Package: BridgeDbR (via r-universe)

November 17, 2026

Type Package

Title Code for using BridgeDb identifier mapping framework from within R

Version 2.22.0

Date 2025-08-02

Author Christ Leemans <christleemans@gmail.com>, Egon Willighagen <egon.willighagen@gmail.com>, Denise Slenter, Anwesha Bohler <anweshabohler@gmail.com>, Lars Eijssen <l.eijssen@maastrichtuniversity.nl>, Tooba Abbassi-Daloii

Maintainer Egon Willighagen <egon.willighagen@gmail.com>

Depends R (>= 3.3.0), rJava

Imports curl

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, testthat

Description Use BridgeDb functions and load identifier mapping databases in R. It uses GitHub, Zenodo, and Figshare if you use this package to download identifier mappings files.

biocViews Software, Annotation, Metabolomics, Cheminformatics

License AGPL-3

LazyLoad yes

URL https://github.com/bridgedb/BridgeDbR

BugReports https://github.com/bridgedb/BridgeDbR/issues

Config/pak/sysreqs make default-jdk libssl-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:40:07 UTC

RemoteUrl https://github.com/bioc/BridgeDbR

RemoteRef RELEASE_3_23

RemoteSha f66739c361d11b9c954d4add95d78a78330a1b0d

Contents

BridgeDbR-package . . . 2
fullNameExists . . . 2
getAttributes . . . 3
getBridgeNames . . . 4
getDatabase . . . 5
getDataSource . . . 5
getFullName . . . 6
getMatchingSources . . . 7
getOrganismCode . . . 7
getProperties . . . 8
getSystemCode . . . 9
loadDatabase . . . 9
map . . . 10
maps . . . 11
registerDataSource . . . 12
systemCodeExists . . . 12
xrefExists . . . 13

Index **14**

`BridgeDbR-package` *BridgeDbR Package.*

Description

R package for using BridgeDb directly from R.

Author(s)

Christ Leemans Anwesha Bohler Denise Slenter Lars Eijssen Tooba Abbassi-Daloii

Maintainer: Egon Willighagen <egon.willighagen@gmail.com>

`fullNameExists` *Determines if the full name corresponds to any known data source.*

Description

Data sources are defined by a title and a short system code. This method determines if a data source is known with the given full name.

Usage

```
fullNameExists(name)
```

Arguments

name	full name of the data source

Value

exists	boolean that indicates if a data is known with the given name

Author(s)

Egon Willighagen

Examples

```
exists <- fullNameExists("HMDB")
```

---

getAttributes *Returns the attributes of the given identifier.*

Description

Returns the attributes of the given identifier.

Usage

```
getAttributes(mapper, source, identifier)
```

Arguments

mapper	loaded BridgeDb identifier mapper
source	system code of the data source
identifier	identifier to return the attributes for

Value

mappings	a vector of attributes

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Homo sapiens")
mapper <- loadDatabase(location)
getAttributes(mapper, "L", "672")

## End(Not run)
```

`getBridgeNames` *Method to list the BridgeDb identifier mapping file names as available from the download site.*

Description

This method returns a list of file names of BridgeDb identifier mapping files as they are available from the mapping file download site. The optional code parameter can be used to restrict the list to a specific organism.

Occassionally, the ID mapping file download website changes the pattern of the file names. In that case, you can provide an alternative filename pattern to recognize files on the download site.

This method uses data from https://bridgedb.github.io/data/gene_database/.

Usage

```
getBridgeNames(code, website, pattern)
```

Arguments

`code`	the organism code
`website`	optional parameter that allows you to provide a different location to download BridgeDb Derby files from.
`pattern`	optional parameter to change the filename pattern of the BridgeDb Derby files.

Value

`list`	the list of available BridgeDb mapping files

Author(s)

Egon Willighagen Christ Leemans

Examples

```
files <- getBridgeNames("Hs")
## Not run:
prereleases <- getBridgeNames(
    website="http://bridgedb.org/data/gene_database/pre-release/",
    pattern="metabolites_\d*.bridge"
)

## End(Not run)
```

`getDatabase` *Downloads a identifier mapping database for the given organism.*

Description

Downloads a identifier mapping database for the given organism. The optional location parameter can be used to indicate where the file should be stored. If not give, it will save it in a temporary folder.

This method extracts links from https://bridgedb.github.io/data/gene_database/ to identifier mapping databases hosted on Zenodo and Figshare, two public data archives.

Usage

```
getDatabase(organism,location,website)
```

Arguments

`organism`	the organism name
`location`	optional parameter to indicate where the database should be saved
`website`	optional URL of the server where to look for BridgeDb identifier mapping databases

Value

`location`	the location where the database was saved

Author(s)

Egon Willighagen Christ Leemans

Examples

```
## Not run: location <- getDatabase("Bacillus subtilis")
## Not run: location <- getDatabase("Homo sapiens")
```

`getDataSource` *Return a DataSource object based on either a name, Bioregistry.io prefix, or system code.*

Description

Return a DataSource object based on either a name, Bioregistry.io prefix, or system code.

Usage

```
getDataSource(name,code,prefix)
```

Arguments

name	the name of the data source
code	the system code of the data source
prefix	the Bioregistry.io prefix of the data source

Value

dataSource	a DataSource Java object

Author(s)

Egon Willighagen

Examples

```
dataSource <- getDataSource(code="Ce")
```

---

getFullName *Return the name of a particular data source.*

Description

Data sources are defined by a title and a short system code. This method converts a system code into a full name.

Usage

```
getFullName(syscode)
```

Arguments

syscode	the system code of the data source

Value

systemCode	the full name of the data source

Author(s)

Egon Willighagen Christ Leemans

Examples

```
name <- getFullName("Ce")
```

`getMatchingSources` *Return a list of data sources of which the identifier pattern matches the identifier given.*

Description

This method runs through all data sources it knows about, and looks for sources of which the identifier pattern matches the given identifier.

Usage

```
getMatchingSources(identifier)
```

Arguments

`identifier`	identifier to test

Value

`list`	list of matching data sources

Author(s)

Egon Willighagen Christ Leemans

Examples

```
list <- getMatchingSources("555")
```

`getOrganismCode` *Return the code of a particular organism.*

Description

This method converts a organism latin name into a code.

Usage

```
getOrganismCode(name)
```

Arguments

`name`	latin name of an organism

Value

`systemCode`	the code of the organism

Author(s)

Egon Willighagen Christ Leemans

Examples

```
systemCode <- getOrganismCode("Mus musculus")
```

`getProperties` *BridgeDb database properties.*

Description

Gets the properties of an BridgeDb ID mapping database.

Usage

```
getProperties(mapper)
```

Arguments

`mapper`	loaded BridgeDb identifier mapper

Value

`properties`	a data frame with the properties

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Bacillus subtilis")
mapper <- loadDatabase(location)
getProperties(mapper)

## End(Not run)
```

`getSystemCode` *Return the system code of a particular data source.*

Description

Data sources are defined by a title and a short system code. This method converts a name into a system code.

Usage

```
getSystemCode(name)
```

Arguments

`name`	name of the data source

Value

`systemCode`	the system code of the data source

Author(s)

Egon Willighagen Christ Leemans

Examples

```
systemCode <- getSystemCode("ChEBI")
```

`loadDatabase` *Loads a BridgeDb identifier mapping database (.bridge file) as an IDMapper Java-Object.*

Description

Loads a BridgeDb identifier mapping database (.bridge file) as an IDMapper Java-Object.

Usage

```
loadDatabase(location)
```

Arguments

`location`	location on the hard disk of the BridgeDb file

Details

The return Java-Object is a Java class implementing the BridgeDb IDMapper interface. It is the required mapper class that is used as input for other methods that need an IDMapper.

Value

mapper	an IDMapper object

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Bacillus subtilis")
mapper <- loadDatabase(location)

## End(Not run)
```

---

map *Converts one identifier into other identifiers for the given target data source.*

Description

Converts one identifier into other identifiers for the given target data source.

Usage

```
map(mapper, source, identifier, target, compactIdentifier)
```

Arguments

mapper	loaded BridgeDb identifier mapper
source	system code of the data source
identifier	identifier to be converted
target	system code of the target data source (optional)
compactIdentifier	the Bioregistry.io compact identifier to convert

Value

mappings	a data frame of mapped identifiers for the target data source

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Bacillus subtilis")
mapper <- loadDatabase(location)
map(mapper, "En", "BSn5_00025", "S")
map(mapper, "En", "BSn5_00025")
location <- getDatabase("Homo sapiens")
mapper <- loadDatabase(location)
map(mapper, "En", "ENSG00000139618")
map(mapper, compactIdentifier="uniprot:P51587")

## End(Not run)
```

`maps` *Converts multiple identifiers into other identifiers for the given target data source.*

Description

Converts one identifier into other identifiers for the given target data source.

Usage

```
maps(mapper, identifiers, target)
```

Arguments

`mapper`	loaded BridgeDb identifier mapper
`identifiers`	data frame with identifiers to be converted, with a source and a identifier column
`target`	system code of the target data source (optional)

Value

`mappings`	a data frame of mapped identifiers for the target data source

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Bacillus subtilis")
mapper <- loadDatabase(location)
input = data.frame(
  source= c("L"),
  identifier= c("885041")
)
```

```
maps(mapper,input)

## End(Not run)
```

`registerDataSource` *Registers a new DataSource based on the given name or system code.*

Description

Return a DataSource of the newly registered data source, based on either a name or system code.

Usage

```
registerDataSource(code,name)
```

Arguments

`code`	the system code of the data source
`name`	the name of the data source

Value

`dataSource`	a DataSource Java object

Author(s)

Egon Willighagen

Examples

```
newDataSource <- registerDataSource(code="Cks", name="KNApSAcK")
```

`systemCodeExists` *Determines if the system code corresponds to any known data source.*

Description

Data sources are defined by a title and a short system code. This method determines if a data source is known with the given system code.

Usage

```
systemCodeExists(code)
```

Arguments

`code`	system code of the data source

Value

exists	boolean that indicates if a data is known with the given system code

Author(s)

Egon Willighagen

Examples

```
exists <- systemCodeExists("Ce")
```

---

xrefExists *Checks if the given identifier exists*

Description

Checks if the given identifier exists in the given mapping file

Usage

```
xrefExists(mapper, source, identifier, compactIdentifier)
```

Arguments

mapper	loaded BridgeDb identifier mapper
source	system code of the data source
identifier	identifier to be searched
compactIdentifier	the Bioregistry.io compact identifier to check

Value

boolean	true, if the given identifier is found in the mapping database

Author(s)

Egon Willighagen

Examples

```
## Not run:
location <- getDatabase("Bacillus subtilis")
mapper <- loadDatabase(location)
xrefExists(mapper, "L", "885041")
xrefExists(mapper, compactIdentifier="ncbigene:885041")

## End(Not run)
```

Index

∗ **BridgeDB**
BridgeDbR-package, 2
∗ **WikiPathways**
BridgeDbR-package, 2
∗ **identifier mapping**
BridgeDbR-package, 2
∗ **package**
BridgeDbR-package, 2

BridgeDbR *(*BridgeDbR-package*)*, 2
BridgeDbR-package, 2

fullNameExists, 2

getAttributes, 3
getBridgeNames, 4
getDatabase, 5
getDataSource, 5
getFullName, 6
getMatchingSources, 7
getOrganismCode, 7
getProperties, 8
getSystemCode, 9

loadDatabase, 9

map, 10
maps, 11

registerDataSource, 12

systemCodeExists, 12

xrefExists, 13