Package ‘MeSHDbi’

January 9, 2025

Title DBI to construct MeSH-related package from sqlite file

Description The package is unified implementation of MeSH.db, MeSH.AOR.db, and MeSH.PCR.db and also is interface to construct Gene-MeSH package (MeSH.XXX.eg.db). loadMeSHDbiPkg import sqlite file and generate MeSH.XXX.eg.db.

Version 1.42.0

Author Koki Tsuyuzaki

Maintainer Koki Tsuyuzaki <`k.t.the-answer@hotmail.co.jp`>

Depends R (>= 3.0.1)

Imports methods, AnnotationDbi (>= 1.31.19), RSQLite, Biobase

Suggests testthat

License Artistic-2.0

biocViews Annotation, AnnotationData, Infrastructure

git_url https://git.bioconductor.org/packages/MeSHDbi

git_branch RELEASE_3_20

git_last_commit 6355ff4

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-01-09

Contents

`MeSHDb-class` *MeSHDb objects*

Description

`MeSHDb` is the simple class for providing the relationship between Entrez gene IDs and MeSH IDs. It provides the database connection and easily accessible with `columns`, `keytypes`, `keys` and `select`. Some users may use additional functions such as `dbconn`, `dbfile`, `dbschema`, `dbInfo`, `species`, `nomenclature`, `listDatabases`, and `meshVersion` for much complex data acquisition.

`columns` shows which kinds of data can be returned for the `MeSHDb` object.

`keytypes` allows the user to discover which keytypes can be passed in to `select` or `keys` and the `keytype` argument.

`keys` returns keys for the database contained in the `MeSHDb` object . This method is already documented in the keys manual page but is mentioned again here because it's usage with `select` is so intimate. By default it will return the primary keys for the database, but if used with the `keytype` argument, it will return the keys from that keytype.

`select` will retrieve the data as a data.frame based on parameters for selected `keys`, `columns`, and `keytype` arguments.

`dbconn` returns the connection with database in the package.

`dbfile` returns the absolute path sqlite file is saved.

`dbschema` returns the database schema.

`dbInfo` returns the many meta information about the package.

`species` returns the species name.

`nomenclature` returns the scientific name.

`listDatabases` returns the list of databases to correspond Gene ID and MeSH ID.

`meshVersion` returns the version of MeSH.

Usage

```
columns(x)
keytypes(x)
keys(x, keytype, ...)
select(x, keys, columns, keytype, ...)
dbconn(x)
dbfile(x)
dbschema(x, file = "", show.indices = FALSE)
dbInfo(x)
species(object)
nomenclature(x)
listDatabases(x)
meshVersion(x)
```

Arguments

`x`	the `MeSHDb` object converted by `MeSHDbi::MeSHDb`.
`object`	same as x
`keys`	the keys to select records for from the database. All possible keys are returned by using the `keys` method.
`columns`	the columns or kinds of things that can be retrieved from the database. As with `keys`, all possible columns are returned by using the `columns` method.

`keytype`	the keytype that matches the keys used. For the `select` methods, this is used to indicate the kind of ID being used with the keys argument. For the `keys` method this is used to indicate which kind of keys are desired from `keys`
`...`	other arguments.
`file`	The `file` argument must be 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.

Value

`keys`, `columns`, `keytypes`, `dbfile`, `dbInfo`, `species`, and `nomenclature` each return a character vector or possible values. `select`, `dbschema`, `listDatabases`, and `meshVersion` each return a data.frame. dbconn returns database connection.

Author(s)

Koki Tsuyuzaki

See Also

`dbConnect`

Examples

```
# library("MeSHDbi")
# library("AnnotationHub")

# # Data retrieval from AnnotationHub
# ah <- AnnotationHub()
# dbfile <- query(ah, c("MeSHDb", "Bombyx mori", "v001"))[[1]]

# # Constructor
# MeSH.Sil.eg.db <- MeSHDbi::MeSHDb(dbfile)

# # show
# MeSH.Sil.eg.db

# # dbconn
# dbconn(MeSH.Sil.eg.db)

# # dbfile
# dbfile(MeSH.Sil.eg.db)

# # dbschema
# dbschema(MeSH.Sil.eg.db)

# # dbInfo
# dbInfo(MeSH.Sil.eg.db)

# # species
# species(MeSH.Sil.eg.db)

# # nomenclature
# nomenclature(MeSH.Sil.eg.db)
```

```
# # listDatabases
# listDatabases(MeSH.Sil.eg.db)

# # meshVersion
# meshVersion(MeSH.Sil.eg.db)

# # columns
# cols <- columns(MeSH.Sil.eg.db)

# # keytypes
# kys <- keytypes(MeSH.Sil.eg.db)

# # keys
# ks <- keys(MeSH.Sil.eg.db, keytype="GENEID")[seq(10)]

# # select
# out <- select(MeSH.Sil.eg.db,
#   columns=cols,
#   keys=ks,
#   keytype="GENEID")
```

Index