Package: homologene (via r-universe)

October 27, 2024

Type Package

Title Quick Access to Homologene and Gene Annotation Updates

Version 1.7.68.23.10.31

Depends R (>= 3.1.2)

Imports dplyr (>= 0.7.4), magrittr (>= 1.5), purrr (>= 0.2.5), readr (>= 1.3.1), R.utils(>= 2.8.0), assertthat (>= 0.2.1), rvest (>= 1.0.0), xml2 (>= 1.3.2)

Suggests testthat (>= 1.0.2)

Date 2023-10-31

BugReports https://github.com/oganm/homologene/issues

URL https://github.com/oganm/homologene

Description A wrapper for the homologene database by the National Center for Biotechnology Information ('NCBI'). It allows searching for gene homologs across species. Data in this package can be found at <ftp://ftp.ncbi.nih.gov/pub/HomoloGene/build68/>. The package also includes an updated version of the homologene database where gene identifiers and symbols are replaced with their latest (at the time of submission) version and functions to fetch latest annotation data to keep updated.

License MIT + file LICENSE

LazyData true

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/oganm/homologene

RemoteRef HEAD

RemoteSha 9a9f99c4b596ccdd05a1ea1d7f62323bffb3b721

Contents

`autoTranslate` *Attempt to automatically translate a gene list*

Description

Given a list of query gene list and a target gene list, the function tries find the homology pairing that matches the query list to the target list. The query list is a short list of genes while the target list is supposed to represent a large number of genes from the target species. The default output will be the largest possible list. If `returnAllPossible = TRUE` then all possible pairings with any matches are returned. It is possible to limit the search by setting `possibleOrigins` and `possibleTargets`. Note that gene symbols of some species are more similar to each other than others. Using this with small gene lists and without providing any `possibleOrigins` or `possibleTargets` might return multiple hits, or if `returnAllPossible = TRUE` a wrong match can be returned.

Usage

```
autoTranslate(
  genes,
  targetGenes,
  possibleOrigins = NULL,
  possibleTargets = NULL,
  returnAllPossible = FALSE,
  db = homologene::homologeneData
)
```

Arguments

genes	A list of genes to match the target. Symbols or NCBI ids
targetGenes	The target list. This list is supposed to represent a large number of genes from the target species.
possibleOrigins	Taxonomic identifiers of possible origin species
possibleTargets	Taxonomic identifiers of possible target species
returnAllPossible	if TRUE returns all possible pairings with non zero gene matches. If FALSE (default) returns the best match
db	Homologene database to use.

Value

A data frame if `returnAllPossibe = FALSE` and a list of data frames if `TRUE`

---

diopt *Query DIOPT database*

Description

Query DIOPT database (https://www.flyrnai.org/cgi-bin/DRSC_orthologs.pl) for orthologues. DIOPT database uses multiple tools to find gene orthologues. Sadly they don't have an API so this function queries by visiting the site and filling up the form. By default each query will take a minimum of 10 seconds due to `delay` parameter. This is taken from their robots.txt at the time this function is written. Note that DIOPT is not necesariy in sync with homologene database as provided in this package.

Usage

```
diopt(genes, inTax, outTax, delay = 10)
```

Arguments

genes	A vector of gene identifiers. Anything that DIOPT accepts
inTax	taxid of the species that the input genes are coming from
outTax	taxid of the species that you are seeking homology. 0 to query all species.
delay	How many seconds of delay should be between queries. Default is 10 based on the robots.txt at the time this function is written.

Details

DIOPT does not support all species available in the homologene database. The supported species are:

4896 Schizosaccharomyces pombe

4932 Saccharomyces cerevisiae

6239 Caenorhabditis elegans

7227 Drosophila melanogaster

7955 Danio rerio

8364 Xenopus (Silurana) tropicalis

9606 Homo sapiens

10090 Mus musculus

10116 Rattus norvegicus

3702 Arabidopsis thaliana

Value

A data frame

`getGeneHistory` *Download gene history file*

Description

Downloads and reads the gene history file from NCBI website. This file is needed for other functions

Usage

```
getGeneHistory(destfile = NULL, justRead = FALSE)
```

Arguments

`destfile`	Path of the output file. If NULL a temp file will be used
`justRead`	If TRUE and destfile exists, it reads the file instead of downloading the latest one from NCBI

Value

A data frame with latest gene history information

getGeneInfo *Download gene symbol information*

Description

This function downloads the gene_info file from NCBI website and returns the gene symbols for current IDs.

Usage

```
getGeneInfo(destfile = NULL, justRead = FALSE, chunk_size = 1e+06)
```

Arguments

destfile	Path of the output file. If NULL a temp file will be used
justRead	If TRUE and destfile exists, it reads the file instead of downloading the latest one from NCBI
chunk_size	Chunk size to be used with link[readr]{read_tsv_chunked}. The gene_info file is big enough to make its intake difficult. If you don't have large amounts of free memory you may have to reduce this number to read the file in smaller chunks

Value

A data frame with gene symbols for each current gene id

getHomologene *Get the latest homologene file*

Description

This function downloads the latest homologene file from NCBI. Note that Homologene has not been updated since 2014 so the output will be identical to homologeneData included in this package. This function is here for futureproofing purposes.

Usage

```
getHomologene(destfile = NULL, justRead = FALSE)
```

Arguments

destfile	Path of the output file. If NULL a temp file will be used
justRead	If TRUE and destfile exists, it reads the file instead of downloading the latest one from NCBI

Value

A data frame with homology groups, gene ids and gene symbols

`homologene` *Get homologues of given genes*

Description

Given a list of genes and a taxid, returns a data frame inlcuding the genes and their corresponding homologues

Usage

```
homologene(genes, inTax, outTax, db = homologene::homologeneData)
```

Arguments

`genes`	A vector of gene symbols or NCBI ids
`inTax`	taxid of the species that the input genes are coming from
`outTax`	taxid of the species that you are seeking homology
`db`	Homologene database to use.

Examples

```
homologene(c('Eno2','17441'), inTax = 10090, outTax = 9606)
```

`homologeneData` *homologeneData*

Description

List of gene homologues used by homologene functions

Usage

```
homologeneData
```

Format

An object of class `data.frame` with 275237 rows and 4 columns.

homologeneData2 *homologeneData2*

Description

A modified copy of the homologene database. Homologene was updated at 2014 and many of its gene IDs and symbols are out of date. Here the IDs and symbols are replaced with their most current version Last update: Tue Oct 31 18:41:52 2023

Usage

```
homologeneData2
```

Format

An object of class `data.frame` with 266573 rows and 4 columns.

homologeneVersion *Version of homologene used*

Description

Version of homologene used

Usage

```
homologeneVersion
```

Format

An object of class `integer` of length 1.

human2mouse *Human/mouse wraper for homologene*

Description

Human/mouse wraper for homologene

Usage

```
human2mouse(genes, db = homologene::homologeneData)
```

Arguments

genes	A vector of gene symbols or NCBI ids
db	Homologene database to use.

Examples

```
human2mouse(c('ENO2','4340'))
```

mouse2human *Mouse/human wraper for homologene*

Description

Mouse/human wraper for homologene

Usage

```
mouse2human(genes, db = homologene::homologeneData)
```

Arguments

genes	A vector of gene symbols or NCBI ids
db	Homologene database to use.

Examples

```
mouse2human(c('Eno2','17441'))
```

taxData *Names and ids of included species*

Description

Names and ids of included species

Usage

```
taxData
```

Format

An object of class `data.frame` with 21 rows and 2 columns.

updateHomologene *Update homologene database*

Description

Creates an updated version of the homologene database. This is done by downloading the latest gene annotation information and tracing changes in gene symbols and identifiers over history. `homologeneData2` was created using this function over the original `homologeneData`. This function requires downloading large amounts of data from the NCBI ftp servers.

Usage

```
updateHomologene(
  destfile = NULL,
  baseline = homologene::homologeneData2,
  gene_history = NULL,
  gene_info = NULL
)
```

Arguments

`destfile`	Optional. Path of the output file.
`baseline`	The baseline homologene file to be used. By default uses the `homologeneData2` that is included in this package. The more ids to update, the more time is needed for the update which is why the default option uses an already updated version of the original database.
`gene_history`	A gene history data frame, possibly returned by `getGeneHistory` function. Use this if you want to have a static gene_history file to update up to a specific date. An up to date gene_history object can be set to update to a specific date by trimming rows that have recent dates. Note that the same is not possible for the gene_info If not provided, the latest file will be downloaded.

gene_info — A gene info data frame that contatins ID-symbol matches, possibly returned by getGeneInfo. Use this if you want a static version. Should be in sync with the gene_history file. Note that there is no easy way to track changes in gene symbols back in time so if you want to update it up to a specific date, make sure you don't lose that file.

Value

Homologene database in a data frame with updated gene IDs and symbols

---

updateIDs *Update gene IDs*

Description

Given a list of gene ids and gene history information, traces changes in the gene's name to get the latest valid ID

Usage

```
updateIDs(ids, gene_history)
```

Arguments

ids	Gene ids
gene_history	Gene history information, probably returned by getGeneHistory

Value

A character vector. New ids for genes that changed ids, or "-" for discontinued genes. the input itself.

Examples

```
## Not run:
gene_history = getGeneHistory()
updateIDs(c("4340964", "4349034", "4332470", "4334151", "4323831"),gene_history)

## End(Not run)
```

Index