PFAM.db

April 29, 2025

PFAM.db *Bioconductor annotation data package*

Description

Welcome to the PFAM.db annotation Package. The purpose of this package is to provide detailed information about the PFAM 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.

## 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:PFAM.db")
```

PFAMCAZY *Mappings from a PFAM Accession number to another kind of ID*

Description

This is an R object that always contains mappings from a PFAM ID to the other ID type named by the object

Details

For each of the aliases listed above, there is a mapping object that corresponds which will map you from the appropriate PFAM accession number to the ID type that is in the environments name. So for example PFAMCAZY will map from PFAM IDs to CAZY IDs while PFAMDE will map from PFAM IDs to Descriptions.

CAZY: The CAZy database (<URL: http://afmb.cnrs-mrs.fr/CAZY/>) describes the families of structurally-related catalytic and carbohydrate-binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds.

DE: Definition for all the Accession number in the database.

ID: Associated Identification for all the Accession number in the database.

HOMSTRAD: HOMSTRAD (HOMologous STRucture Alignment Database, <URL: http://www-cryst.bioc.cam.ac.uk/homstrad/>) is a curated database of structure-based alignments for homologous protein families. Reference: Mizuguchi K, Deane CM, Blundell TL, Overington JP. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. Protein Science 7:2469-2471.

INTERPRO: Associated INTERPRO ID for all the Accession number in the database. <URL: http://www.ebi.ac.uk/interpro/>

LOAD: LOAD ID for all the Accession number in the database.

MEROPS: The MEROPS database (<URL: http://merops.sanger.ac.uk/>) is an information resource for peptidases (also termed proteases, proteinases and proteolytic enzymes) and the proteins that inhibit them. Reference: Rawlings, N.D., Tolle, D.P. & Barrett, A.J. (2004) MEROPS: the peptidase database. Nucleic Acids Res. 32 Database issue, D160-D164

MIM: MIM (a.k.a. OMIM, <URL: https://www.ncbi.nlm.nih.gov/omim/>) is a catalog of human genes and genetic disorders authored and edited by Dr. Victor A. McKusick and his colleagues at Johns Hopkins and elsewhere. Reference: <MIM> MIM: McKusick, V.A.: Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders. Baltimore: Johns Hopkins University Press, 1998 (12th edition). <OMIM> Online Mendelian Inheritance in Man, OMIM (TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), 2000

PRINTS: PRINTS (<URL: http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/>) is a compendium of protein fingerprints.

PROSITEPROFILE: A list of associated PROSITE PROFILE ID.

RM: Reference Medline (<URL: https://www.ncbi.nlm.nih.gov/PubMed/>)

SMART: SMART (a Simple Modular Architecture Research Tool, <URL: http://smart.embl-heidelberg.de/>) allows the identification and annotation of genetically mobile domains and the analysis of domain architectures. Reference: (1) Schultz et al. (1998) Proc. Natl. Acad. Sci. USA 95, 5857-5864. (2) Letunic et al. (2004) Nucleic Acids Res 32, D142-D144

TC: Trusted cutoff for all the Accession number in the database.

TP: A list of associated Type field for the given Accession.

URL: A list of associated URL for all the Accession number in the database.

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from PFAM to CAZYs:
x <- PFAMCAZY
# Get the PFAM identifiers that are mapped to a CAZY
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the CAZY for the first five PFAM IDs
  xx[1:5]
  # Get the first one
  xx[[1]]
}

#Or to use the DE mapping:
x <- PFAMDE
# Get the PFAM identifiers that are mapped to a DE
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the DE for the first five PFAM IDs
  xx[1:5]
  # Get the first one
  xx[[1]]
}

#etc.
```

PFAMMAPCOUNTS *Number of mapped keys for the maps in package PFAM.db*

Description

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

Details

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

See Also

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

Examples

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

## Bimap interface:
PFAMMAPCOUNTS
mapnames <- names(PFAMMAPCOUNTS)
PFAMMAPCOUNTS[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 PFAM.db
checkMAPCOUNTS("PFAM.db")
```

PFAMPDB *Mappings from a PFAM Accession number to a PDB ID*

Description

This is an R object that always contains mappings from a PFAM accession (AC) to a PDB ID

Details

The PDB ID along with the start point and end point have been attached to the PFAM accessions in this object.

More Details: PDB (http://www.rcsb.org/pdb/index.html), the single worldwide repository for the processing and distribution of 3-D biological macromolecular structure data. Reference: H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: The Protein Data Bank. Nucleic Acids Research , 28 pp. 235-242 (2000)

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from PFAM to PDB:
x <- PFAMPDB
# Get the PFAM identifiers that are mapped to a PDB
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the PDB info for the first five PFAM IDs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

PFAMPDB2AC *Mappings from a PDB ID to a PFAM Accession number*

Description

This is an R object that always contains mappings from a PDB ID to a PFAM accession (AC)

Details

The PFAM accession number has been attached to the PDB IDs in this object.

More Details: PDB (http://www.rcsb.org/pdb/index.html), the single worldwide repository for the processing and distribution of 3-D biological macromolecular structure data. Reference: H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: The Protein Data Bank. Nucleic Acids Research , 28 pp. 235-242 (2000)

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from PDB to PFAM IDs:
x <- PFAMPDB2AC
# Get the PDB identifiers that are mapped to a PFAM ID
```

```
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the PDB ID for the first five CAZYs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

PFAMCAZY2AC *Mappings from an ID to a PFAM Accession number*

Description

This is an R object that contains mapppings from an ID to its PFAM Accession number (AC)

Details

For each of the aliases listed above, there is a mapping object that corresponds which will map you from the ID type in the environments name to the appropriate PFAM accession number. So for example PFAMCAZY2AC will map from CAZY IDs to PFAM IDs while PFAMDE2AC will map from Descriptions to PFAM IDs.

Details on supported things that are mapped to:

CAZY: The CAZy database (<URL: http://afmb.cnrs-mrs.fr/CAZY/>) describes the families of structurally-related catalytic and carbohydrate-binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds.

DE: Definition for all the Accession number in the database.

ID: Associated Identification for all the Accession number in the database.

HOMSTRAD: HOMSTRAD (HOMologous STRucture Alignment Database, <URL: http://www-cryst.bioc.cam.ac.uk/homstrad/>) is a curated database of structure-based alignments for homologous protein families. Reference: Mizuguchi K, Deane CM, Blundell TL, Overington JP. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. Protein Science 7:2469-2471.

INTERPRO: Associated INTERPRO ID for all the Accession number in the database. <URL: http://www.ebi.ac.uk/interpro/>

LOAD: LOAD ID for all the Accession number in the database.

MEROPS: The MEROPS database (<URL: http://merops.sanger.ac.uk/>) is an information resource for peptidases (also termed proteases, proteinases and proteolytic enzymes) and the proteins that inhibit them. Reference: Rawlings, N.D., Tolle, D.P. & Barrett, A.J. (2004) MEROPS: the peptidase database. Nucleic Acids Res. 32 Database issue, D160-D164

MIM: MIM (a.k.a. OMIM, <URL: https://www.ncbi.nlm.nih.gov/omim/>) is a catalog of human genes and genetic disorders authored and edited by Dr. Victor A. McKusick and his colleagues at Johns Hopkins and elsewhere. Reference: <MIM> MIM: McKusick, V.A.: Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders. Baltimore: Johns Hopkins University Press, 1998 (12th edition). <OMIM> Online Mendelian Inheritance in Man, OMIM (TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), 2000

PRINTS: PRINTS (<URL: http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/>) is a compendium of protein fingerprints.

PROSITEPROFILE: A list of associated PROSITE PROFILE ID.

RM: Reference Medline (<URL: https://www.ncbi.nlm.nih.gov/PubMed/>)

SMART: SMART (a Simple Modular Architecture Research Tool, <URL: http://smart.embl-heidelberg.de/>) allows the identification and annotation of genetically mobile domains and the analysis of domain architectures. Reference: (1) Schultz et al. (1998) Proc. Natl. Acad. Sci. USA 95, 5857-5864. (2) Letunic et al. (2004) Nucleic Acids Res 32, D142-D144

TC: Trusted cutoff for all the Accession number in the database.

TP: A list of associated Type field for the given Accession.

URL: A list of associated URL for all the Accession number in the database.

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from CAZY to PFAM IDs:
x <- PFAMCAZY2AC
# Get the CAZY identifiers that are mapped to a PFAM ID
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the PFAM ID for the first five CAZYs
  xx[1:5]
  # Get the first one
  xx[[1]]
}

#Or to use the DE2AC mapping:
x <- PFAMDE2AC
# Get the Descriptions that are mapped to a PFAM ID
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the PFAM ID for the first five DEs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

PFAMSCOP *Mappings from a PFAM Accession number to a SCOP ID*

Description

This is an R object that always contains mappings from a PFAM accession (AC) to a SCOP ID

Details

The SCOP ID along with the start point and end point have been attached to the PFAM accessions in this object.

More Details: Structural Classification of Proteins (http://scop.mrc-lmb.cam.ac.uk/scop/index.html). Reference: Murzin A. G., Brenner S. E., Hubbard T., Chothia C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247, 536-540

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from PFAM to SCOP:
x <- PFAMSCOP
# Get the PFAM identifiers that are mapped to a SCOP
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the SCOP info for the first five PFAM IDs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

PFAMSCOP2AC *Mappings from a SCOP ID to a PFAM Accession number*

Description

This is an R object that always contains mappings from a SCOP ID to a PFAM accession (AC)

Details

The PFAM accession number has been attached to the SCOP IDs in this object.

More Details: Structural Classification of Proteins (http://scop.mrc-lmb.cam.ac.uk/scop/index.html). Reference: Murzin A. G., Brenner S. E., Hubbard T., Chothia C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247, 536-540

References

http://www.sanger.ac.uk/Software/Pfam/ and ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/userman.txt

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:
#To map from SCOP to PFAM IDs:
x <- PFAMSCOP2AC
# Get the SCOP identifiers that are mapped to a PFAM ID
mapped_keys <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_keys])
if(length(xx) > 0) {
  # Get the SCOP ID for the first five CAZYs
  xx[1:5]
  # Get the first one
  xx[[1]]
}
```

PFAM_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

```
PFAM_dbconn()
PFAM_dbfile()
PFAM_dbschema(file="", show.indices=FALSE)
PFAM_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

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

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

PFAM_dbschema prints the schema definition of the package annotation DB.

PFAM_dbInfo prints other information about the package annotation DB.

Value

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

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

PFAM_dbschema: none (invisible NULL).

PFAM_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

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

PFAM_dbschema()

PFAM_dbInfo()
```

Index

∗ **datasets**
PFAM.db, 1
PFAM_dbconn, 10
PFAMCAZY, 1
PFAMCAZY2AC, 6
PFAMMAPCOUNTS, 3
PFAMPDB, 4
PFAMPDB2AC, 5
PFAMSCOP, 8
PFAMSCOP2AC, 9
∗ **utilities**
PFAM_dbconn, 10

AnnDbObj, *10*

cat, *10*
checkMAPCOUNTS, *3*, *4*
count.mappedkeys, *4*

dbconn, *10*
dbConnect, *10*
dbDisconnect, *10*
dbfile, *10*
dbGetQuery, *10*
dbInfo, *10*
dbschema, *10*

mappedkeys, *4*

PFAM *(*PFAM.db*)*, 1
PFAM.db, 1
PFAM_dbconn, 10
PFAM_dbfile *(*PFAM_dbconn*)*, 10
PFAM_dbInfo *(*PFAM_dbconn*)*, 10
PFAM_dbschema *(*PFAM_dbconn*)*, 10
PFAMCAZY, 1
PFAMCAZY2AC, 6
PFAMDE *(*PFAMCAZY*)*, 1
PFAMDE2AC *(*PFAMCAZY2AC*)*, 6
PFAMHOMSTRAD *(*PFAMCAZY*)*, 1
PFAMHOMSTRAD2AC *(*PFAMCAZY2AC*)*, 6
PFAMID *(*PFAMCAZY*)*, 1
PFAMID2AC *(*PFAMCAZY2AC*)*, 6
PFAMINTERPRO *(*PFAMCAZY*)*, 1
PFAMINTERPRO2AC *(*PFAMCAZY2AC*)*, 6
PFAMLOAD *(*PFAMCAZY*)*, 1
PFAMLOAD2AC *(*PFAMCAZY2AC*)*, 6
PFAMMAPCOUNTS, 3
PFAMMEROPS *(*PFAMCAZY*)*, 1
PFAMMEROPS2AC *(*PFAMCAZY2AC*)*, 6
PFAMMIM *(*PFAMCAZY*)*, 1
PFAMMIM2AC *(*PFAMCAZY2AC*)*, 6
PFAMPDB, 4
PFAMPDB2AC, 5
PFAMPRINTS *(*PFAMCAZY*)*, 1
PFAMPRINTS2AC *(*PFAMCAZY2AC*)*, 6
PFAMPROSITE *(*PFAMCAZY*)*, 1
PFAMPROSITE2AC *(*PFAMCAZY2AC*)*, 6
PFAMPROSITEPROFILE *(*PFAMCAZY*)*, 1
PFAMPROSITEPROFILE2AC *(*PFAMCAZY2AC*)*, 6
PFAMRM *(*PFAMCAZY*)*, 1
PFAMRM2AC *(*PFAMCAZY2AC*)*, 6
PFAMSCOP, 8
PFAMSCOP2AC, 9
PFAMSMART *(*PFAMCAZY*)*, 1
PFAMSMART2AC *(*PFAMCAZY2AC*)*, 6
PFAMTC *(*PFAMCAZY*)*, 1
PFAMTC2AC *(*PFAMCAZY2AC*)*, 6
PFAMTP *(*PFAMCAZY*)*, 1
PFAMTP2AC *(*PFAMCAZY2AC*)*, 6
PFAMURL *(*PFAMCAZY*)*, 1
PFAMURL2AC *(*PFAMCAZY2AC*)*, 6