

Package ‘LRBaseDbi’

January 20, 2026

Title DBI to construct LRBase-related package

Description Interface to construct LRBase package (LRBase.XXX.eg.db).

Version 2.20.0

Author Koki Tsuyuzaki

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

Depends R (>= 3.5.0)

Imports methods, stats, utils, AnnotationDbi, RSQLite, DBI, Biobase

Suggests testthat, BiocStyle, AnnotationHub

VignetteBuilder utils

biocViews Infrastructure

License Artistic-2.0

git_url <https://git.bioconductor.org/packages/LRBaseDbi>

git_branch RELEASE_3_22

git_last_commit 5edc0aa

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-01-19

Contents

LRBaseDb-class 1

Index 5

LRBaseDb-class *LRBaseDb objects*

Description

LRBaseDb is the simple class for providing the relationship between Entrez gene IDs and LRBase 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`, `lrNomenclature`, `lrListDatabases`, and `lrVersion` for much complex data acquisition.

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

`lrNomenclature` returns the scientific name.

`lrListDatabases` returns the list of databases to correspond Gene ID of ligend gene and Gene ID of receptor gene.

`lrVersion` returns the version of LRBaseDb.

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)
lrNomenclature(x)
lrListDatabases(x)
lrVersion(x)
```

Arguments

<code>x</code>	the LRBaseDb object converted by <code>LRBaseDbi::LRBaseDb</code> .
<code>object</code>	same as <code>x</code>
<code>keys</code>	the keys to select records for from the database. All possible keys are returned by using the <code>keys</code> method.
<code>columns</code>	the columns or kinds of things that can be retrieved from the database. As with <code>keys</code> , all possible columns are returned by using the <code>columns</code> method.

keytype	the keytype that matches the keys used. For the <code>select</code> methods, this is used to indicate the kind of ID being used with the <code>keys</code> argument. For the <code>keys</code> method this is used to indicate which kind of keys are desired from <code>keys</code>
...	other arguments.
file	The <code>file</code> argument must be a connection, or a character string naming the file to print to (see the <code>file</code> argument of the <code>cat</code> function for the details).
show.indices	The CREATE INDEX statements are not shown by default. Use <code>show.indices=TRUE</code> to get them.

Value

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

Author(s)

Koki Tsuyuzaki

See Also

[dbConnect](#)

Examples

```
# library("LRBaseDbi")
# library("AnnotationHub")

# # Data retrieval from AnnotationHub
# ah <- AnnotationHub()
# dbfile <- query(ah, c("LRBaseDb", "Sus scrofa", "v001"))[[1]]

# # Constructor
# LRBase.Ssc.eg.db <- LRBaseDbi::LRBaseDb(dbfile)

# # show
# LRBase.Ssc.eg.db

# # dbconn
# dbconn(LRBase.Ssc.eg.db)

# # dbfile
# dbfile(LRBase.Ssc.eg.db)

# # dbschema
# dbschema(LRBase.Ssc.eg.db)

# # dbInfo
# dbInfo(LRBase.Ssc.eg.db)

# # species
# species(LRBase.Ssc.eg.db)

# # lrNomenclature
# lrNomenclature(LRBase.Ssc.eg.db)
```

```
# # lrListDatabases
# lrListDatabases(LRBase.Ssc.eg.db)

# # lrVersion
# lrVersion(LRBase.Ssc.eg.db)

# # columns
# cols <- columns(LRBase.Ssc.eg.db)

# # keytypes
# kts <- keytypes(LRBase.Ssc.eg.db)

# # keys
# ks <- keys(LRBase.Ssc.eg.db, keytype="GENEID_L")[seq(10)]

# # select
# out <- select(LRBase.Ssc.eg.db,
#                 columns=cols,
#                 keys=ks,
#                 keytype="GENEID_L")
```

Index

cat, [3](#)
class:LRBaseDb (LRBaseDb-class), [1](#)
columns (LRBaseDb-class), [1](#)
columns,LRBaseDb-method
 (LRBaseDb-class), [1](#)

dbconn (LRBaseDb-class), [1](#)
dbconn,LRBaseDb-method
 (LRBaseDb-class), [1](#)
dbConnect, [3](#)
dbfile (LRBaseDb-class), [1](#)
dbfile,LRBaseDb-method
 (LRBaseDb-class), [1](#)
dbInfo (LRBaseDb-class), [1](#)
dbInfo,LRBaseDb-method
 (LRBaseDb-class), [1](#)
dbschema (LRBaseDb-class), [1](#)
dbschema,LRBaseDb-method
 (LRBaseDb-class), [1](#)

keys (LRBaseDb-class), [1](#)
keys,LRBaseDb-method (LRBaseDb-class), [1](#)
keytypes (LRBaseDb-class), [1](#)
keytypes,LRBaseDb-method
 (LRBaseDb-class), [1](#)

LRBaseDb (LRBaseDb-class), [1](#)
LRBaseDb-class, [1](#)
lrListDatabases (LRBaseDb-class), [1](#)
lrListDatabases,LRBaseDb-method
 (LRBaseDb-class), [1](#)
lrNomenclature (LRBaseDb-class), [1](#)
lrNomenclature,LRBaseDb-method
 (LRBaseDb-class), [1](#)
lrVersion (LRBaseDb-class), [1](#)
lrVersion,LRBaseDb-method
 (LRBaseDb-class), [1](#)

select (LRBaseDb-class), [1](#)
select,LRBaseDb-method
 (LRBaseDb-class), [1](#)
species (LRBaseDb-class), [1](#)
species,LRBaseDb-method
 (LRBaseDb-class), [1](#)