

Package ‘metaboGSE’

May 7, 2020

Type Package

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

Version 1.2.4

Depends R (>= 3.4.0), sybil (>= 2.0.4), ape (>= 3.3)

Suggests glpkAPI (>= 1.3.0), topGO (>= 2.24.0), unix, KEGGREST, survival, knitr, rmarkdown

Imports utils, stats, methods, parallel, grDevices, graphics, Matrix, AnnotationDbi (>= 1.32.3)

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:10.1093/bioinformatics/bty929>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-07 13:00:17 UTC

R topics documented:

bestRanking	2
exprMaguire	3
fitness	3
iMK735	4
metaboGSE	5

pwDesc	6
relativeExpr	7
rescue	7
rmGenes	8
scoreGeneDel	9
simulateSubmnet	10
submnet	11
weightReacts	13
yarli2GO	13
yarliGSE	14
yarliSubmnets	14
zscoreExpr	15

Index	16
--------------	-----------

bestRanking	<i>Identify the best ranking</i>
-------------	----------------------------------

Description

This function computes the performance indices of different rankings compared to the random ranking for gene removal and identify the best ranking

Usage

```
bestRanking(fns)
```

Arguments

fns List of fitness objects.

Value

The performance indices for all rankings and the best ranking.

Examples

```
data(Ec_core)
mod <- rescue(Ec_core, target=0.1)
mod.weight <- changeObjFunc(mod$rescue, react=rownames(mod$coef), obj_coef=mod$coef)
ranks <- list(
  rep.1=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4),
      sybil::allGenes(mod.weight)),
  rep.2=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4.1),
      sybil::allGenes(mod.weight)))
fn <- fitness(model=mod.weight, ranks=ranks, step=200, draw.num=1)
bestRanking(list(fn))
```

`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.

`fitness`*Fitness of gene removal-based submodels with different gene rankings*

Description

This function computes the fitness of submodels by removing genes in different gene rankings.

Usage

```
fitness(  
  model,  
  ranks,  
  rescue.weight = NULL,  
  step = 1,  
  draw.num = 0,  
  obj.react = NA,  
  mc.cores = 1,  
  timeout = 12,  
  tol = SYBIL_SETTINGS("TOLERANCE"),  
  solver = SYBIL_SETTINGS("SOLVER"),  
  method = SYBIL_SETTINGS("METHOD")  
)
```

Arguments

model	An object of class <code>modelorg</code> indicating the weighted rescue model obtained from the rescue process.
ranks	A list of data frames of scores for ranking genes, with gene per row, e.g. <code>data.frame(pkm=pkm expression, rel=relative expression)</code> .
rescue.weight	A vector of rescue reaction weights. Default: <code>NULL</code> , the weights are computed from the given model with <code>gene.num=1</code> .
step	An integer indicating the step in numbers of genes to remove. Default: <code>1</code> , gene-by-gene removal. When there are many genes in the model, the step is multiplied by an exponent of <code>2</code> 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: <code>0</code> .
obj.react	A string indicating objective reaction ID. Default: reaction producing <code>BIOMASS</code> .
mc.cores	The number of cores to use (at least <code>1</code>), i.e. at most how many child processes will be run simultaneously. Default: <code>1</code> .
timeout	The maximum time in seconds to allow for LP call to return. Default: <code>12</code> .
tol	The maximum value to be considered null. Default: <code>SYBIL_SETTINGS("TOLERANCE")</code> .
solver	<code>sybil</code> solver. Default: <code>SYBIL_SETTINGS("SOLVER")</code> .
method	<code>sybil</code> method. Default: <code>SYBIL_SETTINGS("METHOD")</code> .

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)
ranks <- list(
  rep.1=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4),
      sybil::allGenes(mod.weight)),
  rep.2=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4.1),
      sybil::allGenes(mod.weight)))
fn <- fitness(model=mod.weight, ranks=ranks, step=200, draw.num=1)
```

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,
  gene.sets = NULL,
  method = "perm",
  test = NA,
  nperm = 1000,
  nrand = 1000,
  mc.cores = 1,
  posthoc = TRUE,
  contrast = FALSE,
  prefix = NA,
  desc.data = NULL,
  cols = NULL,
  ltys = NULL
)
```

Arguments

<code>scores</code>	A list of <code>scoreGeneDe1</code> objects.
<code>gene.sets</code>	A named list of gene sets for gene set enrichment analysis, or a vector of gene set IDs computed in <code>scores</code> . Default: <code>NULL</code> , all gene sets from <code>scores</code> .
<code>method</code>	Statistical testing method <code>c("perm", "survival")</code> . Default: <code>"perm"</code> . <code>"survival"</code> may be used for exploration.
<code>test</code>	Type of test <code>c("likelihood", "logrank", "wald")</code> , when <code>method = "survival"</code> . Default: <code>"likelihood"</code> .

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 plots. Default: NA, no plot.
desc.data	A vector of descriptions of a priori KEGG pathway IDs. Default: NULL, KEGGREST will be called with internet connection required if gene.sets is KEGG pathway.
cols	Colors for conditions. Default: rainbow colors.
ltys	Line types for conditions. Default: incrementing line types in R.

Value

Gene set enrichment information

Examples

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

pwDesc	<i>Description of gene set IDs</i>
--------	------------------------------------

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	<i>Relative expression data (log2 counts)</i>
--------------	---

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: $(x^n / \langle x \rangle)^{1/n}$. Default: 1.

Value

A matrix of relative log2 counts.

Examples

```

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

```

rescue	<i>Rescue a model</i>
--------	-----------------------

Description

The function rescues a given model.

Usage

```
rescue(
  model,
  target,
  react = NULL,
  weight.type = "r",
  timeout = 12,
  prefix.rescue = NA,
  prefix.rescued = NA,
  rescue.threshold = 1e-05
)
```

Arguments

model	An object of class <code>modelorg</code> .
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.
timeout	The maximum time in seconds to allow for LP call to return. Default: 12.
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 to "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.

Examples

```
data(Ec_core)
rmGenes(Ec_core, head(sybil::allGenes(Ec_core)))
```

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.ranks = NULL,  
  fitness.id.random = 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,  
  sub.metas = NULL,  
  rescue.met = NULL  
)
```

Arguments

model	An object of class <code>modelorg</code> indicating the weighted rescue model obtained from the rescue process.
condition	The experimental condition ID.
fitness.random	Random-based fitness with weighting scheme.
fitness.ranks	Ranks-based fitness with weighting scheme.
fitness.id.random	Random-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.
sub.metas	Remaining metabolites in submodels after propagation.
rescue.met	Fraction of every rescued metabolite among random draws.

Value

An object of class `scoreGeneDel`.

Examples

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

simulateSubmnet

Plot fitness of submodels built by gene removal in a condition

Description

This function plots the fitness of submodels built by gene removal in a condition with different rankings.

Usage

```
simulateSubmnet(
  sgd,
  mc.cores = 1,
  ranks.name = NULL,
  njt = NULL,
  cols = NULL,
  ltys = NULL
)
```

Arguments

<code>sgd</code>	An object of class <code>scoreGeneDel</code> .
<code>mc.cores</code>	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
<code>ranks.name</code>	Names of gene expression ranking. Default: NULL.
<code>njt</code>	An object of class <code>phylo</code> for colored plot of fitness weighting schema resulting from <code>weightReacts</code> . Default: NULL.
<code>cols</code>	Colors for conditions. Default: rainbow colors.
<code>ltys</code>	Line types for conditions. Default: incrementing line types in R.

Examples

```
data(yarliSubmnets)

simulateSubmnet(yarliSubmnets$UH)
```

<code>submnet</code>	<i>Simulation of gene removal-based submodel series with a given ranking</i>
----------------------	--

Description

This function simulates the construction of a series of submodels by removing genes in a given ranking.

Usage

```
submnet(
  model,
  fn,
  rank.best = "expr",
  gene.sets = NULL,
  mc.cores = 1,
  obj.react = NA,
```

```

    timeout = 12,
    tol = SYBIL_SETTINGS("TOLERANCE"),
    solver = SYBIL_SETTINGS("SOLVER"),
    method = SYBIL_SETTINGS("METHOD")
  )

```

Arguments

<code>model</code>	An object of class <code>modelorg</code> indicating the weighted rescue model obtained from the rescue process.
<code>fn</code>	An object returned by the fitness function.
<code>rank.best</code>	Name of a ranking among simulated ones. Default: "expr".
<code>gene.sets</code>	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.
<code>mc.cores</code>	The number of cores to use (at least 1), i.e. at most how many child processes will be run simultaneously. Default: 1.
<code>obj.react</code>	A string indicating objective reaction ID. Default: reaction producing BIOMASS.
<code>timeout</code>	The maximum time in seconds to allow for LP call to return. Default: 12.
<code>tol</code>	The maximum value to be considered null. Default: SYBIL_SETTINGS("TOLERANCE").
<code>solver</code>	sybil solver. Default: SYBIL_SETTINGS("SOLVER").
<code>method</code>	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)
ranks <- list(
  rep.1=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4),
      sybil::allGenes(mod.weight)),
  rep.2=data.frame(
    expr=setNames(rnorm(length(sybil::allGenes(mod.weight)), mean=5, sd=4.1),
      sybil::allGenes(mod.weight)))
fn <- fitness(model=mod.weight, ranks=ranks, step=200, draw.num=1)
gene.sets <- list(X1=head(sybil::allGenes(mod.weight)), X2=tail(sybil::allGenes(mod.weight)))
sgd <- submnet(model=mod.weight, fn=fn, rank.best="expr",
  obj.react="Biomass_Ecoli_core_w_GAM", gene.sets=gene.sets)

```

weightReacts	<i>Compute weights of rescue reactions</i>
--------------	--

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 and an object of class phylo for colored plot of fitness weighting schema.

Examples

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

yarli2GO	<i>GO annotation retrieved from Uniprot</i>
----------	---

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. *Bioinformatics* 35(13), 2258-2266.

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. *Bioinformatics* 35(13), 2258-2266.

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. *Bioinformatics* 35(13), 2258-2266.

zscoreExpr	<i>Z-score of expression data</i>
------------	-----------------------------------

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

bestRanking, [2](#)

exprMaguire, [3](#)

fitness, [3](#)

iMK735, [4](#)

metaboGSE, [5](#)

pwDesc, [6](#)

relativeExpr, [7](#)

rescue, [7](#)

rmGenes, [8](#)

scoreGeneDel, [9](#)

simulateSubmnet, [10](#)

submnet, [11](#)

sybil, [4](#)

weightReacts, [13](#)

yarli2GO, [13](#)

yarliGSE, [14](#)

yarliSubmnets, [14](#)

zscoreExpr, [15](#)