

sc.dbsubloc.db

February 3, 2010

sc.dbsubloc.db *annotation data package*

Description

Welcome to the sc.dbsubloc.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 the protein subcellular localization from DBSubLoc database: <http://www.bioinfo.tsinghua.edu.cn/~guotao/data/dbsubloc2.dat.gz> 2005-8-17

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:sc.dbsubloc.db")
```

sc.dbsubloc_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

```
sc.dbsubloc_dbconn()  
sc.dbsubloc_dbfile()  
sc.dbsubloc_dbschema(file="", show.indices=FALSE)  
sc.dbsubloc_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

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

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

`sc.dbsubloc_dbschema` prints the schema definition of the package annotation DB.

`sc.dbsubloc_dbInfo` prints other information about the package annotation DB.

Examples

```
## Show the first three rows.
dbGetQuery(sc.dbsubloc_dbconn(), "select * from dbsubloc limit 3")

## The connection object returned by sc.dbsubloc_dbconn() was created with:
dbConnect(SQLite(), dbname=sc.dbsubloc_dbfile(), cache_size=64000, synchronous=0)

sc.dbsubloc_dbschema()

sc.dbsubloc_dbInfo()
```

sc.dbsublocDE *Maps SysPTM protein to textural descriptions*

Description

`sc.dbsublocDE` maps Swss-Prot protein accession number to their descriptive information.

Details

Mappings were based on data provided by: DBSubLoc (<http://www.bioinfo.tsinghua.edu.cn/~guotao/data/dbsubloc2.dat.gz>) on 2005-8-17

Examples

```
xx <- as.list(sc.dbsublocDE)
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]
}
}
```

```
sc.dbsublocMAPCOUNTS
```

Number of mapped keys for the maps in package sc.dbsubloc.db

Description

sc.dbsublocMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package sc.dbsubloc.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

```
sc.dbsublocMAPCOUNTS
mapnames <- names(sc.dbsublocMAPCOUNTS)
sc.dbsublocMAPCOUNTS[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 sc.dbsubloc.db
checkMAPCOUNTS("sc.dbsubloc.db")
```

```
sc.dbsublocORGANISM
```

Map SysPTM protein to organism

Description

sc.dbsublocORGANISM map Swss-Prot protein accession number to the name the organism.

Details

Mappings were based on data provided by: DBSubLoc (<http://www.bioinfo.tsinghua.edu.cn/~guotao/data/dbsubloc2.dat>) on 2005-8-17

Examples

```
# Convert to a list
xxx <- as.list(sc.dbsublocORGANISM)
# randomly display 10 proteins
sample(xxx, 10)
```

sc.dbsublocSEQ *Map protein to the protein sequence*

Description

sc.dbsublocSEQ provides mappings between Swss-Prot protein accession number and the protein Sequence.

Details

Mappings were based on data provided by: DBSubLoc (<http://www.bioinfo.tsinghua.edu.cn/~guotao/data/dbsubloc2.dat.gz>) on 2005-8-17

Examples

```
# Convert to a list
xx <- as.list(sc.dbsublocSEQ)
# Get the first one
xx[[1]]
```

sc.dbsublocSUBCELL *Map protein to the subcellular location*

Description

sc.dbsublocSUBCELL provides mappings between Swss-Prot protein accession number and the protein subcellular locations.

Details

Mappings were based on data provided by: DBSubLoc (<http://www.bioinfo.tsinghua.edu.cn/~guotao/data/dbsubloc2.dat.gz>) on 2005-8-17

Examples

```
## Convert to a list
xx <- as.list(sc.dbsublocSUBCELL)
## Get the first one
xx[[1]]

## For the reverse map sc.dbsublocSUBCELL2SPAC
## Convert to a list
xx <- as.list(sc.dbsublocSUBCELL2SPAC)
## Get the first one
xx[[1]]
```

Index

*Topic **datasets**

- `sc.dbsublocMAPCOUNTS`, 3
- `AnnDbObj`, 2
- `checkMAPCOUNTS`, 3
- `count.mappedkeys`, 3
- `dbDisconnect`, 2
- `mappedkeys`, 3
- `sc.dbsubloc.db`, 1
- `sc.dbsubloc_dbconn`, 1
- `sc.dbsubloc_dbfile`
 - (sc.dbsubloc_dbconn)*, 1
- `sc.dbsubloc_dbInfo`
 - (sc.dbsubloc_dbconn)*, 1
- `sc.dbsubloc_dbschema`
 - (sc.dbsubloc_dbconn)*, 1
- `sc.dbsublocDE`, 2
- `sc.dbsublocMAPCOUNTS`, 3
- `sc.dbsublocORGANISM`, 3
- `sc.dbsublocSEQ`, 4
- `sc.dbsublocSUBCELL`, 4
- `sc.dbsublocSUBCELL2SPAC`
 - (sc.dbsublocSUBCELL)*, 4