Package ‘GoMiner’

March 25, 2025

Version 1.0

Date 2025-03-16

Title Automate the Mapping Between a List of Genes and Gene Ontology Categories

Maintainer Barry Zeeberg <`barryz2013@gmail.com`>

Author Barry Zeeberg [aut, cre]

Depends R (>= 4.2.0)

Imports minimalistGODB, HGNChelper, stats, gplots, grDevices, utils

LazyData true

LazyDataCompression xz

Description In gene-expression microarray studies, for example, one generally obtains a list of dozens or hundreds of genes that differ in expression between samples and then asks 'What does all of this mean biologically?' Alternatively, gene lists can be derived conceptually in addition to experimentally. For instance, one might want to analyze a group of genes known as housekeeping genes. The work of the Gene Ontology (GO) Consortium <geneontology.org> provides a way to address that question. GO organizes genes into hierarchical categories based on biological process, molecular function and subcellular localization. The role of 'GoMiner' is to automate the mapping between a list of genes and GO, and to provide a statistical summary of the results as well as a visualization.

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 2025-03-25 09:20:02 UTC

Contents

`cluster52` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(cluster52)
```

`FDR` *FDR*

Description

compute the false discovery rate (FDR) of the hypergeometric p values of genes mapping to gene ontology (GO) categories

Usage

```
FDR(sampleList, GOGOA3, nrand, ONT)
```

Arguments

`sampleList`	character vector of user-supplied genes of interest
`GOGOA3`	return value of subsetGOGOA()
`nrand`	integer number of randomizations
`ONT`	c("molecular_function","cellular_component","biological_process")

Value

returns a list with FDR information

Examples

```
#load("data/GOGOA3small.RData")
sampleList<-randSubsetGeneList(GOGOA3small$genes[["biological_process"]],10)
fdr<-FDR(sampleList,GOGOA3small,nrand=100,"biological_process")
```

`GOenrich3` *GOenrich3*

Description

compute the gene enrichment in a GO category

Usage

```
GOenrich3(tableSample3, tablePop3)
```

Arguments

`tableSample3`	sample return value of GOtable3()
`tablePop3`	population return value of GOtable3()

Value

returns a matrix with columns c("SAMPLE","POP","ENRICHMENT")

Examples

```
#load("data/x_tableSample3.RData")
#load("data/x_tablePop3.RData")
m<-GOenrich3(x_tableSample3,x_tablePop3)
```

`GOGOA3small` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(GOGOA3small)
```

`GOheatmap` *GOheatmap*

Description

generate a matrix to be used as input to a heat map

Usage

```
GOheatmap(sampleList, x, thresh, fdrThresh = 0.105)
```

Arguments

`sampleList`	character list of gene names
`x`	DB component of return value of GOtable3()
`thresh`	output of GOthresh()
`fdrThresh`	numeric value of FDR acceptance threshold

Value

returns a matrix to be used as input to a heat map

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# you can generate it using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO
#load("~/GODB_RDATA/GOGOA3.RData")
ONT<-"biological_process"
DB<-GOGOA3$ontologies[[ONT]]

#load("data/cluster52.RData")
sampleList<-cluster52
```

```
#load("data/x_thresh.RData")
heatmap<-GOheatmap(sampleList,DB,x_thresh)

## End(Not run)
```

`GOhypergeometric3` *GOhypergeometric*

Description

compute the hypergeometric p value for gene enrichment in a GO category

Usage

```
GOhypergeometric3(tableSample3, tablePop3)
```

Arguments

`tableSample3`	sample return value of GOtable3()
`tablePop3`	population return value of GOtable3()

Value

returns a matrix with columns c("x","m","n","k","p")

Examples

```
#load("data/x_tableSample3.RData")
#load("data/x_tablePop3.RData")
hyper<-GOhypergeometric3(x_tableSample3,x_tablePop3)
```

`GoMiner` *GoMiner*

Description

driver to generate heatmap

Usage

```
GoMiner(
  title = NULL,
  dir,
  sampleList,
  GOGOA3,
  ONT,
  enrichThresh = 2,
  countThresh = 5,
  fdrThresh = 0.1,
  nrand = 100
)
```

Arguments

`title`	character string descriptive title
`dir`	character string full pathname to the directory acting result repository
`sampleList`	character list of gene names
`GOGOA3`	return value of subsetGOGOA()
`ONT`	character string c("molecular_function", "cellular_component", "biological_process")
`enrichThresh`	numerical acceptance threshold for enrichment
`countThresh`	numerical acceptance threshold for gene count
`fdrThresh`	numerical acceptance threshold for fdr
`nrand`	numeric number of randomizations to compute FDR

Value

returns a matrix suitable to generate a heatmap

Examples

```
## Not run:
# GOGOA3.RData is too large to include in the R package
# you can generate it using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO
load("~/GODB_RDATA/GOGOA3.RData")
load("data/cluster52.RData")
l<-GoMiner("Cluster52",tempdir(),cluster52,
 GOGOA3,ONT="biological_process",enrichThresh=2,
 countThresh=5,fdrThresh=0.10,nrand=10)

## End(Not run)
```

`GOtable3` *GOtable3*

Description

tabulate number of geneList mappings to GO categories

Usage

```
GOtable3(hgncList, DB)
```

Arguments

`hgncList`	character list of gene names
`DB`	selected ontology branch of return value of subsetGOGOA

Value

returns a list whose components are c("DB","table","ngenes") where 'DB' is the GO DB subsetted to the desired ONTOLOGY, and 'table' is tabulation of number of occurrences of each GO category name within the desired ONTOLOGY, and ngenes is the total number of hgncList genes mapping to GOGOA

Examples

```
#load("data/GOGOA3small.RData")
DB<-GOGOA3small$ontologies[["biological_process"]]

# housekeeping genes downloaded from https://housekeeping.unicamp.br/?download
#load("data/Housekeeping_Genes.RData")
hgncList<-Housekeeping_Genes[,"Gene.name"]
x<-GOtable3(hgncList,DB)
```

`GOthresh` *GOthresh*

Description

retrieve lines of m that meet both enrichThresh and countThresh

Usage

```
GOthresh(m, sampleFDR, enrichThresh, countThresh, fdrThresh)
```

Arguments

`m`	return value of GOenrich3()
`sampleFDR`	component of return value of RCPD()
`enrichThresh`	numerical acceptance threshold for enrichment
`countThresh`	numerical acceptance threshold for gene count
`fdrThresh`	numerical acceptance threshold for fdr

Value

returns a subset of matrix (m joined with fdr$sampleFDR) with entries meeting all thresholds

Examples

```
#load("data/x_m.RData")
#load("data/x_fdr.RData")
thresh<-GOthresh(x_m,x_fdr$sampleFDR,enrichThresh=2,countThresh=2,fdrThresh=0.100)
```

`Housekeeping_Genes` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(Housekeeping_Genes)
```

`randSubsetGeneList` *randSubsetGeneList*

Description

retrieve n unique random genes

Usage

```
randSubsetGeneList(geneList, ngenes)
```

Arguments

`geneList`	character vector geneList
`ngenes`	integer desired number of random genes

Value

returns a character vector of genes

Examples

```
#load("data/GOGOA3small.RData")
genes<-randSubsetGeneList(GOGOA3small$genes[["biological_process"]],20)
```

`RCPD` *RCPD*

Description

prepare a cpd of p values from randomized gene sets

Usage

```
RCPD(GOGOA3, ngenes, nrand, ONT)
```

Arguments

`GOGOA3`	return value of subsetGOGOA()
`ngenes`	integer number of genes to randomize
`nrand`	integer number of randomizations
`ONT`	c("molecular_function","cellular_component","biological_process")

Details

the cpd of the randomizations is to be used for estimating the false discovery rate (FDR) of the real sampled genes

Value

returns a histogram of log10(p)

Examples

```
#load("data/GOGOA3small.RData")
rcpd<-RCPD(GOGOA3small,ngenes=100,nrand=10,ONT="biological_process")
```

`validHGNCSymbols` *validHGNCSymbols*

Description

convert outdated HGNC symbols to current HGNC symbols

Usage

```
validHGNCSymbols(geneList)
```

Arguments

`geneList`	character vector of HGNC symbols

Details

removes NA and /// from output of checkGeneSymbols()

Value

returns list of mapping table and vector of current HGNC symbols

Examples

```
geneList<-c("FN1", "tp53", "UNKNOWNGENE","7-Sep",
 "9/7", "1-Mar", "Oct4", "4-Oct","OCT4-PG4", "C19ORF71",
  "C19orf71")
l<-validHGNCSymbols(geneList)
```

`x_fdr` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(x_fdr)
```

`x_m` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(x_m)
```

`x_tablePop3` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(x_tablePop3)
```

`x_tableSample3` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(x_tableSample3)
```

`x_thresh` *GoMiner data set*

Description

GoMiner data set

Usage

```
data(x_thresh)
```

Index