Package: ribiosAnnotation (via r-universe)

May 17, 2026

Type Package

Title Annotation of Genes, RNAs, and Proteins in 'ribios'

Version 3.8.0

Date 2026-02-15

Description Retrieves annotation information of genomic features including genes, RNAs, and proteins from databases. It supports querying by gene identifiers, gene symbols, UniProt accessions, Ensembl identifiers, and RefSeq identifiers, as well as mapping orthologs across species using NCBI data.

Depends R (>= 4.1.0)

Imports ribiosUtils, dplyr, tidyr, rjson, magrittr, mongolite

Suggests testthat

LazyData true

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Additional_repositories https://bedapub.r-universe.dev

Collate 'sortAnnotationByQuery.R' 'removeEnsemblVersion.R' 'utils.R' 'annotateAnyIDs.R' 'annotateHumanOrthologsWithNCBI.R' 'appendHumanOrthologsWithNCBI.R' 'annotateGeneIDs.R' 'annotateGeneSymbols.R' 'annotateProbesets.R' 'annotateProteinGroups.R' 'annotateTaxID.R' 'annotateUniprotAccession.R' 'ensembl.R' 'featureID.R' 'formatIn.R' 'gti2bioc.R' 'humanOrthologsByTaxID.R' 'ribiosAnnotation-package.R' 'taxID.R' 'uniprotByTaxID.R'

Remotes github::bedapub/ribiosUtils

Config/pak/sysreqs libicu-dev libssl-dev libsasl2-dev

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

Date/Publication 2026-02-15 13:42:58 UTC

RemoteUrl https://github.com/bedapub/ribiosAnnotation

RemoteRef HEAD

RemoteSha 331cc913da0be187571e8dfce47302950aeda7eb

Contents

annotateAnyIDs *Annotate any identifiers*

Description

This annotates any identifies that can be recognized by GTI.

Usage

```
annotateAnyIDs(ids, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

ids	A vector of identifiers. They must be of the same type. Supported types include Entrez GeneID, GeneSymbol, Probesets, UniProt identifiers, NCBI RefSeq mRNA identifiers, and Ensembl gene identifiers (with possible version suffixes).
orthologue	Logical, is orthologous mapping needed?
multiOrth	Logical, is more than one orthologs allowed

Value

A `data.frame` containing annotation information. Following columns exist at least:

1. `Input` Input string, it will be in the first column.
2. `IDType` Input ID type
3. `GeneID` (Human) Entrez GeneID
4. `GeneSymbol` (Human) official gene symbol
5. `GeneName` (Human) gene name
6. `TaxID` NCBI taxonomy ID

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

See Also

`annotateGeneIDs`, `annotateGeneSymbols`

Examples

```
## Not run:
# GeneID
annotateAnyIDs(ids=c(780, 5982, 3310, NA))

# GeneSymbol
annotateAnyIDs(ids=c("DDR1", "RFC2", "HSPA6", "HSAP6"))

# Probesets
myprobes <- c("1000_at", "1004_at", "1002_f_at", "nonsense_at")
annotateAnyIDs(myprobes)

# UniProt
annotateAnyIDs(ids=c("P38398", "Q8NDF8"))

# EnsEMBL
ensemblIDs <- c("ENSG00000197535", "ENST00000399231.7", "ENSP00000418960.2")
annotateAnyIDs(ensemblIDs)

# RefSeq
annotateAnyIDs(c("NM_000235", "NM_000498"))

## End(Not run)
```

`annotateEnsemblGeneIDs` *Annotate Enesembl GeneIDs*

Description

Annotate Enesembl GeneIDs

Usage

```
annotateEnsemblGeneIDs(ids, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

`ids`	A vector of EnsemblGeneIDs in form of `ENS(species)(object type)(identifier).(version)`. The version is optional.
`orthologue`	Logical, whether human orthologues should be returned. Default: `FALSE`
`multiOrth`	Logical, whether mutliple orthologues should be returned if exist. Deafult: `FALSE`

Value

A `data.frame` object containing the annotations: * GeneID EntrezGeneID * GeneSymbol Official gene symbol * Description Gene description * TaxID Taxonomy ID * Type Gene type

If `orthologue` is `TRUE`, following columns are appended: * HumanGeneID * HumanGeneSymbol * HumanDescription * HumanType

See Also

`annotateEnsemblGeneIDsWithoutHumanOrtholog` and `annotateEnsemblGeneIDsWithHumanOrtholog`

Examples

```
## Not run:
  annotateEnsemblGeneIDs(ids=c("ENSG00000236453", "ENSG00000170782",
                               "ENSG00000187867"))
  annotateEnsemblGeneIDs(ids=c("ENSG00000236453", "ENSG00000170782",
                               "ENSG00000187867", NA), orthologue=TRUE)
  annotateEnsemblGeneIDs(ids=c("ENSG00000174827", "ENSMUSG00000038298",
                               "ENSG00000198483", "ENSMUSG00000038354",
                               "ENSRNOG00000054947", "ENSG00000278099"),
                         orthologue=TRUE)

## End(Not run)
```

`annotateEnsemblGeneIDsWithEnsembl`
Annotate EnsEMBL GeneID with data from EnsEMBL

Description

Annotate EnsEMBL GeneID with data from EnsEMBL

Usage

```
annotateEnsemblGeneIDsWithEnsembl(ids)
```

Arguments

`ids`	Character strings, Ensembl GeneIDs in form of `ENS(species)(object type)(identifier).(version)` The version is optional.

Details

The `ensembl_genes` collection is used. Note that Ensembl IDs often refer to novel transcripts which do not have identifiers in other databases like NCBI Genes. If an EnsemblID is invalid or obsolete, the fields `GeneName` and `TaxID` will be NA.

Value

A `data.frame` containing following columns:

- EnsemblID: The input EnsemblID
- GeneID: NCBI GeneID
- GeneSymbol: Official gene symbol
- Description: Gene description
- TaxID: Taxonomy ID

This function uses data from EnsEMBL to annotate EnsEMBL GeneIDs. For most users, it is recommended to use `annotateEnsemblGeneIDs`, because it uses both data from EnsEMBL and data from NCBI to perform the task.

See Also

Function `annotateEnsemblGeneIDsWithNCBI` annotates EnsEMBL GeneIDs with data from NCBI, and `annotateEnsemblGeneIDs` annotates EnsEMBL GeneIDs with both data from EnsEMBL and data from NCBI.

Examples

```
## Not run:
ensIDs <- readLines(system.file(file.path("extdata/ribios_annotate_testdata",
                    "ensemble_geneids.txt"), package="ribiosAnnotation"))
ensAnno <- annotateEnsemblGeneIDsWithEnsembl(ensIDs)

## End(Not run)
```

---

annotateEnsemblGeneIDsWithHumanOrtholog

Annotate Ensembl GeneIDs while appending human orthologs

Description

Annotate Ensembl GeneIDs while appending human orthologs

Usage

```
annotateEnsemblGeneIDsWithHumanOrtholog(ids, multiOrth = FALSE)
```

Arguments

`ids`	A vector of character strings, Ensembl GeneIDs in form of `ENS(species)(object type)(identifier).(version)`. The version is optional.
`multiOrth`	Logical, whether mutliple orthologues should be returned if exist. Deafult: `FALSE`

Value

A `data.frame` containing following columns:

- EnsemblID: The input EnsemblID
- GeneID: NCBI GeneID
- GeneSymbol: Official gene symbol
- Description: Gene description
- TaxID: Taxonomy ID
- Type: Gene type
- HumanGeneID: NCBI GeneID of the human orthologue
- HumanGeneSymbol: Official gene symbol of the human orthologue
- HumanDescription: Gene description of the human orthologue
- HumanType: Gene type of the human orthologue

Note

Currently the human orthologs are looked up in NCBI. It remains to be changed to EnsEMBL

Examples

```
## Not run:
ensIDs <- readLines(system.file(file.path("extdata/ribios_annotate_testdata",
                    "ensemble_geneids.txt"), package="ribiosAnnotation"))
enAnnoHumanOrt <- annotateEnsemblGeneIDsWithHumanOrtholog(ensIDs)

## End(Not run)
```

`annotateEnsemblGeneIDsWithNCBI`
Annotate EnsEMBL GeneID with data from NCBI

Description

Annotate EnsEMBL GeneID with data from NCBI

Usage

```
annotateEnsemblGeneIDsWithNCBI(ids)
```

Arguments

`ids`	Character strings, Ensembl GeneIDs in form of `ENS(species)(object type)(identifier).(version)` The version is optional.

Details

The ncbi_gene2ensembl collection is used.

Value

A data.frame containing following columns:

- EnsemblID: The input EnsemblID
- GeneID: NCBI GeneID
- GeneSymbol: Official gene symbol
- Description: Gene description
- TaxID: Taxonomy ID
- Type: Gene type

This function uses data from NCBI to annotate EnsEMBL GeneIDs. For most users, it is recommended to use annotateEnsemblGeneIDs, because it uses both data from EnsEMBL and data from NCBI to perform the task.

See Also

Function annotateEnsemblGeneIDsWithEnsembl annotates EnsEMBL GeneIDs with data from Ensembl, and annotateEnsemblGeneIDs annotates EnsEMBL GeneIDs with both data from EnsEMBL and data from NCBI.

Examples

```
## Not run:
ensIDs <- readLines(system.file(file.path("extdata/ribios_annotate_testdata",
                    "ensemble_geneids.txt"), package="ribiosAnnotation"))
ncbiAnno <- annotateEnsemblGeneIDsWithNCBI(ensIDs)

## End(Not run)
```

---

annotateEnsemblGeneIDsWithoutHumanOrtholog

Annotate Ensembl GeneIDs with data from both EnsEMBL and NCBI

Description

Annotate Ensembl GeneIDs with data from both EnsEMBL and NCBI

Usage

```
annotateEnsemblGeneIDsWithoutHumanOrtholog(ids)
```

Arguments

ids	A vector of character strings, Ensembl GeneIDs in form of ENS(species)(object type)(identifier).(version). The version is optional.

Details

First, both EnsEMBL and NCBI annotation is queried. Next, we use the NCBI annotation as the template. Finally, we take the EnsEMBL annotation for those genes that are annotated by EnsEMBL but not by NCBI, merging the information from both sources.

Value

A data.frame containing following columns:

- EnsemblID: The input EnsemblID
- GeneID: NCBI GeneID
- GeneSymbol: Official gene symbol
- Description: Gene description
- TaxID: Taxonomy ID
- Type: Gene type

Examples

```
## Not run:
ensIDs <- readLines(system.file(file.path("extdata/ribios_annotate_testdata",
                    "ensemble_geneids.txt"), package="ribiosAnnotation"))
enAnno <- annotateEnsemblGeneIDsWithoutHumanOrtholog(ensIDs)

## End(Not run)
```

---

annotateGeneIDs *Annotate Entrez GeneIDs*

Description

Annotate Entrez GeneIDs

Usage

```
annotateGeneIDs(ids, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

ids	A vector of integers or characters, encoding NCBI Entrez GeneIDs. It can contain NA or NULL.
orthologue	Logical, whether human orthologues should be returned. Default: FALSE
multiOrth	Logical, whether mutliple orthologues should be returned if exist. Deafult: FALSE

Value

A data.frame object containing the annotations: * GeneID EntrezGeneID * GeneSymbol Official gene symbol * Description Gene description * TaxID Taxonomy ID * Type Gene type

If orthologue is TRUE, following columns are appended: * HumanGeneID * HumanGeneSymbol * HumanDescription * HumanType

See Also

annotateGeneIDsWithoutHumanOrtholog and annotateGeneIDsWithHumanOrtholog

Examples

```
## Not run:
  annotateGeneIDs(ids=c(780, 5982, 3310))
  annotateGeneIDs(ids=c(780, 5982, 3310, NA), orthologue=TRUE)
  annotateGeneIDs(ids=c(780, 1506, 1418,
                        114483548, 57300,
                        20, 1506, 102129055),
                        orthologue=TRUE)

## End(Not run)
```

---

annotateGeneIDsWithHumanOrtholog

Annotate Entrez GeneIDs with the query of human orthologs

Description

Annotate Entrez GeneIDs with the query of human orthologs

Usage

```
annotateGeneIDsWithHumanOrtholog(ids, multiOrth = FALSE)
```

Arguments

ids	Vector of integer or character strings, EntrezIDs to be annotated
multiOrth	Logical, whether mutliple orthologues should be returned if exist. Deafult: FALSE

Value

A data.frame object containing the annotations: * GeneID EntrezGeneID * GeneSymbol Official gene symbol * Description Gene description * TaxID Taxonomy ID * Type Gene type * HumanGeneID * HumanGeneSymbol * HumanDescription * HumanType

Examples

```
## Not run:
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310, NA))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310, NULL))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310, "1418"))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310, "NotValidGeneID"))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 3310, 1418, 5982))
  annotateGeneIDsWithHumanOrtholog(ids=c(780, 5982, 1418, 5982, 25120,
                        114483548, 57300, 20, 1506,
                        1545, 102129055))

## End(Not run)
```

---

annotateGeneIDsWithoutHumanOrtholog

Annotate Entrez GeneIDs without querying human orthologs

Description

Annotate Entrez GeneIDs without querying human orthologs

Usage

```
annotateGeneIDsWithoutHumanOrtholog(ids)
```

Arguments

ids	A vector of integers or characters, encoding NCBI Entrez GeneIDs. It can contain NA or NULL.

Details

The collection ncbi_gene_info is used.

Value

A data.frame object containing the annotations: * GeneID EntrezGeneID * GeneSymbol Official gene symbol * Description Gene description * TaxID Taxonomy ID * Type Gene type

Note

annotatemRNAs is an alias of annotateRefSeqs

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
## Not run:
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310, NA))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310, NULL))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310, "1418"))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310, "NotValidGeneID"))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 3310, 1418, 5982))
  annotateGeneIDsWithoutHumanOrtholog(ids=c(780, 5982, 1418, 5982, 25120,
                        114483548, 57300, 20, 1506,
                        1545, 102129055))

## End(Not run)
```

---

annotateGeneSymbols *Annotate GeneSymbols*

Description

Annotate GeneSymbols

Usage

```
annotateGeneSymbols(ids, taxId = 9606, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

ids	Character strings, gene symbols
taxId	Integer, NCBI taxonomy ID. Default value: 9606 (human). See commonSpecies for tax id of common species.
orthologue	Logical, whether orthologues are to be returned
multiOrth	Logical, only valid when orthologue is set to TRUE, whether multiple orthologues are returned

Value

A data.frame containing following columns

GeneID Entrez Gene ID

GeneSymbol Official gene symbols

Description Description

TaxID NCBI Taxonomy ID

Type Gene type

If orthologue is TRUE, then additional columns are appended:

HumanGeneID Human orthologue Entrez GeneID

HumanGeneSymbol Human orthologue official gene symbol

HumanDescription Human orthologue gene description

HumanType Human orthologue gene type

See Also

The function is a convenient wrapper of two functions: `annotateGeneSymbolsWithoutHumanOrtholog` and `annotateGeneSymbolsWithHumanOrtholog`.

Examples

```
## Not run:
    annotateGeneSymbols(c("AKT1", "ERBB2", "NoSuchAGene", "TGFBR1"), 9606)
    annotateGeneSymbols(c("Akt1", "Erbb2", "NoSuchAGene", "Tlr7"),
                        taxId=10116, orthologue=FALSE)
    annotateGeneSymbols(c("Akt1", "Erbb2", "NoSuchAGene", "Tlr7"),
                        taxId=10116, orthologue=TRUE)
    annotateGeneSymbols(c("Akt1", "Erbb2", "NoSuchAGene", "Tlr7"),
                        taxId=10116, orthologue=TRUE, multiOrth=TRUE)

## End(Not run)
```

---

annotateGeneSymbolsWithHumanOrtholog

Annotate GeneSymbol with human ortholog

Description

Annotate GeneSymbol with human ortholog

Usage

```
annotateGeneSymbolsWithHumanOrtholog(ids, taxId, multiOrth = FALSE)
```

Arguments

`ids`	Character strings, gene symbols
`taxId`	Integer, NCBI taxonomy ID. Default value: 9606 (human). See `commonSpecies` for tax id of common species.
`multiOrth`	Logical, only valid when orthologue is set to TRUE, whether multiple orthologues are returned

Value

A data.frame containing following columns:

GeneID Entrez Gene ID

GeneSymbol Official gene symbols

Description Description

TaxID NCBI Taxonomy ID

Type Gene type

HumanGeneID Human orthologue Entrez GeneID

HumanGeneSymbol Human orthologue official gene symbol

HumanDescription Human orthologue gene description

HumanType Human orthologue gene type

Examples

```
## Not run:
    annotateGeneSymbolsWithHumanOrtholog(c("Akt1", "Erbb2",
                                           "NoSuchAGene", "Tgfbr1"),
                                           taxId=10090, multiOrth=FALSE)

## End(Not run)
```

---

annotateGeneSymbolsWithoutHumanOrtholog

Annotate gene symbols without human ortholog

Description

Annotate gene symbols without human ortholog

Usage

```
annotateGeneSymbolsWithoutHumanOrtholog(ids, taxId = 9606)
```

Arguments

ids	Character vector, gene symbols to be queried
taxId	Integer, NCBI taxonomy ID of the species. Default value: 9606 (human). See commonSpecies for tax id of common species.

Value

A data.frame of following columns

GeneID Entrez Gene ID

GeneSymbol Official gene symbols

Description Description

TaxID NCBI Taxonomy ID

Type Gene type

Examples

```
## Not run:
    annotateGeneSymbolsWithoutHumanOrtholog(c("AKT1", "ERBB2",
                                              "NoSuchAGene", "TGFBR1"), 9606)
    annotateGeneSymbolsWithoutHumanOrtholog(c("Akt1", "Erbb2",
                                              "NoSuchAGene", "Tgfbr1"), 10090)

## End(Not run)
```

---

annotateHumanOrthologsWithNCBI

Annotate human orthologs with data from NCBI

Description

Annotate human orthologs with data from NCBI

Usage

```
annotateHumanOrthologsWithNCBI(geneids, multiOrth = FALSE)
```

Arguments

geneids	Integer GeneIDs, can contain human GeneIDs.
multiOrth	Logical, whether one gene is allowed to map to multiple human orthologs? Default value is FALSE, i.e. only the first human ortholog (random choice) is returned.

Details

ncbi_gene_info and ncbi_gene_orthologs collections are used.

Value

A data.frame containing following columns: * GeneID: Input GeneID * TaxID: Taxonomy ID of the input gene * HumanGeneID: Human Entrez GeneID

This function annotates human orthologs for any GeneID, including human genes, in which case the ortholog will be itself. Use annotateNonHumanGenesHumanOrthologsWithNCBI if you are sure that input GeneIDs do not come from human.

See Also

annotateNonHumanGenesHumanOrthologsWithNCBI

Examples

```
## Not run:
annotateHumanOrthologsWithNCBI(c(25120, 114483548, 57300, 20))
annotateHumanOrthologsWithNCBI(c(25120, 114483548, 57300, 20, 1506, 1545,
                                 102129055))

## End(Not run)
```

---

annotateNonHumanGenesHumanOrthologsWithNCBI

Annotate human orthologs with data from NCBI

Description

Annotate human orthologs with data from NCBI

Usage

```
annotateNonHumanGenesHumanOrthologsWithNCBI(geneids, multiOrth = FALSE)
```

Arguments

geneids	Integer GeneIDs, can contain human GeneIDs.
multiOrth	Logical, whether one gene is allowed to map to multiple human orthologs? Default value is FALSE, i.e. only the first human ortholog (random choice) is returned.

Details

ncbi_gene_info and ncbi_gene_orthologs collections are used.

Value

A data.frame containing following columns: * GeneID: Input GeneID * TaxID: Taxonomy ID of the input gene * HumanGeneID: Human Entrez GeneID

This function annotates human orthologs for any non-human genes. Use annotateHumanOrthologsWithNCBI if you are not sure whether all input GeneIDs are non-human.

See Also

annotateHumanOrthologsWithNCBI

Examples

```
## Not run:
annotateNonHumanGenesHumanOrthologsWithNCBI(c(25120, 114483548, 57300, 20))
annotateNonHumanGenesHumanOrthologsWithNCBI(c(25120, 114483548, 57300,
                                              20, 1506, 1545, 102129055))

## End(Not run)
```

---

annotateProteinGroups *Annotate protein groups for proteomics studies*

Description

Annotate protein groups for proteomics studies

Usage

```
annotateProteinGroups(
  ids,
  delimiter = ";",
  orthologue = FALSE,
  multiOrth = FALSE
)
```

Arguments

ids	Character, Protein groups with identifiers (e.g. UniProt/SwissProt IDs) by the delimiter
delimiter	Character, delimiter, default semicolon
orthologue	Logical, whether human orthologues should be queried.
multiOrth	Logical, in case of multiple orthologues, whether all of them should be returned The function queries proteins in protein groups, and annotate all proteins that cannot be annotated. For protein groups in which no protein can be annotated, all proteins will be returned as they are, without annotation

Value

A data.frame with following columns: * ProteinGroup * Protein * GeneID * GeneSymbol * GeneName * TaxID In case orthologue is TRUE, human orthologue information is returned as well.

Examples

```
## Not run:
annotateProteinGroups(c("A0A024RBG1;Q9NZJ9", "A0A0B4J2D5;P0DPI2",
                        "A0A0B4J2F0;A0A0U1RRL7"))

## End(Not run)
```

---

annotateTaxID *Annotations of all genes associated with the given TaxID*

Description

The function returns annotations (see details below) of all features (probably probesets) associated with the given taxon.

Usage

```
annotateTaxID(taxId, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

taxId	Integer, the TaxID of the species in interest. For instance '9606' for Homo sapiens.
orthologue	Logical, whether human orthologues should be returned
multiOrth	Logical, in case orthologue is set to TRUE, whether to return multiple orthologues for one gene. Default: FALSE

Details

The function reads from the backend, the MongoDB bioinfo database.

Value

A data.frame object with very similar structure as the EG_GENE_INFO table in the database. In case orthologue is TRUE, additional columns containing human orthologue information are returned.

Rownames of the data.frame are set to NULL.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
## Not run:
hsAnno <- annotateTaxID("9606")
dim(hsAnno)
head(hsAnno)

hsMtAnno <- annotateTaxID("10092")
dim(hsMtAnno)
head(hsMtAnno)

mtOrthAnno <- annotateTaxID(10090, orthologue=TRUE)
dim(mtOrthAnno)
head(mtOrthAnno)

pigMultiOrthAnno <- annotateTaxID(9823, orthologue=TRUE, multiOrth=TRUE)
dim(pigMultiOrthAnno)
head(pigMultiOrthAnno)

## End(Not run)
```

`annotateUniprotAccession`

Annotate UniProt accessions or names

Description

Annotate UniProt accessions or names

Usage

```
annotateUniprotAccession(accessions, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

`accessions`	Character strings, UniProt accessions or names
`orthologue`	Logical, whether orthologues are returned
`multiOrth`	Logical, only valid if `orthologue` is `TRUE`, whether multiple orthologues are returned instead of only one.

Examples

```
## Not run:
annotateUniprotAccession(c("B4E0K5"))

## End(Not run)
```

appendHumanOrthologsWithNCBI *Append human orthologs to an existing annotation dataframe*

Description

Append human orthologs to an existing annotation dataframe

Usage

```
appendHumanOrthologsWithNCBI(anno, multiOrth = FALSE)
```

Arguments

anno	A data.frame containing at least two columns GeneID and TaxID. The column GeneID stores Entrez GeneIDs, which are integers (NA values and characters are tolerated). The column TaxID stores taxonomy ID, which are integers, too (again, NA values and characters are tolerated).
multiOrth	Logical, whether one row is allowed to map to multiple orthologues
	The function appends human orthologs to an existing annotation data.frame. It is usually called by another function. Please make sure of what you are doing if you call it directly.

Value

A data.frame with annotation and human orthologs appended.

Note

The function does not sort the rows by GeneID. It is the responsibility of the calling function to do so.

Examples

```
## Not run:
anno <- data.frame(GeneID=c(780, 1506, 114483548, 102129055, NA),
                   TaxID=c(9606, 9606, 10116, 9541, NA))
appendHumanOrthologsWithNCBI(anno)

tol_anno <- data.frame(GeneID=c(780, 1506, 114483548, 102129055, NA, "NotV"),
                   TaxID=c(9606, 9606, 10116, 9541, NA, NA))
appendHumanOrthologsWithNCBI(tol_anno)

## End(Not run)
```

checkSingleIntegerTaxId *check single integer Tax ID*

Description

check single integer Tax ID

Usage

```
checkSingleIntegerTaxId(taxId)
```

Arguments

taxId	Integer tax identifier, or character that can be converted to an integer

Value

An integer tax ID if successful, otherwise the function stops and prints error

commonSpecies *Common species taxonomy IDs*

Description

Common species taxonomy IDs

Usage

```
commonSpecies
```

Format

A `data.frame` containing three columns:

TaxID NCBI taxonomy ID

ScientificName Scientific name

CommonName Common name

connectMongoDB *Connect to a MongoDB instance*

Description

Connect to a MongoDB instance

Usage

```
connectMongoDB(
  instance = "bioinfo_read",
  collection = "ncbi_gene_info",
  verbose = FALSE
)
```

Arguments

instance	Character string, the MongoDB instance to connect to
collection	Character string, the collection to be used
verbose	Logical

Value

A pointer to a collection on the server, as returned by mongo.

See Also

loadMongodbSecrets

Examples

```
## Not run:
giCon <- connectMongoDB(instance="bioinfo_read",
                        collection="ncbi_gene_info")

## End(Not run)
```

formatIn *Formatting a vector for SQL SELECT query with IN syntax*

Description

Prepare a vector for SQL SELECT query with the IN syntax

Usage

```
formatIn(x)
```

Arguments

x	A vector to be queried with the IN syntax

Value

A character string to be used after IN. See examples.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
myvec <- c("HH", "HM", "TH")
formatIn(myvec)
mysel <- "SELECT * FROM table WHERE city IN"
paste(mysel,formatIn(myvec))
```

getAllTaxIDs *Get all taxonomy ID and scientific names offered by NCBI*

Description

Get all taxonomy ID and scientific names offered by NCBI

Usage

```
getAllTaxIDs()
```

Value

A data.frame containing two columns, `TaxID` and `ScientificName`.

Examples

```
## Not run:
    all_tax_ids <- getAllTaxIDs()

## End(Not run)
```

---

gti2bioc *Translate chiptypes between GTI Bioconductor naming conventions*

Description

gti2bioc converts chip types from GTI array names into Bioconductor names, and bioc2gti converts Bioconductor array names to GTI names. If the array name is not valid or not found, NA will be returned.

Usage

```
gti2bioc(chipname)
```

Arguments

chipname	Character vector, chip names (types). If missing, chip types supported by both GTI and Bioconductor will be printed, see details.

Details

The translation table gtibioc was compiled manually in December 2011.

When the parameter 'chipname' is missing, chip types supported by both GTI and Bioconductor will be printed: gti2bioc returns a character vector of the Bioconductor names, and bioc2gti returns such a vector of the GTI names. Both vectors have the chip types in the other system as names. See examples.

Value

Chracter vector of the same length as the input

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
bioc2gti("hgu133plus2")
bioc2gti(c("hgu133plus2", "hgu95av2", "bad_array"))
gti2bioc("HG_U95AV2")
gti2bioc(c("HG_U95AV2", "CANINE", "HG_U95A"))

## supporting empty option
```

```
bioc2gti()
gti2bioc()
```

`gtibioc` *Translation table between GTI and Bioconductor chip type names*

Description

A data frame mapping GTI array names to Bioconductor array names.

Usage

```
gtibioc
```

Format

A data frame with columns:

GTI GTI chip type name

Bioconductor Bioconductor chip type name

Source

Compiled manually in December 2011.

`guessAndAnnotate` *Guess feature ID type by majority voting and annotate them*

Description

Guess feature ID type by majority voting and annotate them

Usage

```
guessAndAnnotate(
  featureIDs,
  majority = 0.5,
  orthologue = FALSE,
  multiOrth = FALSE,
  taxId = 9606
)
```

Arguments

featureIDs	A vector of character strings. Other input types will be converted to character strings.
majority	Numeric value between 0 and 1. If the proportion of valid feature IDs in the input matching the pattern of a certain feature type exceeds this value, the function returns a character string representing the feature ID type.
orthologue	Logical, whether orthologue should be returned if the input features are not of human
multiOrth	Logical, in case multiple human orthologues are available, should they all be returned?
taxId	Integer, in case the input identifiers are gene symbols, the user can specify the organism to be used with the NCBI taxonomy ID. The option is passed to annotateGeneSymbols. Check out commonSpecies for matches between taxonomy ID and species names.

Value

A data.frame, containing annotations of following ID types

- GeneID
- GeneSymbol
- RefSeq
- EnsemblGeneID
- Ensembl
- UniProt
- Unknown

. In case of Unknown, a data.frame with one column (FeatureName), containing input ids, is returned.

The difference between guessAndAnnotate and annotateAnyIDs is that the later does not assume that all IDs are of the same type.

See Also

annotateAnyIDs

Examples

```
## Not run:
  guessAndAnnotate(c("AKT1", "AKT2", "MAPK14"))
  guessAndAnnotate(c(1,2,14,149))
  guessAndAnnotate(c("NM_000259", "NM_000331"))
  guessAndAnnotate(c("ENST00000613858.4", "ENST00000553916.5",
       "ENST00000399229.6"))
  guessAndAnnotate(c("O60583", "P05997", "Q7Z624"))
  guessAndAnnotate(c("CM000677.2", "AB003434.2"))

## End(Not run)
```

guessFeatureType *Guess feature ID type by majority voting*

Description

Guess feature ID type by majority voting

Usage

```
guessFeatureType(featureIDs, majority = 0.5)
```

Arguments

featureIDs	A vector of character strings. Other input types will be converted to character strings.
majority	Numeric value between 0 and 1. If the proportion of valid feature IDs in the input matching the pattern of a certain feature type exceeds this value, the function returns a character string representing the feature ID type.

Value

A character string, one of the following values:

- GeneID
- GeneSymbol
- RefSeq
- EnsemblGeneID
- Ensembl
- UniProt
- Unknown

. The majority voting is done in the same order

Examples

```
guessFeatureType(c("AKT1", "AKT2", "MAPK14"))
guessFeatureType(c(1,2,14,149))
guessFeatureType(c("NM_000259", "NM_000331"))
guessFeatureType(c("ENST00000613858.4", "ENST00000553916.5",
    "ENST00000399229.6"))
guessFeatureType(c("A2BC19", "P12345", "A0A023GPI8"))
guessFeatureType(c("CM000677.2"))
```

`humanOrthologsByTaxID` *Retrieve human orthologs of genes of another species with its Taxonomy ID*

Description

Retrieve human orthologs of genes of another species with its Taxonomy ID

Usage

```
humanOrthologsByTaxID(taxid)
```

Arguments

`taxid`	An integer, a NCBI taxonomy ID to identify a species, for instance `10116` for rat, `10090` for mouse, and `9541` for cyno (crab-eating macaque).

Value

A `data.frame` contains following columns:

`GeneID` NCBI Gene ID of the query species

`GeneSymbol` NCBI Gene symbol of the query species

`Description` Gene description of the query species

`HumanGeneID` NCBI Gene ID of the human homolog

`HumanGeneSymbol` NCBI Gene symbol of the human homolog

`HumanDescription` Gene description of the human homolog

Note

To query NCBI taxonomy IDs from free-text search, visit [NCBI Taxonomy Browser](https://www.ncbi.nlm.nih.gov/Taxonor

Examples

```
## Not run:
## human orthologs of rat genes
ratOrths <- humanOrthologsByTaxID(10116)
## human orthologs of mouse genes
mouseOrths <- humanOrthologsByTaxID(10090)
## human orthologs of cyno genes (crab-eating macaque, Macaca fascicularis)
cynoOrths <- humanOrthologsByTaxID(9541)

## End(Not run)
```

isValidFeatureID *Whether input character strings are valid feature IDs*

Description

Whether input character strings are valid feature IDs

Usage

```
isValidFeatureID(featureIDs)
```

Arguments

featureIDs	A vector of character strings

Value

Logical vector of the same lenght as the input

Invalid feature IDs include NA, "-", and empty string. Other features are deedm as valid.

See Also

validFeatureIDs

Examples

```
featureIDs <- c("AMPK", "", "ACTB", "-")
isValidFeatureID(featureIDs)
```

likeGeneID *Whether input strings look like Entrez GeneIDs*

Description

Whether input strings look like Entrez GeneIDs

Usage

```
likeGeneID(featureIDs)

likeRefSeq(featureIDs)

likeEnsembl(featureIDs)

likeEnsemblGeneID(featureIDs)
```

```
likeUniProt(featureIDs)

likeGeneSymbol(featureIDs)

likeHumanGeneSymbol(featureIDs)
```

Arguments

featureIDs	Character strings. Input of other types are converted to them.

Value

A logical vector of the same length as input

Functions

- likeRefSeq(): tests whether input strings look like NCBI RefSeq IDs
- likeEnsembl(): tests whether input strings look like Ensembl IDs
- likeEnsemblGeneID(): tests whether input strings look like EnsemblGeneIDs
- likeUniProt(): tests whether input strings look like UniProt IDs
- likeGeneSymbol(): tests whether input strings look like gene symbols
- likeHumanGeneSymbol(): tests whether input strings look like human gene symbols

References

Regular expression of UniProt accesion numbers is available at https://www.uniprot.org/help/accession_numbers. We requirea a whole-string match additionally

The HGNC guideline is available at https://www.genenames.org/about/guidelines/

Examples

```
feats <- c("1234", "LOX", "345", "-", "", "NKX-1", "CXorf21", "Snail",
   "A2BC19", "P12345", "A0A023GPI8",
   "NM_000259", "NM_000259.3", "ENSG00000197535", "ENST00000399231.7")
likeGeneID(feats)
likeGeneSymbol(feats)
likeRefSeq(feats)
likeEnsembl(feats)
likeUniProt(feats)
likeHumanGeneSymbol(feats)
```

loadMongodbSecrets *Get secrets for MongoDB connections*

Description

Get secrets for MongoDB connections

Usage

```
loadMongodbSecrets(file = locateSecretsFile(), instance = "bioinfo_read")
```

Arguments

file	The secret JSON file.
instance	String, which must be found under the mongodb section of the JSON file

Value

A list of the following items:

hostname Hostname of the MongoDB

port Port of the MongoDB

dbname Database of the MongoDB

username User name

password Password

Examples

```
loadMongodbSecrets(instance="bioinfo_read")
## Not run:
    loadMongodbSecrets(instance="decoy")

## End(Not run)
```

majorityLikeHumanGeneSymbol *Guess the majority members of a character string look like human gene symbols*

Description

Guess the majority members of a character string look like human gene symbols

Usage

```
majorityLikeHumanGeneSymbol(x, majority = 0.8)
```

Arguments

x	A vector of character strings
majority	A numeric value between 0 and 1, the threshold of majority voting

Value

A logical value TRUE is only returned if at least a proportion of majority members look like human gene symbols

Examples

```
majorityLikeHumanGeneSymbol(c("AKT1", "AKT2", "MYOA")) # TRUE
majorityLikeHumanGeneSymbol(c("Akt1", "Akt2", "Myoa")) # FALSE
majorityLikeHumanGeneSymbol(c("AKT1", "Akt2", "MYOA"), majority=0.5) # TRUE
```

---

removeEnsemblVersion *Remove version suffix from Ensembl IDs*

Description

Remove version suffix from Ensembl IDs

Usage

```
removeEnsemblVersion(ensemblIDs)
```

Arguments

ensemblIDs	A vector of character strings. Other types of inputs are converted.

Value

A character vector of the same length as input

Examples

```
ensemblIDs <- c("ENSG00000197535", "ENST00000399231.7", "ENSP00000418960.2")
removeEnsemblVersion(ensemblIDs)
```

returnFieldsJson *Construct a JSON string to indicate returned fields from a MongoDB query*

Description

Construct a JSON string to indicate returned fields from a MongoDB query

Usage

```
returnFieldsJson(fields, include_id = FALSE)
```

Arguments

fields	A vector of character strings that should be included
include_id	Logical, whether _id should be returned. Default is FALSE

Value

A JSON string that represents the fields to be returned

Examples

```
returnFieldsJson(c("name", "birthday"))
returnFieldsJson(c("name", "birthday"), include_id=TRUE)
```

ribiosAnnotationSecretEnvVar *Locate ribiosAnnotation secrets file in JSON*

Description

ribiosAnnotation needs to access databases to fetch annotations, the process of which requires credentials for these databases. The package looks for a file in JSON format, either specified in environment variable RIBIOS_ANNOTATION_SECRETS_JSON, or in the file '~/.credentials/ribiosAnnotation-secrets.json', which contains the credentials. If this file is not found, no queries can be made.

Usage

```
ribiosAnnotationSecretEnvVar

locateSecretsFile(path)
```

Arguments

path	Path to the secret file. If not set, in case the environmental variable RIBIOS_ANNOTATION_SECRETS_JSON is set, its value is used as the file path; if not, '~/.credentials/ribiosAnnotation-secrets.json' is used. In any case, if the file does not exist, a message will be printed.

Format

An object of class character of length 1.

Details

The function locates the file and returns the normalized path of the file.

Value

String, the normalized path of the file

---

ribiosAnnotationSecretFile *ribiosAnnotation secret file*

Description

ribiosAnnotation secret file

Usage

```
ribiosAnnotationSecretFile
```

Format

An object of class character of length 1.

---

sortAnnotationByQuery *Sort the annotation table by query IDs*

Description

Sort the annotation table by query IDs

Usage

```
sortAnnotationByQuery(anno, ids, id_column = "GeneID", multi = FALSE)
```

Arguments

anno	A data.frame containing annotations
ids	A vector of character or integer, identifiers used to query the annotation
id_column	Character, column of the data frame where ids can be found.
multi	In case that an identifier appears more than once in id_column, should all rows be returned or only the first row? Default: FALSE, namely only the first row is returned.

Value

A data.frame sorted by the query identifiers, with the column id_column containing exactly the same value as ids. If the identifiers are unique and if they do not contain NA, they are used as the row names of the data.frame; otherwise, NULL will be used.

Examples

```
myAnno <- data.frame(GeneID=c(4,6,5), GeneName=c("Gene4", "Gene6", "Gene5"))
inputIds <- c("6", "5", "6", "4", "NotAGeneID")
sortAnnotationByQuery(myAnno, inputIds, "GeneID")

myAnno2 <- data.frame(GeneID=c(4,6,5, 5),
                      GeneName=c("Gene4", "Gene6", "Gene5", "Gene5.V2"))
inputIds <- c("6", "5", "6", "4", "NotAGeneID")
sortAnnotationByQuery(myAnno2, inputIds, "GeneID")
sortAnnotationByQuery(myAnno2, inputIds, "GeneID", multi=TRUE)
```

---

uniprotByTaxID *Get Uniprot annotation with NCBI Taxonomy ID*

Description

Get Uniprot annotation with NCBI Taxonomy ID

Usage

```
uniprotByTaxID(taxid, orthologue = FALSE, multiOrth = FALSE)
```

Arguments

taxid	NCBI Taxonomy ID
orthologue	Logical, whether human orthologues should be appended to the annotation
multiOrth	Logical, whether to return all orthologues or the (randomly) selected top one if multiple exist. Only valid when orthologue is set as TRUE.

Value

A data.frame with UniProt accessions and gene annotations.

See Also

* annotateUniprotAccession, which annotates Uniprot accessions * annotateTaxID, which annotates genes given TaxID.

Examples

```
## Not run:
humanUniprot <- uniprotByTaxID(9606)

## End(Not run)
```

---

validFeatureIDs *Return valid features in a vector*

Description

Return valid features in a vector

Usage

```
validFeatureIDs(featureIDs)
```

Arguments

featureIDs	A vector of character strings

Value

A filtered vector containing only valid feature IDs.

Factor input will remain factors as output, but with invalid levels dropped. The output class will remain the same in case of integer or character input.

See Also

isValidFeatureID

Examples

```
featureIDs <- c("AMPK", "", "ACTB", "-")
validFeatureIDs(featureIDs)
```

Index