Package ‘randomGODB’

July 23, 2025

Version 1.0

Date 2025-05-05

Title Random GO Database

Maintainer Barry Zeeberg <barryz2013@gmail.com>

Author Barry Zeeberg [aut, cre]

Depends R (>= 4.2.0)

Imports minimalistGODB, GO.db, graphics, stats

Description The Gene Ontology (GO) Consortium <https://geneontology.org/> organizes genes into hierarchical categories based on biological process (BP), molecular function (MF) and cellular component (CC, i.e., subcellular localization). Tools such as 'GoMiner' (see Zeeberg, B.R.,
Feng, W., Wang, G. et al. (2003) <doi:10.1186/gb-2003-4-4-r28>) can leverage GO to perform ontological analysis of microarray and proteomics studies, typically generating a list of significant functional categories. The significance is traditionally determined by randomizing the input gene list to computing the false discovery rate (FDR) of the enrichment p-value for each category. We explore here the novel alternative of randomizing the GO database rather than the gene list.

License GPL (>= 2)

Encoding UTF-8

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2025-05-07 12:10:02 UTC

Contents

`addName2List` *addName2List*

Description

add the leaf category name to the list of ancestor categories

Usage

```
addName2List(GOBPANCESTOR)
```

Arguments

`GOBPANCESTOR`	GO.db data set

Value

returns an augmented list of ancestor categories

Examples

```
BP_ANCESTOR<-addName2List(as.list(GO.db::GOBPANCESTOR))
```

characterizeDB *characterizeDB*

Description

compute distribution of GO category sizes, and fraction of a leaf's ancestors containing a bait gene

Usage

```
characterizeDB(
  GOGOA3,
  ontology = "biological_process",
  ngene = 2,
  GOBPCHILDREN,
  GOBPANCESTOR,
  hitters = "all",
  verbose = TRUE
)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
ontology	character c("biological_process","molecular_function","cellular_component")
ngene	integer number of genes to examine within range of 'hitters'
GOBPCHILDREN	GO.db data set
GOBPANCESTOR	GO.db data set
hitters	character c("big","mid","lo","all") designate which portion of gene table to look at
verbose	BOOLEAN if TRUE print out some information

Value

returns the sorted number of GO category sizes, and also has side effect of printing out some information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
```

```
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
ontology<-"biological_process"
t<-characterizeDB(GOGOA3,ontology,ngene=3,GO.db::GOBPCHILDREN,GO.db::GOBPANCESTOR,hitters="all")

## End(Not run)
```

compare2DB *compare2DB*

Description

compare pairs of GO_HGNC in 2 databases

Usage

```
compare2DB(GOGOA3, GOGOA3R, verbose = TRUE)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
GOGOA3R	a supposedly randomized version of GOGOA3
verbose	Boolean if TRUE print out some information

Value

returns no values, but has side effect of printing information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
GOGOA3R<-randomGODB(GOGOA3)
compare2DB(GOGOA3,GOGOA3R)

## End(Not run)
```

`DBstats` *DBstats*

Description

display some gene and category stats

Usage

```
DBstats(DB, title = NULL, ontology = "biological_process", verbose = TRUE)
```

Arguments

`DB`	GOGOA3 or a randomized version of it
`title`	character if not null, title for output
`ontology`	character c("biological_process","molecular_function","cellular_component")
`verbose`	Boolean if TRUE print out some information

Value

returns no values, but prints out some stats

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
DBstats(GOGOA3,ontology="biological_process")

## End(Not run)
```

`fractAncest` *fractAncest*

Description

analysis of fraction of ancestor categories to which a leaf gene maps

Usage

```
fractAncest(
  genes,
  GOGOA3,
  ontology = "biological_process",
  GOBPCHILDREN,
  GOBPANCESTOR,
  verbose = TRUE
)
```

Arguments

genes	character vector list of gene names
GOGOA3	return value of minimalistGODB::buildGODatabase()
ontology	character c("biological_process","molecular_function","cellular_component")
GOBPCHILDREN	GO.db data set
GOBPANCESTOR	GO.db data set
verbose	Boolean if TRUE print out some information

Value

returns no values, but has side effect of printing out some results

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
ontology<-"biological_process"
genes<-c("CDC45","CELF2")
fractAncest(genes,GOGOA3,ontology,GO.db::GOBPCHILDREN,GO.db::GOBPANCESTOR)
```

```
## End(Not run)
```

---

geneListDistHitters *geneListDistHitters*

Description

compute number of GOGOA3 mappings for genes in geneList

Usage

```
geneListDistHitters(geneList, GOGOA3, ontologies = NULL, verbose = TRUE)
```

Arguments

geneList	character vector listg of gene names
GOGOA3	return value of minimalistGODB::buildGODatabase()
ontologies	character c("biological_process","molecular_function","cellular_component")
verbose	Boolean if TRUE print out some information

Value

returns no value, but has side effect of printing information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
geneList<-GOGOA3$ontologies[["biological_process"]][1:10,"HGNC"]
geneListDistHitters(geneList,GOGOA3)

## End(Not run)
```

hitters *hitters*

Description

pick genes of a size range and submit to fractAncest()

Usage

```
hitters(
  GOGOA3,
  ontology,
  hitters,
  ngene,
  GOBPCHILDREN,
  GOBPANCESTOR,
  verbose = TRUE
)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
ontology	character c("biological_process","molecular_function","cellular_component")
hitters	character c("big","mid","lo","all") designate which portion of gene table to look at
ngene	integer number of genes to examine within range of 'hitters'
GOBPCHILDREN	GO.db data set
GOBPANCESTOR	GO.db data set
verbose	Boolean if TRUE print out some information

Value

returns no values, but has side effect of printing out some information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
```

```
hitters(GOGOA3,ontology="biological_process",hitters="all",
 5,GO.db::GOBPCHILDREN,GO.db::GOBPANCESTOR)

## End(Not run)
```

leafList *leafList*

Description

retrieve leaf nodes

Usage

```
leafList(GOBPCHILDREN)
```

Arguments

GOBPCHILDREN	GO.db dataset

Value

returns a list of leaf nodes

Examples

```
BP_LEAF<-leafList(GO.db::GOBPCHILDREN)
```

mapPerGene *mapPerGene*

Description

characterize number of mappings per gene

Usage

```
mapPerGene(GOGOA3, ontology, verbose = TRUE)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
ontology	character c("biological_process","molecular_function","cellular_component")
verbose	Boolean if TRUE print out some information

Value

returns no values, but has side effect of printing out information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
mapPerGene(GOGOA3,ontology="biological_process")

## End(Not run)
```

postProcess *postProcess*

Description

adds secondary components to database like GOGOA3$genes etc

Usage

```
postProcess(l)
```

Arguments

`l`	return value of randomGODB2()

Value

returns a database like GOGOA3

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
```

```
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
pp<-postProcess(randomGODB(GOGOA3))

## End(Not run)
```

randomGODB *randomGODB*

Description

driver to construct a randomized version of GOGOA3

Usage

```
randomGODB(GOGOA3, verbose = TRUE)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
verbose	Boolean if TRUE print out some information

Details

The results of characterizeDB() show that a gene mapping to a leaf node maps to only around 10% of the ancestors. So I do not need to use a more sophisticated method to generate a random database. That is, I do not need to maintain a consistency between leaf and ancestor mappings. Therefore a very simple randomization

- simply scrambling the genes in an ontology of GOGOA3 will suffice.

Value

description

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
GOGOA3R<-randomGODB(GOGOA3)
```

```
## End(Not run)
```

sizeGOcats *sizeGOcats*

Description

characterize size of GO categories

Usage

```
sizeGOcats(GOGOA3, ontology, verbose = TRUE)
```

Arguments

GOGOA3	return value of minimalistGODB::buildGODatabase()
ontology	character c("biological_process","molecular_function","cellular_component")
verbose	Boolean if TRUE print out some information

Value

returns no values, but has side effect of printing out information

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# so I need to load it from a file that is not in the package.
# Since this is in a file in my own file system, I could not
# include this as a regular example in the package.
# This example is given in full detail in the package vignette.
# You can generate GOGOA3.RData using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/databases
dir<-"/Users/barryzeeberg/personal/GODB_RDATA/goa_human/"
load(sprintf("%s/%s",dir,"GOGOA3_goa_human.RData"))
sizeGOcats(GOGOA3,ontology="biological_process")

## End(Not run)
```

Index