Package 'UniProt.ws'

April 9, 2024

Type Package

Title R Interface to UniProt Web Services

Version 2.42.0

Depends BiocGenerics, methods, RSQLite, utils

Imports AnnotationDbi, BiocFileCache, BiocBaseUtils, httr, httpcache, jsonlite, progress, rjsoncons

Suggests BiocStyle, knitr, rmarkdown, RUnit

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.

Collate AllGenerics.R AllClasses.R getFunctions.R methods-select.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

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

git_branch RELEASE_3_18

git_last_commit 72fb35b

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-04-08

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

Maintainer Marcel Ramos <marcel.ramos@roswellpark.org>

R topics documented:

mapping-and-querying 2
UniProt.ws-objects 4
utilities 7

Index **9**

`mapping-and-querying` *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

```
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 = " OR ",
    n = Inf,
    pageSize = 25L
)
allToKeys(fromName = "UniProtKB_AC-ID")
allFromKeys()
returnFields()
```

Arguments

`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 that would indicate the target accession identifiers but can also be a named list based on the available query fields. See https://www.uniprot.org/help/query-fields for a list of query fields. The typical query might only include a character vector of UniProt accession identifiers, e.g., `c("A0A0C5B5G6", "A0A1B0GTW7", "A0JNW5", "A0JP26", "A0PK11", "A1A4S6")`

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

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

fromName — character(1) A `from` key to use as the basis of mapping to other keys, by default, `"UniProtKB_AC-ID"`.

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.

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

- mapUniProtA data.frame of returned results
- allToKeysA sorted character vector of possible "To" keytypes based on the given "From" type
- allFromKeysA sorted character vector of possible "From" keytypes
- returnFieldsA data.frame of entries for the columns input in `mapUniprot`; see 'name' column

Author(s)

Marcel 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",
    query = c("P31946", "P62258"),
    columns = c("accession", "id", "xref_pdb", "xref_hgnc", "sequence")
)

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

allToKeys(fromName = "UniRef100")

head(allFromKeys())

head(returnFields())
```

`UniProt.ws-objects` *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.

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.

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.

Usage

```
columns(x)
keytypes(x)
select(x, keys, columns, keytype, ...)
species(object)
taxId(x)

availableUniprotSpecies(pattern)
lookupUniprotSpeciesFromTaxId(taxId)
UniProt.ws(taxId, ...)
```

Arguments

x	the UniProt.ws object.
object	the UniProt.ws object.

keys	the keys to select records for from the database. All possible keys are returned by using the keys method.
columns	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.
keytype	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
pattern	A string passed in to limit the results
taxId	a taxonomy id
...	other arguments

Value

keys,columns,keytypes, species and lookupUniprotSpeciesFromTaxId each return a character vector of possible values.

taxId returns a numeric value that corresponds to the taxonomy ID.

select and availableUniprotSpecies each return a data.frame.

Author(s)

Marc Carlson

See Also

select

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
```

```
## 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` *Utility functions*

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)\
- updatespecfile(): The updatespecfile helper function attempts to download the current version of the controlled vocabulary of species table from UniProt controlled vocabulary of species. If it fails to download, an archived version of the table in (in extdata/) will be used.

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

Function taxname2species returns a character vector of scientific taxon names matching to the UniProt taxon names supplied as taxname.

Function taxname2taxid returns a numeric vector of Taxonomy IDs matching to the UniProt taxon names supplied as taxname.

Function taxname2domain returns a character vector of one letter domain symbols matching to the UniProt taxon names supplied as taxname.

Function updatespecfile is deprecated and no longer required as the specfile is self updating based on BiocFileCache's bfcneedsupdate when necessary.

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

∗ **classes**
    UniProt.ws-objects, 4
∗ **methods**
    UniProt.ws-objects, 4

allFromKeys *(*mapping-and-querying*)*, 2
allToKeys *(*mapping-and-querying*)*, 2
availableUniprotSpecies
        *(*UniProt.ws-objects*)*, 4

class:UniProt.ws *(*UniProt.ws-objects*)*, 4
cols *(*mapping-and-querying*)*, 2
columns *(*UniProt.ws-objects*)*, 4
columns,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4

keys *(*UniProt.ws-objects*)*, 4
keys,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4
keytypes *(*UniProt.ws-objects*)*, 4
keytypes,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4

lookupUniprotSpeciesFromTaxId
        *(*UniProt.ws-objects*)*, 4

mapping-and-querying, 2
mapUniProt *(*mapping-and-querying*)*, 2

queryUniProt *(*mapping-and-querying*)*, 2

returnFields *(*mapping-and-querying*)*, 2

select *(*UniProt.ws-objects*)*, 4
select,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4
show,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4
species *(*UniProt.ws-objects*)*, 4
species,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4

taxId *(*UniProt.ws-objects*)*, 4
taxId,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4
taxId<- *(*UniProt.ws-objects*)*, 4
taxId<-,UniProt.ws-method
        *(*UniProt.ws-objects*)*, 4
taxname2domain *(*utilities*)*, 7
taxname2species *(*utilities*)*, 7
taxname2taxid *(*utilities*)*, 7

UniProt.ws *(*UniProt.ws-objects*)*, 4
UniProt.ws-class *(*UniProt.ws-objects*)*, 4
UniProt.ws-objects, 4
utilities, 7