

# Package ‘BridgeDbR’

October 17, 2017

**Type** Package

**Title** Code for using BridgeDb identifier mapping framework from within R

**Version** 1.10.2

**Date** 2017-08-28

**Author** Christ Leemans <christleemans@gmail.com>, Egon Willighagen  
<egon.willighagen@gmail.com>, Anwasha Bohler  
<anweshabohler@gmail.com>, Lars Eijssen  
<l.eijssen@maastrichtuniversity.nl>

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

**Depends** R (>= 3.3.0), rJava

**Imports** RCurl

**Suggests** testthat

**Description** Use BridgeDb functions and load identifier mapping databases in R.

**biocViews** Software, Annotation

**License** AGPL-3

**LazyLoad** yes

**URL** <https://github.com/bridgedb/BridgeDb>,  
<https://github.com/BiGCAT-UM/bridgedb-r>

**BugReports** <https://github.com/BiGCAT-UM/bridgedb-r/issues>

**NeedsCompilation** no

## R topics documented:

BridgeDbR-package	2
fullNameExists	2
getBridgeNames	3
getDatabase	4
getDataSource	4
getFullName	5
getMatchingSources	5
getOrganismCode	6

getSystemCode . . . . .	7
loadDatabase . . . . .	7
map . . . . .	8
registerDataSource . . . . .	9
systemCodeExists . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

BridgeDbR-package	<i>BridgeDBR Package.</i>
-------------------	---------------------------

---

### Description

R package for using BridgeDB directly from R.

### Author(s)

Christ Leemans

Maintainer: Anwasha Bohler <anweshabohler@gmail.com>

---

fullNameExists	<i>Determines if the full name corresponds to any known data source.</i>
----------------	--

---

### 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")
```

---

getBridgeNames	<i>Method to list the BridgeDb identifier mapping file names as available from the download site.</i>
----------------	---

---

### 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.

Occasionally, 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.

### 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	<i>Downloads a identifier mapping database for the given organism.</i>
-------------	--

---

**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.

**Usage**

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

**Arguments**

organism	the organism name
location	optional parameter to indicate where the database should be saved

**Value**

location	the location where the database was saved
----------	---

**Author(s)**

Egon Willighagen Christ Leemans

**Examples**

```
location <- getDatabase("Mycobacterium tuberculosis")  
## Not run: location <- getDatabase("Homo sapiens")
```

---

getDataSource	<i>Return a DataSource object based on either a name or system code.</i>
---------------	--

---

**Description**

Return a DataSource object based on either a name or system code.

**Usage**

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

**Arguments**

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

**Value**

dataSource	a DataSource Java object
------------	--------------------------

**Author(s)**

Egon Willighagen

**Examples**

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

---

getFullName	<i>Return the name of a particular data source.</i>
-------------	---

---

**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	<i>Return a list of data sources of which the identifier pattern matches the identifier given.</i>
--------------------	--

---

**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")
```

---

getSystemCode	<i>Return the system code of a particular data source.</i>
---------------	--

---

**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	<i>Loads a BridgeDb identifier mapping database (.bridge file) as an IDMapper Java-Object.</i>
--------------	--

---

**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**

```
location <- getDatabase("Mycobacterium tuberculosis")
mapper <- loadDatabase(location)
```

---

map	<i>Converts one identifier into other identifiers for the given target data source.</i>
-----	---

---

**Description**

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

**Usage**

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

**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)

**Value**

mappings            a vector of mapped identifiers for the target data source

**Author(s)**

Egon Willighagen

**Examples**

```
location <- getDatabase("Mycobacterium tuberculosis")
mapper <- loadDatabase(location)
map(mapper, "L", "885041", "X")
map(mapper, "L", "885041")
```



---

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")
```

# Index

\*Topic **BridgeDB**

BridgeDbR-package, [2](#)

\*Topic **WikiPathways**

BridgeDbR-package, [2](#)

\*Topic **identifier mapping**

BridgeDbR-package, [2](#)

\*Topic **package**

BridgeDbR-package, [2](#)

BridgeDbR (BridgeDbR-package), [2](#)

BridgeDbR-package, [2](#)

fullNameExists, [2](#)

getBridgeNames, [3](#)

getDatabase, [4](#)

getDataSource, [4](#)

getFullName, [5](#)

getMatchingSources, [5](#)

getOrganismCode, [6](#)

getSystemCode, [7](#)

loadDatabase, [7](#)

map, [8](#)

registerDataSource, [9](#)

systemCodeExists, [9](#)