Package ‘HTGM4D’

May 2, 2026

Version 1.0

Date 2026-04-29

Title Four Dimensional High Throughput 'GoMiner'

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

Author Barry Zeeberg [aut, cre]

Depends R (>= 4.2.0)

Imports minimalistGODB, GoMiner, HTGM, HTGM2D, grDevices, stats,
gplots, vprint, randomGODB, HGNChelper, png, magick, svglite,
utils, graphics

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 now present an enhancement of the existing two-dimensional version of 'High-Throughput GoMiner' ('HTGM2D'),
which is called 'HTGM4D'. The original 'HTGM2D' is augmented by adding two instances of the original
'GoMiner' genes versus categories heatmaps, aligned with the categories axes of the 'HTGM2D' heatmap.

License GPL (>= 2)

Encoding UTF-8

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-05-02 09:50:08 UTC

Contents

`buildSubdirs` *buildSubdirs*

Description

create the directory structure for the output file

Usage

```
buildSubdirs(dir)
```

Arguments

`dir`	character string path name for the top level output directory

Value

returns no values, but has side effect of creating some subdirectories

Examples

```
stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
buildSubdirs(dir)
```

`cluster52` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(cluster52)
```

`combine3SVG` *combine3SVG*

Description

combine 3 svg objects into a single svg object

Usage

```
combine3SVG(svg1, svg2, svg3, browser = "Safari")
```

Arguments

`svg1`	an svg object, presumably the result of HTGM2D()
`svg2`	an svg object, presumably the result of GoMiner()
`svg3`	an svg object, presumably the result of GoMiner()
`browser`	character string c("Safari","Google Chrome","Firefox","Brave Browser")

Details

sort of works, but rendering of axis labels are distorted for the "below" GoMiner heatmap

Value

returns the composite svg object

Examples

```
## Not run:
# must be run in the R Console, NOT from RStudio Console
# because the window for edit() is not rendered properly
# in RStudio Console

svg9<-combine3SVG(x_svg1,x_svg2,x_svg3)

## End(Not run)
```

`editor` *editor*

Description

iteratively invoke edit() to allow user to adjust relative positions of svg's

Usage

```
editor(svg, lines, m, browser)
```

Arguments

`svg`	an svg object
`lines`	integer vector of svg line numbers
`m`	numerical matrix of parameter values for svg "transform" tag
`browser`	character string c("Safari","Google Chrome","Firefox","Brave Browser")

Value

returns an svg object

Examples

```
## Not run:
# must be run in the R Console, NOT from RStudio Console
# because the window for edit() is not rendered properly
# in RStudio Console

m<-matrix(nrow=2,ncol=4)
colnames(m)<-c("x","y","scale","rotate")
m[1,]<-c(300,0,.5,0)
m[2,]<-c(150,200,.5,90)
svg9<-editor(x_svg3,x_lines,m,browser="Safari")

## End(Not run)
```

`fillLines` *fillLines*

Description

insert <g transform line into svg

Usage

```
fillLines(svg, lines, m)
```

Arguments

`svg`	an svg object
`lines`	integer vector of svg line numbers
`m`	numerical matrix of parameter values for svg "transform" tag

Value

returns an svg object

Examples

```
m<-matrix(nrow=2,ncol=4)
colnames(m)<-c("x","y","scale","rotate")
m[1,]<-c(300,0,.5,0)
m[2,]<-c(150,200,.5,90)
svg9<-fillLines(x_svg3,x_lines,m)
```

`findSVGtags` *findSVGtags*

Description

locate <svg and </svg tags

Usage

```
findSVGtags(svg)
```

Arguments

`svg`	svg object

Value

returns a vector containing the line numbers of the tags

Examples

```
findSVGtags(x_svg1)
```

`GoMiner4` *GoMiner4*

Description

version of GoMiner() to use with HTGM4D

Usage

```
GoMiner4(
  title = NULL,
  dir,
  sampleList,
  GOGOA3,
  ontology,
  enrichThresh = 2,
  countThresh = 5,
  pvalThresh = 0.1,
  fdrThresh = 0.1,
  nrand = 100,
  mn = 2,
  mx = 200,
```

```
  opt,
  verbose = 1
)
```

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()
`ontology`	character string c("molecular_function", "cellular_component", "biological_process")
`enrichThresh`	numerical acceptance threshold for enrichment
`countThresh`	numerical acceptance threshold for gene count
`pvalThresh`	numerical acceptance threshold for pval
`fdrThresh`	numerical acceptance threshold for fdr
`nrand`	numeric number of randomizations to compute FDR
`mn`	integer param passed to trimGOGOA3, min size threshold for a category
`mx`	integer param passed to trimGOGOA3, max size threshold for a category
`opt`	integer 0:1 parameter used to select randomization method
`verbose`	integer vector representing classes

Details

see full details in the original GoMiner() this version differs from the original GoMiner in returning "heatmap" now. I should make the original GoMiner, but I am afraid of breaking several programs that invoke the original GoMiner.

Value

returns a list whose elements are an svg object and a heatmap

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.
# you can generate it using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/tree/main/databases
load("/Users/barryzeeberg/personal/GODB_RDATA/goa_human/GOGOA3_goa_human.RData")

stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
```

```
buildSubdirs(dir)

l<-GoMiner4("Cluster52",dir,cluster52,
 GOGOA3=GOGOA3,ontology="biological_process",enrichThresh=2,
 countThresh=5,pvalThresh=0.10,fdrThresh=0.10,nrand=2,mn=2,mx=200,opt=0,verbose=1)

## End(Not run)
```

`HTGM4D` *HTGM4D*

Description

run 4D version of GoMiner

Usage

```
HTGM4D(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 a list containing the return value of Jaccard() and JaccardHeatMap4()

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.
# you can generate it using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/tree/main/databases
load("/Users/barryzeeberg/personal/GODB_RDATA/goa_human/GOGOA3_goa_human.RData")

stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
buildSubdirs(dir)

geneList<-cluster52
```

```
ontologies<-c("biological_process","cellular_component")
mat<-HTGM4D(dir,geneList,ontologies,GOGOA3)

## End(Not run)
```

`HTGM4Ddriver` *HTGM4Ddriver*

Description

driver to invoke GoMiner and HTGM4D, and compare the results

Usage

```
HTGM4Ddriver(
  dir,
  geneList,
  ontologies,
  GOGOA3,
  enrichThresh = 2,
  countThresh = 5,
  pvalThresh = 0.1,
  fdrThresh = 0.1,
  nrand = 100,
  mn,
  mx,
  opt = 0,
  SVG3 = FALSE,
  browser = "Safari",
  verbose = 1
)
```

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
`pvalThresh`	numerical acceptance threshold for pval
`fdrThresh`	numerical acceptance threshold for fdr
`nrand`	numeric number of randomizations to compute FDR
`mn`	integer param passed to trimGOGOA3, min size threshold for a category

`mx`	integer param passed to trimGOGOA3, max size threshold for a category
`opt`	integer 0:1 parameter used to select randomization method
`SVG3`	Boolean if TRUE invoke combine3SVG()
`browser`	character string c("Safari","Google Chrome","Firefox","Brave Browser")
`verbose`	integer vector representing classes

Value

returns the path name of the output subdirectory

Examples

```
## Not run:
# must be run in the R Console, NOT from RStudio Console
# because the window for edit() is not rendered properly
# in RStudio Console

# 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.
# you can generate it using the package 'minimalistGODB'
# or you can retrieve it from https://github.com/barryzee/GO/tree/main/databases
load("/Users/barryzeeberg/personal/GODB_RDATA/goa_human/GOGOA3_goa_human.RData")
geneList<-cluster52
ontologies<-c("biological_process","cellular_component")
dir<-tempdir()
odir<-HTGM4Ddriver(dir,geneList,ontologies,GOGOA3,enrichThresh=2,
 countThresh=5,pvalThresh=0.10,fdrThresh=0.10,nrand=100,mn=2,mx=2000)

## End(Not run)
```

`JaccardHeatMap4` *JaccardHeatMap4*

Description

use the Jaccard metric to construct HTGM2D heat map

Usage

```
JaccardHeatMap4(dir, mat)
```

Arguments

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

Details

computes heatmap twice, once with and once without axis labels. for the composite svg, the axis labels generate too much screen clutter. the version with axis labels is used only for quality assurance checking.

Value

returns the return value of heatmap.2, and also saves hyperlinked SVG heatmap files to a results directory

Examples

```
# takes too long to run in CRAN check
stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
buildSubdirs(dir)

jHeatMap<-JaccardHeatMap4(dir,x_jmat)
```

`mergeGoMinerHTGM2D` *mergeGoMinerHTGM2D*

Description

match the categories in the results of GoMiner and HTGM2D

Usage

```
mergeGoMinerHTGM2D(
  dir,
  htgm4d,
  l,
  ontologies,
  fdrThresh,
  doRow = FALSE,
  SVG3 = FALSE,
  browser = "Safari",
  verbose = 4
)
```

Arguments

`dir`	character string path name of output directory
`htgm4d`	return value of HTGM4D()
`l`	return value of GoMiner4()
`ontologies`	character vector of 2 ontologies e.g. c("biological_process","cellular_component")
`fdrThresh`	numeric background value to initialize matrix m
`doRow`	Boolean param passed to heatmap.2
`SVG3`	Boolean if TRUE invoke combine3SVG()
`browser`	character string c("Safari","Google Chrome","Firefox","Brave Browser")
`verbose`	param passed to vprint()

Value

returns no values but has side effect of saving some .png and .svg images

---

`parseSVGline` *parseSVGline*

Description

scale the width and height parameters of <svg line in svg object

Usage

```
parseSVGline(svg)
```

Arguments

`svg`	an svg object

Value

returns modified svg object

Examples

```
s<-parseSVGline(x_svg1)
```

`pngBorder` *pngBorder*

Description

manual adjustment of border in composite png

Usage

```
pngBorder(dir, border = "500", bcol = "white")
```

Arguments

`dir`	character string path name for main output directory
`border`	character string containing border width
`bcol`	character string containing border color

Value

returns path name of ofile

`pngDims` *pngDims*

Description

carefully compute the parameters for heatmap.2()

Usage

```
pngDims(
  mat,
  margins = c(50, 50),
  cex = 4,
  inchPerDendro = 0.25,
  inchPerChar = 0.05,
  verbose = 4
)
```

Arguments

`mat`	numerical matrix input for heatmap.2()
`margins`	param passed to heatmap.2()
`cex`	param passed to heatmap.2()
`inchPerDendro`	numerical value how many inches to allocate to dendrogram
`inchPerChar`	empirically determined value for character size, that results in square heatmap elements
`verbose`	integer param passed to vprint()

Details

the goal here is to compute the parameters that result in square, rather than rectangular, heatmap elements

Value

returns a list of parameter values to be used by heatmap.2()

Examples

```
m<-matrix(1:28,nrow=4,ncol=7)
pngDims(m)
```

`pngDisplay3` *pngDisplay3*

Description

use magick package to display arrangement of aligned 3 pngs

Usage

```
pngDisplay3(HTGM2D.png, side.png, below.png, border = "500", bcol = "white")
```

Arguments

`HTGM2D.png`	character string path name for HTGM2D .png
`side.png`	character string path name for side .png
`below.png`	character string path name for below .png
`border`	character string border size like "500"
`bcol`	character string border color like "white"

Value

returns the path name of the output file .png

Examples

```
# takes too long to run in CRAN check
HTGM2D.png <- system.file("extdata/files", "HTGM2D.png", package = "HTGM4D")
side.png <- system.file("extdata/files", "side.png", package = "HTGM4D")
below.png <- system.file("extdata/files", "below.png", package = "HTGM4D")
png<-pngDisplay3(HTGM2D.png,side.png,below.png)
```

`pngHeat` *pngHeat*

Description

generate .png heatmap using parameters carefully computed by pngDims()

Usage

```
pngHeat(
  filename,
  mat,
  labs = FALSE,
  Rowv = TRUE,
  Colv = TRUE,
  res = 300,
  border = FALSE,
  verbose = 4
)
```

Arguments

`filename`	character string path name for .png output file
`mat`	numerical matrix input for heatmap.2()
`labs`	Boolean if TRUE display row and col labels
`Rowv`	parameter for heatmap.2()
`Colv`	parameter for heatmap.2()
`res`	numeric screen resolution
`border`	Boolean if TRUE add red border to periphery of png
`verbose`	integer param passed to vprint()

Value

returns the path name for the output heatmap .png file

Examples

```
# takes too long to run in CRAN check
stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
buildSubdirs(dir)

m<-matrix(1:28,nrow=4,ncol=7)
pngHeat(filename=sprintf("%s/%s/%s",dir,"standards","xpng.png"),mat=m)
```

`renderSVG` *renderSVG*

Description

cluster a GoMiner (but not HTGM2D) heatmap and save it as an SVG file

Usage

```
renderSVG(dir, heatmap, identifier, doRow = TRUE, verbose)
```

Arguments

`dir`	character string containing path name of output directory
`heatmap`	matrix to be clustered
`identifier`	character string "side" or "below"
`doRow`	Boolean if TRUE cluster row
`verbose`	input param for vprint()

Value

returns a list containing the SVG file and the SVG also has side effect of saving the svg file

Examples

```
# takes too long to run in CRAN check
stamp<-gsub(":","_",format(Sys.time(), "%a_%b_%d_%Y_%X"))
dir<-sprintf("%s/%s/",tempdir(),stamp)
dir.create(dir)
print(c("dir: ",dir))
buildSubdirs(dir)

m<-matrix(data=1:16,nrow=4,ncol=4)
rownames(m)<-c("a","b","c","d")
colnames(m)<-c("a","b","c","d")
```

```
svg<-renderSVG(sprintf("%s/%s/",dir,"standards"),m,"example",verbose=-1)
```

`setSVGtransparency` *setSVGtransparency*

Description

modify svg code so that background opacity is set to 0

Usage

```
setSVGtransparency(svg)
```

Arguments

`svg`	svg object

Details

this prevents the background of one svg from overlaying the content of a second svg in combine3SVG()

Value

returns modified version of svg object

Examples

```
svgt<-setSVGtransparency(x_svg1)
```

`whoRanMe` *whoRanMe*

Description

wrapper for commandArgs()

Usage

```
whoRanMe()
```

Value

Returns a character string containing the identity of who ran the program in which this is called.

Examples

```
whoRanMe()
```

`x_jmat` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_jmat)
```

`x_l` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_l)
```

`x_lines` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_lines)
```

`x_svg1` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_svg1)
```

`x_svg2` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_svg2)
```

`x_svg3` *HTGM4D data set*

Description

HTGM4D data set

Usage

```
data(x_svg3)
```

Index