Package ‘convertid’

January 9, 2026

Type Package

Title Convert Gene IDs Between Each Other and Fetch Annotations from Biomart

Version 0.1.12

Date 2026-01-09

Author Vidal Fey [aut, cre],
Henrik Edgren [aut]

Maintainer Vidal Fey <`vidal.fey@gmail.com`>

Description Gene Symbols or Ensembl Gene IDs are converted using the Bimap interface in 'AnnotationDbi' in convertId2() but
that function is only provided as fallback mechanism for the most common use cases in data analysis. The main function
in the package is convert.bm() which queries BioMart using the full capacity of the API provided through the
'biomaRt' package. Presets and defaults are provided for convenience but all ``marts'', ``filters'' and ``attributes''
can be set by the user. Function convert.alias() converts Gene Symbols to Aliases and vice versa and function likely_symbol()
attempts to determine the most likely current Gene Symbol.

Depends AnnotationDbi

Imports plyr, stringr, biomaRt, stats, xml2, utils, rappdirs,
assertthat, methods, httr, BiocFileCache

Suggests BiocManager, org.Hs.eg.db, org.Mm.eg.db

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-01-09 15:50:02 UTC

Contents

`.addToCache` *Add values to cache*

Description

Add values to cache

Usage

```
.addToCache(bfc, result, hash)
```

Arguments

`bfc`	Object of class BiocFileCache, created by a call to BiocFileCache::BiocFileCache()
`result`	character; name of the file written to chache
`hash`	unique hash representing a query.

`.checkInCache` *Check whether value in cache exists*

Description

Check whether value in cache exists

Usage

```
.checkInCache(bfc, hash, verbose = FALSE)
```

Arguments

`bfc`	Object of class BiocFileCache, created by a call to BiocFileCache::BiocFileCache()
`hash`	unique hash representing a query.
`verbose`	logical; should additional verbose output be printed? Not currently used. This function returns TRUE if a record with the requested hash already exists in the file cache, otherwise returns FALSE.

`.readFromCache` *Read values from cache*

Description

Read values from cache

Usage

```
.readFromCache(bfc, hash)
```

Arguments

`bfc`	Object of class BiocFileCache, created by a call to BiocFileCache::BiocFileCache()
`hash`	unique hash representing a query.

`convert.alias` *Convert Symbols to Aliases and Vice Versa.*

Description

`convert.alias()` attempts to find all possible symbol-alias combinations for a given gene symbol, i.e., it assumes the input ID to be either an Alias or a Symbol and performs multiple queries to find all possible counterparts. The input IDs are converted to title and upper case before querying and all possibilities are tested. There are species presets for Human and Mouse annotations.

Usage

```
convert.alias(id, species = c("Human", "Mouse"), db = NULL)
```

Arguments

`id`	(`character`). Vector of gene symbols.
`species`	(`character`). One of "Human" and "Mouse". Defaults to "Human".
`db`	(`AnnotationDb object`). Annotation package object.

Value

A `data.frame` with two columns:

’SYMBOL’: The official gene symbol.
’ALIAS’: All possible aliases.

See Also

`select`

Examples

```
convert.alias("TRPV4")
```

`convert.bm` *Retrieve Additional Annotations from Biomart*

Description

`convert.bm()` is a wrapper for `get.bm()` which in turn makes use of `getBM()` from the *biomaRt* package. It takes a matrix or data frame with the IDs to be converted in one column or as row names as input and returns a data frame with additional annotations after cleaning the fetched annotations and merging them with the input data frame.

Usage

```
convert.bm(
  dat,
  id = "ID",
  biom.data.set = c("human", "mouse"),
  biom.mart = c("ensembl", "mouse", "snp", "funcgen", "plants"),
  host = "https://www.ensembl.org",
  biom.filter = "ensembl_gene_id",
  biom.attributes = c("ensembl_gene_id", "hgnc_symbol", "description"),
  biom.cache = rappdirs::user_cache_dir("biomaRt"),
  use.cache = TRUE,
  sym.col = "hgnc_symbol",
  rm.dups = FALSE,
  verbose = FALSE
)
```

Arguments

`dat`	`matrix` or `data.frame`. Matrix or data frame with the ids to be converted in a column or as row names.
`id`	`character`. Name of the column with the ids to be converted, special name "rownames" will use the row names.
`biom.data.set`	`character` of length one. Biomart data set to use.
`biom.mart`	`character` vector. Biomart to use (uses the first element of the vector), defaults to "ensembl".
`host`	`character` of length one. Host URL.
`biom.filter`	`character` of length one. Name of biomart filter, i.e., type of query ids, defaults to "ensembl_gene_id".
`biom.attributes`	`character` vector. Biomart attributes, i.e., type of desired result(s); make sure query id type is included!
`biom.cache`	`character`. Path name giving the location of the cache `getBM()` uses if `use.cache=TRUE`. Defaults to the value in the *BIOMART_CACHE* environment variable.
`use.cache`	(`logical`). Should `getBM()` use the cache? Defaults to `TRUE` as in the `getBM()` function and is passed on to that.
`sym.col`	`character`. Name of the column in the query result with gene symbols.
`rm.dups`	`logical`. Should duplicated input IDs (‘`biom.filter`’) be removed from the result?
`verbose`	(`logical`). Should verbose output be written to the console? Defaults to `FALSE`.

Details

Wrapped around ‘get.bm‘.

Value

A data frame with the retrieved information.

Author(s)

Vidal Fey

See Also

`getBM`

Examples

```
## Not run:
dat <- data.frame(ID=c(”ENSG00000111199”, ”ENSG00000134121”, ”ENSG00000176102”, ”ENSG00000171611”))
bm <- convert.bm(dat)
bm

## End(Not run)
```

`convertId2` *Convert Gene Symbols to Ensembl Gene IDs or vice versa*

Description

`convertId2()` uses the Bimap interface in AnnotationDbi to extract information from annotation packages. The function is limited to Human and Mouse annotations and is provided only as fallback mechanism for the most common use cases in data analysis. Please use the Biomart interface function `convert.bm()` for more flexibility.

Usage

```
convertId2(id, species = c("Human", "Mouse"))
```

Arguments

`id`	(`character`). Vector of gene symbols.
`species`	(`character`). One of "Human" and "Mouse". Defaults to "Human".

Value

A named character vector where the input IDs are the names and the query results the values.

See Also

`Bimap-envirAPI`

Examples

```
convertId2("ENSG00000111199")
convertId2("TRPV4")
```

`get.bm` *Make a Query to Biomart.*

Description

`get.bm()` is a user-friendly wrapper for `getBM()` from the *biomaRt* package with default settings for Human and Mouse. It sets all needed variables and performs the query.

Usage

```
get.bm(
  values,
  biom.data.set = c("human", "mouse"),
  biom.mart = c("ensembl", "mouse", "snp", "funcgen", "plants"),
  host = "https://www.ensembl.org",
  biom.filter = "ensembl_gene_id",
  biom.attributes = c("ensembl_gene_id", "hgnc_symbol", "description"),
  biom.cache = rappdirs::user_cache_dir("biomaRt"),
  use.cache = TRUE,
  verbose = FALSE
)
```

Arguments

`values`	`character` vector of ids to be converted.
`biom.data.set`	`character` of length one. Biomart data set to use. Defaults to 'human' (internally translated to "hsapiens_gene_ensembl" if `biom.mart="ensembl"`).
`biom.mart`	`character` vector. Biomart to use (uses the first element of the vector), defaults to "ensembl".
`host`	`character` of length one. Host URL.
`biom.filter`	`character` of length one. Name of biomart filter, i.e., type of query ids, defaults to "ensembl_gene_id".
`biom.attributes`	`character` vector. Biomart attributes, i.e., type of desired result(s); make sure query id type is included!
`biom.cache`	`character`. Path name giving the location of the cache `getBM()` uses if `use.cache=TRUE`. Defaults to the value in the *BIOMART_CACHE* environment variable.
`use.cache`	(`logical`). Should `getBM()` use the cache? Defaults to `TRUE` as in the `getBM()` function and is passed on to that.
`verbose`	(`logical`). Should verbose output be written to the console? Defaults to `FALSE`.

Value

A data frame with the retrieved information.

Author(s)

Vidal Fey

See Also

`getBM`

Examples

```
## Not run:
val <- c("ENSG00000111199", "ENSG00000134121", "ENSG00000176102", "ENSG00000171611")
bm <- get.bm(val)
bm

## End(Not run)
```

---

likely_symbol

Retrieve Symbol Aliases and Previous symbols to determine a likely current symbol

Description

`likely_symbol()` downloads the latest version of the HGNC gene symbol database as a text file and query it to obtain symbol aliases, previous symbols and all symbols currently in use. (Optionally) assuming the input ID to be either an Alias or a Symbol or a Previous Symbol it performs multiple queries and compares the results of all possible combinations to determine a likely current Symbol.

Usage

```
likely_symbol(
  syms,
  alias_sym = TRUE,
  prev_sym = TRUE,
  orgnsm = "human",
  hgnc = NULL,
  hgnc_url = NULL,
  output = c("likely", "symbols", "all"),
  verbose = TRUE
)
```

Arguments

`syms`	(`character`). Vector of Gene Symbols to be tested.
`alias_sym`	(`logical`). Should the input be assumed to be an Alias? Defaults to `TRUE`.
`prev_sym`	(`logical`). Should the input be assumed to be a Previous Symbol? Defaults to `TRUE`.
`orgnsm`	(`character`). The organism for which the Symbols are tested.
`hgnc`	(`data.frame`). An optional data frame with the needed HGNC annotations. (Needs to match the format available at `hgnc_utl`!)
`hgnc_url`	(`character`). URL where to download the HGNC annotation dataset. Defaults to `"https://storage.googleapis.com/public-download-files/hgnc/tsv/tsv/hgnc_complete_s`

`output`	(`character`). One of "likely", "symbols" and "all". Determines the scope of the output data frame. Defaults to `"likely"` which will return the inout Symbol and the determined likely Symbol.
`verbose`	(`logical`). Should messages be written to the console? Defaults to `TRUE`.

Details

Please note that the algorithm is very slow for large input vectors.

Value

A `data.frame` with the following columns depending on the `output` setting. `output="likely"`:

’likely_symbol’
’input_symbol’

`output="symbols"`:

’current_symbols’
’likely_symbol’
’input_symbol’
’all_symbols’

`output="all"`:

’orig_input’
’organism’
’current_symbols’
’likely_symbol’
’input_symbol’
’all_symbols’

Note

Only fully implemented for Human for now.

Examples

```
## Not run:
likely_symbol(c("ABCC4", "ACPP", "KIAA1524"))

## End(Not run)
```

`todisp2` *Convenience Function to Convert Ensembl Gene IDs to Gene Symbols*

Description

`todisp2()` uses Biomart by employing `get.bm()` to retrieve Gene Symbols for a set of Ensembl Gene IDs. It is mainly meant as a fast way to convert IDs in standard gene expression analysis output to Symbols, e.g., for visualisation, which is why the input ID type is hard coded to ENSG IDs. If Biomart is not available the function can fall back to use `convertId2()` or a user-provided data frame with corresponding ENSG IDs and Symbols.

Usage

```
todisp2(ensg, lab = NULL, biomart = TRUE, verbose = FALSE)
```

Arguments

`ensg`	(`character`). Vector of Ensemble Gene IDs. Other ID types are not yet supported.
`lab`	(`data.frame`). A data frame with Ensembl Gene IDs as row names and Gene Symbols in the only column.
`biomart`	(`logical`). Should Biomart be used? Defaults to `TRUE`.
`verbose`	(`logical`). Should verbose output be written to the console? Defaults to `FALSE`.

Value

A character vector of Gene Symbols.

See Also

`get.bm`

Examples

```
## Not run:
val <- c("ENSG00000111199", "ENSG00000134121", "ENSG00000176102", "ENSG00000171611")
sym <- todisp2(val)
sym

## End(Not run)
```

Index