

# Package ‘mskcc.oncotree’

October 13, 2022

**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

**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>

**NeedsCompilation** no

**Author** Ramiro Magno [aut, cre] (<<https://orcid.org/0000-0001-5226-3441>>),  
Isabel Duarte [aut] (<<https://orcid.org/0000-0003-0060-2936>>),  
Ana-Teresa Maia [aut] (<<https://orcid.org/0000-0002-0454-9207>>),  
CINTESIS [cph, fnd]

**Maintainer** Ramiro Magno <ramiro.magno@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-10-11 18:10:06 UTC

## R topics documented:

get_tumor_types . . . . .	2
get_versions . . . . .	3
map_ontology_code . . . . .	3
nci_to_oncotree . . . . .	5
oncotree_to_nci . . . . .	6
oncotree_to_umls . . . . .	7
open_in_nci_thesaurus . . . . .	8
umls_to_oncotree . . . . .	9

**Index****10**


---

get_tumor_types	<i>Get tumor types</i>
-----------------	------------------------

---

**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 hierarchical 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	<i>Get OncoTree versions</i>
--------------	------------------------------

---

### 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	<i>Map tumor types across ontologies</i>
-------------------	--

---

### 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 NCI 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. Nevertheless, 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](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 = 'ontotree_code',
                  to = 'nci_code',
                  collapse = list)

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

# `map_ontology_code()` is vectorized over `code`
map_ontology_code(
  c('AASTR', 'MDEP'),
  from = 'ontotree_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\_ontotree      *Map NCI to OncoTree codes*

---

## Description

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

## Usage

```

nci_to_ontotree(
  nci_code = NULL,
  ontotree_version = "ontotree_latest_stable",
  expand = FALSE
)

```

## Arguments

nci_code	NCI codes.
ontotree_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 NCI 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**

<code>oncotree_code</code>	OncoTree codes.
<code>oncotree_version</code>	OncoTree database release version.
<code>expand</code>	Whether to expand one-to-many mappings. If TRUE, one-to-many mappings are expanded into several rows in the output.
<code>keep_empty</code>	OncoTree codes that do not map to NCI have the <code>nci_code</code> with NA if <code>keep_empty = TRUE</code> . Use <code>keep_empty = FALSE</code> , 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	<i>Map OncoTree to UMLS codes</i>
------------------	-----------------------------------

---

**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**

<code>oncotree_code</code>	OncoTree codes.
<code>oncotree_version</code>	OncoTree database release version.
<code>expand</code>	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 NCI*

---

### 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_ontotree	<i>Map UMLS to OncoTree codes</i>
------------------	-----------------------------------

---

**Description**

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

**Usage**

```
umls_to_ontotree(
  umls_code = NULL,
  ontotree_version = "ontotree_latest_stable",
  expand = FALSE
)
```

**Arguments**

umls_code	UMLS codes.
ontotree_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 ontotree\_code.

**Examples**

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

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

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

## End(Not run)
```

# Index

`get_tumor_types`, 2  
`get_versions`, 3  
`get_versions()`, 2  
  
`map_ontology_code`, 3  
  
`nci_to_ontotree`, 5  
`nci_to_ontotree()`, 3, 4  
  
`ontotree_to_nci`, 6  
`ontotree_to_nci()`, 3, 4  
`ontotree_to_umls`, 7  
`ontotree_to_umls()`, 3, 4  
`open_in_nci_thesaurus`, 8  
  
`tibble`, 2–4, 6–9  
  
`umls_to_ontotree`, 9  
`umls_to_ontotree()`, 3, 4