Package 'UniProt.ws'

November 12, 2026

Type Package

Title R Interface to UniProt Web Services

Version 2.52.1

Description The Universal Protein Resource (UniProt) is a comprehensive resource for protein sequence and annotation data. This package provides a collection of functions for retrieving, processing, and re-packaging UniProt web services. The package makes use of UniProt's modernized REST API and allows mapping of identifiers accross different databases.

Depends R (>= 4.5.0)

Imports AnnotationDbi, BiocFileCache, BiocBaseUtils, BiocGenerics, httr2, jsonlite, methods, progress, rjsoncons, rlang, utils

Suggests BiocStyle, knitr, rmarkdown, tinytest

Collate UniProt.ws-package.R UniProt.ws-class.R mapUniProt.R UniProt.ws-methods.R utilities.R

License Artistic-2.0

biocViews Annotation, Infrastructure, GO, KEGG, BioCarta

BugReports https://github.com/Bioconductor/UniProt.ws/issues

URL https://github.com/Bioconductor/UniProt.ws

Encoding UTF-8

VignetteBuilder knitr

Date 2026-07-14

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

git_url https://git.bioconductor.org/packages/UniProt.ws

git_branch RELEASE_3_23

git_last_commit 3ca9213

git_last_commit_date 2026-07-14

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Marc Carlson [aut],
Csaba Ortutay [ctb],
Marcel Ramos [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3242-0582>)

Maintainer Marcel Ramos <marcel.ramos@sph.cuny.edu>

Contents

UniProt.ws-package 2
allFromKeys 3
UniProt.ws-class 6
UniProt.ws-methods 8
utilities 10

Index **12**

`UniProt.ws-package` *UniProt.ws: R Interface to UniProt Web Services*

Description

The Universal Protein Resource (UniProt) is a comprehensive resource for protein sequence and annotation data. This package provides a collection of functions for retrieving, processing, and re-packaging UniProt web services. The package makes use of UniProt's modernized REST API and allows mapping of identifiers accross different databases.

Author(s)

Maintainer: Marcel Ramos <marcel.ramos@sph.cuny.edu> (ORCID)

Authors:

- Marc Carlson <maintainer@bioconductor.org>

Other contributors:

- Csaba Ortutay [contributor]

See Also

Useful links:

- https://github.com/Bioconductor/UniProt.ws
- Report bugs at https://github.com/Bioconductor/UniProt.ws/issues

allFromKeys *Mapping identifiers with the UniProt API*

Description

These functions are the main workhorses for mapping identifiers from one database to another. They make use of the latest UniProt API (seen at https://www.uniprot.org/help/api).

Usage

```
allFromKeys()

allToKeys(fromName = "UniProtKB_AC-ID")

returnFields()

mapUniProt(
  from = "UniProtKB_AC-ID",
  to = "UniRef90",
  columns = character(0L),
  query,
  verbose = FALSE,
  debug = FALSE,
  paginate = TRUE,
  pageSize = 500L
)

queryUniProt(
  query = character(0L),
  fields = c("accession", "id"),
  collapse = c("OR", "AND"),
  n = Inf,
  pageSize = 25L
)
```

Arguments

fromName	character(1) A from key to use as the basis of mapping to other keys, by default, "UniProtKB_AC-ID".
from	character(1) The identifier type to map from, by default "UniProtKB_AC-ID", short for UniProt accession identifiers. See a list of all 'from' type identifiers with allFromKeys.
to	character(1) The target mapping identifier, by default "UniRef90". It can be any one of those returned by allToKeys from the appropriate fromName argument.

columns, fields — character() Additional information to be retreived from UniProt service. See a full list of possible input return fields at https://www.uniprot.org/help/return_fields. Example fields include, "accession", "id", "gene_names", "xref_pdb", "xref_hgnc", "sequence", etc.

query — character() or named list() Typically, a string of from= identifiers for ID mapping (mapUniProt). For the uniprotkb/search endpoint (queryUniProt), query can be a string of colon separated key-value pairs (e.g., "organism_id:9606") or a named list of available query fields (queryUniProt). See https://www.uniprot.org/help/query-fields for a list of query fields.

verbose — logical(1) Whether the operations should provide verbose updates (default FALSE).

debug — logical(1) Whether to display the URL API endpoints, for advanced debugging (default FALSE)

paginate — logical(1) Whether to use the pagination API (i.e., "results" vs "stream") in the request responses. For performance, it is set to TRUE by default.

pageSize — integer(1) number of records per page. It corresponds to the size parameter in the API request.

collapse — character(1) A string indicating either "OR" or "AND" for combining query clauses (case-insensitive).

n — numeric(1) Maximum number of rows to return

Details

Note that mapUniProt is used internally by the select method but made available for API queries with finer control. Provide values from the name column in returnFields as the columns input in either mapUniProt or select method.

When using from='Gene_Name', you may restrict the search results to a specific organism by including e.g., taxId=9606 in the query as a named list element. See examples below.

Value

- mapUniProt: A data.frame of returned results
- allToKeys: A sorted character vector of possible "To" keytypes based on the given "From" type
- allFromKeys: A sorted character vector of possible "From" keytypes
- returnFields: A data.frame of entries for the columns input in mapUniProt; see 'name' column

Author(s)

M. Ramos

Examples

```
mapUniProt(
    from="UniProtKB_AC-ID",
    to='RefSeq_Protein',
    query=c('P13368','Q9UM73','P97793','Q17192')
)

mapUniProt(
    from='GeneID', to='UniProtKB', query=c('1','2','3','9','10')
)

mapUniProt(
    from = "UniProtKB_AC-ID",
    to = "UniProtKB",
    columns = c("accession", "id"),
    query = list(organism_id = 10090, ids = c('Q7TPG8', 'P63318'))
)

## restrict 'from = Gene_Name' result to taxId 9606
mapUniProt(
    from = "Gene_Name",
    to = "UniProtKB-Swiss-Prot",
    columns = c("accession", "id"),
    query = list(taxId = 9606, ids = 'TP53')
)

mapUniProt(
    from = "UniProtKB_AC-ID", to = "UniProtKB",
    columns = c("accession", "id", "xref_pdb", "xref_hgnc", "sequence"),
    query = c("P31946", "P62258")
)

## query as character
queryUniProt(
    query = c("accession:A5YMT3", "organism_id:9606"),
    fields = c("accession", "id", "reviewed"),
    collapse = "AND"
)

## query as list
queryUniProt(
    query = list(organism_id = 9606, gene_exact = "A2M"),
    fields = c(
        "id", "accession", "gene_primary",
        "organism_name", "protein_name", "reviewed"
    ),
    collapse = "OR", n = 3, pageSize = 3
)

allToKeys(fromName = "UniRef100")

head(allFromKeys())
```

```
head(returnFields())
```

`UniProt.ws-class` *UniProt.ws objects and their related methods and functions*

Description

`UniProt.ws` is the base class for interacting with the UniProt web services from Bioconductor.

Usage

```
UniProt.ws(taxId = 9606, ...)

## S4 method for signature 'UniProt.ws'
show(object)

## S4 method for signature 'UniProt.ws'
taxId(x)

availableUniprotSpecies(pattern = "")

lookupUniprotSpeciesFromTaxId(taxId)

## S4 replacement method for signature 'UniProt.ws'
taxId(x) <- value

## S4 method for signature 'UniProt.ws'
species(object)
```

Arguments

`taxId`	`numeric(1)` a taxonomy identifier
`...`	other arguments
`x, object`	a `UniProt.ws` object.
`pattern`	character string containing a regular expression (or character string for `fixed = TRUE`) to be matched in the given character vector. Coerced by `as.character` to a character string if possible. If a character vector of length 2 or more is supplied, the first element is used with a warning. Missing values are allowed except for `regexpr`, `gregexpr` and `regexec`.
`value`	`numeric(1)` the new taxId to set

Details

`UniProt.ws` is a class that is used to interact with the UniProt web services. It makes use of `AnnotationDbi` methods similarly to `AnnotationDb` objects.

The `UniProt.ws` will be loaded whenever you load the `UniProt.ws` package. This object will be set up to retrieve information from Homo sapiens by default, but this value can be changed to any of the species supported by UniProt. The `species` and `taxId` methods allow users to see what species is currently being accessed, and `taxId<-` allows them to change this value.

`species` shows the genus and species label currently attached to the `UniProt.ws` objects database.

`taxId` shows the NCBI taxonomy ID currently attached to the `AnnotationDb` objects database. Using the equivalently names replace method (`taxId<-`) allows the user to change the taxon ID, and the species represented along with it.

`availableUniprotSpecies` is a helper function to list out the available Species along with their official taxonomy IDs that are available by UniProt. Because there are so many species represented at UniProt, there is also a pattern argument that can be used to restrict the range of things returned to be only those whose species names match the searth term. Please remember when using this argument that the Genus is always capitalized and the species never is.

`lookupUniprotSpeciesFromTaxId` is another helper that will look up the species of any tax ID that is supported by UniProt.

Value

- `species` and `lookupUniprotSpeciesFromTaxId` each return a character vector of possible values
  - `taxId` returns a numeric value that corresponds to the taxonomy ID
  - `availableUniprotSpecies` returns a `data.frame`

Functions

- `show(UniProt.ws)`: Show method for UniProt.ws objects
- `taxId(UniProt.ws)`: Get the taxonomy ID from a UniProt.ws object
- `taxId(UniProt.ws) <- value`: Set or chnage the taxonomy ID for a UniProt.ws object
- `species(UniProt.ws)`: Get the species name from a UniProt.ws object

Author(s)

Marc Carlson

See Also

UniProt.ws-methods

Examples

```
## Make a UniProt.ws object
up <- UniProt.ws(taxId=9606)

## look at the object
up

## get the current species
species(up)

## look up available species with their tax ids
availableUniprotSpecies("musculus")

## get the current taxId
taxId(up)

## look up the species that goes with a tax id
lookupUniprotSpeciesFromTaxId(9606)

## set the taxId to something else
taxId(up) <- 10090
up
```

UniProt.ws-methods *UniProt.ws methods from AnnotationDbi*

Description

Various methods from AnnotationDbi such as `select`, `columns`, `keys`, `keytypes`, and `species` are made available for UniProt.ws objects.

Usage

```
## S4 method for signature 'UniProt.ws'
keytypes(x)

## S4 method for signature 'UniProt.ws'
columns(x)

## S4 method for signature 'UniProt.ws'
keys(x, keytype)

## S4 method for signature 'UniProt.ws'
select(x, keys, columns, keytype, ...)
```

Arguments

x	a UniProt.ws object.
keytype	character(1) The keytype that matches the keys used. For the select methods, this is used to indicate the kind of ID being used with the keys argument. For the keys method this is used to indicate which kind of keys are desired from keys
keys	character() the keys to select records for from the database. All possible keys are returned by using the keys method.
columns	character() The columns or kinds of things that can be retrieved from the database. As with keys, all possible columns are returned by using the columns method.
...	Additional arguments passed to lower level functions, mainly used for the to input to mapUniProt.

Details

In much the same way as an AnnotationDb object allows acces to select for many other annotation packages, UniProt.ws is meant to allow usage of select methods and other supporting methods to enable the easy extraction of data from the UniProt web services.

select, columns and keys are used together to extract data via an UniProt.ws object.

columns shows which kinds of data can be returned for the UniProt.ws object.

keytypes allows the user to discover which keytypes can be passed in to select or keys via the keytype argument.

keys returns keys for the database contained in the UniProt.ws object . By default it will return the primary keys for the database, which are UniProtKB keys, but if used with the keytype argument, it will return the keys from that keytype.

select will retrieve the data as a data.frame based on parameters for selected keys and columns and keytype arguments.

Value

- keys,columns,keytypes, return a character vector of possible values
  - select returns a data.frame

Functions

- keytypes(UniProt.ws): Get keytypes for a UniProt.ws object
- columns(UniProt.ws):
- keys(UniProt.ws): Get keys for a UniProt.ws object
- select(UniProt.ws): Select columns from keys

See Also

UniProt.ws

Examples

```
## Make a UniProt.ws object
up <- UniProt.ws(taxId=9606)

## list the possible key types
head(keytypes(up))

## list of possible columns
head(columns(up))

## list all possible keys of type entrez gene ID
egs <- keys(up, "GeneID")

## use select to extract some data
res <- select(
    x = up,
    keys = c("22627","22629"),
    columns = c("xref_pdb","xref_hgnc","sequence"),
    keytype = "GeneID"
)
res

univals <- c("A0A0C5B5G6", "A0A1B0GTW7", "A0JNW5", "A0JP26", "A0PK11")
res <- select(
    x = up,
    keys = univals,
    to = "Ensembl"
)
res
```

`utilities` *Translate UniProt taxon names to scientific names, taxids, or domain codes*

Description

UniProt uses custom coding of organism names from which protein sequences they store. These taxon names are used also in the protein names (not in the UniProt IDs!). These functions help to translate those names to standard scientific (Latin) taxon names and other useful identifiers.

- taxname2species(): converts UniProt taxonomy names to scientific species names
- taxname2taxid(): converts UniProt taxonomy names to NCBI Taxonomy IDs
- taxname2domain(): converts UniProt taxonomy names to the following taxonomical domains:
  - 'A' for archaea (=archaebacteria)
  - 'B' for bacteria (=prokaryota or eubacteria)
  - 'E' for eukaryota (=eukarya)
  - 'V' for viruses and phages (=viridae)
  - 'O' for others (such as artificial sequences)

Usage

```
taxname2species(taxname, specfile)

taxname2taxid(taxname, specfile)

taxname2domain(taxname, specfile)
```

Arguments

taxname	Character string up to 6 uppercase characters, like HUMAN, MOUSE, or AERPX. Also works for a vector of such taxon names.
specfile	An optional local file where speclist.RData is saved from UniProt.org. When specfile is missing, a cached file from the extdata/ package directory is used.

Value

- taxname2species: a character vector of scientific taxon names matching to the UniProt taxon names supplied as taxname.
  - taxname2taxid: a numeric vector of Taxonomy IDs matching to the UniProt taxon names supplied as taxname.
  - taxname2domain: a character vector of one letter domain symbols matching to the UniProt taxon names supplied as taxname.

Author(s)

Csaba Ortutay

See Also

UniProt controlled vocabulary of species, which defines the taxon names.

Examples

```
taxname2species("PIG")
taxname2species(c("PIG","HUMAN","TRIHA"))

taxname2taxid("PIG")
taxname2taxid(c("PIG","HUMAN","TRIHA"))

taxname2domain("PIG")
taxname2domain(c("PIG","HUMAN","TRIHA"))
```

Index

∗ **internal**
    UniProt.ws-package, 2
.UniProt.ws *(*UniProt.ws-class*)*, 6

allFromKeys, 3
allToKeys *(*allFromKeys*)*, 3
as.character, *6*
availableUniprotSpecies
        *(*UniProt.ws-class*)*, 6

columns,UniProt.ws-method
        *(*UniProt.ws-methods*)*, 8

keys,UniProt.ws-method
        *(*UniProt.ws-methods*)*, 8
keytypes,UniProt.ws-method
        *(*UniProt.ws-methods*)*, 8

lookupUniprotSpeciesFromTaxId
        *(*UniProt.ws-class*)*, 6

mapUniProt *(*allFromKeys*)*, 3

queryUniProt *(*allFromKeys*)*, 3

regular expression, *6*
returnFields *(*allFromKeys*)*, 3

select,UniProt.ws-method
        *(*UniProt.ws-methods*)*, 8
show,UniProt.ws-method
        *(*UniProt.ws-class*)*, 6
species,UniProt.ws-method
        *(*UniProt.ws-class*)*, 6

taxId *(*UniProt.ws-class*)*, 6
taxId,UniProt.ws-method
        *(*UniProt.ws-class*)*, 6
taxId<- *(*UniProt.ws-class*)*, 6
taxId<-,UniProt.ws-method
        *(*UniProt.ws-class*)*, 6
taxname2domain *(*utilities*)*, 10
taxname2species *(*utilities*)*, 10
taxname2taxid *(*utilities*)*, 10

UniProt.ws, *9*
UniProt.ws *(*UniProt.ws-class*)*, 6
UniProt.ws-class, 6
UniProt.ws-methods, *7*, 8
UniProt.ws-package, 2
utilities, 10