Package ‘HTGM2D’

April 7, 2025

Version 1.0

Date 2025-04-03

Title Two Dimensional High Throughput 'GoMiner'

Maintainer Barry Zeeberg <barryz2013@gmail.com>

Author Barry Zeeberg [aut, cre]

Depends R (>= 4.2.0)

Imports minimalistGODB, GoMiner, HTGM, grDevices, stats, gplots, jaccard

LazyData true

LazyDataCompression xz

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. Microarray studies are usually analyzed with BP, whereas proteomics researchers often prefer CC. To capture the benefit of both of those ontologies, I developed a two-dimensional version of 'High-Throughput GoMiner' ('HTGM2D'). I generate a 2D heat map whose axes are any two of BP, MF, or CC, and the value within a picture element of the heat map reflects the Jaccard metric p-value for the number of genes in common for the corresponding pair.

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 2025-04-07 16:10:08 UTC

Contents

`catGenes` *catGenes*

Description

match up genes in gene list with categories in GOGOA3 database

Usage

```
catGenes(geneList, GOGOA3, ontology)
```

Arguments

`geneList`	character vector of user-supplied genes of interest
`GOGOA3`	return value of subsetGOGOA()
`ontology`	c("molecular_function","cellular_component","biological_process")

Value

returns a matrix of 1's and 0's indicating the presence or absence of gene-category pairs

Examples

```
#load("data/GOGOA3small.RData")
geneList<-cluster52
m1<-catGenes(geneList,GOGOA3small,"biological_process")
```

`cluster52` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(cluster52)
```

`compareGoMinerHTGM2D` *compareGoMinerHTGM2D*

Description

compare the results of GoMiner and HTGM2D

Usage

```
compareGoMinerHTGM2D(subd, mat, l, ontologies)
```

Arguments

`subd`	character string full path name to the output subdirectory
`mat`	return value of Jaccard()
`l`	of return values of GoMiner()
`ontologies`	character vector of 2 ontologies e.g. c("biological_process","cellular_component")

Value

returns no value, but saves files that list category difference between GoMiner and HTG2D

Examples

```
ontologies<-c("biological_process","cellular_component")
#load("data/x_l.Rdata")
#load("data/x_mat.Rdata")
subd<-tempdir()
compareGoMinerHTGM2D(subd,x_mat,x_l,ontologies)
```

`GOGOA3small` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(GOGOA3small)
```

`Housekeeping_Genes` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(Housekeeping_Genes)
```

`HTGM2D` *HTGM2D*

Description

run 2D version of GoMiner

Usage

```
HTGM2D(dir, geneList, ontologies, GOGOA3)
```

Arguments

`dir`	character string full path name to the directory acting as result repository
`geneList`	character vector of user-supplied genes of interest
`ontologies`	character vector of 2 ontologies e.g. c("biological_process","cellular_component")
`GOGOA3`	return value of subsetGOGOA()

Value

returns the return value of Jaccard()

Examples

```
## Not run:
# this example takes too long to run, and
# 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
# load("~/GODB_RDATA/GOGOA3.RData")
subd<-tempdir()
geneList<-cluster52
ontologies<-c("biological_process","cellular_component")
mat<-HTGM2D(subd,geneList,ontologies,GOGOA3)

## End(Not run)
```

`HTGM2Ddriver` *HTGM2Ddriver*

Description

driver to invoke GoMiner and HTGM2D, and compare the results

Usage

```
HTGM2Ddriver(
  dir,
  geneList,
  ontologies,
  GOGOA3,
  enrichThresh = 2,
  countThresh = 5,
  fdrThresh = 0.1,
  nrand = 100
)
```

Arguments

`dir`	character string full path name to the directory acting as result repository
`geneList`	character vector of user-supplied genes of interest
`ontologies`	character vector of 2 ontologies e.g. c("biological_process","cellular_component")
`GOGOA3`	return value of subsetGOGOA()
`enrichThresh`	numerical acceptance threshold for enrichment passed to GoMiner
`countThresh`	numerical acceptance threshold for gene count passed to GoMiner

`fdrThresh`	numerical acceptance threshold for fdr passed to GoMiner
`nrand`	integer number of randomizations passed to GoMiner

Value

returns no value, but saves hyperlinked SVG heatmap files to a results directory

Examples

```
## Not run:
# this example takes too long to run, and
# 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
# load("~/GODB_RDATA/GOGOA3.RData")
geneList<-cluster52
ontologies<-c("biological_process","cellular_component")
dir<-tempdir()
HTGM2Ddriver(dir,geneList,ontologies,GOGOA3,enrichThresh=2,countThresh=5,fdrThresh=0.10,nrand=100)

## End(Not run)
```

`Jaccard` *Jaccard*

Description

create the heat map data that is needed as input to JaccardHeatMap()

Usage

```
Jaccard(dir, m1, m2, thresh1 = 2, thresh2 = 3, B = 100)
```

Arguments

`dir`	character string full pathname to the directory acting as result repository
`m1`	return value of catGenes
`m2`	return value of catGenes
`thresh1`	integer acceptance threshold for the number of genes in a cat
`thresh2`	integer acceptance threshold for the number of common genes in 2 cats
`B`	integer a total bootstrap iteration

Value

returns a numerical matrix containing number of genes and associated p value in the intersection of 2 categories

Examples

```
#load("data/x_m1.RData")
#load("data/x_m2.RData")
mat<-Jaccard(dir=tempdir(),x_m1,x_m2)
```

`JaccardHeatMap` *JaccardHeatMap*

Description

use the Jaccard metric to construct 2D heat map

Usage

```
JaccardHeatMap(dir, mat)
```

Arguments

`dir`	character string containing path name of output directory
`mat`	return value of Jaccard()

Value

returns a Jaccard matrix of cat1 vs cat2 FDR, and also saves hyperlinked SVG heatmap files to a results directory

Examples

```
#load("data/x_jmat.RData")
dir<-tempdir()
jHeatMap<-JaccardHeatMap(dir,x_jmat)
```

`x_jmat` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(x_jmat)
```

`x_l` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(x_l)
```

`x_m1` *HTGM2D data set*

Description

HTGM2D data set

Usage

```
data(x_m1)
```

`x_m2` *HTGM2D data set*

Description

HTGM2D

Usage

```
data(x_m2)
```

`x_mat` *HTGM2D data set*

Description

HTGM2D

Usage

```
data(x_mat)
```

Index