Package 'metaboGSE'

April 13, 2018

Type Package

Title Gene Set Enrichment Analysis via Integration of Metabolic Networks and RNA-Seq Data

Version 1.0.2

Depends R (>= 3.4.0), sybil (>= 2.0.4), AnnotationDbi (>= 1.32.3), ape (>= 3.3), topGO (>= 2.24.0)

Suggests clpAPI (>= 1.2.6), glpkAPI (>= 1.2.11), KernSmooth, KEGGREST, survival, ctc, knitr, rmarkdown

Imports utils, stats, methods, parallel, grDevices, graphics, Matrix

Author Van Du T. Tran [aut, cre], Marco Pagni [aut]

Maintainer Van Du T. Tran <thuong.tran@sib.swiss>

Description Integrates metabolic networks and RNA-seq data to construct condition-specific series of metabolic sub-networks and applies to gene set enrichment analysis (Tran et al. (2018) <doi:10.1101/200964>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-13 09:46:26 UTC

R topics documented:

`exprMaguire` *Gene expression from Maguire et al. RNA-seq data*

Description

Gene expression from Maguire et al. RNA-seq data

Usage

```
data(exprMaguire)
```

Author(s)

Maguire et al.

References

Maguire, S. L. et al. (2014) Zinc finger transcription factors displaced SREBP proteins as the major Sterol regulators during Saccharomycotina evolution. PLoS Genet. 10, e1004076.

`iMK735` *iMK735 metabolic models in hypoxia and normoxia*

Description

iMK735 metabolic models in hypoxia and normoxia

Usage

```
data(iMK735)
```

Author(s)

Kavscek et al.

References

Kavšček, M., Bhutada, G., Madl, T. & Natter, K. (2015) Optimization of lipid production with a genome-scale model of Yarrowia lipolytica. BMC Syst. Biol. 9, 72.

`metaboGSE` *Gene set enrichment analysis*

Description

This function performs the gene set enrichment analysis.

Usage

```
metaboGSE(scores, geneset.ID = NULL, method = "perm", test = NA,
  nperm = 1000, nrand = 1000, mc.cores = 1, posthoc = TRUE,
  contrast = FALSE, prefix = "sub", desc.data = NULL)
```

Arguments

`scores`	A list of `scoreGeneDel` objects.
`geneset.ID`	A character vector of gene set IDs. Default: all from `scores`.
`method`	Statistical testing method `c("perm", "survival")`. Default: `"perm"`. `"survial"` may be used for exploration.
`test`	Type of test `c("likelihood", "logrank", "wald")`, when method = `"survival"`. Default: `"likelihood"`.
`nperm`	Number of permutations for testing, when method = `"perm"`. Default: 10000.
`nrand`	Number of draws for random gene set generation, when method = `"perm"`. Default: 1000.
`mc.cores`	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
`posthoc`	A logical value indicating if pairwise tests are performed. Default: TRUE.
`contrast`	A logical value indicating if the Newick-based contrast will be computed. Default: FALSE.
`prefix`	A string indicating prefix of output files. Default: "sub".
`desc.data`	A vector of descriptions of a priori KEGG pathway IDs. Default: NULL, KEGGREST will be called with internet connection required if geneset.ID is KEGG pathway ID.

Value

Gene set enrichment information

Examples

```
data(yarliSubmnets)
metaboGSE(yarliSubmnets[c('SH','SN')], geneset.ID = "GO:0006696", method="perm", nperm=10, nrand=10)
```

`pwDesc` *Description of gene set IDs*

Description

This function returns the description of given gene set IDs, e.g. GO, KEGG.

Usage

```
pwDesc(x, desc.data = NULL)
```

Arguments

`x`	A vector of gene set IDs.
`desc.data`	A named vector of descriptions of all studied gene set IDs. Default: NULL, AnnotationDbi is used if x is a GO term ID. KEGGREST will be called with internet connection required if x is a KEGG pathway ID.

Value

Description

Examples

```
pwDesc("GO:0006696")
pwDesc("genesetX", desc.data=setNames("processX", "genesetX"))

pwDesc("hsa04930")
```

`relativeExpr` *Relative expression data (log2 counts)*

Description

This function computes the relative gene expression (log2 counts) in a sample w.r.t all samples.

Usage

```
relativeExpr(expr, power = 1)
```

Arguments

expr	A matrix of expression data (log2 counts), with gene per row and sample per column.
power	The power n of relative expression: (x^n/<x>)^(1/n). Default: 1.

Value

A matrix of relative log2 counts.

Examples

```
data(exprMaguire)
relExpr1 <- relativeExpr(exprMaguire$expr, 1)
```

---

rescue *Rescue a model*

Description

The function rescues a given model.

Usage

```
rescue(model, target, react = NULL, weight.type = "r", prefix.rescue = NA,
  prefix.rescued = NA, rescue.threshold = 1e-05)
```

Arguments

model	An object of class modelorg.
target	A numeric vector for growth target.
react	A numeric vector or a character vector containing reaction id's. Default: reactions in objective function.
weight.type	A character indicating which type of weighting to use in model objective modification, i: 1, r: 1/coefficient, s: 1/sqrt(coefficient). Default: r.
prefix.rescue	A string indicating the prefix of output rescue model. Default: no output.
prefix.rescued	A string indicating the prefix of output rescued model. Default: no output.
rescue.threshold	A numeric value indicating the threshold to consider a rescue. Default: 1e-5.

Value

The rescue and rescued models, as well as the coefficient set to rescue reactions. SYBIL_SETTINGS("OPT_DIRECTION") is set as "min".

Examples

```
data(Ec_core)
rescue(Ec_core, target=0.1)
```

rmGenes *Generate a submodel by removing genes*

Description

This functions creates a submodel by removing genes from a given model. It is similar to `deleteModelGenes` from the COBRA Toolbox.

Usage

```
rmGenes(model, genes)
```

Arguments

`model`	An object of class `modelorg`.
`genes`	A vector of genes to remove.

Value

The submodel.

scoreGeneDel *Structure of Class "scoreGeneDel"*

Description

Structure of the class `scoreGeneDel`. Objects of this class are returned by the function submnet.

Usage

```
scoreGeneDel(model = NULL, condition = NA, fitness.random = NULL,
  fitness.ranked = NULL, fitness.ranks = NULL, fitness.id.random = NULL,
  fitness.id.ranked = NULL, fitness.id.ranks = NULL, ess.gene = NULL,
  ess.reaction = NULL, gene.del = NULL, gene.sets = NULL,
  ratio.GS = NULL, sub.genes = NULL, sub.reacs = NULL,
  rescue.met = NULL)
```

Arguments

`model`	An object of class `modelorg` indicating the weighted `rescue` model obtained from the rescue process.
`condition`	The experimental condition ID.
`fitness.random`	Random-based fitness with weighting scheme.
`fitness.ranked`	Ranked-based fitness with weighting scheme.

fitness.ranks	Ranks-based fitness with weighting scheme.
fitness.id.random	Random-based fitness without weighting scheme.
fitness.id.ranked	Ranked-based fitness without weighting scheme.
fitness.id.ranks	Ranks-based fitness without weighting scheme.
ess.gene	Percentages of essential genes. The computation of essentiality is deprecated in this version.
ess.reaction	Percentages of essential reactions. The computation of essentiality is deprecated in this version.
gene.del	Number of deleted genes.
gene.sets	Gene sets.
ratio.GS	Percentages of remaining genes in each gene set.
sub.genes	Remaining genes in submodels after propagation.
sub.reacs	Remaining reactions in submodels after propagation.
rescue.met	Fraction of every rescued metabolite among random draws.

Examples

```
data(yarliSubmnets)
attributes(yarliSubmnets[[1]])
```

---

simulateSubmnet *Assess submodels built by gene removal in a condition*

Description

This function computes the significance levels of different rankings compared to the random ranking for gene removal in a condition.

Usage

```
simulateSubmnet(sgd, mc.cores = 1, njt = NULL)
```

Arguments

sgd	An object of class scoreGeneDel.
mc.cores	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
njt	An object of class phylo for colored plot of fitness weighting schema resulting from weightReacts. Default: NULL.

Value

The significance levels

Examples

```
data(yarliSubmnets)
simulateSubmnet(yarliSubmnets$DN)
```

---

submnet *Simulation of gene removal-based submodels*

Description

This function simulates the construction of submodels by removing genes.

Usage

```
submnet(model, expr, rescue.weight = NULL, ranks = NULL, step = 1,
  draw.num = 0, gene.sets = NULL, mc.cores = 1, obj.react = NA,
  tol = SYBIL_SETTINGS("TOLERANCE"), solver = SYBIL_SETTINGS("SOLVER"),
  method = SYBIL_SETTINGS("METHOD"))
```

Arguments

model	An object of class modelorg indicating the weighted rescue model obtained from the rescue process.
expr	A matrix of gene expression in a given condition, with replicate per column.
rescue.weight	A vector of rescue reaction weights. Default: NULL, the weights are computed from the given model with gene.num=1.
ranks	A matrix of score vectors for ranking genes, with gene per row, e.g. cbind(pkm expression, relative expression). Default: NULL.
step	An integer indicating the step in numbers of genes to remove. Default: 1, gene-by-gene removal. When there are many genes in the model, the step is multiplied by an exponent of 2 for later removals. This is to reduce the computing time for non-informative sub-models at the end of the series.
draw.num	Number of random draws. Default: 0.
gene.sets	Named list of gene sets for gene set enrichment analysis. Default: NULL, depletion fraction of gene sets should be further computed for gene set enrichment analysis.
mc.cores	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
obj.react	A string indicating objective reaction ID. Default: reaction producing BIOMASS.
tol	The maximum value to be considered null. Default: SYBIL_SETTINGS("TOLERANCE").
solver	sybil solver. Default: SYBIL_SETTINGS("SOLVER").
method	sybil method. Default: SYBIL_SETTINGS("METHOD").

Value

An object of class scoreGeneDel for the submodel construction simulation.

Examples

```
data(Ec_core)
mod <- rescue(Ec_core, target=0.1)
mod.weight <- changeObjFunc(mod$rescue, react=rownames(mod$coef), obj_coef=mod$coef)
expr <- matrix(replicate(2, rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4)),
               ncol=2,
               dimnames=list(sybil::allGenes(mod.weight), c('rep.1', 'rep.2')))
gene.sets <- list(X1=head(rownames(expr)), X2=tail(rownames(expr)))
sgd <- submnet(model=mod.weight,
            expr=expr, step=200, obj.react="Biomass_Ecoli_core_w_GAM", gene.sets=gene.sets)
```

---

weightReacts *Compute weights of rescue reactions*

Description

The function weightReacts computes the weights of rescue reactions.

Usage

```
weightReacts(model, mc.cores = 1, gene.num = 1, draw.num = 1000)
```

Arguments

model	An object of class modelorg indicating the weighted rescue model obtained from the rescue process.
mc.cores	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
gene.num	The number of genes to remove. If 1, oneGeneDel will be performed and draw.num will be ignored. Default: 1.
draw.num	The number of random draws. Default: 1000.

Value

A vector of weights for rescue reactions.

Examples

```
data(Ec_core)
mod <- rescue(Ec_core, target=0.1)
weightReacts(changeObjFunc(mod$rescue, react=rownames(mod$coef), obj_coef=mod$coef))
```

yarli2GO *GO annotation retrieved from Uniprot*

Description

GO annotation retrieved from Uniprot

Usage

```
data(yarli2GO)
```

Source

Uniprot

References

Tran, V.D.T. et al. (2018) Condition-specific series of metabolic sub-networks and its application for gene set enrichment analysis. bioRxiv 200964.

yarliGOdata *topGOdata object built from yarli2GO*

Description

topGOdata object built from yarli2GO

Usage

```
data(yarliGOdata)
```

Format

An object of class `"topGOdata"`.

Source

topGO_2.24.0, GO.db_3.2.2

References

Tran, V.D.T. et al. (2018) Condition-specific series of metabolic sub-networks and its application for gene set enrichment analysis. bioRxiv 200964.

yarliGSE *Gene set enrichment example for Yarrowia lipolytica*

Description

Gene set enrichment example for Yarrowia lipolytica

Usage

```
data(yarliGSE)
```

Author(s)

Van Du T. Tran

References

Tran, V.D.T. et al. (2018) Condition-specific series of metabolic sub-networks and its application for gene set enrichment analysis. bioRxiv 200964.

yarliSubmnets *Submodel construction example for Yarrowia lipolytica*

Description

Submodel construction example for Yarrowia lipolytica

Usage

```
data(yarliSubmnets)
```

Author(s)

Van Du T. Tran

References

Tran, V.D.T. et al. (2018) Condition-specific series of metabolic sub-networks and its application for gene set enrichment analysis. bioRxiv 200964.

zscoreExpr *Z-score of expression data*

Description

This function computes the z-score of expression data across different conditions.

Usage

```
zscoreExpr(expr)
```

Arguments

expr	A matrix of expression data (log2 counts), with gene per row and experiment per column.

Value

A matrix of z-score.

Examples

```
data(exprMaguire)
zExpr <- zscoreExpr(exprMaguire$expr)
```

Index