Package: mskcc.oncotree (via r-universe)

August 15, 2024

Type Package

Title Interface to the 'OncoTree' API

Version 0.1.1

Description Programmatic access to 'OncoTree' API <http://oncotree.mskcc.org/>. Get access to tumor main types, identifiers and utility routines to map across to other tumor classification systems.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Imports dplyr, glue, httr2, magrittr, memoise, purrr, readr, rlang, tibble, tidyjson, tidyr

URL https://maialab.org/mskcc.oncotree/

BugReports https://github.com/maialab/mskcc.oncotree/issues

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

RemoteUrl https://github.com/maialab/mskcc.oncotree

RemoteRef HEAD

RemoteSha 03f30b9886b00feaf81606cb083f069d2923f636

Contents

`get_tumor_types` *Get tumor types*

Description

Get tumor types according to OncoTree's ontology.

Usage

```
get_tumor_types(oncotree_version = "oncotree_latest_stable")
```

Arguments

`oncotree_version`
: OncoTree version. Check available options with `get_versions()`.

Value

A tibble of 13 variables:

`oncotree_version` OncoTree tumor classification system version.

`oncotree_code` Tumor type code: a unique identifier for a tumor type within the classification system of the OncoTree.

`oncotree_name` Tumor type name: a brief description of the tumor type.

`oncotree_main_type` Tumor main type: a category under which the tumor type can be grouped.

`tissue` Tissue associated with the tumor type.

`level` OncoTree is a hierachical classification system with 5 levels. At the root level (level 0) there is the single `"TISSUE"` tumor type. At level 1, there are 32 tissue sites, e.g., `"BREAST"`.

`parent` The `parent` is the parent `oncotree_code` for this tumor type.

`umls_code` The corresponding tumor type identifier(s) in the Unified Medical Language System (UMLS).

`nci_code` The corresponding tumor type identifier(s) in the National Cancer Institute (NCI) Thesaurus.

`history` Previous tumor type codes (from previous OncoTree versions) used to identify this tumor type.

`revocations` TODO.

`precursors` TODO.

`color` Color associated with the tumor type.

Examples

```
## Not run:
get_tumor_types()

## End(Not run)
```

get_versions *Get OncoTree versions*

Description

Get OncoTree versions

Usage

```
get_versions()
```

Value

A tibble of four variables:

`oncotree_version` OncoTree tumor classification system version.

`description` OncoTree release description.

`visible` A logical indicating whether this OncoTree version is visible, i.e. a forefront option at the website.

`release_date` OncoTree release date.

Examples

```
## Not run:
get_versions()

## End(Not run)
```

map_ontology_code *Map tumor types across ontologies*

Description

This function maps codes (identifiers) across tumor classification systems. Use the arguments `from` and `to` to choose the source and target ontologies. Available options are: `'oncotree_code'`, `'nci_code'`, `'umls_code'`, `'icdo_topography_code'`, `'icdo_morphology_code'`, and `'hemeonc_code'`.

Note that you can also use the functions `oncotree_to_nci()`, `nci_to_oncotree()`, `oncotree_to_umls()` and `umls_to_oncotree()` to map between OncoTree and NCIt systems. The difference is that these functions use the OncoTree API, and the output can be made to depend on older versions of OncoTree. `map_ontology_code()` relies on a static file provided by the OncoTree team that is not as up to date as the data provided by the web API. Nevetheless, the scope of the mappings provided by `map_ontology_code()` is broader. The file used by `map_ontology_code()` can be directly imported into R using the function `read_ontology_mappings()`.

Usage

```
map_ontology_code(code, from, to, collapse = NULL)
```

Arguments

code	A character vector with identifier codes of the from ontology that are meant to be mapped to the to ontology.
from	The source ontology. One of: 'oncotree_code', 'nci_code', 'umls_code', 'icdo_topography_code', 'icdo_morphology_code', and 'hemeonc_code'.
to	The target ontology. One of: 'oncotree_code', 'nci_code', 'umls_code', 'icdo_topography_code', 'icdo_morphology_code', and 'hemeonc_code'.
collapse	A function that expects one argument, it will be the character vector of codes in the to variable, that are to be "collapsed". When the mapping is one-to-many, passing a collapsing function will allow you to make the mapping one-to-one. See examples.

Value

A tibble of two variables: first column is corresponds to the from variable and the second is the to variable.

Source

The mappings here provided are based on the file https://github.com/cBioPortal/oncotree/blob/master/scripts/ontology_to_ontology_mapping_tool/ontology_mappings.txt.

See Also

oncotree_to_nci(), nci_to_oncotree(), oncotree_to_umls() and umls_to_oncotree().

Examples

```
## Not run:
# Omit the `code` argument to get all possible mappings. Note that
# one-to-many mappings will generate more than one row per `from` code.
map_ontology_code(from = 'oncotree_code', to = 'nci_code')

# Simple example
map_ontology_code('MMB', from = 'oncotree_code', to = 'nci_code')

# Some mappings are one-to-many, e.g. "SRCCR", which means repeated rows for
# the same input code.
map_ontology_code('SRCCR', from = 'oncotree_code', to = 'nci_code')

# Using the `collapse` argument to "collapse" one-to-many mappings makes sure
# that the output has as many rows as the `from` vector.
map_ontology_code('SRCCR',
                  from = 'oncotree_code',
                  to = 'nci_code',
                  collapse = toString)
```

```
map_ontology_code('SRCCR',
                  from = 'oncotree_code',
                  to = 'nci_code',
                  collapse = list)

map_ontology_code(
  'SRCCR',
  from = 'oncotree_code',
  to = 'nci_code',
  collapse = \(x) paste(x, collapse = ' ')
)

# `map_ontology_code()` is vectorized over `code`
map_ontology_code(
  c('AASTR', 'MDEP'),
  from = 'oncotree_code',
  to = 'nci_code'
  )

# Map from ICDO topography to ICDO morphology codes
map_ontology_code(
  'C72.9',
  from = 'icdo_topography_code',
  to = 'icdo_morphology_code'
  )

## End(Not run)
```

`nci_to_oncotree` *Map NCI to OncoTree codes*

Description

This function maps National Cancer Institute Thesaurus (NCIt) codes to OncoTree codes.

Usage

```
nci_to_oncotree(
  nci_code = NULL,
  oncotree_version = "oncotree_latest_stable",
  expand = FALSE
)
```

Arguments

`nci_code`	NCI codes.
`oncotree_version`	OncoTree database release version.
`expand`	Whether to expand one-to-many mappings. If `TRUE`, one-to-many mappings are expanded into several rows in the output.

Value

A tibble of two variables: nci_code and oncotree_code.

Examples

```
## Not run:
# Leave `nci_code` empty to return mappings for all NCI codes
nci_to_oncotree()

# Map a few selected OncoTree codes
nci_to_oncotree(nci_code = c('C8969', 'C4862', 'C9168', 'C7967'))

# Use `expand` to make sure the column `oncotree_code` is a character vector
# and not a list-column. One-to-many mappings will result in more than row
# with `oncotree_code` values repeated.
nci_to_oncotree(nci_code = c('C8969', 'C4862', 'C9168', 'C7967'), expand =
TRUE)

## End(Not run)
```

---

oncotree_to_nci *Map OncoTree to NCIt codes*

Description

This function maps OncoTree codes to National Cancer Institute Thesaurus (NCIt) codes.

Usage

```
oncotree_to_nci(
  oncotree_code = NULL,
  oncotree_version = "oncotree_latest_stable",
  expand = FALSE,
  keep_empty = TRUE
)
```

Arguments

oncotree_code	OncoTree codes.
oncotree_version	OncoTree database release version.
expand	Whether to expand one-to-many mappings. If TRUE, one-to-many mappings are expanded into several rows in the output.
keep_empty	OncoTree codes that do not map to NCI have the nci_code with NA if keep_empty = TRUE. Use keep_empty = FALSE, to remove the mapping (row) altogether from the output.

Value

A tibble of two variables: oncotree_code and nci_code.

Examples

```
## Not run:
# Leave `oncotree_code` empty to return mappings for all OncoTree codes
oncotree_to_nci()

# Map a few selected OncoTree codes
oncotree_to_nci(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'))

# Use `expand` to make sure the column `nci_code` is a character vector and
# not a list-column. One-to-many mappings will result in more than row with
# `oncotree_code` values repeated.
oncotree_to_nci(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'), expand
= TRUE)

# Use `keep_empty` to drop or keep one-to-none mappings
oncotree_to_nci(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'), expand
= TRUE, keep_empty = FALSE)

## End(Not run)
```

---

oncotree_to_umls *Map OncoTree to UMLS codes*

Description

This function maps OncoTree codes to Unified Medical Language System (UMLS) codes.

Usage

```
oncotree_to_umls(
  oncotree_code = NULL,
  oncotree_version = "oncotree_latest_stable",
  expand = FALSE,
  keep_empty = TRUE
)
```

Arguments

oncotree_code	OncoTree codes.
oncotree_version	OncoTree database release version.
expand	Whether to expand one-to-many mappings. If TRUE, one-to-many mappings are expanded into several rows in the output.

keep_empty — OncoTree codes that do not map to UMLS have the umls_code with NA if keep_empty = TRUE. Use keep_empty = FALSE, to remove the mapping (row) altogether from the output.

Value

A tibble of two variables: oncotree_code and umls_code.

Examples

```
## Not run:
# Leave `oncotree_code` empty to return mappings for all OncoTree codes
oncotree_to_umls()

# Map a few selected OncoTree codes
oncotree_to_umls(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'))

# Use `expand` to make sure the column `umls_code` is a character vector and
# not a list-column. One-to-many mappings will result in more than row with
# `oncotree_code` values repeated.
oncotree_to_umls(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'),
  expand = TRUE)

# Use `keep_empty` to drop or keep one-to-none mappings
oncotree_to_umls(oncotree_code = c('PAOS', 'SCST', 'ITLPDGI', 'SRCCR'),
  expand = TRUE, keep_empty = FALSE)

## End(Not run)
```

---

open_in_nci_thesaurus *Browse the NCIt*

Description

Opens the web browser at NCI Thesaurus for the entries provided as NCI codes.

Usage

```
open_in_nci_thesaurus(nci_code)
```

Arguments

nci_code — A character vector of NCI codes.

Value

Run for its side effect.

Examples

```
## Not run:
open_in_nci_thesaurus('C3107')

## End(Not run)
```

`umls_to_oncotree` *Map UMLS to OncoTree codes*

Description

This function maps Unified Medical Language System (UMLS) codes to OncoTree codes.

Usage

```
umls_to_oncotree(
  umls_code = NULL,
  oncotree_version = "oncotree_latest_stable",
  expand = FALSE
)
```

Arguments

`umls_code`	UMLS codes.
`oncotree_version`	OncoTree database release version.
`expand`	Whether to expand one-to-many mappings. If `TRUE`, one-to-many mappings are expanded into several rows in the output.

Value

A tibble of two variables: `umls_code` and `oncotree_code`.

Examples

```
## Not run:
# Leave `umls_code` empty to return mappings for all UMLS codes
umls_to_oncotree()

# Map a few selected OncoTree codes
umls_to_oncotree(umls_code = c('C0206642', 'C0600113', 'C0279654', 'C1707436'))

# Use `expand` to make sure the column `oncotree_code` is a character vector and
# not a list-column. One-to-many mappings will result in more than row with
# `oncotree_code` values repeated.
umls_to_oncotree(umls_code = c('C0206642', 'C0600113', 'C0279654', 'C1707436'), expand = TRUE)

## End(Not run)
```

Index