GO.db

November 11, 2026

GO.db *Bioconductor annotation data package*

Description

Welcome to the GO.db annotation Package. The purpose of this package is to provide detailed information about the latest version of the Gene Ontologies. This package is updated biannually.

Objects in this package are accessed using the `select()` interface. See ?select in the AnnotationDbi package for details.

See Also

- `AnnotationDb-class` for use of `keys()`, `columns()` and `select()`.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.
columns(GO.db)

## Bimap interface:
## The 'old style' of interacting with these objects is manipulation as
## bimaps. While this approach is still available we strongly encourage the
## use of select().
ls("package:GO.db")
```

GOBPANCESTOR *Annotation of GO Identifiers to their Biological Process Ancestors*

Description

This data set describes associations between GO Biological Process (BP) terms and their ancestor BP terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO BP terms to all ancestor terms, where an ancestor term is a more general GO term that precedes the given GO term in the DAG (in other words, the parents, and all their parents, etc.).

Details

Each GO BP term is mapped to a vector of ancestor GO BP terms.

Biological process is defined as the broad biological goals, such as mitosis or purine metabolism, that are accomplished by ordered assemblies of molecular functions as defined by Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the `select()` interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOBPANCESTOR)
# Remove GO IDs that do not have any ancestor
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the ancestor GO IDs for the first two elents of xx
  goids <- xx[1:2]
}
```

GOBPCHILDREN *Annotation of GO Identifiers to their Biological Process Children*

Description

This data set describes associations between GO molecular function (BP) terms and their direct children BP terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO BP terms to all direct children terms, where a direct child term is a more specific GO term that is immediately preceded by the given GO term in the DAG.

Details

Each GO BP term is mapped to a vector of children GO BP terms.

Biological process is defined as the broad biological goals, such as mitosis or purine metabolism, that are accomplished by ordered assemblies of molecular functions as defined by Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOBPCHILDREN)
# Remove GO IDs that do not have any children
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # Get the parent GO IDs for the first elents of xx
    goids <- xx[[1]]
    # Find out the GO terms for the first parent goid
    GOID(GOTERM[[goids[1]]])
    Term(GOTERM[[goids[1]]])
    Synonym(GOTERM[[goids[1]]])
    Secondary(GOTERM[[goids[1]]])
```

```
    Definition(GOTERM[[goids[1]]])
    Ontology(GOTERM[[goids[1]]])
}
```

GOBPOFFSPRING *Annotation of GO Identifiers to their Biological Process Offspring*

Description

This data set describes associations between GO molecular function (BP) terms and their offspring BP terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO BP terms to all offspring terms, where an offspring term is a more specific GO term that is preceded by the given GO term in the DAG (in other words, the children and all their children, etc.).

Details

Each GO BP term is mapped to a vector of offspring GO BP terms.

Biological process is defined as the broad biological goals, such as mitosis or purine metabolism, that are accomplished by ordered assemblies of molecular functions as defined by Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOBPOFFSPRING)
# Remove GO IDs that do not have any offspring
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the offspring GO IDs for the first two elents of xx
  goids <- xx[1:2]
}
```

GOBPPARENTS *Annotation of GO Identifiers to their Biological Process Parents*

Description

This data set describes associations between GO molecular function (BP) terms and their direct parent BP terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO BP terms to all direct parent terms, where a direct parent term is a more general GO term that immediately precedes the given GO term in the DAG.

Details

Each GO BP term is mapped to a named vector of GO BP terms. The name associated with the parent term will be either *isa*, *hasa* or *partof*, where *isa* indicates that the child term is a more specific version of the parent, and *hasa* and *partof* indicate that the child term is a part of the parent. For example, a telomere is part of a chromosome.

Biological process is defined as the broad biological goals, such as mitosis or purine metabolism, that are accomplished by ordered assemblies of molecular functions as defined by Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the `select()` interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOBPPARENTS)
# Remove GO IDs that do not have any parent
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # Get the children GO IDs for the first elents of xx
    goids <- xx[[1]]
    # Find out the GO terms for the first parent goid
    GOID(GOTERM[[goids[1]]])
    Term(GOTERM[[goids[1]]])
```

```
    Synonym(GOTERM[[goids[1]]])
    Secondary(GOTERM[[goids[1]]])
    Definition(GOTERM[[goids[1]]])
    Ontology(GOTERM[[goids[1]]])
}
```

GOCCANCESTOR *Annotation of GO Identifiers to their Cellular Component Ancestors*

Description

This data set describes associations between GO molecular function (CC) terms and their ancestor CC terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO CC terms to all ancestor terms, where an ancestor term is a more general GO term that precedes the given GO term in the DAG (in other words, the parents, and all their parents, etc.).

Details

Each GO CC term is mapped to a vector of ancestor GO C terms.

Cellular component is defined as the subcellular structures, locations, and macromolecular complexes; examples include nucleus, telomere, and origin recognition complex as defined by Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the `select()` interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOCCANCESTOR)
# Remove GO IDs that do not have any ancestor
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the ancestor GO IDs for the first two elents of xx
  goids <- xx[1:2]
}
```

GOCCCHILDREN *Annotation of GO Identifiers to their Cellular Component Children*

Description

This data set describes associations between GO molecular function (CC) terms and their direct children CC terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO CC terms to all direct children terms, where a direct child term is a more specific GO term that is immediately preceded by the given GO term in the DAG.

Details

Each GO CC term is mapped to a vector of children GO CC terms.

Cellular component is defined as the subcellular structures, locations, and macromolecular complexes; examples include nucleus, telomere, and origin recognition complex as defined by Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOCCCHILDREN)
# Remove GO IDs that do not have any children
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
goids <- xx[[1]]
# Find out the GO terms for the first parent goid
GOID(GOTERM[[goids[1]]])
Term(GOTERM[[goids[1]]])
Synonym(GOTERM[[goids[1]]])
Secondary(GOTERM[[goids[1]]])
Definition(GOTERM[[goids[1]]])
```

```
Ontology(GOTERM[[goids[1]]])
}
```

GOCCOFFSPRING *Annotation of GO Identifiers to their Cellular Component Offspring*

Description

This data set describes associations between GO molecular function (MF) terms and their offspring MF terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO MF terms to all offspring terms, where an offspring term is a more specific GO term that is preceded by the given GO term in the DAG (in other words, the children and all their children, etc.).

Details

Each GO CC term is mapped to a vector of offspring GO MF terms.

Cellular component is defined as the subcellular structures, locations, and macromolecular complexes; examples include nucleus, telomere, and origin recognition complex as defined b y Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the `select()` interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOCCOFFSPRING)
# Remove GO identifiers that do not have any offspring
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the offspring GO identifiers for the first two elents of xx
  goidentifiers <- xx[1:2]
}
```

GOCCPARENTS *Annotation of GO Identifiers to their Cellular Component Parents*

Description

This data set describes associations between GO molecular function (CC) terms and their direct parent CC terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO CC terms to all direct parent terms, where a direct parent term is a more general GO term that immediately precedes the given GO term in the DAG.

Details

Each GO CC term is mapped to a named vector of GO CC terms. The name associated with the parent term will be either *isa*, *hasa* or *partof*, where *isa* indicates that the child term is a more specific version of the parent, and *hasa* and *partof* indicate that the child term is a part of the parent. For example, a telomere is part of a chromosome.

Cellular component is defined as the subcellular structures, locations, and macromolecular complexes; examples include nucleus, telomere, and origin recognition complex as defined b y Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOCCPARENTS)
# Remove GO IDs that do not have any parent
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
   goids <- xx[[1]]
   # Find out the GO terms for the first parent go ID
   GOID(GOTERM[[goids[1]]])
   Term(GOTERM[[goids[1]]])
   Synonym(GOTERM[[goids[1]]])
```

```
  Secondary(GOTERM[[goids[1]]])
  Definition(GOTERM[[goids[1]]])
  Ontology(GOTERM[[goids[1]]])
}
```

GOMAPCOUNTS *Number of mapped keys for the maps in package GO.db*

Description

GOMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package GO.db.

Details

This "map count" information is precalculated and stored in the package annotation DB. This allows some quality control and is used by the checkMAPCOUNTS function defined in AnnotationDbi to compare and validate different methods (like `count.mappedkeys(x)` or `sum(!is.na(as.list(x)))`) for getting the "map count" of a given map.

See Also

- mappedkeys,
- count.mappedkeys,
- checkMAPCOUNTS
- AnnotationDb-class for use of the `select()` interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
GOMAPCOUNTS
mapnames <- names(GOMAPCOUNTS)
GOMAPCOUNTS[mapnames[1]]
x <- get(mapnames[1])
sum(!is.na(as.list(x)))
count.mappedkeys(x)   # much faster!

## Check the "map count" of all the maps in package GO.db
checkMAPCOUNTS("GO.db")
```

GOMFANCESTOR *Annotation of GO identifiers to their Molecular Function Ancestors*

Description

This data set describes associations between GO molecular function (MF) terms and their ancestor MF terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO MF terms to all ancestor terms, where an ancestor term is a more general GO term that precedes the given GO term in the DAG (in other words, the parents, and all their parents, etc.).

Details

Each GO MF term is mapped to a vector of ancestor GO MF terms.

Molecular function is defined as the tasks performed by individual gene products; examples are transcription factor and DNA helicase as defined by Gene Ontology Consortium.

Mappings were based on data provided: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOMFANCESTOR)
# Remove GO identifiers that do not have any ancestor
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the ancestor GO identifiers for the first two elents of xx
  goids <- xx[1:2]
}
```

GOMFCHILDREN *Annotation of GO Identifiers to their Molecular Function Children*

Description

This data set describes associations between GO molecular function (MF) terms and their direct children MF terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO MF terms to all direct children terms, where a direct child term is a more specific GO term that is immediately preceded by the given GO term in the DAG.

Details

Each GO MF term is mapped to a vector of children GO MF terms.

Molecular function is defined as the tasks performed by individual gene products; examples are transcription factor and DNA helicase as defined by Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOMFCHILDREN)
# Remove GO identifiers that do not have any children
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # Get the children GO identifiers for the first elents of xx
    goids <- xx[[1]]
    # Find out the GO terms for the first parent goid
    GOID(GOTERM[[goids[1]]])
    Term(GOTERM[[goids[1]]])
    Synonym(GOTERM[[goids[1]]])
    Secondary(GOTERM[[goids[1]]])
    Definition(GOTERM[[goids[1]]])
```

```
    Ontology(GOTERM[[goids[1]]])
}
```

GOMFOFFSPRING *Annotation of GO Identifiers to their Molecular Function Offspring*

Description

This data set describes associations between GO molecular function (MF) terms and their offspring MF terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO MF terms to all offspring terms, where an offspring term is a more specific GO term that is preceded by the given GO term in the DAG (in other words, the children and all their children, etc.).

Details

Each GO MF term is mapped to a vector of offspring GO MF terms.

Molecular function is defined as the tasks performed by individual gene products; examples are transcription factor and DNA helicase as defined by Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOMFOFFSPRING)
# Remove GO identifiers that do not have any offspring
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
  # Get the offspring GO identifiers for the first two elents of xx
goids <- xx[1:2]
}
```

GOMFPARENTS *Annotation of GO Identifiers to their Molecular Function Parents*

Description

This data set describes associations between GO molecular function (MF) terms and their direct parent MF terms, based on the directed acyclic graph (DAG) defined by the Gene Ontology Consortium. The format is an R object mapping the GO MF terms to all direct parent terms, where a direct parent term is a more general GO term that immediately precedes the given GO term in the DAG.

Details

Each GO MF term is mapped to a named vector of GO MF terms. The name associated with the parent term will be either *isa*, *hasa* or *partof*, where *isa* indicates that the child term is a more specific version of the parent, and *hasa* and *partof* indicate that the child term is a part of the parent. For example, a telomere is part of a chromosome.

Molecular function is defined as the tasks performed by individual gene products; examples are transcription factor and DNA helicase as defined by the Gene Ontology Consortium.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

http://www.geneontology.org/ and https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOMFPARENTS)
# Remove GO identifiers that do not have any parent
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # Get the parent GO identifiers for the first elents of xx
    goids <- xx[[1]]
    # Find out the GO terms for the first parent goid
    GOID(GOTERM[[goids[1]]])
    Term(GOTERM[[goids[1]]])
```

```
    Synonym(GOTERM[[goids[1]]])
    Secondary(GOTERM[[goids[1]]])
    Definition(GOTERM[[goids[1]]])
    Ontology(GOTERM[[goids[1]]])
}
```

GOOBSOLETE *Annotation of GO identifiers by terms defined by Gene Ontology Consortium and their status are obsolete*

Description

This is an R object mapping GO identifiers to the specific terms in defined by Gene Ontology Consortium and their definition are obsolete

Details

All the obsolete GO terms that are collected in this index will no longer exist in other mapping objects.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

https://www.ncib.nlm.nih.gov/LocusLink

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOTERM)
if(length(xx) > 0){
    # Get the TERMS for the first elent of xx
    GOID(xx[[1]])
    Ontology(xx[[1]])
}
```

GOSYNONYM *Map from GO synonyms to GO terms*

Description

GOSYNONYM is an R object that provides mapping from GO synonyms to GO terms

Details

[TODO: Put some details here]

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- GOSYNONYM
sample(x, 3)
# GO ID "GO:0009435" has a lot of synonyms
GOTERM[["GO:0009435"]]
# GO ID "GO:0006736" is a synonym of GO ID "GO:0009435"
GOID(GOSYNONYM[["GO:0006736"]])
```

GOTERM *Annotation of GO Identifiers to GO Terms*

Description

This data set gives mappings between GO identifiers and their respective terms.

Details

Each GO identifier is mapped to a GOTerms object that has 6 slots: GOID: GO Identifier Term: The term for that GO id Synonym: Synonymous terms Secondary: Secondary terms that have been merged into this term Definition: Further definition of the GO term Ontology: One of MF - molecular function, BP - biological process, or CC - cellular component

All the obsolete GO terms are under the nodes "obsolete molecular function" (GO:0008369), "obsolete cellular component" (GO id GO:0008370), and "obsolete biological process" (GO:0008371). Each of these GO identifiers has a group of GO identifiers as their direct children with GO terms

that were defined by GO but are deprecated in the current build. These deprecated GO terms were appended by "(obsolete)" when the data package was built.

Mappings were based on data provided by: Gene Ontology https://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2026-01-23

References

https://www.ncib.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(GOTERM)
if(length(xx) > 0){
    # Get the TERMS for the first elent of xx
    GOID(xx[[1]])
    Term(xx[[1]])
    Synonym(xx[[1]])
    Secondary(xx[[1]])
    Definition(xx[[1]])
    Ontology(xx[[1]])
}
```

GO_dbconn *Collect information about the package annotation DB*

Description

Some convenience functions for getting a connection object to (or collecting information about) the package annotation DB.

Usage

```
GO_dbconn()
GO_dbfile()
GO_dbschema(file="", show.indices=FALSE)
GO_dbInfo()
```

Arguments

file	A connection, or a character string naming the file to print to (see the file argument of the cat function for the details).
show.indices	The CREATE INDEX statements are not shown by default. Use show.indices=TRUE to get them.

Details

GO_dbconn returns a connection object to the package annotation DB. IMPORTANT: Don't call dbDisconnect on the connection object returned by GO_dbconn or you will break all the AnnDbObj objects defined in this package!

GO_dbfile returns the path (character string) to the package annotation DB (this is an SQLite file).

GO_dbschema prints the schema definition of the package annotation DB.

GO_dbInfo prints other information about the package annotation DB.

Value

GO_dbconn: a DBIConnection object representing an open connection to the package annotation DB.

GO_dbfile: a character string with the path to the package annotation DB.

GO_dbschema: none (invisible NULL).

GO_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

```
library(DBI)
## Count the number of rows in the "go_term" table:
dbGetQuery(GO_dbconn(), "SELECT COUNT(*) FROM go_term")

GO_dbschema()

GO_dbInfo()
```

Index

∗ **datasets**
GO.db, 1
GO_dbconn, 17
GOBPANCESTOR, 2
GOBPCHILDREN, 3
GOBPOFFSPRING, 4
GOBPPARENTS, 5
GOCCANCESTOR, 6
GOCCCHILDREN, 7
GOCCOFFSPRING, 8
GOCCPARENTS, 9
GOMAPCOUNTS, 10
GOMFANCESTOR, 11
GOMFCHILDREN, 12
GOMFOFFSPRING, 13
GOMFPARENTS, 14
GOOBSOLETE, 15
GOSYNONYM, 16
GOTERM, 16
∗ **utilities**
GO_dbconn, 17

AnnDbObj, *18*

cat, *18*
checkMAPCOUNTS, *10*
count.mappedkeys, *10*

dbconn, *18*
dbConnect, *18*
dbDisconnect, *18*
dbfile, *18*
dbGetQuery, *18*
dbInfo, *18*
dbschema, *18*

GO *(*GO.db*)*, 1
GO.db, 1
GO_dbconn, 17
GO_dbfile *(*GO_dbconn*)*, 17
GO_dbInfo *(*GO_dbconn*)*, 17
GO_dbschema *(*GO_dbconn*)*, 17
GOBPANCESTOR, 2
GOBPCHILDREN, 3
GOBPOFFSPRING, 4
GOBPPARENTS, 5
GOCCANCESTOR, 6
GOCCCHILDREN, 7
GOCCOFFSPRING, 8
GOCCPARENTS, 9
GOMAPCOUNTS, 10
GOMFANCESTOR, 11
GOMFCHILDREN, 12
GOMFOFFSPRING, 13
GOMFPARENTS, 14
GOOBSOLETE, 15
GOSYNONYM, 16
GOTERM, 16

mappedkeys, *10*