Package ‘minimalistGODB’

May 4, 2025

Version 1.1.0

Date 2025-05-04

Title Build a Minimalist Gene Ontology (GO) Database (GODB)

Maintainer Barry Zeeberg <barryz2013@gmail.com>

Author Barry Zeeberg [aut, cre]

Depends R (>= 4.2.0)

LazyData true

LazyDataCompression xz

Description Normally building a GODB is fairly complicated, involving downloading multiple database files and using these to build e.g. a 'mySQL' database. Accessing this database is also complicated, involving an intimate knowledge of the database in order to construct reliable queries. Here we have a more modest goal, generating GOGOA3, which is a stripped down version of the GODB that was originally restricted to human genes as designated by the HUGO Gene Nomenclature Committee (HGNC) (see <https://geneontology.org/>). I have now added about two dozen additional species, namely all species represented on the Gene Ontology download page <https://current.geneontology.org/products/pages/downloads.html>. This covers most of the model organisms that are commonly used in bio-medical and basic research (assuming that anyone still has a grant to do such research). This can be built in a matter of seconds from 2 easily downloaded files (see <https://current.geneontology.org/products/pages/downloads.html> and <https://geneontology.org/docs/download-ontology/>), and it can be queried by e.g. w<-which(GOGOA3[,``HGNC"] %in% hgncList) where GOGOA3 is a matrix representing the minimalist GODB and hgncList is a list of gene identifiers. This database will be used in my upcoming package 'GoMiner' which is based on my previous publication (see Zeeberg, B.R., Feng, W., Wang, G. et al. (2003)<doi:10.1186/gb-2003-4-4-r28>). Relevant .RData files are available from GitHub (<https://github.com/barryzee/GO/tree/main/databases>).

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 2025-05-04 14:00:02 UTC

Contents

`buildGODatabase` *buildGODatabase*

Description

driver to build GO database

Usage

```
buildGODatabase(goa, gobasic, dir = NULL, verbose = FALSE)
```

Arguments

`goa`	character string path name to downloaded goa_human.gaf
`gobasic`	character string path name to downloaded go-basic.obo
`dir`	character string path name to directory to hold subdirectory GODB_RDATA
`verbose`	Boolean if TRUE print out some diagnostic info

Details

download goa_human.gaf from https://current.geneontology.org/products/pages/downloads.html download go-basic.obo from https://geneontology.org/docs/download-ontology/ parameter dir should be omitted or NULL except for the developer harvesting the updated .RData DBs

The output GOGOA was saved as an .RData file. This was too large for CRAN. It is available from https://github.com/barryzee/GO/tree/main/databases

Value

returns no value but has side effect of saving GOGOA3 to a subdirectory

Examples

```
## Not run:
# replace my path names for goa and gobasic with your own!!
# these were obtained from the download sites listed in 'details' section
goa<-"~/goa_human.gaf"
gobasic<-"~/go-basic.obo"
buildGODatabase(goa,gobasic,dir="~/",verbose=TRUE)
# > dim(GOGOA)
# [1] 720139      5
# > GOGOA[1:5,]
#      HGNC          GO           RELATION      NAME                    ONTOLOGY
# [1,] "NUDT4B"     "GO:0003723" "enables"    "RNA binding"          "molecular_function"
# [2,] "NUDT4B"     "GO:0005515" "enables"    "protein binding"      "molecular_function"
# [3,] "NUDT4B"     "GO:0046872" "enables"    "metal ion binding"    "molecular_function"
# [4,] "NUDT4B"     "GO:0005829" "located_in"  "cytosol"             "cellular_component"
# [5,] "TRBV20OR9-2" "GO:0002376" "involved_in" "immune system process" "biological_process"

## End(Not run)

# here is a small example that you can run
f1<-system.file("extdata","goa_human.small.gaf",package="minimalistGODB")
f2<-system.file("extdata","go-basic.small.obo",package="minimalistGODB")
buildGODatabase(f1,f2,verbose=TRUE)
```

`buildGODatabaseDriver` *buildGODatabaseDriver*

Description

driver to build multiple GO databases for many species

Usage

```
buildGODatabaseDriver(goaDir, gobasic, dir = NULL, verbose = FALSE)
```

Arguments

`goaDir`	character string path name to directory containing downloaded goa .gaf files
`gobasic`	character string path name to downloaded go-basic.obo
`dir`	character string path name to directory to hold species database subdirectories
`verbose`	Boolean if TRUE print out some diagnostic info

Details

download goa .gaf files from https://current.geneontology.org/products/pages/downloads.html download go-basic.obo from https://geneontology.org/docs/download-ontology/

The output GOGOA3 was saved as an .RData file. This was too large for CRAN. It is available from https://github.com/barryzee/GO/tree/main/databases

Value

returns GO database with columns c("HGNC","GO","RELATION","NAME","ONTOLOGY")

Examples

```
## Not run:
# replace my path names for goa and gobasic with your own!!
# these were obtained from the download sites listed in 'details' section
goaDir<-"/Users/barryzeeberg/Downloads/gaf/"
gobasic<-"~/go-basic.obo"
buildGODatabaseDriver(goaDir,gobasic,dir="~/personal",verbose=TRUE)

## End(Not run)

# here is a small example that you can run
goaDir<-system.file("extdata",package="minimalistGODB")
gobasic<-system.file("extdata","go-basic.small.obo",package="minimalistGODB")
dir<-tempdir()
buildGODatabaseDriver(goaDir,gobasic,dir,verbose=TRUE)
```

`GO` *minimalistGODB data set*

Description

minimalistGODB data set generated by parseGOBASIC()

Usage

```
data(GO)
```

`GOA` *minimalistGODB data set*

Description

minimalistGODB data set generated by parseGOA()

Usage

```
data(GOA)
```

`GOGOAsmall` *minimalistGODB data set*

Description

small version of minimalistGODB data set generated by buildGODatabase()

Usage

```
data(GOGOAsmall)
```

`grepList` *grepList*

Description

determine the correct pattern to grep for depending on the species

Usage

```
grepList(gaf)
```

Arguments

`gaf`	character string containing the basename of the gaf file downloaded from https://current.geneontology.org/

Value

returns the correct pattern to grep for

Examples

```
pattern<-grepList("tair.gaf")
```

`joinGO` *joinGO*

Description

join the outputs of parseGOA and parseGOBASIC to add the GO category name and the ontology to GOA

Usage

```
joinGO(GOA, GO)
```

Arguments

`GOA`	output of parseGOA()
`GO`	output of parseGOBASIC()

Value

returns a matrix with columns c("HGNC","GO","RELATION","NAME","ONTOLOGY")

Examples

```
GOGOA<-joinGO(GOA,GO)
# GOGOA[1:5,]
# HGNC          GO           RELATION      NAME                    ONTOLOGY
# [1,] "NUDT4B"     "GO:0003723" "enables"    "RNA binding"         "molecular_function"
# [2,] "NUDT4B"     "GO:0005515" "enables"    "protein binding"     "molecular_function"
# [3,] "NUDT4B"     "GO:0046872" "enables"    "metal ion binding"   "molecular_function"
# [4,] "NUDT4B"     "GO:0005829" "located_in" "cytosol"             "cellular_component"
# v[5,] "TRBV20OR9-2" "GO:0002376" "involved_in" "immune system process" "biological_process"
# GO_NAME
# [1,] "GO_0003723__RNA_binding"
# [2,] "GO_0005515__protein_binding"
# [3,] "GO_0046872__metal_ion_binding"
# [4,] "GO_0005829__cytosol"
# [5,] "GO_0002376__immune_system_process"

# querying GOGOA to compute gene enrichment of some GO categories
hgncList<-GOGOA[1:1000,"HGNC"]
ontology<-"biological_process"
w<-which(GOGOA[,"ONTOLOGY"] == ontology)
GOGOA<-GOGOA[w,]
w<-which(GOGOA[,"HGNC"] %in% hgncList)
t<-sort(table(GOGOA[w,"NAME"]),decreasing=TRUE)[1:10]
```

`parseGOA` *parseGOA*

Description

parse goa_human.gaf

Usage

```
parseGOA(goa)
```

Arguments

`goa`	character string path name to downloaded goa_human.gaf

Details

download goa_human.gaf from https://current.geneontology.org/products/pages/downloads.html

Value

returns matrix with columns c("HGNC","GO","RELATION")

Examples

```
## Not run:
# replace my path name for goa with your own!!
# this was obtained from the download sites listed in 'details' section
GOA<-parseGOA("~/goa_human.gaf")
# GOA[1:5,]
#      HGNC          GO           RELATION
# [1,] "NUDT4B"      "GO:0003723" "enables"
# [2,] "NUDT4B"      "GO:0005515" "enables"
# [3,] "NUDT4B"      "GO:0046872" "enables"
# [4,] "NUDT4B"      "GO:0005829" "located_in"
# [5,] "TRBV20OR9-2" "GO:0002376" "involved_in"

## End(Not run)
# here is a small example that you can run
f<-system.file("extdata","goa_human.small.gaf",package="minimalistGODB")
GOAsmall<-parseGOA(f)
```

`parseGOBASIC` *parseGOBASIC*

Description

parse go-basic.obo

Usage

```
parseGOBASIC(gobasic, verbose = FALSE)
```

Arguments

`gobasic`	character string path name to downloaded go-basic.obo
`verbose`	Boolean if TRUE print out some diagnostic info

Details

download go-basic.obo from https://geneontology.org/docs/download-ontology/

Value

returns a list whose components are c("m", "bp", "mf", "cc")

Examples

```
## Not run:
# replace my path name for gobasic with your own!!
# this was obtained from the download sites listed in 'details' section
GO<-parseGOBASIC("~/go-basic.obo",verbose=FALSE)
# GO$bp[1:5,]
#            GO           NAME                                 ONTOLOGY
# GO:0000001 "GO:0000001" "mitochondrion inheritance"        "biological_process"
# GO:0000002 "GO:0000002" "mitochondrial genome maintenance" "biological_process"
# GO:0000011 "GO:0000011" "vacuole inheritance"              "biological_process"
# GO:0000012 "GO:0000012" "single strand break repair"       "biological_process"
# GO:0000017 "GO:0000017" "alpha-glucoside transport"        "biological_process"

## End(Not run)

# here is a small example that you can run
f<-system.file("extdata","go-basic.small.obo",package="minimalistGODB")
GOsmall<-parseGOBASIC(f)
```

`restrictGOA` *restrictGOA*

Description

restrict GO categories in GOA to those in GO

Usage

```
restrictGOA(GOA, GO)
```

Arguments

`GOA`	output of parseGOA()
`GO`	output of parseGOBASIC()

Value

returns a restricted version of GOA

Examples

```
GOA<-restrictGOA(GOA,GO)
```

`subsetGOGOA` *subsetGOGOA*

Description

split GOGOA into 3 separate ontologies

Usage

```
subsetGOGOA(GOGOA)
```

Arguments

`GOGOA`	return value of minimalistGODB::joinGO()

Value

returns a list containing subsets of GOGOA for each ontology, unique gene and cat lists, and stats

Examples

```
#load("data/GOGOAsmall.RData")
GOGOA3small<-subsetGOGOA(GOGOAsmall)
```

Index