a single column data frame with description string for each unique pathway id memberships - a list of character vectors of gene ids indexed by pathway id, containing the list of hgnc symbols of genes in each pathway.

Examples

```
## Not run:
     dataFolder <- tempdir()
     rawPathwayData<-downloadReactomePathways(dataFolder)
     pdata<-loadPathwayDataReactome(pdata)

## End(Not run)
```

logPairedTTestFunction

Paired t-test based test function

Description

Perform a paired t-test on expression data for a single gene It is meant to be used with `runTestOnData` This function is used to obtain a p-value on the fold change between normal and disease samples for a single gene. If the data is already logarithm transformed in this function, then the exponent must be passed in the list extraArgs

Usage

```
logPairedTTestFunction(nData, dData, hgncSymbol, extraArgs)
```

Arguments

`nData` A vector with numSamples/2 normal expression values for a gene

`dData` A vector with numSamples/2 disease expression values for a gene

`hgncSymbol` The HGNC Symbol of the gene

`extraArgs` #' @param extraArgs A list of named extra arguments, the only name supported is exponent (assumed to be 1 if not provided for non log transformed data). An exponent not equal to 1 assumes that the expression data has been log transformed with base exponent

Value

A single row data frame with two columns foldChange and pValue

---

`meanAbsoluteDeviationDistanceFunction`

Mean Absolute Deviation between two vectors.

Description

Computes the mean absolute deviation between two vectors. This is meant to be used as a distance function for `computePathwayScores`

Usage

```
meanAbsoluteDeviationDistanceFunction(A, B, extraArgs)
```

Arguments

`A` A vector of real values

`B` A vector of real values of the same size as A

`extraArgs` NA

Value

A data frame containing the mean absolute deviation between A and B as a single value

outdegreeNormalizedFCOneMinusPScoreFunction *Outdegree normalized fold change and p-value based scoring*

Description

Uses the fold change, p-values and the outdegree of the nodes to define a score function for use with `computePersonalizationVectors`

Usage

```
outdegreeNormalizedFCOneMinusPScoreFunction(experimentalData, PPIGraph,
  extraArgs)
```

Arguments

`experimentalData`
: experimentalData A data frame with as many rows as HGNC Gene Symbols, with the columns defined by the output of testFunction. The output of `runTestOnData`.

`PPIGraph`
: PPIGraph A list containing igraph_object - the igraph object and vertex_map - a frame containing node attributes currently indegree and outdegree. The output of `createIGraphObject`

`extraArgs`
: NA

Value

A data frame with two columns npv - normal personalization vector and dpv - disease personalization vector. There are as many rows as number of HGNC Symbols in PPIGraph

outdegreeNormalizedFCScoreFunction *Outdegree normalized fold change based scoring*

Description

Uses the fold change values and the outdegree of the nodes to define a score function for use with `computePersonalizationVectors`

Usage

```
outdegreeNormalizedFCScoreFunction(experimentalData, PPIGraph, extraArgs)
```

Arguments

experimentalData
: A data frame with as many rows as HGNC Gene Symbols, with the columns defined by the output of testFunction. The output of `runTestOnData`.

PPIGraph
: A list containing igraph_object - the igraph object and vertex_map - a frame containing node attributes currently indegree and outdegree. The output of `createIGraphObject`

extraArgs
: NA

Value

A data frame with two columns npv - normal personalization vector and dpv - disease personalization vector. There are as many rows as number of HGNC Symbols in PPIGraph

`probeCombinerMean` *Combine probes by mean expression value*

Description

Multiple probes may map to the same HGNC Symbol, and this function that is to be used as input to the `loadExpressionData` combines these probes by taking the mean of the expression values

Usage

```
probeCombinerMean(pls, expdata)
```

Arguments

expdata
: a matrix of expression values returned from functions such as `'http://svitsrv25.epfl.ch/R-doc/library/Biobase/html/exprs.html'`

A
: string containing comma separated probe names

Value

a single value representing the gene expression

probeSelectorATSAT *Select only _at and _a_at probe sets for analysis*

Description

There are probe sets with different suffixes depending on probe behavior,(See http://www.affymetrix.com/support/help/faqs/hgu133/index.jsp for more information) and this function that is to be used as input to the loadExpressionData selects only the expression values corresponding probe set suffixes _at and _a_at for analysis

Usage

```
probeSelectorATSAT(probeName)
```

Arguments

probeName	the name of the probe set

Value

a binary value indicating if probeName is to be selected

runTestOnData *Test expression data*

Description

Runs the specified test function on the expression data to obtain fold change and p-values.

Usage

```
runTestOnData(data, normalSampleIndexes, diseaseSampleIndexes, testFunction,
  testFunctionExtraArgs)
```

Arguments

data	A data frame with columns as sample names, and rows containing the expression values of each as HGNC Symbol. For example the output of importExpressionDataGEO
normalSampleIndexes	The indices of the columns which contain the normal samples
diseaseSampleIndexes	The indices of the columns which contain the disease samples
testFunction	The test function - see logPairedTTestFunction
testFunctionExtraArgs	Extra arguments to the test function

Value

A data frame with as many rows as HGNC Gene Symbols, with the columns defined by the output of testFunction.

Examples

```
## Not run:
gds = getGEO('GDS3837',AnnotGPL = TRUE, getGPL = TRUE,destdir = "/tmp")
gpl = getGEO(Meta(gds)$platform,destdir = "/tmp")
eset = GDS2eSet(gds, GPL=gpl,do.log2=FALSE)
numPairs = dim(pData(eset))[1]/2
expressionDataGEO=importExpressionDataGEO(eset,gpl,defaultProbeSelector,probeCombinerMean)
experimentDataGEO=runTestOnData(expressionDataGEO,1:60,61:120,logPairedTTestFunction)

## End(Not run)
```

Index