Package: brendaDb (via r-universe)

September 28, 2024

Type Package

Title The BRENDA Enzyme Database

Version 1.19.0

Description R interface for importing and analyzing enzyme information from the BRENDA database.

License MIT + file LICENSE

Encoding UTF-8

biocViews ThirdPartyClient, Annotation, DataImport

URL https://github.com/y1zhou/brendaDb

BugReports https://github.com/y1zhou/brendaDb/issues

Suggests testthat, BiocStyle, knitr, rmarkdown, devtools

Imports dplyr, Rcpp, tibble, stringr, magrittr, purrr, BiocParallel, crayon, utils, tidyr, grDevices, rlang, BiocFileCache, rappdirs

LinkingTo Rcpp

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

SystemRequirements C++11

VignetteBuilder knitr

Repository https://bioc.r-universe.dev

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

RemoteRef HEAD

RemoteSha 1d08ce5b5820150698d4f089474019e76bc55c06

Contents

`brendaDb-package` *brendaDb: the BRENDA enzyme database.*

Description

brendaDb provides an R interface to download, clean and extract enzyme information from the BRENDA database.

Details

The main aims of brendaDb include:

- Read text file downloaded from BRENDA into an R `tibble`
- Retrieve information for specific enzymes
- Query enzymes using their synonyms, gene symbols, etc.
- Query enzyme information for specific BioCyc pathways

To learn more about brendaDb, please refer to the vignette. `browseVignettes(package = "brendaDb")`

Author(s)

Maintainer: Yi Zhou <yi.zhou@uga.edu> (ORCID)

See Also

Useful links:

- https://github.com/y1zhou/brendaDb
- Report bugs at https://github.com/y1zhou/brendaDb/issues

DownloadBrenda — *Download and unzip the BRENDA text file.*

Description

By default, the function downloads a zipped BRENDA text file to a local cache directory, and extracts a `brenda_download.txt` file.

Usage

```
DownloadBrenda(force.download = FALSE)
```

Arguments

`force.download`	Boolean value. If TRUE, ignore the cache and force re-download of the BRENDA text file. Default is FALSE.

Value

A string of the path to the downloaded BRENDA text file.

Examples

```
## Not run: DownloadBrenda()
```

ExtractField — *Extract a specific field from a brenda.entries object.*

Description

Retrieve one field from all the brenda.entry objects. A column of EC numbers will be added to distinguish different enzymes.

Usage

```
ExtractField(res, field, entries = NULL)
```

Arguments

`res`	A brenda.entries object from `QueryBrenda()`.
`field`	A string indicating the field to extract. Nested fields should be separated by `$`, e.g. `parameters$ph.optimum`.
`entries`	A character vector with values of EC numbers. This should be a subset of `names(res)`.

Value

A tibble with all columns from the tibble in the given field, and extra columns containing the EC numbers and organisms.

Examples

```
df <- ReadBrenda(system.file("extdata", "brenda_download_test.txt",
                             package = "brendaDb"))
res <- QueryBrenda(brenda = df, EC = c("1.1.1.1", "6.3.5.8"),
                   n.core = 2)
ExtractField(res, field = "molecular$stability$general.stability")
ExtractField(res, field = "structure$subunits")
```

ID2Enzyme — *A helper function for converting names/synonyms to EC numbers.*

Description

A helper function for converting names/synonyms to EC numbers.

Usage

```
ID2Enzyme(brenda, ids)
```

Arguments

brenda	A `tibble` generated from `ReadBrenda()`.
ids	A character vector of IDs to be converted.

Details

The function goes through "RECOMMENDED_NAME", "SYSTEMATIC_NAME", and "SYNONYMS" in the BRENDA file, and uses regexes to look for the given IDs. Values in the three columns are kept if the regex had a hit, otherwise NA is filled. The function can take in IDs of multiple sources, e.g. `c("ADH4", "CD38", "pyruvate dehydrogenase")`. Note that using aliases instead of symbols could lead to false positives in the output table.

Value

A tibble with columns ID, EC, and at least one of (RECOMMENDED_NAME, SYSTEMATIC_NAME and SYNONYMS).

Examples

```
df <- ReadBrenda(system.file("extdata", "brenda_download_test.txt",
                             package = "brendaDb"))
ID2Enzyme(df, c("CD38", "ADH4", "pyruvate dehydrogenase"))
```

print.brenda.entries — *Show the number of regular and transferred/deleted brenda.entry objects in the brenda.entries list.*

Description

Show the number of regular and transferred/deleted brenda.entry objects in the brenda.entries list.

Usage

```
## S3 method for class 'brenda.entries'
print(x, ..., verbose = FALSE)
```

Arguments

x	A brenda.entries list returned by QueryBrenda().
...	Other arguments passed to the generic function.
verbose	Boolean; if TRUE, print tree views of brenda.query objects.

Value

Nothing; print summary information to the terminal.

print.brenda.entry — *Show the non-empty fields in the query result.*

Description

For details, see PrettyPrintBrendaEntry().

Usage

```
## S3 method for class 'brenda.entry'
print(x, full.output = FALSE, ...)
```

Arguments

x	A brenda.entry object (elements in the list returned by the function QueryBrenda()).
full.output	A boolean default to FALSE. If TRUE, include all entries even if they are empty (NA or 0 rows).
...	Other arguments passed to the generic function.

Value

Nothing; print object information to the terminal.

QueryBrenda *Query for multiple enzymes.*

Description

Use a vector of EC numbers to retrieve information from the BRENDA `tibble` read in by `ReadBrenda()`. Invalid EC numbers will be removed and a message will be generated.

Usage

```
QueryBrenda(brenda, EC, n.core = 0, fields = FALSE, ...)
```

Arguments

`brenda`	A `tibble` containing information from BRENDA.
`EC`	A string of the EC number.
`n.core`	Integer specifying the number of cores to use. Default is 0, which would result in using all available cores.
`fields`	A character vector indicating fields to parse. Default is FALSE, which would be returning all fields.
`...`	Other parameters passed to `QueryBrendaBase()`.

Value

A list of `brenda.entry` objects.

See Also

`QueryBrendaBase()` `ConfigBPCores()` `SelectOrganism()`

Examples

```
df <- ReadBrenda(system.file("extdata", "brenda_download_test.txt",
                             package = "brendaDb"))
res <- QueryBrenda(brenda = df, EC = c("1.1.1.1", "1.1.1.10", "6.3.5.8"),
                   n.core = 2, organisms = "Homo sapiens")
```

ReadBrenda — *Read BRENDA text file into matrix.*

Description

For each EC entry, split the annotations into three columns:

- ID: EC number, e.g. 1.1.1.1
- field: the content of the information, e.g. protein, localization
- description: everything else

Usage

```
ReadBrenda(filepath, clean = TRUE)
```

Arguments

`filepath`	A string indicating the path to the text file.
`clean`	Boolean; if TRUE, run `CleanECNumber()` after reading the file.

Value

A matrix containing information about the EC entries.

Examples

```
brenda_txt <- system.file("extdata", "brenda_download_test.txt",
                          package = "brendaDb")
df <- ReadBrenda(brenda_txt)
```

ReadBrendaFile — *Read raw BRENDA text file.*

Description

Read file into buffer, and load all non-empty lines. Comment lines (starting with *) are skipped. The text file should be downloaded from https://www.brenda-enzymes.org/download_brenda_without_registration.php

Usage

```
ReadBrendaFile(filepath)
```

Arguments

`filepath`	A string indicating the path to the text file.

Value

A vector with each element being a line in the file.

SeparateEntries *Convert vector of lines to matrix.*

Description

For each EC entry, split the annotations into three columns:

- ID: EC number, e.g. 1.1.1.1
- field: the content of the information, e.g. protein, localization
- description: everything else

Usage

```
SeparateEntries(lines)
```

Arguments

`lines`	The output vector from `read_brenda_file`.

Value

A vector<vector> containing information about the EC entries. In R this is a list of 3 lists.

ShowFields *Show all unique BRENDA fields and their corresponding acronyms.*

Description

Show all unique BRENDA fields and their corresponding acronyms.

Usage

```
ShowFields(df)
```

Arguments

`df`	A data.frame with columns "field" and "description"

Value

A data.frame with columns "field" and "acronym".

Examples

```
df <- ReadBrenda(system.file("extdata", "brenda_download_test.txt",
                             package = "brendaDb"))
ShowFields(df)
```

Index