org.Gg.eg.db

November 11, 2026

`org.Gg.egACCNUM` *Map Entrez Gene identifiers to GenBank Accession Numbers*

Description

org.Gg.egACCNUM is an R object that contains mappings between Entrez Gene identifiers and GenBank accession numbers.

Details

This object is a simple mapping of Entrez Gene identifiers https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene to all possible GenBank accession numbers.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

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 <- org.Gg.egACCNUM
# Get the entrez gene identifiers that are mapped to an ACCNUM
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the ACCNUM for the first five genes
  xx[1:5]
  # Get the first one
```

```
  xx[[1]]
}
#For the reverse map ACCNUM2EG:
# Convert to a list
xx <- as.list(org.Gg.egACCNUM2EG)
if(length(xx) > 0){
   # Gets the entrez gene identifiers for the first five Entrez Gene IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

`org.Gg.egALIAS2EG` *Map between Common Gene Symbol Identifiers and Entrez Gene*

Description

org.Gg.egALIAS is an R object that provides mappings between common gene symbol identifiers and entrez gene identifiers.

Details

Each gene symbol maps to a named vector containing the corresponding entrez gene identifier. The name of the vector corresponds to the gene symbol. Since gene symbols are sometimes redundantly assigned in the literature, users are cautioned that this map may produce multiple matching results for a single gene symbol. Users should map back from the entrez gene IDs produced to determine which result is the one they want when this happens.

Because of this problem with redundant assigment of gene symbols, is it never advisable to use gene symbols as primary identifiers.

This mapping includes ALL gene symbols including those which are already listed in the SYMBOL map. The SYMBOL map is meant to only list official gene symbols, while the ALIAS maps are meant to store all used symbols.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

References

https://www.ncbi.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(org.Gg.egALIAS2EG)
# Remove pathway identifiers that do not map to any entrez gene id
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # The entrez gene identifiers for the first two elements of XX
    xx[1:2]
    # Get the first one
    xx[[1]]
}
```

`org.Gg.eg.db` *Bioconductor annotation data package*

Description

Welcome to the org.Gg.eg.db annotation Package. This is an organism specific package. The purpose is to provide detailed information about the species abbreviated in the second part of the package name org.Gg.eg.db. 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(org.Gg.eg.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:org.Gg.eg.db")
```

org.Gg.egCHR *Map Entrez Gene IDs to Chromosomes*

Description

org.Gg.egCHR is an R object that provides mappings between entrez gene identifiers and the chromosome that contains the gene of interest.

Details

Each entrez gene identifier maps to a vector of a chromosome.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

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 <- org.Gg.egCHR
# Get the entrez gene identifiers that are mapped to a chromosome
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the CHR for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

org.Gg.egCHRLENGTHS *A named vector for the length of each of the chromosomes*

Description

org.Gg.egCHRLENGTHS provides the length measured in base pairs for each of the chromosomes.

Details

This is a named vector with chromosome numbers as the names and the corresponding lengths for chromosomes as the values.

Total lengths of chromosomes were derived by calculating the number of base pairs on the sequence string for each chromosome.

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:
tt <- org.Gg.egCHRLENGTHS
# Length of chromosome 1
tt["1"]
```

---

org.Gg.egCHRLOC *Entrez Gene IDs to Chromosomal Location*

Description

org.Gg.egCHRLOC is an R object that maps entrez gene identifiers to the starting position of the gene. The position of a gene is measured as the number of base pairs.

The CHRLOCEND mapping is the same as the CHRLOC mapping except that it specifies the ending base of a gene instead of the start.

Details

Each entrez gene identifier maps to a named vector of chromosomal locations, where the name indicates the chromosome.

Chromosomal locations on both the sense and antisense strands are measured as the number of base pairs from the p (5' end of the sense strand) to q (3' end of the sense strand) arms. Chromosomal locations on the antisense strand have a leading "-" sign (e. g. -1234567).

Since some genes have multiple start sites, this field can map to multiple locations.

Mappings were based on data provided by: UCSC Genome Bioinformatics (Gallus gallus) ftp://hgdownload.cse.ucsc.edu/gol
With a date stamp from the source of: UTC-Sep26

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 <- org.Gg.egCHRLOC
# Get the entrez gene identifiers that are mapped to chromosome locations
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the CHRLOC for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

`org.Gg.egENSEMBL` *Map Ensembl gene accession numbers with Entrez Gene identifiers*

Description

org.Gg.egENSEMBL is an R object that contains mappings between Entrez Gene identifiers and Ensembl gene accession numbers.

Details

This object is a simple mapping of Entrez Gene identifiers https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene to Ensembl gene Accession Numbers.

Mappings were based on data provided by BOTH of these sources: http://www.ensembl.org/biomart/martview/ ftp://ftp.ncbi.nlm.nih.gov/gene/DATA

For most species, this mapping is a combination of NCBI to ensembl IDs from BOTH NCBI and ensembl. Users who wish to only use mappings from NCBI are encouraged to see the ncbi2ensembl table in the appropriate organism package. Users who wish to only use mappings from ensembl are encouraged to see the ensembl2ncbi table which is also found in the appropriate organism packages. These mappings are based upon the ensembl table which is contains data from BOTH of these sources in an effort to maximize the chances that you will find a match.

For worms and flies however, this mapping is based only on sources from ensembl, as these organisms do not have ensembl to entrez gene mapping data at NCBI.

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 <- org.Gg.egENSEMBL
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the Ensembl gene IDs for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
#For the reverse map ENSEMBL2EG:
# Convert to a list
xx <- as.list(org.Gg.egENSEMBL2EG)
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

`org.Gg.egENSEMBLPROT` *Map Ensembl protein acession numbers with Entrez Gene identifiers*

Description

org.Gg.egENSEMBL is an R object that contains mappings between Entrez Gene identifiers and Ensembl protein accession numbers.

Details

This object is a simple mapping of Entrez Gene identifiers https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene to Ensembl protein accession numbers.

Mappings were based on data provided by: ftp://ftp.ensembl.org/pub/current_fasta ftp://ftp.ncbi.nlm.nih.gov/gene/DATA

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 <- org.Gg.egENSEMBLPROT
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the Ensembl gene IDs for the first five proteins
  xx[1:5]
  # Get the first one
  xx[[1]]
}
#For the reverse map ENSEMBLPROT2EG:
# Convert to a list
xx <- as.list(org.Gg.egENSEMBLPROT2EG)
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

---

org.Gg.egENSEMBLTRANS *Map Ensembl transcript acession numbers with Entrez Gene identifiers*

Description

org.Gg.egENSEMBL is an R object that contains mappings between Entrez Gene identifiers and Ensembl transcript accession numbers.

Details

This object is a simple mapping of Entrez Gene identifiers https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene to Ensembl transcript accession numbers.

Mappings were based on data provided by: ftp://ftp.ensembl.org/pub/current_fasta ftp://ftp.ncbi.nlm.nih.gov/gene/DATA

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 <- org.Gg.egENSEMBLTRANS
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the Ensembl gene IDs for the first five proteins
  xx[1:5]
  # Get the first one
  xx[[1]]
}
#For the reverse map ENSEMBLTRANS2EG:
# Convert to a list
xx <- as.list(org.Gg.egENSEMBLTRANS2EG)
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

---

org.Gg.egENZYME *Map between Entrez Gene IDs and Enzyme Commission (EC) Numbers*

Description

org.Gg.egENZYME is an R object that provides mappings between entrez gene identifiers and EC numbers.

Details

Each entrez gene identifier maps to a named vector containing the EC number that corresponds to the enzyme produced by that gene. The name corresponds to the entrez gene identifier. If this information is unknown, the vector will contain an NA.

Enzyme Commission numbers are assigned by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology http://www.chem.qmw.ac.uk/iubmb/enzyme/ to allow enzymes to be identified.

An Enzyme Commission number is of the format EC x.y.z.w, where x, y, z, and w are numeric numbers. In org.Gg.egENZYME2EG, EC is dropped from the Enzyme Commission numbers.

Enzyme Commission numbers have corresponding names that describe the functions of enzymes in such a way that EC x is a more general description than EC x.y that in turn is a more general description than EC x.y.z. The top level EC numbers and names are listed below:

EC 1 oxidoreductases

EC 2 transferases

EC 3 hydrolases

EC 4 lyases

EC 5 isomerases

EC 6 ligases

The EC name for a given EC number can be viewed at http://www.chem.qmul.ac.uk/iupac/jcbn/index.html#6

Mappings between entrez gene identifiers and enzyme identifiers were obtained using files provided by: KEGG GENOME ftp://ftp.genome.jp/pub/kegg/genomes With a date stamp from the source of: 2011-Mar15

For the reverse map, each EC number maps to a named vector containing the entrez gene identifier that corresponds to the gene that produces that enzyme. The name of the vector corresponds to the EC number.

References

ftp://ftp.genome.ad.jp/pub/kegg/pathways

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 <- org.Gg.egENZYME
# Get the entrez gene identifiers that are mapped to an EC number
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the ENZYME for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
# For the reverse map:
# Convert to a list
xx <- as.list(org.Gg.egENZYME2EG)
if(length(xx) > 0){
```

```
  # Gets the entrez gene identifiers for the first five enzyme
  #commission numbers
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

org.Gg.egGENENAME *Map between Entrez Gene IDs and Genes*

Description

org.Gg.egGENENAME is an R object that maps entrez gene identifiers to the corresponding gene name.

Details

Each entrez gene identifier maps to a named vector containing the gene name. The vector name corresponds to the entrez gene identifier. If the gene name is unknown, the vector will contain an NA.

Gene names currently include both the official (validated by a nomenclature committee) and preferred names (interim selected for display) for genes. Efforts are being made to differentiate the two by adding a name to the vector.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

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 <- org.Gg.egGENENAME
# Get the gene names that are mapped to an entrez gene identifier
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the GENE NAME for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

org.Gg.egGENETYPE *Map between Entrez Gene Identifiers and Gene Type*

Description

org.Gg.egGENETYPE is an R object that provides mappings between entrez gene identifiers and gene types.

Details

Each entrez gene identifier is mapped to the reported type of the corresponding gene. An NA is reported if there is no known type for a given gene.

References

https://www.ncbi.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:
x <- org.Gg.egGENETYPE
# Get the gene symbol that are mapped to an entrez gene identifiers
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the SYMBOL for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
# Get the entrez gene identifiers that are mapped to a gene type
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the entrez gene ID for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

org.Gg.egGO *Maps between Entrez Gene IDs and Gene Ontology (GO) IDs*

Description

org.Gg.egGO is an R object that provides mappings between entrez gene identifiers and the GO identifiers that they are directly associated with. This mapping and its reverse mapping do NOT associate the child terms from the GO ontology with the gene. Only the directly evidenced terms are represented here.

org.Gg.egGO2ALLEGS is an R object that provides mappings between a given GO identifier and all of the Entrez Gene identifiers annotated at that GO term OR TO ONE OF IT'S CHILD NODES in the GO ontology. Thus, this mapping is much larger and more inclusive than org.Gg.egGO2EG.

Details

If org.Gg.egGO is cast as a list, each Entrez Gene identifier is mapped to a list of lists. The names on the outer list are GO identifiers. Each inner list consists of three named elements: GOID, Ontology, and Evidence.

The GOID element matches the GO identifier named in the outer list and is included for convenience when processing the data using 'lapply'.

The Ontology element indicates which of the three Gene Ontology categories this identifier belongs to. The categories are biological process (BP), cellular component (CC), and molecular function (MF).

The Evidence element contains a code indicating what kind of evidence supports the association of the GO identifier to the Entrez Gene id. Some of the evidence codes in use include:

IMP: inferred from mutant phenotype

IGI: inferred from genetic interaction

IPI: inferred from physical interaction

ISS: inferred from sequence similarity

IDA: inferred from direct assay

IEP: inferred from expression pattern

IEA: inferred from electronic annotation

TAS: traceable author statement

NAS: non-traceable author statement

ND: no biological data available

IC: inferred by curator

A more complete listing of evidence codes can be found at:

http://www.geneontology.org/GO.evidence.shtml

If org.Gg.egGO2ALLEGS or org.Gg.egGO2EG is cast as a list, each GO term maps to a named vector of entrez gene identifiers and evidence codes. A GO identifier may be mapped to the same entrez gene identifier more than once but the evidence code can be different. Mappings between

Gene Ontology identifiers and Gene Ontology terms and other information are available in a separate data package named GO.

Whenever any of these mappings are cast as a data.frame, all the results will be output in an appropriate tabular form.

Mappings between entrez gene identifiers and GO information were obtained through their mappings to Entrez Gene identifiers. NAs are assigned to entrez gene identifiers that can not be mapped to any Gene Ontology information. Mappings between Gene Ontology identifiers an Gene Ontology terms and other information are available in a separate data package named GO.

All mappings were based on data provided by: Gene Ontology http://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2025-07-22

For GO2ALL style mappings, the intention is to return all the genes that are the same kind of term as the parent term (based on the idea that they are more specific descriptions of the general term). However because of this intent, not all relationship types will be counted as offspring for this mapping. Only "is a" and "has a" style relationships indicate that the genes from the child terms would be the same kind of thing.

References

ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/

See Also

- org.Gg.egGO2ALLEGS
- 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 <- org.Gg.egGO
# Get the entrez gene identifiers that are mapped to a GO ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
    # Try the first one
    got <- xx[[1]]
    got[[1]][["GOID"]]
    got[[1]][["Ontology"]]
    got[[1]][["Evidence"]]
}
# For the reverse map:
# Convert to a list
xx <- as.list(org.Gg.egGO2EG)
if(length(xx) > 0){
    # Gets the entrez gene ids for the top 2nd and 3nd GO identifiers
```

```
    goids <- xx[2:3]
    # Gets the entrez gene ids for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}
# For org.Gg.egGO2ALLEGS
xx <- as.list(org.Gg.egGO2ALLEGS)
if(length(xx) > 0){
    # Gets the Entrez Gene identifiers for the top 2nd and 3nd GO identifiers
    goids <- xx[2:3]
    # Gets all the Entrez Gene identifiers for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}
```

`org.Gg.egMAPCOUNTS` *Number of mapped keys for the maps in package org.Gg.eg.db*

Description

org.Gg.egMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package org.Gg.eg.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:
org.Gg.egMAPCOUNTS
mapnames <- names(org.Gg.egMAPCOUNTS)
```

```
org.Gg.egMAPCOUNTS[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 org.Gg.eg.db
checkMAPCOUNTS("org.Gg.eg.db")
```

`org.Gg.egORGANISM` *The Organism for org.Gg.eg*

Description

org.Gg.egORGANISM is an R object that contains a single item: a character string that names the organism for which org.Gg.eg was built.

Details

Although the package name is suggestive of the organism for which it was built, org.Gg.egORGANISM provides a simple way to programmatically extract the organism name.

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:
org.Gg.egORGANISM
```

`org.Gg.egPATH` *Mappings between Entrez Gene identifiers and KEGG pathway identifiers*

Description

KEGG (Kyoto Encyclopedia of Genes and Genomes) maintains pathway data for various organisms. org.Gg.egPATH maps entrez gene identifiers to the identifiers used by KEGG for pathways

Details

Each KEGG pathway has a name and identifier. Pathway name for a given pathway identifier can be obtained using the KEGG data package that can either be built using AnnBuilder or downloaded from Bioconductor http://www.bioconductor.org.

Graphic presentations of pathways are searchable at url http://www.genome.ad.jp/kegg/pathway.html by using pathway identifiers as keys.

Mappings were based on data provided by: KEGG GENOME ftp://ftp.genome.jp/pub/kegg/genomes With a date stamp from the source of: 2011-Mar15

References

http://www.genome.ad.jp/kegg/

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 <- org.Gg.egPATH
# Get the entrez gene identifiers that are mapped to a KEGG pathway ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the PATH for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}

# For the reverse map:
# Convert the object to a list
xx <- as.list(org.Gg.egPATH2EG)
# Remove pathway identifiers that do not map to any entrez gene id
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # The entrez gene identifiers for the first two elements of XX
    xx[1:2]
    # Get the first one
    xx[[1]]
}
```

org.Gg.egPFAM *Maps between Manufacturer Identifiers and PFAM Identifiers*

Description

org.Gg.egPFAM is an R object that provides mappings between manufacturer identifiers and PFAM identifiers.

Details

The bimap interface for PFAM is defunct. Please use select() interface to PFAM identifiers. See ?AnnotationDbi::select for details.

org.Gg.egPMID *Map between Entrez Gene Identifiers and PubMed Identifiers*

Description

org.Gg.egPMID is an R object that provides mappings between entrez gene identifiers and PubMed identifiers.

Details

Each entrez gene identifier is mapped to a named vector of PubMed identifiers. The name associated with each vector corresponds to the entrez gene identifier. The length of the vector may be one or greater, depending on how many PubMed identifiers a given entrez gene identifier is mapped to. An NA is reported for any entrez gene identifier that cannot be mapped to a PubMed identifier.

Titles, abstracts, and possibly full texts of articles can be obtained from PubMed by providing a valid PubMed identifier. The pubmed function of annotate can also be used for the same purpose.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

References

https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed

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 <- org.Gg.egPMID
# Get the entrez gene identifiers that are mapped to any PubMed ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0){
    # The entrez gene identifiers for the first two elements of XX
    xx[1:2]
    # Get the first one
    xx[[1]]
    if(interactive() && !is.null(xx[[1]]) && !is.na(xx[[1]])
       && require(annotate)){
        # Gets article information as XML files
        xmls <- pubmed(xx[[1]], disp = "data")
        # Views article information using a browser
        pubmed(xx[[1]], disp = "browser")
    }
}
# For the reverse map:
# Convert the object to a list
xx <- as.list(org.Gg.egPMID2EG)
if(length(xx) > 0){
    # The entrez gene identifiers for the first two elements of XX
    xx[1:2]
    # Get the first one
    xx[[1]]
    if(interactive() && require(annotate)){
        # Gets article information as XML files for a PubMed id
        xmls <- pubmed(names(xx)[1], disp = "data")
        # Views article information using a browser
        pubmed(names(xx)[1], disp = "browser")
    }
}
```

`org.Gg.egPROSITE` *Maps between Manufacturer Identifiers and PROSITE Identifiers*

Description

org.Gg.egPROSITE is an R object that provides mappings between manufacturer identifiers and PROSITE identifiers.

Details

The bimap interface for PROSITE is defunct. Please use select() interface to PROSITE identifiers. See ?AnnotationDbi::select for details.

`org.Gg.egREFSEQ` *Map between Entrez Gene Identifiers and RefSeq Identifiers*

Description

org.Gg.egREFSEQ is an R object that provides mappings between entrez gene identifiers and RefSeq identifiers.

Details

Each entrez gene identifier is mapped to a named vector of RefSeq identifiers. The name represents the entrez gene identifier and the vector contains all RefSeq identifiers that can be mapped to that entrez gene identifier. The length of the vector may be one or greater, depending on how many RefSeq identifiers a given entrez gene identifier can be mapped to. An `NA` is reported for any entrex gene identifier that cannot be mapped to a RefSeq identifier at this time.

RefSeq identifiers differ in format according to the type of record the identifiers are for as shown below:

NG_XXXXX: RefSeq accessions for genomic region (nucleotide) records

NM_XXXXX: RefSeq accessions for mRNA records

NC_XXXXX: RefSeq accessions for chromosome records

NP_XXXXX: RefSeq accessions for protein records

XR_XXXXX: RefSeq accessions for model RNAs that are not associated with protein products

XM_XXXXX: RefSeq accessions for model mRNA records

XP_XXXXX: RefSeq accessions for model protein records

Where XXXXX is a sequence of integers.

NCBI https://www.ncbi.nlm.nih.gov/RefSeq/ allows users to query the RefSeq database using RefSeq identifiers.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

References

https://www.ncbi.nlm.nih.gov https://www.ncbi.nlm.nih.gov/RefSeq/

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 <- org.Gg.egREFSEQ
# Get the entrez gene identifiers that are mapped to any RefSeq ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the REFSEQ for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}

# For the reverse map:
x <- org.Gg.egREFSEQ2EG
# Get the RefSeq identifier that are mapped to an entrez gene ID
mapped_seqs <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_seqs])
if(length(xx) > 0) {
  # Get the entrez gene for the first five Refseqs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

`org.Gg.egSYMBOL` *Map between Entrez Gene Identifiers and Gene Symbols*

Description

org.Gg.egSYMBOL is an R object that provides mappings between entrez gene identifiers and gene abbreviations.

Details

Each entrez gene identifier is mapped to the a common abbreviation for the corresponding gene. An `NA` is reported if there is no known abbreviation for a given gene.

Symbols typically consist of 3 letters that define either a single gene (ABC) or multiple genes (ABC1, ABC2, ABC3). Gene symbols can be used as key words to query public databases such as Entrez Gene.

Mappings were based on data provided by: Entrez Gene ftp://ftp.ncbi.nlm.nih.gov/gene/DATA With a date stamp from the source of: 2025-Sep24

References

https://www.ncbi.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:
x <- org.Gg.egSYMBOL
# Get the gene symbol that are mapped to an entrez gene identifiers
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the SYMBOL for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
# For the reverse map:
x <- org.Gg.egSYMBOL2EG
# Get the entrez gene identifiers that are mapped to a gene symbol
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the entrez gene ID for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

org.Gg.egUNIPROT *Map Uniprot accession numbers with Entrez Gene identifiers*

Description

org.Gg.egUNIPROT is an R object that contains mappings between Entrez Gene identifiers and Uniprot accession numbers.

Details

This object is a simple mapping of Entrez Gene identifiers https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene to Uniprot Accession Numbers.

Mappings were based on data provided by NCBI (link above) with an exception for fly, which required retrieving the data from ensembl http://www.ensembl.org/biomart/martview/

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 <- org.Gg.egUNIPROT
# Get the entrez gene IDs that are mapped to a Uniprot ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
  # Get the Uniprot gene IDs for the first five genes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

org.Gg.eg_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

```
org.Gg.eg_dbconn()
org.Gg.eg_dbfile()
org.Gg.eg_dbschema(file="", show.indices=FALSE)
org.Gg.eg_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

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

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

org.Gg.eg_dbschema prints the schema definition of the package annotation DB.

org.Gg.eg_dbInfo prints other information about the package annotation DB.

Value

org.Gg.eg_dbconn: a DBIConnection object representing an open connection to the package annotation DB.

org.Gg.eg_dbfile: a character string with the path to the package annotation DB.

org.Gg.eg_dbschema: none (invisible NULL).

org.Gg.eg_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

```
library(DBI)
## Count the number of rows in the "genes" table:
dbGetQuery(org.Gg.eg_dbconn(), "SELECT COUNT(*) FROM genes")

org.Gg.eg_dbschema()

org.Gg.eg_dbInfo()
```

Index

∗ **datasets**
org.Gg.eg.db, 3
org.Gg.eg_dbconn, 23
org.Gg.egACCNUM, 1
org.Gg.egALIAS2EG, 2
org.Gg.egCHR, 4
org.Gg.egCHRLENGTHS, 4
org.Gg.egCHRLOC, 5
org.Gg.egENSEMBL, 6
org.Gg.egENSEMBLPROT, 7
org.Gg.egENSEMBLTRANS, 8
org.Gg.egENZYME, 9
org.Gg.egGENENAME, 11
org.Gg.egGENETYPE, 12
org.Gg.egGO, 13
org.Gg.egMAPCOUNTS, 15
org.Gg.egORGANISM, 16
org.Gg.egPATH, 16
org.Gg.egPFAM, 18
org.Gg.egPMID, 18
org.Gg.egPROSITE, 19
org.Gg.egREFSEQ, 20
org.Gg.egSYMBOL, 21
org.Gg.egUNIPROT, 22
∗ **utilities**
org.Gg.eg_dbconn, 23

AnnDbObj, *24*

cat, *24*
checkMAPCOUNTS, *15*
count.mappedkeys, *15*

dbconn, *24*
dbConnect, *24*
dbDisconnect, *24*
dbfile, *24*
dbGetQuery, *24*
dbInfo, *24*
dbschema, *24*

mappedkeys, *15*

org.Gg.eg *(*org.Gg.eg.db*)*, 3
org.Gg.eg.db, 3
org.Gg.eg_dbconn, 23
org.Gg.eg_dbfile *(*org.Gg.eg_dbconn*)*, 23
org.Gg.eg_dbInfo *(*org.Gg.eg_dbconn*)*, 23
org.Gg.eg_dbschema *(*org.Gg.eg_dbconn*)*, 23
org.Gg.egACCNUM, 1
org.Gg.egACCNUM2EG *(*org.Gg.egACCNUM*)*, 1
org.Gg.egALIAS2EG, 2
org.Gg.egCHR, 4
org.Gg.egCHRLENGTHS, 4
org.Gg.egCHRLOC, 5
org.Gg.egCHRLOCEND *(*org.Gg.egCHRLOC*)*, 5
org.Gg.egENSEMBL, 6
org.Gg.egENSEMBL2EG *(*org.Gg.egENSEMBL*)*, 6
org.Gg.egENSEMBLPROT, 7
org.Gg.egENSEMBLPROT2EG *(*org.Gg.egENSEMBLPROT*)*, 7
org.Gg.egENSEMBLTRANS, 8
org.Gg.egENSEMBLTRANS2EG *(*org.Gg.egENSEMBLTRANS*)*, 8
org.Gg.egENZYME, 9
org.Gg.egENZYME2EG *(*org.Gg.egENZYME*)*, 9
org.Gg.egGENENAME, 11
org.Gg.egGENETYPE, 12
org.Gg.egGO, 13
org.Gg.egGO2ALLEGS, *14*
org.Gg.egGO2ALLEGS *(*org.Gg.egGO*)*, 13
org.Gg.egGO2EG *(*org.Gg.egGO*)*, 13
org.Gg.egMAPCOUNTS, 15
org.Gg.egORGANISM, 16
org.Gg.egPATH, 16
org.Gg.egPATH2EG *(*org.Gg.egPATH*)*, 16
org.Gg.egPFAM, 18
org.Gg.egPMID, 18
org.Gg.egPMID2EG *(*org.Gg.egPMID*)*, 18

org.Gg.egPROSITE, 19
org.Gg.egREFSEQ, 20
org.Gg.egREFSEQ2EG *(*org.Gg.egREFSEQ*)*, 20
org.Gg.egSYMBOL, 21
org.Gg.egSYMBOL2EG *(*org.Gg.egSYMBOL*)*, 21
org.Gg.egUNIPROT, 22