org.Rn.sp.db

October 16, 2009

R topics documented:

---

org.Rn.sp_dbconn   *Collect information about the package annotation DB*

Description

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

Usage

```
org.Rn.sp_dbconn()
org.Rn.sp_dbfile()
org.Rn.sp_dbschema(file="", show.indices=FALSE)
org.Rn.sp_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.Rn.sp_dbconn returns a connection object to the package annotation DB. IMPORTANT: Don't call dbDisconnect on the connection object returned by org.Rn.sp_dbconn or you will break all the AnnDbObj objects defined in this package!

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

org.Rn.sp_dbschema prints the schema definition of the package annotation DB.

org.Rn.sp_dbInfo prints other information about the package annotation DB.

Examples

```
## Show the first three rows.
dbGetQuery(org.Rn.sp_dbconn(), "select * from basic limit 3")

## The connection object returned by org.Rn.sp_dbconn() was created with:
dbConnect(SQLite(), dbname=org.Rn.sp_dbfile(), cache_size=64000, synchronous=0)

org.Rn.sp_dbschema()

org.Rn.sp_dbInfo()
```

```
org.Rn.sp.db          annotation data package
```

Description

Welcome to the org.Rn.sp.db annotation Package. The annotation package was built using a downloadable R package - PAnnBuilder (download and build your own). The purpose is to provide detailed information about proteins in SwissProt databases: ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Each of these objects has their own manual page detailing where relevant data was obtained along with examples of how to use it. Many of these objects also have a reverse map available. When this is true, expect to usually find relevant information on the same manual page as the forward map.

Examples

```
# You can learn what objects this package supports with the following command:
ls("package:org.Rn.sp.db")
```

```
org.Rn.spDE          Maps protein identifier to textural descriptions
```

Description

org.Rn.spDE maps protein identifiers to their descriptive information.

Details

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spDE
# Get the protein identifiers that are mapped to textural descriptions.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
            # Get the values for a few keys
            if(length(xx) >= 3){
                    xx[1:3]
            }
    }
```

---

org.Rn.spDISEASE   *Map SysPTM protein identifier to disease(s)*

Description

org.Rn.spDISEASE maps protein identifier to the description of the disease(s) associated with a deficiency of a protein.

Details

Each protein identifier maps to the description of the disease(s) associated with a deficiency of a protein. This information is extractd from "DISEASE" topic in SwissProt "comments" lines.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spDISEASE
# Get the protein identifiers that are mapped to disease(s).
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

---

org.Rn.spFUNCTION   *Map SysPTM protein identifier to function(s)*

Description

org.Rn.spFUNCTION maps protein identifier to the general description of the Function(s) of a protein.

Details

Each protein identifier maps to the general description of the Function(s) of a protein. This information is extractd from "FUNCTION" topic in SwissProt "comments" lines.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spFUNCTION
# Get the protein identifiers that are mapped to function(s).
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spGENEID	*Map protein identifier to Entrez gene identifier*

Description

org.Rn.spGENEID maps protein identifiers to Entrez Gene identifiers.

Details

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spGENEID
# Get the protein identifiers that are mapped to Entrez gene identifier.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
              # Get the values for a few keys
              if(length(xx) >= 3){
                      xx[1:3]
              }
      }
```

---

org.Rn.spGN	*Map SysPTM protein identifier to gene names*

Description

org.Rn.spGN maps protein identifier to the corresponding gene names.

Details

Each protein identifier maps to a vector containing the gene name or alias.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spGN
# Get the protein identifiers that are mapped to gene names.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

---

org.Rn.spGO   *Map protein identifier to GO*

Description

org.Rn.spGO maps protein identifiers to Gene Ontology identifiers .

Details

Each Protein identifier is mapped to a list. Each component contain : GO ID, Evidence and Ontology (C,F,P). NAs are assigned to probe identifiers that can not be mapped to any Gene Ontology information.

The Evidence element contains a code indicating what kind of evidence supports the association of the GO id to the protein id. 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

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spGO
# Get the protein identifiers that are mapped to a GO ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes])
if(length(xx) > 0) {
    # Try the first one
    got <- xx[[1]]
    got[[1]][["GOID"]]
```

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

# For the reverse map:
xx <- as.list(org.Rn.spGO2SPID)
if(length(xx) > 0){
    goids <- xx[2:3]
}
```

---

org.Rn.spINTERPRO   *Map protein identifier to Interpro doamins*

Description

org.Rn.spINTERPRO maps protein identifiers to Interpro identifiers .

Details

Each protein identifier maps to a vector of InterPro identifiers.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spINTERPRO
# Get the protein identifiers that are mapped to Interpro doamins.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

---

org.Rn.spINT   *Map protein identifier to interacted proteins*

Description

org.Rn.spINT maps protein identifiers to interacted proteins.

Details

Each protein identifier is mapped to a vector of protein identifiers which may have interaction between them.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spINT
# Get the protein identifiers that are mapped to interacted proteins.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

`org.Rn.spKEGG`   *Map protein identifier to KEGG gene identifiers*

Description

org.Rn.spKEGG maps protein identifiers to KEGG gene identifiers.

Details

Each protein identifier maps to KEGG gene identifiers.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
    x <- org.Rn.spKEGG
# Get the protein identifiers that are mapped to KEGG gene identifiers.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

`org.Rn.spLEN`   *Map protein identifier to the length of protein sequence*

Description

org.Rn.spLEN maps protein identifiers to the length of protein sequence.

Details

Each protein identifier maps to the length of protein sequence.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spLEN
# Get the protein identifiers that are mapped to the length of protein sequence.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

---

org.Rn.spMAPCOUNTS   *Number of mapped keys for the maps in package org.Rn.sp.db*

Description

org.Rn.spMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package org.Rn.sp.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

Examples

```
org.Rn.spMAPCOUNTS
mapnames <- names(org.Rn.spMAPCOUNTS)
org.Rn.spMAPCOUNTS[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.Rn.sp.db
checkMAPCOUNTS("org.Rn.sp.db")
```

---

org.Rn.spMW *Map protein identifier to its molecular weight*

Description

org.Rn.spMW maps protein identifiers to its molecular weight.

Details

Each protein identifier maps to its molecular weight.

Mappings were based on data provided by: IPI (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spMW
# Get the protein identifiers that are mapped to its molecular weight.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spORGANISM *The Organism for org.Rn.sp.db*

Description

org.Rn.spORGANISM is an R object that contains a single item: a character string that names the organism for which org.Rn.sp.db was built.

Details

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

Examples

```
org.Rn.spORGANISM
```

`org.Rn.spPATH` *Map protein identifier to KEGG pathway*

Description

org.Rn.spPATH maps protein identifiers to KEGG pathway identifiers.

Details

Each protein identifier maps to KEGG pathway identifiers.

Mappings were based on data provided by: KEGG (`ftp://ftp.genome.jp/pub/kegg/genes/organisms/rno/rno_pathway.list`) on

Examples

```
      x <- org.Rn.spPATH
# Get the protein identifiers that are mapped to KEGG pathway.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }

      ## For the reverse map org.Rn.spPATH2SPID
xx <- as.list(org.Rn.spPATH2SPID)
if(length(xx) > 0){
    ## Get the first one
    xx[[1]]
}
```

`org.Rn.spPDB` *Map protein identifier to PDB structure*

Description

org.Rn.spPDB maps protein identifiers to PDB structure.

Details

Each protein identifier is mapped to a vector of PDB 3D structure identifiers.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spPDB
# Get the protein identifiers that are mapped to PDB structure.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spPFAM   *Map protein identifier to Pfam domain*

Description

org.Rn.spPFAM maps protein identifiers to Pfam domain identifiers.

Details

Each protein identifier maps to a vector of Pfam domain identifiers.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spPFAM
# Get the protein identifiers that are mapped to Pfam domain.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spPMID   *Map protein identifier to PubMed Identifiers*

Description

org.Rn.spPMID maps protein identifiers to PubMed Identifiers.

Details

Each protein identifier is mapped to a vector of PubMed identifiers. The length of the vector may be one or greater, depending on how many PubMed identifiers a given manufacturer identifier is mapped to.

Titles, abstracts, and possibly full texts of articles can be obtained from PubMed by providing a valid PubMed identifier.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spPMID
# Get the protein identifiers that are mapped to PubMed Identifiers.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spPROSITE   *Map protein identifier to PROSITE domain*

Description

org.Rn.spPROSITE maps protein identifiers to PROSITE domain identifiers.

Details

Each protein identifier maps to a vector of PROSITE domain identifiers.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spPROSITE
# Get the protein identifiers that are mapped to PROSITE domain.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

`org.Rn.spREFSEQ` *Map protein identifier to RefSeq identifiers*

Description

org.Rn.spREFSEQ maps protein identifiers to RefSeq identifiers.

Details

Each protein identifier maps to RefSeq identifiers.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spREFSEQ
# Get the protein identifiers that are mapped to the RefSeq identifiers.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }

      ## For the reverse map org.Rn.spREFSEQ2SPID
xx <- as.list(org.Rn.spREFSEQ2SPID)
if(length(xx) > 0){
    ## Get the first one
    xx[[1]]
}
```

`org.Rn.spSEQ` *Map protein identifier to the protein sequence*

Description

org.Rn.spSEQ provides mappings between a protein identifier and the protein Sequence.

Details

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spSEQ
# Get the protein identifiers that are mapped to protein sequence.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
# randomly display 10 proteins
sample(xx, 10)
```

---

org.Rn.spSPAC   *Map protein identifier to SwissProt primary accession number*

Description

org.Rn.spSPAC maps protein identifiers to SwissProt primary accession number.

Details

Each protein identifier maps to SwissProt primary accession number.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/ uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
     x <- org.Rn.spSPAC
# Get the protein identifiers that are mapped to SwissProt primary accession number.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
     if(length(xx) > 0){
             # Get the value of the first key
             xx[[1]]
     }
```

---

org.Rn.spSPACs   *Map protein identifier to SwissProt accession numbers*

Description

org.Rn.spSPACs maps protein identifiers to SwissProt accession numbers.

Details

Each protein identifier maps to a vector of SwissProt accession numbers.

Mappings were based on data provided by: SwissProt (ftp://ftp.ebi.ac.uk/pub/databases/ uniprot/knowledgebase/uniprot_sprot.dat.gz) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spSPACs
# Get the protein identifiers that are mapped to SwissProt accession numbers.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spSUBCELL   *Map protein identifier to its subcellular location*

Description

org.Rn.spSUBCELL maps protein identifiers to its subcellular location.

Details

Each protein identifier maps to the description of the Subcellular Location of the chain/peptide/isoform. This information is extractd from "SUBCELLULAR LOCATION" topic in SwissProt "comments" lines.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spSUBCELL
# Get the protein identifiers that are mapped to subcellular location.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spSYMBOL   *Map protein identifier to gene symbols*

Description

org.Rn.spSYMBOL maps protein identifiers to gene symbols.

Details

Each protein identifier maps to an abbreviation for the corresponding gene. .

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/`
`uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release
56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spSYMBOL
# Get the protein identifiers that are mapped to gene symbols.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

---

org.Rn.spTISSUE   *Map protein identifier to tissue*

Description

org.Rn.spTISSUE maps protein identifiers to tissue.

Details

Each protein identifier maps to the description of the tissue-specific expression of mRNA or protein.
This information is extractd from "TISSUE SPECIFICITY" topic in UniProt "comments" lines.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/`
`uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release
56.9 of 03-Mar-2009

Examples

```
      x <- org.Rn.spTISSUE
# Get the protein identifiers that are mapped to tissue.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
      if(length(xx) > 0){
              # Get the value of the first key
              xx[[1]]
      }
```

org.Rn.spUNIGENE	*Map protein identifier to UniGene cluster identifiers*

Description

org.Rn.spUNIGENE maps protein identifiers to UniGene cluster identifiers.

Details

Each protein identifier maps to UniGene cluster identifiers. A UniGene identifier represents a cluster of sequences of a gene. Using UniGene identifiers one can query the UniGene database for information about the sequences or the Entrez Gene database for information about the genes.

Mappings were based on data provided by: SwissProt (`ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.dat.gz`) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009

Examples

```
x <- org.Rn.spUNIGENE
# Get the protein identifiers that are mapped to UniGene cluster identifiers.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
    if(length(xx) > 0){
            # Get the value of the first key
            xx[[1]]
    }
```

Index