## org.Dm.eg.db

February 26, 2025

org.Dm.egACCNUM

Map Entrez Gene identifiers to GenBank Accession Numbers

## **Description**

org.Dm.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 <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> 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: 2024-Sep20

## See Also

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

```
## select() interface:
## Objects in this package can be accessed using the select() interface
\mbox{\tt \#\#} from the AnnotationDbi package. See ?select for details.
## Bimap interface:
x <- org.Dm.egACCNUM
# Get the entrez gene identifiers that are mapped to an ACCNUM
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
if(length(xx) > 0) {
  \mbox{\tt\#} 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.Dm.egACCNUM2EG)</pre>
```

```
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.Dm.egALIAS2EG

Map between Common Gene Symbol Identifiers and Entrez Gene

## **Description**

org.Dm.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 assignment 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: 2024-Sep20

## References

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

## See Also

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

```
## 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.Dm.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]
```

org.Dm.eg.db

```
# Get the first one
xx[[1]]
}
```

org.Dm.eg.db

Bioconductor annotation data package

## **Description**

Welcome to the org.Dm.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.Dm.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.Dm.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.Dm.eg.db")
```

org.Dm.egCHR

Map Entrez Gene IDs to Chromosomes

## Description

org.Dm.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: 2024-Sep20

## 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.Dm.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.Dm.egCHRLENGTHS

A named vector for the length of each of the chromosomes

## **Description**

org.Dm.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.

```
## 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.Dm.egCHRLENGTHS
# Length of chromosome 1
tt["1"]</pre>
```

org.Dm.egCHRLOC 5

org.Dm.egCHRLOC

Entrez Gene IDs to Chromosomal Location

## **Description**

org.Dm.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 (Drosophila melanogaster) ftp://hgdownload.cse.ucsc.edu/goldenPath/dm6/database With a date stamp from the source of: 2024-Sep22

## See Also

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

```
## 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.Dm.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.Dm.egENSEMBL

Map Ensembl gene accession numbers with Entrez Gene identifiers

## **Description**

org.Dm.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 <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> 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.

```
## 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.Dm.egENSEMBL
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egENSEMBL2EG)</pre>
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   xxΓ1:57
   # Get the first one
```

```
xx[[1]]
}
```

org.Dm.egENSEMBLPROT Map Ensembl protein acession numbers with Entrez Gene identifiers

## **Description**

org.Dm.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 <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> 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.

```
## 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.Dm.egENSEMBLPROT
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egENSEMBLPROT2EG)</pre>
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   # Get the first one
   xx[[1]]
}
```

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

## **Description**

org.Dm.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 <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> 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.

```
## 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.Dm.egENSEMBLTRANS
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egENSEMBLTRANS2EG)</pre>
if(length(xx) > 0){
   \mbox{\tt\#} Gets the entrez gene IDs for the first five Ensembl IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

org.Dm.egENZYME 9

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

## Description

org.Dm.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 <a href="http://www.chem.qmw.ac.uk/iubmb/enzyme/">http://www.chem.qmw.ac.uk/iubmb/enzyme/</a> 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.Dm.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.

org.Dm.egFLYBASE

#### **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.Dm.egENZYME
# Get the entrez gene identifiers that are mapped to an EC number
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egENZYME2EG)</pre>
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.Dm.egFLYBASE

Map FlyBase acession numbers with Entrez Gene identifiers

## **Description**

org.Dm.egFLYBASE is an R object that contains mappings between Entrez Gene identifiers and FlyBase accession numbers.

## **Details**

This object is a simple mapping of Entrez Gene identifiers <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> to Flybase 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: 2024-Sep20

## See Also

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

```
## 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.Dm.egFLYBASE
# Get the entrez gene identifiers that are mapped to a Flybase ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
if(length(xx) > 0) {
  # Get the FlyBase IDs for the first five genes
  xxΓ1:57
  # Get the first one
  xx[[1]]
#For the reverse map FLYBASE2EG:
# Convert to a list
xx <- as.list(org.Dm.egFLYBASE2EG)</pre>
if(length(xx) > 0){
   # Gets the entrez gene identifiers for the first five FlyBase IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

org.Dm.egFLYBASECG

Map FlyBase CG Acession numbers with Entrez Gene identifiers

## **Description**

org.Dm.egFLYBASE is an R object that contains mappings between Entrez Gene identifiers and FlyBase CG accession numbers. These accessions are used by ensembl and supported by flybase.

## **Details**

This object is a simple mapping of Entrez Gene identifiers <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> to Flybase CG 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: 2024-Sep20

#### See Also

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

```
## 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.Dm.egFLYBASECG
# Get the entrez gene IDs that are mapped to a Flybase CG ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
```

```
# Get the FlyBase CG IDs for the first five genes
xx[1:5]
# Get the first one
xx[[1]]
}
#For the reverse map FLYBASECG2EG:
# Convert to a list
xx <- as.list(org.Dm.egFLYBASECG2EG)
if(length(xx) > 0){
    # Gets the entrez gene IDs for the first five FlyBase CG IDs
    xx[1:5]
    # Get the first one
    xx[[1]]
}
```

org.Dm.egFLYBASEPROT Map FlyBase Protein Acession numbers with Entrez Gene identifiers

## **Description**

org.Dm.egFLYBASE is an R object that contains mappings between Entrez Gene identifiers and FlyBase protein accession numbers.

#### **Details**

This object is a simple mapping of Entrez Gene identifiers <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> to Flybase protein 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: 2024-Sep20

## See Also

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

```
## 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.Dm.egFLYBASEPROT
# Get the entrez gene IDs that are mapped to a Flybase prot ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
if(length(xx) > 0) {
  # Get the FlyBase protein IDs for the first five genes
  xxΓ1:57
  # Get the first one
  xx[[1]]
}
#For the reverse map FLYBASEPROT2EG:
# Convert to a list
```

```
xx <- as.list(org.Dm.egFLYBASEPROT2EG)
if(length(xx) > 0){
    # Gets the entrez gene IDs for the first five FlyBase protein IDs
    xx[1:5]
    # Get the first one
    xx[[1]]
}
```

org.Dm.egGENENAME

Map between Entrez Gene IDs and Genes

## **Description**

org.Dm.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: 2024-Sep20

## See Also

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

```
## 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.Dm.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.Dm.egGENETYPE

Map between Entrez Gene Identifiers and Gene Type

## **Description**

org.Dm.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.

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

org.Dm.egGO

org.Dm.egGO

Maps between Entrez Gene IDs and Gene Ontology (GO) IDs

## **Description**

org.Dm.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.Dm.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.Dm.egGO2EG.

#### **Details**

If org.Dm.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.Dm.egGO2ALLEGS or org.Dm.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.

org.Dm.egGO

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:

With a date stamp from the source of:

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.Dm.egGO2ALLEGS
- AnnotationDb-class for use of the select() interface.

```
## 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.Dm.egGO
# Get the entrez gene identifiers that are mapped to a GO ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egG02EG)</pre>
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.Dm.egGO2ALLEGS
xx <- as.list(org.Dm.egGO2ALLEGS)</pre>
if(length(xx) > 0){
    # Gets the Entrez Gene identifiers for the top 2nd and 3nd GO identifiers
```

org.Dm.egMAP

```
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]])
}</pre>
```

org.Dm.egMAP

Map between Entrez Gene Identifiers and cytogenetic maps/bands

## Description

org.Dm.egMAP is an R object that provides mappings between entrez gene identifiers and cytoband locations.

#### Details

Each entrez gene identifier is mapped to a vector of cytoband locations. The vector length may be one or longer, if there are multiple reported chromosomal locations for a given gene. An NA is reported for any entrez gene identifiers that cannot be mapped to a cytoband at this time.

Cytogenetic bands for most higher organisms are labeled p1, p2, p3, q1, q2, q3 (p and q are the p and q arms), etc., counting from the centromere out toward the telomeres. At higher resolutions, sub-bands can be seen within the bands. The sub-bands are also numbered from the centromere out toward the telomere. Thus, a label of 7q31.2 indicates that the band is on chromosome 7, q arm, band 3, sub-band 1, and sub-sub-band 2.

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: 2024-Sep20

#### References

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

## See Also

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

```
## 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.Dm.egMAP
# Get the entrez gene identifiers that are mapped to any cytoband
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
    # Get the ids for the first five genes
    xx[1:5]
    # Get the first one
    xx[1]]
```

```
#For the reverse map org.Dm.egMAP2EG
x <- org.Dm.egMAP2EG
# Get the cytobands that are mapped to any entrez gene id
mapped_bands <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_bands])
if(length(xx) > 0) {
    # Get the bands for the first five genes
    xx[1:5]
    # Get the first one
    xx[[1]]
}
```

org.Dm.egMAPCOUNTS

Number of mapped keys for the maps in package org.Dm.eg.db

## **Description**

org.Dm.egMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package org.Dm.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.

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

## Bimap interface:
org.Dm.egMAPCOUNTS
mapnames <- names(org.Dm.egMAPCOUNTS)
org.Dm.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.Dm.eg.db
checkMAPCOUNTS("org.Dm.eg.db")</pre>
```

g
---

## **Description**

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

## **Details**

Although the package name is suggestive of the organism for which it was built, org.Dm.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.Dm.egORGANISM

org.Dm.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.Dm.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 <a href="http://www.bioconductor.org">http://www.bioconductor.org</a>.

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/
```

20 org.Dm.egPMID

#### 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.Dm.egPATH
# Get the entrez gene identifiers that are mapped to a KEGG pathway ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egPATH2EG)</pre>
# Remove pathway identifiers that do not map to any entrez gene id
xx <- xx[!is.na(xx)]</pre>
if(length(xx) > 0){
    \# The entrez gene identifiers for the first two elements of XX
    # Get the first one
    xx[[1]]
}
```

org.Dm.egPMID

Map between Entrez Gene Identifiers and PubMed Identifiers

## **Description**

org.Dm.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: 2024-Sep20

org.Dm.egREFSEQ 21

#### 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.Dm.egPMID
\mbox{\tt\#} Get the entrez gene identifiers that are mapped to any PubMed ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
if(length(xx) > 0){
    # The entrez gene identifiers for the first two elements of XX
    # 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 \leftarrow 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.Dm.egPMID2EG)</pre>
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")</pre>
        # Views article information using a browser
        pubmed(names(xx)[1], disp = "browser")
    }
}
```

org.Dm.egREFSEQ

Map between Entrez Gene Identifiers and RefSeq Identifiers

## **Description**

org.Dm.egREFSEQ is an R object that provides mappings between entrez gene identifiers and Ref-Seq identifiers.

22 org.Dm.egREFSEQ

#### **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: 2024-Sep20

## References

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

## See Also

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

```
## select() interface:
## Objects in this package can be accessed using the select() interface
\ensuremath{\mbox{\#\#}} from the AnnotationDbi package. See ?select for details.
## Bimap interface:
x <- org.Dm.egREFSEQ
# Get the entrez gene identifiers that are mapped to any RefSeq ID
mapped_genes <- mappedkeys(x)</pre>
# Convert to a list
xx <- as.list(x[mapped_genes])</pre>
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.Dm.egREFSEQ2EG
# Get the RefSeq identifier that are mapped to an entrez gene ID
```

org.Dm.egSYMBOL 23

```
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.Dm.egSYMBOL

Map between Entrez Gene Identifiers and Gene Symbols

## **Description**

org.Dm.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: 2024-Sep20

#### References

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

## See Also

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

```
## 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.Dm.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]]
```

org.Dm.egUNIPROT

```
# For the reverse map:
x <- org.Dm.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.Dm.egUNIPROT

Map Uniprot accession numbers with Entrez Gene identifiers

## **Description**

org.Dm.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 <a href="https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene">https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene</a> 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 <a href="http://www.ensembl.org/biomart/martview/">http://www.ensembl.org/biomart/martview/</a>

## See Also

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

```
## 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.Dm.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.Dm.eg\_dbconn 25

org Dm eg	dbconn	Collect
org.Dm.eg_dbconn		Coneci

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.Dm.eg_dbconn()
org.Dm.eg_dbfile()
org.Dm.eg_dbschema(file="", show.indices=FALSE)
org.Dm.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.Dm.eg\_dbconn returns a connection object to the package annotation DB. IMPORTANT: Don't call dbDisconnect on the connection object returned by org.Dm.eg\_dbconn or you will break all the AnnDbObj objects defined in this package!

org.Dm.eg\_dbfile returns the path (character string) to the package annotation DB (this is an SQLite file).

org.Dm.eg\_dbschema prints the schema definition of the package annotation DB.

org.Dm.eg\_dbInfo prints other information about the package annotation DB.

## Value

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

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

```
org.Dm.eg\_dbschema: none (invisible NULL).
```

```
org.Dm.eg_dbInfo: none (invisible NULL).
```

## See Also

```
dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo
```

26 org.Dm.eg\_dbconn

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

# Index

org. Dm. eg. db, 3 org. Dm. eg. db, 3 org. Dm. eg. dbconn, 25 org. Dm. eg.dcCNUM, 1 org. Dm. eg.dcCRUM, 1 org.	* datasets	org.Dm.eg(org.Dm.eg.db), 3
org. Dm. eg_dbconn, 25 org. Dm. egACKONM, 1 org. Dm. egACKASZEG, 2 org. Dm. egCHR, 3 org. Dm. egCHRENGTHS, 4 org. Dm. egCHSEMBL, 6 org. Dm. egENSEMBL, 6 o	org.Dm.eg.db,3	
org. Dm. egACCNUM, 1 org. Dm. egChR, 3 org. Dm. egChR, 3 org. Dm. egChRLENGTHS, 4 org. Dm. egChRLENGTHS, 4 org. Dm. egChRLENGTHS, 4 org. Dm. egChRLENGTHS, 4 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 7 org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBL PROT, 7 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASE, 10 org. Dm. egGENEXME, 13 org. Dm. egGANISM, 19 org. Dm. egMAPCOUNTS, 18 org. Dm. egPATH, 19 org. Dm. egPMID, 20 org. Dm. egFNEMBL, 23 org. Dm. egWNIPROT, 24 * utilities org. Dm. egLATAS2EG (org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL TRANS, 8 org. Dm. egENSEMBLTRANS, 8 org. Dm.		
org. Dm. egALIAS2EG, 2 org. Dm. egCHR, 3 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLOC, 5 org. Dm. egCHRLOC, 5 org. Dm. egCHRLOC, 5 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 7 org. Dm. egENSEMBL, 8 org. Dm. egENSEMBL, 8 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egGLNENGTH, 13 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGACCNUM, 1 org. Dm. egCHRLENGTHS, 4 org. Dm. egGCHRLOC, 5 org. Dm. egCHRLENGTHS, 4 org. Dm. egGLYBASECR, 11 org. Dm. egGLYBASECR, 12 org. Dm. egGACCNUM, 1 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLOC, 5 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 10 org. Dm. egENSE		
org. Dm. egCHR, 3 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGT, 5 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCMRLENGTHS, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBL PROT, 7 org. Dm. egENZYME, 9 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGON, 15 org. Dm. egGON, 15 org. Dm. egGNANISM, 19 org. Dm. egCMRLENGTHS, 8 org. Dm. egCMRLENGTHS, 8 org. Dm. egCMRLENGTHS, 4 org. Dm. egCMRLOC, 5 org. Dm. egENESMBL, 6 org. Dm. egCMRLOC, 5 org. Dm. egENSEMBL, 6 org. Dm. egCMRLOC, 5 org. Dm. egENESMBL, 6 org. Dm. egCMRLENGTHS, 4 org. Dm. egCMRLENGTHS, 4 org. Dm. egCON, 15 org. Dm. egCMRLENGTHS, 4 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLOC, 5 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLENGTHS, 6 org. Dm. egCHRLOCED org. Dm. egCHNENGTHS, 7 org. Dm. egCHRLENGTHS, 6 org. Dm.		
org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLECC, 5 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBLPROT, 7 org. Dm. egENSEMBLTRANS, 8 org. Dm. egENSEMBLTRANS, 8 org. Dm. egENSEMBLTRANS, 8 org. Dm. egFLYBASEC, 10 org. Dm. egFLYBASEC, 11 org. Dm. egFLYBASEC, 11 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egMAPCOUNTS, 18 org. Dm. egMAPCOUNTS, 18 org. Dm. egPATHD, 20 org. Dm. egPATHD, 20 org. Dm. egRSYMBOL, 23 org. Dm. egCNPT, 24 * utilities org. Dm. eg JENSEMBL, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconn, 25 dbconnect, 25 dbconnect, 25 dbconnect, 25 dbconnect, 25 dbconnect, 25 dbconnect, 25 dbcohema, 25  dbschema, 25  org. Dm. egGONE, Dm. egGON, 15 org. Dm. egGONE, Dm. egGON, 15 org. Dm. egCNAPC, 10 org. Dm. egCNAPC, 11 org. Dm. egCNAPC, 10 org. Dm. egCNAPC		
org. Dm. egENSEMBL, 6 org. Dm. egENSEMBLPROT, 7 org. Dm. egENSEMBLTRANS, 8 org. Dm. egENSEMBLTRANS, 8 org. Dm. egENSYME, 9 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASEROT, 12 org. Dm. egGENENAME, 13 org. Dm. egGENETYPE, 14 org. Dm. egGENETYPE, 14 org. Dm. egGALTAS, 8 org. Dm. egGENETYPE, 14 org. Dm. egGENETYPE, 14 org. Dm. egGALTAS, 8 org. Dm. egCHRLENGTHS, 4 org. Dm. egCHRLOC, 5 org. Dm. egCHRLOC, 5 org. Dm. egCHRLOC, 5 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 6 org. Dm. egENSEMBL, 7 org. Dm. egENSEMBL, 7 org. Dm. egENSEMBL, 7 org. Dm. egENSEMBLPROT, 7 org. Dm. egENSEM	org.Dm.egCHRLENGTHS,4	
org. Dm. egENSEMBL PROT, 7 org. Dm. egENSEMBL TRANS, 8 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGNBETYPE, 14 org. Dm. egMAP, 17 org. Dm. egMAPCOUNTS, 18 org. Dm. egBATH, 19 org. Dm. eg		org.Dm.egACCNUM, 1
org. Dm. egENSEMBLTRANS, 8 org. Dm. egENZYME, 9 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egENSEMBL, 12 org. Dm. egENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGENENAME, 13 org. Dm. egGENETYPE, 14 org. Dm. egGO, 15 org. Dm. egMAP, 17 org. Dm. egMAPCOUNTS, 18 org. Dm. egPATH, 19 org. Dm. egPATH, 19 org. Dm. egBATH, 19 org. Dm. egSYMBOL, 23 org. Dm. egSYMBOL, 23 org. Dm. egSYMBOL, 24 * utilities org. Dm. eg_Abconn, 25  AnnDbObj, 25  cat, 25 checkMaPCOUNTS, 18 count. mappedkeys, 18  dbconn, 25 dbconnect, 25 dbfile, 25 dbGetQuery, 25 dbSchema, 25  org. Dm. egGNAP, 17 org. Dm. egGNAP, 17 org. Dm. egGNEREMBLTRANS org. Dm. egFLYBASEPGO, 11 org. Dm. egFLYBASEPROT, 12 org. Dm. egGNENAME, 13 org. Dm. egGNENAME, 13 org. Dm. egGNENAME, 13 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASEPROT, 12 org. Dm. egFLYBASEPROT, 12 org. Dm. egGNENAME, 13 org. Dm. egGNE		org.Dm.egACCNUM2EG(org.Dm.egACCNUM), 1
org. Dm. egENZYME, 9 org. Dm. egFLYBASE, 10 org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASEROT, 12 org. Dm. egGENENME, 13 org. Dm. egGENENME, 13 org. Dm. egGENETYPE, 14 org. Dm. egGO, 15 org. Dm. egMAP, 17 org. Dm. egMAP, 17 org. Dm. egMAP, 18 org. Dm. egPATH, 19 org. Dm. egPATH, 19 org. Dm. egBYBBC, 21 org. Dm. egBYBBC, 21 org. Dm. egBYBBC, 23 org. Dm. egBYBBC, 24 **utilities* org. Dm. eg Lymprot, 24 **utilities* org. Dm. eg Lymprot, 25 cat, 25 checkMAPCOUNTS, 18 count. mappedkeys, 18  dbconn, 25 dbconnect, 25 dbfile, 25 dbfile, 25 dbschema, 25  dbschema, 25  org. Dm. egGMAP, 17 org. Dm. egGO2ALLEGS, 16 org. Dm. egGMAP, 17 org. Dm. egGO2D, 15 org. Dm. egGO2D, 15 org. Dm. egGO2ALLEGS (org. Dm. egGO), 15 org. Dm. egGMAP, 17 org. Dm. egGO2D, 15 org. Dm. egGCNENTYPE, 14 org. Dm. egGCNETYPE, 14 org. Dm. egGCNETYPE, 14 org. Dm. egGCNETYPE, 14 org. Dm. egGO2ALLEGS (org. Dm. egGO), 15 org. Dm. egGO2ALLEGS (org. Dm. egGO), 15 org. Dm. egGO2ALLEGS (org. Dm. egGO), 15 org. Dm. egGO2D,		org.Dm.egALIAS2EG, 2
org.Dm.egFLYBASE, 10 org.Dm.egFLYBASEG, 11 org.Dm.egFLYBASEGG, 11 org.Dm.egGENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGENETYPE, 14 org.Dm.egGANISM, 17 org.Dm.egBAPCOUNTS, 18 org.Dm.egPAPCOUNTS, 18 org.Dm.egPAPCOUNTS, 18 org.Dm.egPABI, 19 org.Dm.egPABID, 20 org.Dm.egREFSEQ, 21 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_Dbeck, 25 cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18 dbconn, 25 dbconnect, 25 dbfile, 25 dbconnect, 25 dbclnfo, 25 dbschema, 25 dbschema, 25 dbschema, 25 dbcschema, 25 dbcschema, 25 drag.Dm.egMAPCID, 12 org.Dm.egMAPCID, 12 org.Dm.egCARLOCC, 5 org.Dm.egCHRLOC, 5 org.Dm.egCHRLOCLOC, 5 org.Dm.egCHRLOCLOC, 5 org.Dm.egCHRLOCLOC, 5 org.Dm.egCHRLOCLOCLOCLOCLOCLOCLOCLOCLOCLOCLOCLOCLOCL		org.Dm.egCHR, 3
org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASEPROT, 12 org. Dm. egGNENAME, 13 org. Dm. egGNENAME, 14 org. Dm. egGNENAME, 15 org. Dm. egGNENAME, 17 org. Dm. egMAPCOUNTS, 18 org. Dm. egMAPCOUNTS, 18 org. Dm. egPMID, 20 org. Dm. egPMID, 20 org. Dm. egFNSEMBL 23 org. Dm. egSYMBOL, 23 org. Dm. egSYMBOL, 24 * utilities org. Dm. eg_Abconn, 25  AnnDbObj, 25  AnnDbObj, 25  Cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  Org. Dm. egFLYBASECG, 11 org. Dm. egFLYBASEPROT, 12 org. Dm. egGNENEMBL, 13 org. Dm. egGNENEMBL, 13 org. Dm. egGO2ALLEGS, 16 org. Dm. egGO2ALLEGS, 16 org. Dm. egGO2ALLEGS, 16 org. Dm. egGO2ALLEGS, 16 org. Dm. egGO2ALLEGS (org. Dm. egGO), 15 org. Dm. egMAP, 17 org. Dm. egMAP, 17 org. Dm. egMAP, 17		org.Dm.egCHRLENGTHS,4
org.Dm.egFLYBASEPROT, 12 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egMap, 17 org.Dm.egMap, 17 org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egSYMBOL, 23 org.Dm.egLDrap, 25 and stillites org.Dm.eg_Abconn, 25  AnnDbObj, 25  cat, 25 checkMapCOUNTS, 18 count.mappedkeys, 18 dbconnect, 25 dbConnect, 25 dbGetQuery, 25 dbSchema, 25  org.Dm.egFLYBASEPROT, 12 org.Dm.egGOAPP, 17 org.Dm.egCSYMPD, 12 org.Dm.egCSYMPD, 25 org.Dm.egCSYMPD, 26 org.Dm.egCSYMPD, 27 org.Dm.egCSYMPOL, 28 org.Dm.egCSYMBOL, 29 org.Dm.egCSYMBOL, 29 org.Dm.egCSYMBOL, 20 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egCSYMBASEPROT, 12 org		org.Dm.egCHRLOC, 5
org.Dm.egFLYBASEPROT, 12 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGO, 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egOMAPCOUNTS, 18 org.Dm.egOMAPCOUNTS, 18 org.Dm.egPATH, 19 org.Dm.egPATH, 19 org.Dm.egSYMBOL, 23 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbConnect, 25 dbGetQuery, 25 dbSschema, 25  org.Dm.egFLYBASEPOO, 15 org.Dm.egGNERMBL, 6 org.Dm.egENSEMBLPROT, 7 org.Dm.egENSEMBLPROT, 7 org.Dm.egENSEMBLPROT, 7 org.Dm.egENSEMBLPROT), 7 org.Dm.egENSEMBLPROT), 7 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egFLYBASEQG (org.Dm.egENZYME), 9 org.Dm.egFLYBASEQG (org.Dm.egFLYBASE), 10 org.Dm.egFLYBASECG (org.Dm.egFLYBASE), 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17		org.Dm.egCHRLOCEND(org.Dm.egCHRLOC), 5
org.Dm.egGENETYPE, 14 org.Dm.egGO, 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAPCOUNTS, 18 org.Dm.egPMAPCOUNTS, 18 org.Dm.egPMAP, 19 org.Dm.egPMID, 20 org.Dm.egPMID, 20 org.Dm.egSYMBOL, 23 org.Dm.egSYMBOL, 23 org.Dm.egLNSYME, 9 org.Dm.egLNSYME, 9 org.Dm.egLNSYME, 9 org.Dm.egLNSYME, 9 org.Dm.egLNSYME, 9 org.Dm.egLNSYME, 9 org.Dm.egLYME, 9 org.Dm.egLYME, 9 org.Dm.egFLYBASE, 10 org.Dm.egGLYBASE, 10 org.Dm.egClYBASE, 10 org.D		org.Dm.egENSEMBL, 6
org.Dm.egGENETYPE, 14 org.Dm.egGO, 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAPCOUNTS, 18 org.Dm.egMAPCOUNTS, 18 org.Dm.egPASEMBLPROT, 7 org.Dm.egENSEMBLPROT, 7 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGENETYPE, 14 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAP, 17		org.Dm.egENSEMBL2EG(org.Dm.egENSEMBL),
org.Dm.egMAP, 17 org.Dm.egCNSEMBLPROT2EG org.Dm.egCNGANISM, 19 org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egPSYMBOL, 23 org.Dm.egLNSEMBLPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18 dbConnect, 25 dbGotnect, 25 dbGotnect, 25 dbGotnect, 25 dbSchema, 25  org.Dm.egMAPCOUNTS, 18 org.Dm.egCNSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE2G (org.Dm.egFLYBASE), 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egGNENEMME, 13 org.Dm.egGENENAME, 13 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAP, 17		
org.Dm.egMAP, 17 org.Dm.egMAPCOUNTS, 18 org.Dm.egORGANISM, 19 org.Dm.egORGANISM, 19 org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egSYMBOL, 23 org.Dm.egSYMBOL, 23 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconn, 25 dbConnect, 25 dbGetQuery, 25 dbSchema, 25  dbschema, 25  org.Dm.egMAPCOUNTS, 18 org.Dm.egENSEMBLTRANS, 8 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASECG, 11 org.Dm.egGOPLEG, 10 org.Dm.egCOPLEG,		org.Dm.egENSEMBLPROT, 7
org.Dm.egAAPCOUNTS, 18 org.Dm.egORGANISM, 19 org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egEFSEQ, 21 org.Dm.egSYMBOL, 23 org.Dm.egSVMBOL, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  count.mappedkeys, 18  count.mappedkeys, 18  count.mappedkeys, 18  count.mappedkeys, 25  dbconnect, 25  dcconnect,		
org.Dm.egORGANISM, 19 org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egREFSEQ, 21 org.Dm.egSYMBOL, 23 org.Dm.egSVMBOL, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbconnect, 25 dbfile, 25 dbfile, 25 dbschema, 25  org.Dm.egONE, 21 org.Dm.egGNZYME, 9 org.Dm.egENZYME, 9 org.Dm.egENZYME, 9 org.Dm.egENZYME2EG (org.Dm.egENZYME), 9 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE2EG (org.Dm.egFLYBASE), 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECGEG (org.Dm.egFLYBASECG), 11 org.Dm.egFLYBASECGPC (org.Dm.egFLYBASECG), 11 org.Dm.egFLYBASECGD, 11 org.Dm.egFLYBASECGD, 11 org.Dm.egFLYBASECGD, 11 org.Dm.egFLYBASECGD, 11 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17		
org.Dm.egPATH, 19 org.Dm.egPMID, 20 org.Dm.egENSEMBLTRANS2EG (org.Dm.egENSEMBLTRANS), 8 org.Dm.egSYMBOL, 23 org.Dm.egSYMBOL, 23 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbconnect, 25 dbfile, 25 dbGetQuery, 25 dbschema, 25  org.Dm.egFATH, 19 org.Dm.egENSEMBLTRANS2EG (org.Dm.egENZYME, 9 org.Dm.egENZYME), 9 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE, 10 org.Dm.egFLYBASE2EG (org.Dm.egFLYBASE), 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egGNespENAME, 13 org.Dm.egGNespENAME, 13 org.Dm.egGO, 15 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP2EG (org.Dm.egMAP), 17		, -
org.Dm.egPMID, 20 org.Dm.egRFSEQ, 21 org.Dm.egSYMBOL, 23 org.Dm.egSYMBOL, 24  * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbConnect, 25 dbDisconnect, 25 dbGetQuery, 25 dbSchema, 25  org.Dm.egRNIPROT, 20 (org.Dm.egENZYME), 9 org.Dm.egENZYME), 9 org.Dm.egFLYBASECG, 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG), 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egGNENAME, 13 org.Dm.egGNENAME, 13 org.Dm.egGNENAME, 13 org.Dm.egGO, 15 org.Dm.egGO, 15 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGDEG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17		
org.Dm.egRFSEQ, 21 org.Dm.egSYMBOL, 23 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbDisconnect, 25 dbfile, 25 dbGetQuery, 25 dbschema, 25  org.Dm.egRENZYME, 9 org.Dm.egENZYME2EG (org.Dm.egENZYME), 9 org.Dm.egFLYBASE2EG (org.Dm.egFLYBASE), 10  org.Dm.egFLYBASE2EG (org.Dm.egFLYBASE), 10  org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT2EG (org.Dm.egFLYBASEPROT), 12  org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17		
org.Dm.egSYMBOL, 23 org.Dm.egUNIPROT, 24 * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbDisconnect, 25 dbfile, 25 dbGetQuery, 25 dbschema, 25  org.Dm.egSYMBOL, 23 org.Dm.egENZYME2EG (org.Dm.egENZYME), 9 org.Dm.egFLYBASE2EG (org.Dm.egFLYBASE), 10  org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT2EG (org.Dm.egFLYBASEPROT), 12 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGO, 15 org.Dm.egGO, 15 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17		
org.Dm.egUNIPROT, 24  * utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbConnect, 25 dbDisconnect, 25 dbGetQuery, 25 dbSchema, 25  dbschema, 25 dbschema, 25  org.Dm.egFLYBASE, 10 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG2EG (org.Dm.egFLYBASEROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egGNENAME, 13 org.Dm.egGNENENAME, 13 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17 org.Dm.egMAP, 17	org.Dm.egSYMBOL, 23	
* utilities org.Dm.eg_dbconn, 25  AnnDbObj, 25  annDbObj, 25  cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconn, 25  dbconnect, 25 dbfile, 25 dbGetQuery, 25 dbschema, 25  wrg.Dm.egFLYBASECG, 11 org.Dm.egFLYBASECG), 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT), 12 org.Dm.egGENENAME, 13 org.Dm.egGENENAME, 13 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17	org.Dm.egUNIPROT, 24	
org.Dm.eg_dbconn, 25  AnnDbObj, 25  org.Dm.egFLYBASECG, 11  org.Dm.egFLYBASECG2EG  (org.Dm.egFLYBASECG), 11  org.Dm.egFLYBASECG), 11  org.Dm.egFLYBASEPROT, 12  org.Dm.egFLYBASEPROT2EG  (org.Dm.egFLYBASEPROT2EG  (org.Dm.egFLYBASEPROT), 12  dbconn, 25  dbConnect, 25  dbDisconnect, 25  dbDisconnect, 25  dbfile, 25  org.Dm.egGO, 15  org.Dm.egGO2ALLEGS, 16  org.Dm.egGO2ALLEGS (org.Dm.egGO), 15  dbInfo, 25  dbSchema, 25  org.Dm.egMAP, 17  org.Dm.egMAP, 17  org.Dm.egMAP, 17	* utilities	
org.Dm.egFLYBASECG2EG (org.Dm.egFLYBASECG), 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT2EG (org.Dm.egFLYBASEPROT), 12 dbconn, 25 dbConnect, 25 dbDisconnect, 25 dbfile, 25 dbGetQuery, 25 dbInfo, 25 dbSchema, 25 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGAPP, 17 org.Dm.egMAP2EG (org.Dm.egMAP), 17	org.Dm.eg_dbconn, 25	
cat, 25 checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconnect, 25 dbDisconnect, 25 dbGetQuery, 25 dbGnfo, 25 dbInfo, 25 dbschema, 25  cat, 25 corg.Dm.egFLYBASECG, 11 org.Dm.egFLYBASEPROT, 12 org.Dm.egFLYBASEPROT2EG (org.Dm.egFLYBASEPROT), 12 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGO, 15 org.Dm.egGO, 15 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egMAP, 17 org.Dm.egMAP, 17	AnnDhOhi 25	org.Dm.egFLYBASECG, 11
checkMAPCOUNTS, 18 count.mappedkeys, 18  dbconn, 25 dbConnect, 25 dbDisconnect, 25 dbfile, 25 dbGetQuery, 25 dbInfo, 25 dbSchema, 25  checkMAPCOUNTS, 18 corg.Dm.egFLYBASEPROT, 12 org.Dm.egGENENAME, 13 org.Dm.egGENETYPE, 14 org.Dm.egGO, 15 org.Dm.egGO2ALLEGS, 16 org.Dm.egGO2ALLEGS (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15 org.Dm.egGO2EG (org.Dm.egGO), 15	Allibood, 25	org.Dm.egFLYBASECG2EG
checkMAPCOUNTS, 18       org.Dm.egFLYBASEPROT, 12         count.mappedkeys, 18       org.Dm.egFLYBASEPROT2EG         (org.Dm.egFLYBASEPROT), 12         dbconn, 25       org.Dm.egGENENAME, 13         dbConnect, 25       org.Dm.egGENETYPE, 14         dbDisconnect, 25       org.Dm.egG0, 15         dbfile, 25       org.Dm.egG02ALLEGS, 16         dbGetQuery, 25       org.Dm.egG02ALLEGS (org.Dm.egG0), 15         dbInfo, 25       org.Dm.egG02EG (org.Dm.egG0), 15         dbschema, 25       org.Dm.egMAP, 17         org.Dm.egMAP2EG (org.Dm.egMAP), 17	cat 25	(org.Dm.egFLYBASECG), 11
count.mappedkeys, 18       org.Dm.egFLYBASEPROT2EG		org.Dm.egFLYBASEPROT, 12
dbconn, 25  dbConnect, 25  dbDisconnect, 25  dbfile, 25  dbGetQuery, 25  dbInfo, 25  dbSchema, 25  dbschema, 25  dcorg. Dm. egGENENAME, 13  org. Dm. egGENETYPE, 14  org. Dm. egGO, 15  org. Dm. egGO2ALLEGS, 16  org. Dm. egGO2ALLEGS (org. Dm. egGO), 15  org. Dm. egGO2EG (org. Dm. egGO), 15  org. Dm. egMAP, 17  org. Dm. egMAP, 17		org.Dm.egFLYBASEPROT2EG
dbConnect, 25  dbDisconnect, 25  dbfile, 25  dbGetQuery, 25  dbInfo, 25  dbSchema, 25  dbSchema, 25  org.Dm.egGENETYPE, 14  org.Dm.egGO, 15  org.Dm.egGO2ALLEGS, 16  org.Dm.egGO2ALLEGS (org.Dm.egGO), 15  org.Dm.egGO2EG (org.Dm.egGO), 15  org.Dm.egMAP, 17  org.Dm.egMAP, 17		(org.Dm.egFLYBASEPROT), 12
dbConnect, 25       org.Dm.egGENETYPE, 14         dbDisconnect, 25       org.Dm.egGO, 15         dbfile, 25       org.Dm.egGO2ALLEGS, 16         dbGetQuery, 25       org.Dm.egGO2ALLEGS (org.Dm.egGO), 15         dbInfo, 25       org.Dm.egGO2EG (org.Dm.egGO), 15         dbschema, 25       org.Dm.egMAP, 17         org.Dm.egMAP2EG (org.Dm.egMAP), 17	dbconn. 25	org.Dm.egGENENAME, 13
dbDisconnect, 25       org.Dm.egGO, 15         dbfile, 25       org.Dm.egG02ALLEGS, 16         dbGetQuery, 25       org.Dm.egG02ALLEGS (org.Dm.egGO), 15         dbInfo, 25       org.Dm.egG02EG (org.Dm.egGO), 15         dbschema, 25       org.Dm.egMAP, 17         org.Dm.egMAP2EG (org.Dm.egMAP), 17		org.Dm.egGENETYPE, 14
dbfile, 25org.Dm.egGO2ALLEGS, 16dbGetQuery, 25org.Dm.egGO2ALLEGS (org.Dm.egGO), 15dbInfo, 25org.Dm.egGO2EG (org.Dm.egGO), 15dbschema, 25org.Dm.egMAP, 17org.Dm.egMAP2EG (org.Dm.egMAP), 17		org.Dm.egGO, 15
dbGetQuery, 25       org.Dm.egGO2ALLEGS (org.Dm.egGO), 15         dbInfo, 25       org.Dm.egGO2EG (org.Dm.egGO), 15         dbschema, 25       org.Dm.egMAP, 17         org.Dm.egMAP2EG (org.Dm.egMAP), 17		org.Dm.egGO2ALLEGS, 16
$\begin{array}{ll} \mbox{dbInfo, $25$} & \mbox{org.Dm.egGO2EG (org.Dm.egGO), $15$} \\ \mbox{dbschema, $25$} & \mbox{org.Dm.egMAP, $17$} \\ \mbox{org.Dm.egMAP2EG (org.Dm.egMAP), $17$} \end{array}$		org.Dm.egGO2ALLEGS(org.Dm.egGO), 15
dbschema, 25 org.Dm.egMAP, 17 org.Dm.egMAP2EG (org.Dm.egMAP), 17		org.Dm.egGO2EG (org.Dm.egGO), 15
org.Dm.egMAP2EG(org.Dm.egMAP), 17		
	,	
	mappedkeys, 18	org.Dm.egMAPCOUNTS, 18

28 INDEX

```
org.Dm.egORGANISM, 19
org.Dm.egPATH, 19
org.Dm.egPATH2EG (org.Dm.egPATH), 19
org.Dm.egPMID, 20
org.Dm.egPMID2EG (org.Dm.egPMID), 20
org.Dm.egREFSEQ, 21
org.Dm.egREFSEQ2EG (org.Dm.egREFSEQ), 21
org.Dm.egSYMBOL, 23
org.Dm.egSYMBOL2EG (org.Dm.egSYMBOL), 23
org.Dm.egUNIPROT, 24
```