Package 'MeSHDbi'

October 12, 2016

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 GeneMeSH package (MeSH.XXX.eg.db). loadMeSHDbiPkg import sqlite file and generate MeSH.XXX.eg.db.

Version 1.8.0

Author Koki Tsuyuzaki

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

Depends R (>= 3.0.1), BiocGenerics (>= 0.15.10)

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

Suggests RUnit

License Artistic-2.0

biocViews Annotation, AnnotationData, Infrastructure

NeedsCompilation no

R topics documented:

listDatabases *A function to return the scientific name of package*

Description

This function returns the scientific name of species used in the package.

Usage

```
listDatabases(x)
```

Arguments

x	MeSHDb object such as MeSH.Mmu.eg.db

Author(s)

Koki Tsuyuzaki

Examples

```
# library("MeSH.Mmu.eg.db")
# listDatabases(MeSH.Mmu.eg.db)
```

makeGeneMeSHPackage *Making MeSHDb packages from corresponding table as single data frame.*

Description

makeGeneMeSHPackage is a method that generates a package that will load an appropriate MeSHDb object that will in turn point to existing annotation packages.

Usage

```
makeGeneMeSHPackage(pkgname,
data,
          metadata,
          organism,
version,
maintainer,
author,
destDir,
license="Artistic-2.0")
```

Arguments

pkgname	What is the desired package name.
data	Data frame contains GENEID (e.g., 100036770), MESHID (e.g.D000465), CATEGORY (e.g., G), SOURCEID (pubmed id), and SOURCEDB (e.g., gendoo)
metadata	Data frame contains metadata of the package
organism	The name of the organism this package represents
version	What is the version number for this package?
maintainer	Who is the package maintainer? (must include email to be valid)
author	Who is the creator of this package?
destDir	A path where the package source should be assembled.
license	What is the license (and it's version)

Details

The purpose of this method is to create a special package that will depend on existing annotation packages and which will load a special MeSHDb object that will allow proper dispatch of special select methods. These methods will allow the user to easily query across multiple annotation resources via information contained by the MeSHDb object. Because the end result will be a package that treats all the data mapped together as a single source, the user is encouraged to take extra care to ensure that the different packages used are from the same build etc.

Value

A special package to load an MeSHDb object.

Author(s)

Koki Tsuyuzaki

See Also

MeSHDb

Examples

```
## makeGeneMeSHPackage enable users to construct
## user's own custom MeSH package

## this is test data which means the relationship between
## Entrez gene IDs of Pseudomonas aeruginosa PAO1
## and its MeSH IDs.
data(PAO1)
head(PAO1)

# We are also needed to prepare meta data as follows.
data(metaPAO1)
metaPAO1
```

```
## sets up a temporary directory for this example
## (users won't need to do this step)
destination <- tempfile()
dir.create(destination)

## makes an Organism package for human called Homo.sapiens
makeGeneMeSHPackage(pkgname = "MeSH.Pae.eg.db",
data = PAO1,
          metadata = metaPAO1,
organism = "Pseudomonas aeruginosa PAO1",
version = "1.0.0",
maintainer = "Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>",
author = "Koki Tsuyuzaki",
destDir = destination,
license="Artistic-2.0")
```

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, and species 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.

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)
```

Arguments

x	the MeSHDb object. But in practice this will mean an object derived from an MeSHDb object such as a MeSH.Hsa.eg.db, MeSH.Mmu.eg.db or many other MeSH.XXX.eg.db (XXX means abbreviation of species name).
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, and species each return a character vector or possible values. select and dbschema each return a data.frame. dbconn returns database connection.

Author(s)

Koki Tsuyuzaki

See Also

dbConnect

Examples

```
# # load a package that creates an MeSHDb object
# library(MeSH.Mmu.eg.db)
# MeSH.Mmu.eg.db

# ## then the methods can be used on this object.
# cls <- columns(MeSH.Mmu.eg.db)
```

```
# cls
# kts <- keytypes(MeSH.Mmu.eg.db)
# kt <- kts[2]
# kts
# ks <- head(keys(MeSH.Mmu.eg.db, keytype=kts[2]))
# ks
# res <- select(MeSH.Mmu.eg.db, keys=ks, columns=cls, keytype=kt)
# head(res)

# dbconn(MeSH.Mmu.eg.db)
# dbfile(MeSH.Mmu.eg.db)
# dbschema(MeSH.Mmu.eg.db)
# dbInfo(MeSH.Mmu.eg.db)
# species(MeSH.Mmu.eg.db)
```

`meshVersion` *A function to return the MeSH version of package*

Description

This function returns the version of MeSH used in the package.

Usage

```
meshVersion(x)
```

Arguments

`x`	`MeSHDb` object such as `MeSH.Mmu.eg.db`

Author(s)

Koki Tsuyuzaki

Examples

```
# library("MeSH.Mmu.eg.db")
# meshVersion(MeSH.Mmu.eg.db)
```

metaPAO1 *Metadata to construct user's original MeSHDb package*

Description

Meta data to construct user's custom MeSHDb

Usage

```
data(metaPAO1)
```

Details

- SOURCEDATE: The date the source data is retrived
- SOURCENAME: Type of source data
- SOURCEURL: The URL of source data
- DBSCHEMA: Database schema
- DBSCHEMAVERSION: The version of database schema
- ORGANISM: The scientific name
- SPECIES: The common name of the species
- package: The pacakge name
- Db type: The type of name (or class name)

Examples

```
data(metaPAO1)
head(metaPAO1)
```

nomenclature *A function to return the scientific name of package*

Description

This function returns the scientific name of species used in the package.

Usage

```
nomenclature(x)
```

Arguments

x	MeSHDb object such as MeSH.Mmu.eg.db

Author(s)

Koki Tsuyuzaki

Examples

```
# library("MeSH.Mmu.eg.db")
# nomenclature(MeSH.Mmu.eg.db)
```

---

packageName *A function to return the name of package*

Description

This function returns the name of package

Usage

```
packageName(x)
```

Arguments

x	MeSHDb object such as MeSH.Mmu.eg.db

Author(s)

Koki Tsuyuzaki

Examples

```
# library("MeSH.Mmu.eg.db")
# packageName(MeSH.Mmu.eg.db)
```

PAO1 *Data to construct user's original MeSHDb package*

Description

Correspondance between Entrez gene IDs of Pseudomonas aeruginosa PAO1 and MeSH IDs is provided as demo data. This is to demonstrate how to construct user's original MeSHDb package. The data is based on reciprocal BLASTP best hit (E-value < 200) against Bacillus subtilis subsp. spizizenii str. 168.

Usage

```
data(PAO1)
```

Details

- 1st Column: Entrez gene ID
- 2nd Column: MeSH ID
- 3rd Column: MeSH Category
- 4th Column: Source ID. In this case, this is the Entrez gend ID of Bacillus subtilis subsp. spizizenii str. 168
- 5th Column: Source Database. In this case, species name.

Examples

```
data(PAO1)
head(PAO1)
```

Index