yeast2.db

November 11, 2026

`yeast2ALIAS` *Map Open Reading Frame (ORF) Identifiers to Alias Gene Names*

Description

A set of gene names may have been used to report yeast genes represented by ORF identifiers. One of these names is chosen to be the primary gene name, and the others are considered aliases. This R object provides mappings between the primary name and aliases.

Details

Each primary name maps to a vector of alias names. If there are no aliases, the vector will contain `NA`.

Annotation based on data provided by: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

References

http://www.yeastgenome.org/DownloadContents.shtml

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 <- yeast2ALIAS
# Get the probe identifiers that are mapped to alias names
mapped_probes <- mappedkeys(x)
# Convert to a list
```

```
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
    # Get the alias names for the first five probes
    xx[1:5]
    # For the first probe
    xx[[1]]
}
```

---

yeast2.db *Bioconductor annotation data package*

Description

Welcome to the yeast2.db annotation Package. The purpose of this package is to provide detailed information about the yeast2 platform. 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(yeast2.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:yeast2.db")
```

---

yeast2CHR *Map Manufacturer IDs to Chromosomes*

Description

yeast2CHR is an R object that provides mappings between a manufacturer identifier and the chromosome that contains the gene of interest.

Details

Each manufacturer identifier maps to a vector of chromosomes. Due to inconsistencies that may exist at the time the object was built, the vector may contain more than one chromosome (e.g., the identifier may map to more than one chromosome). If the chromosomal location is unknown, the vector will contain an NA.

Mappings were based on data provided by: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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 <- yeast2CHR
# Get the probe identifiers that are mapped to a chromosome
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the CHR for the first five probes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

yeast2CHRLENGTHS *A named vector for the length of each of the chromosomes*

Description

yeast2CHRLENGTHS 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 <- yeast2CHRLENGTHS
# Length of chromosome 1
tt["1"]
```

---

yeast2CHRLOC *Map Manufacturer IDs to Chromosomal Location*

Description

yeast2CHRLOC is an R object that maps manufacturer 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 manufacturer identifier maps to a named vector of chromosomal locations, where the name indicates the chromosome. Due to inconsistencies that may exist at the time the object was built, these vectors may contain more than one chromosome and/or location. If the chromosomal location is unknown, the vector will contain an NA.

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: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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 <- yeast2CHRLOC
# Get the probe identifiers that are mapped to chromosome locations
mapped_probes <- mappedkeys(x)
```

```
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the CHRLOC for the first five probes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

yeast2DESCRIPTION *An annotation data file that maps Open Reading Frame (ORF) identifiers to textural descriptions of the corresponding genes*

Description

yeast2DESCRIPTION maps yeast ORF identifiers to descriptive information about genes corresponding to the ORF identifiers

Details

This is an R object containing key and value pairs. Keys are ORF identifiers and values are the corresponding descriptions of genes. Values are vectors of length 1. Probe identifiers that can not be mapped to descriptive information are assigned a value NA.

Annotation based on data provided by: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

References

http://www.yeastgenome.org/DownloadContents.shtml

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 <- yeast2DESCRIPTION
# Get the probe identifiers that are mapped to gene descriptions
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the gene descriptions for the first five probes
  xx[1:5]
```

```
  # For the first probe
  xx[[1]]
}
```

---

yeast2ENSEMBL *Map Ensembl gene accession numbers with Entrez Gene identifiers*

Description

yeast2ENSEMBL 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

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.

Mappings were based on data provided by: Ensembl ftp://ftp.ensembl.org/pub/current_fasta With a date stamp from the source of: 2021-Feb16

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 <- yeast2ENSEMBL
# 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 ENSEMBL2PROBE:
# Convert to a list
xx <- as.list(yeast2ENSEMBL2PROBE)
if(length(xx) > 0){
   # Gets the entrez gene IDs for the first five Ensembl IDs
   xx[1:5]
   # Get the first one
   xx[[1]]
}
```

---

yeast2ENZYME *Map between Manufacturer IDs and Enzyme Commission (EC) Numbers*

Description

yeast2ENZYME is an R object that provides mappings between manufacturer identifiers and EC numbers.

Details

Each manufacturer identifier maps to a named vector containing the EC number that corresponds to the enzyme produced by that gene. The name corresponds to the manufacturer 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 yeast2ENZYME2PROBE, 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 probe 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

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 <- yeast2ENZYME
# Get the probe identifiers that are mapped to an EC number
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the ENZYME for the first five probes
  xx[1:5]
  # For the first probe
  xx[[1]]
}
```

---

yeast2ENZYME2PROBE *Map between Enzyme Commission Numbers and Manufacturer Identifiers*

Description

yeast2ENZYME2PROBE is an R object that maps Enzyme Commission (EC) numbers to manufacturer identifiers.

Details

Each EC number maps to a named vector containing all of the manufacturer identifiers that correspond to the gene that produces that enzyme. The name of the vector corresponds to the EC number.

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 yeast2ENZYME2PROBE, 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 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

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:
# Convert to a list
xx <- as.list(yeast2ENZYME2PROBE)
if(length(xx) > 0){
    # Gets the probe identifiers for the first five enzyme
    #commission numbers
    xx[1:5]
    # Get the first one
    xx[[1]]
}
```

---

yeast2GENENAME *Map between Manufacturer IDs and Genes*

Description

yeast2GENENAME is an R object that maps manufacturer identifiers to the corresponding gene name.

Details

Each manufacturer identifier maps to a named vector containing the gene name. The vector name corresponds to the manufacturer 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: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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 <- yeast2GENENAME
# Get the probe identifiers that are mapped to a gene name
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the GENENAME for the first five probes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

yeast2GO *Map between Manufacturer IDs and Gene Ontology (GO)*

Description

yeast2GO is an R object that provides mappings between manufacturer identifiers and the GO identifiers that they are directly associated with.

Details

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

Mappings between probe identifiers and GO information were obtained through their mappings to Entrez Gene identifiers. NAs are assigned to probe 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.

Mappings were based on data provided by: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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 <- yeast2GO
# Get the probe identifiers that are mapped to a GO ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
    # Try the firest one
    got <- xx[[1]]
    got[[1]][["GOID"]]
```

```
    got[[1]][["Ontology"]]
    got[[1]][["Evidence"]]

}
```

---

yeast2GO2ALLPROBES *Map between Gene Ontology (GO) Identifiers and all Manufacturer Identifiers in the subtree*

Description

yeast2GO2ALLPROBES is an R object that provides mappings between a given GO identifier and all manufactuerer identifiers annotated at that GO term or one of its children in the GO ontology.

Details

GO consists of three ontologies—molecular function (MF), biological process (BP), and cellular component (CC). All ontologies are structured as directed acyclic graphs (DAGs). Each node in each DAG (tree) is a GO term (id) associated with a named vector of manufacturer identifiers. The name associated with each manufacturer id corresponds to the evidence code for that GO identifier. This object yeast2GO2ALLPROBES maps between a given GO identifier and all manufactuerer identifiers annotated at that GO term or one of its children in the GO ontology.

The evidence code indicates what kind of evidence supports the association between the GO and Entrez Gene identifiers. Evidence codes currently 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 GO identifier may be mapped to the same manufacturer 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.

Mappings were based on data provided by:

Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25 and Gene Ontology http://current.geneontology.org/ontology/go-basic.obo With a date stamp from the source of: 2021-02-01

References

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:
# Convert to a list
xx <- as.list(yeast2GO2ALLPROBES)
if(length(xx) > 0){
    # Gets the probe identifiers for the top 2nd and 3nd GO identifiers
    goids <- xx[2:3]
    # Gets all the probe identifiers for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}
```

---

yeast2GO2PROBE *Map between Gene Ontology (GO) and Manufacturer Identifiers*

Description

yeast2GO2PROBE is an R object that provides mappings between GO identifiers and manufacturer identifiers.

Details

Each GO term maps to a named vector of manufacturer identifiers. The name associated with each manufacturer identifier corresponds to the evidence code for that GO identifier. The evidence code indicates what kind of evidence supports the association between the GO and Entrez Gene identifiers. Evidence codes currently 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 GO identifier may be mapped to the same probe identifier more than once but the evidence code can be different. Mappings between Gene Ontology identifiers an Gene Ontology terms and other information are available in a separate data package named GO.

Mappings were based on data provided by:

Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

References

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:
# Convert to a list
xx <- as.list(yeast2GO2PROBE)
if(length(xx) > 0){
    # Gets the probe identifiers for the top 2nd and 3nd GO identifiers
    goids <- xx[2:3]
    # Gets the probe identifiers for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}
```

---

yeast2MAPCOUNTS *Number of mapped keys for the maps in package yeast2.db*

Description

DEPRECATED. Counts in the MAPCOUNT table are out of sync and should not be used.

yeast2MAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package yeast2.db.

Details

DEPRECATED. Counts in the MAPCOUNT table are out of sync and should not be used.

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.

`yeast2ORF` *Map Manufacturer Identifiers to Open Reading Frame (ORF) Identifiers*

Description

yeast2ORF is an R object that provides mappings between manufacturer and ORF identifiers.

Details

Each manufacturer identifier is mapped to a vector of ORF identifiers. The length of the vector may be one or longer, depending on how many ORF identifiers the manufacturer identifier can be mapped to. An NA is reported for any manufacturer identifier that cannot be mapped to an ORF identifier at this time.

Annotation based on data provided by: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

References

http://www.yeastgenome.org/DownloadContents.shtml

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 <- yeast2ORF
# Get the probe identifiers that are mapped to ORF identifiers
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the ORF identifiers for the first five probes
  xx[1:5]
  # For the first probe
```

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

---

yeast2ORGANISM *The Organism information for yeast2*

Description

yeast2ORGANISM is an R object that contains a single item: a character string that names the organism for which yeast2 was built. yeast2ORGPKG is an R object that contains a chararcter vector with the name of the organism package that a chip package depends on for its gene-centric annotation.

Details

Although the package name is suggestive of the organism for which it was built, yeast2ORGANISM provides a simple way to programmatically extract the organism name. yeast2ORGPKG provides a simple way to programmatically extract the name of the parent organism package. The parent organism package is a strict dependency for chip packages as this is where the gene cetric information is ultimately extracted from. The full package name will always be this string plus the extension ".db". But most programatic acces will not require this extension, so its more convenient to leave it out.

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:
yeast2ORGANISM
yeast2ORGPKG
```

---

yeast2PATH *Mappings between probe identifiers and KEGG pathway identifiers*

Description

KEGG (Kyoto Encyclopedia of Genes and Genomes) maintains pathway data for various organisms. yeast2PATH maps probe identifiers to the identifiers used by KEGG for pathways in which the genes represented by the probe identifiers are involved

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 <- yeast2PATH
# Get the probe identifiers that are mapped to a KEGG pathway ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0) {
  # Get the PATH for the first five probes
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

---

yeast2PATH2PROBE *Map between Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway identifiers and Manufacturer Identifiers*

Description

yeast2PATH2PROBE is an R object that provides mappings between KEGG identifiers and manufacturer identifiers.

Details

Each KEGG identifier is mapped to a named vector of manufacturer identifiers. The name represents the KEGG identifier and the vector contains all manufacturer identifiers that are found in that particular pathway. An NA is reported for any KEGG identifier that cannot be mapped to any manufacturer identifiers.

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 http://www.genome.ad.jp/kegg/pathway.html 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:
# Convert the object to a list
xx <- as.list(yeast2PATH2PROBE)
# Remove pathway identifiers that do not map to any probe id
xx <- xx[!is.na(xx)]
if(length(xx) > 0){
    # The probe identifiers for the first two elements of XX
    xx[1:2]
    # Get the first one
    xx[[1]]
}
```

---

yeast2PMID *Map between Manufacturer Identifiers and PubMed Identifiers*

Description

yeast2PMID is an R object that provides mappings between manufacturer identifiers and PubMed identifiers.

Details

Each manufacturer identifier is mapped to a named vector of PubMed identifiers. The name associated with each vector corresponds to the manufacturer identifier. The length of the vector may be one or greater, depending on how many PubMed identifiers a given manufacturer identifier is mapped to. An NA is reported for any manufacturer 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: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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 <- yeast2PMID
# Get the probe identifiers that are mapped to any PubMed ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes])
if(length(xx) > 0){
    # The probe 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")
    }
}
```

yeast2PMID2PROBE *Map between PubMed Identifiers and Manufacturer Identifiers*

Description

yeast2PMID2PROBE is an R object that provides mappings between PubMed identifiers and manufacturer identifiers.

Details

Each PubMed identifier is mapped to a named vector of manufacturer identifiers. The name represents the PubMed identifier and the vector contains all manufacturer identifiers that are represented by that PubMed identifier. The length of the vector may be one or longer, depending on how many manufacturer identifiers are mapped to a given 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: Yeast Genome http://sgd-archive.yeastgenome.org With a date stamp from the source of: 2019-Oct25

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(yeast2PMID2PROBE)
if(length(xx) > 0){
    # The probe 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")
    }
}
```

yeast2_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

```
yeast2_dbconn()
yeast2_dbfile()
yeast2_dbschema(file="", show.indices=FALSE)
yeast2_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

yeast2_dbconn returns a connection object to the package annotation DB. IMPORTANT: Don't call dbDisconnect on the connection object returned by yeast2_dbconn or you will break all the AnnDbObj objects defined in this package!

yeast2_dbfile returns the path (character string) to the package annotation DB (this is an SQLite file).

yeast2_dbschema prints the schema definition of the package annotation DB.

yeast2_dbInfo prints other information about the package annotation DB.

Value

yeast2_dbconn: a DBIConnection object representing an open connection to the package annotation DB.

yeast2_dbfile: a character string with the path to the package annotation DB.

yeast2_dbschema: none (invisible NULL).

yeast2_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

```
library(DBI)
## Count the number of rows in the "probes" table:
dbGetQuery(yeast2_dbconn(), "SELECT COUNT(*) FROM probes")

yeast2_dbschema()

yeast2_dbInfo()
```

Index

∗ **datasets**
  yeast2.db, 2
  yeast2_dbconn, 21
  yeast2ALIAS, 1
  yeast2CHR, 2
  yeast2CHRLENGTHS, 3
  yeast2CHRLOC, 4
  yeast2DESCRIPTION, 5
  yeast2ENSEMBL, 6
  yeast2ENZYME, 7
  yeast2ENZYME2PROBE, 8
  yeast2GENENAME, 9
  yeast2GO, 10
  yeast2GO2ALLPROBES, 12
  yeast2GO2PROBE, 13
  yeast2MAPCOUNTS, 14
  yeast2ORF, 15
  yeast2ORGANISM, 16
  yeast2PATH, 16
  yeast2PATH2PROBE, 17
  yeast2PMID, 18
  yeast2PMID2PROBE, 20
∗ **utilities**
  yeast2_dbconn, 21

AnnDbObj, *21*

cat, *21*
checkMAPCOUNTS, *15*

dbconn, *21*
dbConnect, *21*
dbDisconnect, *21*
dbfile, *21*
dbGetQuery, *21*
dbInfo, *21*
dbschema, *21*

yeast2 *(*yeast2.db*)*, 2
yeast2.db, 2
yeast2_dbconn, 21
yeast2_dbfile *(*yeast2_dbconn*)*, 21
yeast2_dbInfo *(*yeast2_dbconn*)*, 21
yeast2_dbschema *(*yeast2_dbconn*)*, 21
yeast2ALIAS, 1
yeast2ALIAS2PROBE *(*yeast2ALIAS*)*, 1
yeast2CHR, 2
yeast2CHRLENGTHS, 3
yeast2CHRLOC, 4
yeast2CHRLOCEND *(*yeast2CHRLOC*)*, 4
yeast2DESCRIPTION, 5
yeast2ENSEMBL, 6
yeast2ENSEMBL2PROBE *(*yeast2ENSEMBL*)*, 6
yeast2ENZYME, 7
yeast2ENZYME2PROBE, 8
yeast2GENENAME, 9
yeast2GO, 10
yeast2GO2ALLPROBES, 12
yeast2GO2PROBE, 13
yeast2MAPCOUNTS, 14
yeast2ORF, 15
yeast2ORGANISM, 16
yeast2ORGPKG *(*yeast2ORGANISM*)*, 16
yeast2PATH, 16
yeast2PATH2PROBE, 17
yeast2PMID, 18
yeast2PMID2PROBE, 20