Package 'rotl'

July 24, 2015

Title Interface to the 'Open Tree of Life' API

Version 0.4.1

Description An interface to the 'Open Tree of Life' API to retrieve phylogenetic trees, information about studies used to assemble the synthetic tree, and utilities to match taxonomic names to 'Open Tree identifiers'. The 'Open Tree of Life' aims at assembling a comprehensive phylogenetic tree for all named species.

URL https://github.com/ropensci/rotl

BugReports https://github.com/ropensci/rotl/issues

Depends R (>= 3.1.1)

Imports httr, jsonlite, rncl (>= 0.6.0), ape

License BSD_2_clause + file LICENSE

Suggests knitr (>= 1.10.5), rmarkdown (>= 0.7), testthat, RNeXML, phylobase

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Francois Michonneau [aut, cre], Joseph Brown [aut], David Winter [aut]

Maintainer Francois Michonneau <francois.michonneau@gmail.com>

Repository CRAN

Date/Publication 2015-07-24 18:21:38

R topics documented:

`get_study` *Get Study*

Description

Returns a study for given ID

Usage

```
get_study(study_id = NULL, object_format = c("phylo", "nexml"), file_format,
  file, ...)
```

Arguments

`study_id`	the study ID for the study of interest (character)
`object_format`	the class of the object the query should return (either `phylo` or `nexml`). Ignored if `file_format` is specified.
`file_format`	the format of the file to be generated (`newick`, `nexus`, `nexml` or `json`).
`file`	the file name where the output of the function will be saved.
`...`	additional arguments to customize the API request (see `rotl` package documentation).

Details

If `file_format` is missing, the function returns an object of the class `phylo` from the ape package (default), or an object of the class `nexml` from the `RNeXML` package.

Otherwise `file_format` can be either `newick`, `nexus`, `nexml` or `json`, and the function will generate a file of the selected format. In this case, a file name needs to be provided using the argument `file`. If a file with the same name already exists, it will be silently overwritten.

Value

if `file_format` is missing, an object of class `phylo` or `nexml`, otherwise a logical indicating whether the file was successfully created.

See Also

`get_study_meta`

Examples

```
## Not run:
that_one_study <- get_study(study_id="pg_719", object_format="phylo")
if (require(RNeXML)) { ## if RNeXML is installed get the object directly
   nexml_study <- get_study(study_id="pg_719", object_format="nexml")
} else { ## otherwise write it to a file
   get_study(study_id="pg_719", file_format="nexml", file=tempfile(fileext=".nexml"))
}

## End(Not run)
```

`get_study_meta` *Study Metadata*

Description

Retrieve metadata about a study in the Open Tree of Life datastore.

Usage

```
get_study_meta(study_id, ...)

get_tree_ids(sm)

get_publication(sm)

candidate_for_synth(sm)

## S3 method for class 'study_meta'
get_tree_ids(sm)
```

```
## S3 method for class 'study_meta'
get_publication(sm)

## S3 method for class 'study_meta'
candidate_for_synth(sm)
```

Arguments

study_id	the study identifier (character)
...	additional arguments to customize the API request (see rotl package documentation).
sm	an object created by get_study_meta

Details

get_study_meta returns a long list of attributes for the studies that are contributing to the synthetic tree. To help with the extraction of relevant information from this list, several helper functions exists:

- get_tree_ids The identifiers of the trees associated with the study
- get_publication The citation information of the publication for the study. The DOI (or URL) for the study is available as an attribute to the returned object (i.e., `attr(object, "DOI")`) .
- candidate_for_synth The identifier of the tree(s) from the study used in the synthetic tree. This is a subset of the result of get_tree_ids.

Value

named-list containing the metadata associated with the study requested

Examples

```
## Not run:
req <- get_study_meta("pg_719")
get_tree_ids(req)
candidate_for_synth(req)
get_publication(req)

## End(Not run)
```

---

get_study_subtree *Study subtree*

Description

Retrieve subtree from a specific tree in the Open Tree of Life data store

Usage

```
get_study_subtree(study_id, tree_id, subtree_id, object_format = c("phylo"),
  file_format, file, ...)
```

Arguments

study_id	the study identifier (character)
tree_id	the tree identifier (character)
subtree_id,	either a node id that specifies a subtree or "ingroup" which returns the ingroup for this subtree.
object_format	the class of the object returned by the function (default, and currently only possibility phylo from the ape package)
file_format	character, the file format to use to save the results of the query (possible values, 'newick', 'nexus', 'json').
file	character, the path and file name where the output should be written.
...	additional arguments to customize the API request (see rotl package documentation).

Examples

```
## Not run:
small_tr <- get_study_subtree(study_id="pg_1144", tree="tree2324", subtree_id="node552052")
ingroup  <- get_study_subtree(study_id="pg_1144", tree="tree2324", subtree_id="ingroup")
nexus_file <- tempfile(fileext=".nex")
get_study_subtree(study_id="pg_1144", tree="tree2324", subtree_id="ingroup", file=nexus_file,
                  file_format="nexus")

## End(Not run)
```

---

get_study_tree *Study Tree*

Description

Returns a specific tree from within a study

Usage

```
get_study_tree(study_id = NULL, tree_id = NULL,
  object_format = c("phylo"), tip_label = c("original_label", "ott_id",
  "ott_taxon_name"), file_format, file, ...)
```

Arguments

study_id	the identifier of a study (character)
tree_id	the identifier of a tree within the study
object_format	the class of the object to be returned (default and currently only possible value phylo from the ape package).
tip_label	the format of the tip labels. "original_label" (default) returns the original labels as provided in the study, "ott_id" labels are replaced by their ott IDs, "ott_taxon_name" labels are replaced by their Open Tree Taxonomy taxon name.
file_format	the format of the file to be generated (newick default, nexus, or json).
file	the file name where the output of the function will be saved.
...	additional arguments to customize the API request (see rotl package documentation).

Value

if file_format is missing, an object of class phylo, otherwise a logical indicating whether the file was successfully created.

Examples

```
## Not run:
 tree <- get_study_tree(study_id="pg_1144", tree="tree2324")

 ## comparison of the first few tip labels depending on the options used
 head(get_study_tree(study_id="pg_1144", tree="tree2324", tip_label="original_label")$tip.label)
 head(get_study_tree(study_id="pg_1144", tree="tree2324", tip_label="ott_id")$tip.label)
 head(get_study_tree(study_id="pg_1144", tree="tree2324", tip_label="ott_taxon_name")$tip.label)

## End(Not run)
```

---

gol_about *Information about the graph of life*

Description

Graph of Life

Usage

```
gol_about(...)
```

Arguments

...	additional arguments to customize the API call (see ?rotl for more information)

Details

Basic information about the graph

Returns summary information about the entire graph database, including identifiers for the taxonomy and source trees used to build it.

Value

An invisible list of graph attributes:

- graph_num_source_trees The number of unique source trees in the graph.
- graph_taxonomy_version The version of the taxonomy used to initialize the graph.
- graph_num_tips The number of terminal (tip) taxa in the graph.
- graph_root_name The taxonomic name of the root node of the graph.
- graph_root_node_id The node ID of the root node of the graph.
- graph_root_ott_id The OpenTree Taxonomy ID (ottID) of the root node of the graph.

Examples

```
## Not run:
    res <- gol_about()

## End(Not run)
```

---

gol_node_info *Node info*

Description

Get summary information about a node in the graph database

Usage

```
gol_node_info(ott_id = NULL, include_lineage = FALSE, ...)

## S3 method for class 'gol_node'
tax_rank(tax)

## S3 method for class 'gol_node'
ott_id(tax, ...)

synth_sources(tax)

## S3 method for class 'gol_node'
synth_sources(tax)
```

Arguments

ott_id	The OpenTree taxonomic identifier.
include_lineage	Boolean. Whether to return the lineage of the node from the synthetic tree. Optional; default = FALSE.
...	additional arguments to customize the API call (see ?rotl for more information)
tax	an object returned by gol_node_info.

Details

Summary information about a queried node, including 1) whether it is in the graph database, 2) whether it is in the synthetic tree, 3) supporting study sources, 4) number of descendant tip taxa, 5) graph node ID, and 6) taxonomic information (if it is a named node in the graph), including: name, rank, OpenTree Taxonomy ID (ottID), and source taxonomy IDs.

Value

gol_node_info returns a list of summary information about the queried node.

- tree_id The tree identifier for a given study.
- num_synth_tips Numeric. The number of synthetic tree tip descendants.
- name String. The taxonomic name of the queried node (if it is a named node).
- rank String. The taxonomic rank of the queried node (if it is a named node).
- ott_id Numeric. The OpenTree Taxonomy ID (ottID) of the queried node (if it is a named node).
- num_tips Numeric. The number of taxonomic tip descendants.
- tree_sources A list of supporting source trees in the graph. May differ from "synth_sources", if trees are in the graph, but were not used in constructing the synthetic tree. Each source has:
  - study_id The study identifier. Will typically include a prefix ("pg_" or "ot_").
  - tree_id The tree identifier for a given study.
  - git_sha The git SHA identifying a particular source version.
- tax_source String. Source taxonomy IDs (if it is a named node), e.g. "ncbi:9242,gbif:5289,irmng:104628".
- synth_sources A list of supporting synthesis source trees, each with:
  - git_sha The git SHA identifying a particular source version.
  - tree_id The tree id associated with the study id used.
  - study_id The study identifier. Will typically include a prefix ("pg_" or "ot_").
- node_id The identifier for the node used by the neo4j database. These identifiers are not persistent and shouldn't be used.
- in_synth_tree Boolean. Whether the ott_id is included in the synthetic tree.

tax_rank and ott_id return vectors (character, and numeric respectively).

Examples

```
## Not run:
birds <- gol_node_info(ott_id=81461)
synth_sources(birds)
tax_rank(birds)
ott_id(birds)

## End(Not run)
```

`inspect.match_names` *Inspect and Update alternative matches for a name returned by tnrs_match_names*

Description

Taxonomic names may have different meanings in different taxonomic contexts, as the same genus name can be applied to animals and plants for instance. Additionally, the meaning of a taxonomic name may have change throughout its history, and may have referred to a different taxon in the past. In such cases, a given names might have multiple matches in the Open Tree Taxonomy. These functions allow users to inspect (and update) alternative meaning of a given name and its current taxonomic status according to the Open Tree Taxonomy.

Usage

```
## S3 method for class 'match_names'
inspect(response, row_number, taxon_name, ott_id, ...)

inspect(response, ...)

## S3 method for class 'match_names'
update(object, row_number, taxon_name, ott_id,
  new_row_number, new_ott_id, ...)
```

Arguments

`response`	an object generated by the `tnrs_match_names` function
`row_number`	the row number corresponding to the name to inspect
`taxon_name`	the taxon name corresponding to the name to inspect
`ott_id`	the ott id corresponding to the name to inspect
`...`	currently ignored
`object`	an object created by `tnrs_match_names`
`new_row_number`	the row number in the output of `inspect` to replace the taxa specified by `row_number`, `taxon_name`, or `ott_id`.
`new_ott_id`	the ott id of the taxon to replace the taxa specified by `row_number`, `taxon_name`, or `ott_id`.

Details

To inspect alternative taxonomic meanings of a given name, you need to provide the object resulting from a call to the tnrs_match_names function, as well as one of either the row number corresponding to the name in this object, the name itself (as used in the original query), or the ott_id listed for this name.

To update one of the name, you also need to provide the row number in which the name to be replaced appear or its ott id.

Value

a data frame

See Also

tnrs_match_names

Examples

```
## Not run:
   matched_names <- tnrs_match_names(c("holothuria", "diadema", "boletus"))
   inspect(matched_names, taxon_name="diadema")
   new_matched_names <- update(matched_names, taxon_name="diadema",
                               new_ott_id = 631176)
   new_matched_names

## End(Not run)
```

---

ott_id.match_names *ott_id and flags for taxonomic names matched by tnrs_match_names*

Description

rotl provides a collection of functions that allows users to extract relevant information from an object generated by tnrs_match_names function.

Usage

```
## S3 method for class 'match_names'
ott_id(tax, row_number, taxon_name, ott_id,
  only_current = TRUE, ...)

flags(tax, ...)

## S3 method for class 'match_names'
flags(tax, row_number, taxon_name, ott_id,
  only_current = TRUE, ...)
```

Arguments

tax	an object returned by tnrs_match_names
row_number	the row number corresponding to the name for which to list the synonyms
taxon_name	the taxon name corresponding to the name for which to list the synonyms
ott_id	the ott id corresponding to the name for which to list the synonyms
only_current	logical (default TRUE), should the results include data for all matched names, or only the one listed in the object returned by tnrs_match_names?
...	currently ignored

Details

These methods optionally accept one of the arguments row_number, taxon_name or ott_id to retrieve the corresponding information for one of the matches in the object returned by the tnrs_match_names function.

If these arguments are not provided, these methods can return information for the matches currently listed in the object returned by tnrs_match_names (the default) or all the matches (using only_current = FALSE).

Value

A list of the ott ids or flags for the taxonomic names matched with tnrs_match_names, for either one or all the names.

Examples

```
## Not run:
  rsp <- tnrs_match_names(c("Diadema", "Tyrannosaurus"))
  rsp$ott_id    # ott id for match currently in use
  ott_id(rsp)   # similar as above but elements are named
  ott_id(rsp, only_current=FALSE) # ott id for all possible taxonomic matches

  ## flags() is useful for instance to determine if a taxon is extinct
  flags(rsp, taxon_name="Tyrannosaurus")

## End(Not run)
```

---

rotl *An Interface to the Open Tree of Life API*

Description

The Open Tree of Life is an NSF funded project that is generating an online, comprehensive phylogenetic tree for 1.8 million species. rotl provides an interface that allows you to query and retrive the parts of the tree of life that is of interest to you.

Details

rotl provides function to most of the end points the API provides. The documentation of the API is available at: https://github.com/OpenTreeOfLife/opentree/wiki/Open-Tree-of-Life-APIs

Customizing API calls

All functions that use API end points can take 2 arguments to customize the API call and are passed as ... arguments.

- otl_v This argument controls which version of the API your call is using. The default value for this argument is a call to the non-exported function otl_version() which returns the current version of the Open Tree of Life APIs (v2).
- dev_url This argument controls whether to use the development version of the API. By default, dev_url is set to FALSE, using dev_url = TRUE in your function calls will use the development version.

For example, to use the development version of the API, you could use: `tnrs_match_names("anas", dev_url=TRUE)`

Additional arguments can also be passed to the GET and POST methods.

Acknowledgments

This package was started during the Open Tree of Life Hackathon organized by OpenTree, the NESCent Hackathon Interoperability Phylogenetic group, and Arbor.

---

studies_find_studies *find_study*

Description

Return a list of studies that match given properties

Usage

```
studies_find_studies(property = NULL, value = NULL, verbose = FALSE,
  exact = FALSE, ...)
```

Arguments

property	The property to be searched on (character)
value	The property-value to be searched on (character)
verbose	Should the output include all metadata (logical default FALSE)
exact	Should exact matching be used? (logical, default FALSE)
...	additional arguments to customize the API request (see rotl package documentation).

See Also

studies_properties which lists properties against which the studies can be searched

Examples

```
## Not run:
study <- studies_find_studies(property="ot:studyId", value="pg_719")

## End(Not run)
```

`studies_find_trees` *find trees*

Description

Return a list of trees that match a given properties

Usage

```
studies_find_trees(property = NULL, value = NULL, verbose = FALSE,
  exact = FALSE, ...)
```

Arguments

`property`	The property to be searched on (character)
`value`	The property-value to be searched on (character)
`verbose`	Should the output include all metadata? (logical, default `FALSE`)
`exact`	Should exact matching be used? (logical, default `FALSE`)
`...`	additional arguments to customize the API request (see `rotl` package documentation).

Details

The list of possible values to be used as values for the argument `property` can be found using the function studies_properties.

See Also

studies_properties which lists properties against which the studies can be searched

Examples

```
## Not run:
res <- studies_find_trees(property="ot:ottTaxonName", value="Garcinia")

## End(Not run)
```

`studies_properties` *Studies properties*

Description

Return the list of study properties that can be used to search studies and trees used in the synthetic tree.

Usage

```
studies_properties(...)
```

Arguments

`...`	additional arguments to customize the API request (see `rotl` package documentation).

Details

The list returned has 2 elements `tree_properties` and `studies_properties`. Each of these elements lists theadditional arguments to customize the API request properties that can be used to search for trees and studies that are contributing to the synthetic tree.

Value

A list of the study properties that can be used to find studies and trees that are contributing to the synthetic tree.

See Also

`studies_find_trees`

Examples

```
## Not run:
 all_the_properties <- studies_properties()
 unlist(all_the_properties$tree_properties)

## End(Not run)
```

study_list *List of studies used for the Tree of Life*

Description

Retrieve the detailed information for the list of studies used in the Tree of Life.

Usage

```
study_list(tol)

## S3 method for class 'tol_summary'
study_list(tol)
```

Arguments

tol	an object created using `tol_about(study_list = TRUE)`

Details

This function takes the object resulting from `tol_about(study_list = TRUE)` and returns a data frame listing the `tree_id`, `study_id` and `git_sha` for the studies currently included in the Tree of Life.

Value

a data frame

synonyms.match_names *List the synonyms for a given name*

Description

When querying the Taxonomic Name Resolution Services for a particular taxonomic name, the API returns as possible matches all names that include the queried name as a possible synonym. This function allows you to explore other synonyms for an accepted name, and allows you to determine why the name you queried is returning an accepted synonym.

Usage

```
## S3 method for class 'match_names'
synonyms(tax, row_number, taxon_name, ott_id,
  only_current = TRUE, ...)
```

Arguments

tax	a data frame generated by the tnrs_match_names function
row_number	the row number corresponding to the name for which to list the synonyms
taxon_name	the taxon name corresponding to the name for which to list the synonyms
ott_id	the ott id corresponding to the name for which to list the synonyms
only_current	logical (default TRUE), should the results include data for all matched names, or only the one listed in the object returned by tnrs_match_names?
...	currently ignored

Details

To list synonyms for a given taxonomic name, you need to provide the object resulting from a call to the tnrs_match_names function, as well as one of either the row number corresponding to the name in this object, the name itself (as used in the original query), or the ott_id listed for this name. Otherwise, the synonyms for all the currently matched names are returned. Using only_current = FALSE will return all synonyms for all possible matches.

Value

a list whose elements are all synomym names (as vectors of character) for the taxonomic names that match the query (the names of the elements of the list).

Examples

```
## Not run:
   echino <- tnrs_match_names(c("Diadema", "Acanthaster", "Fromia"))
   ## These 3 calls are identical
   synonyms(echino, taxon_name="Acanthaster")
   synonyms(echino, row_number=2)
   synonyms(echino, ott_id=337928)

## End(Not run)
```

---

taxonomy_about *About the Open Tree Taxonomy*

Description

Summary information about the Open Tree Taxaonomy (OTT)

Usage

```
taxonomy_about(...)
```

Arguments

...	additional arguments to customize the API request (see rotl package documentation).

Details

Return metadata and information about the taxonomy itself. Currently, the available metadata is fairly sparse, but includes (at least) the eversion, and the location from which the complete taxonomy source files can be downloaded.

Value

A list with at least the version and the location of the taxonomy source files

Examples

```
## Not run:
taxonomy_about()

## End(Not run)
```

---

taxonomy_lica *Taxonomic LICA*

Description

Taxonomic Least Inclusive Common Ancestor

Usage

```
taxonomy_lica(ott_ids = NULL, ...)

## S3 method for class 'taxon_lica'
tax_rank(tax)

## S3 method for class 'taxon_lica'
ott_taxon_name(tax)

## S3 method for class 'taxon_lica'
ott_id(tax, ...)
```

Arguments

ott_ids	a vector of ott ids for the taxa whose LICA is to be found (numeric).
...	additional arguments to customize the API request (see rotl package documentation).
tax	an object generated by the taxonomy_lica function

Details

Given a set of ott ids, get the taxon that is the least inclusive common ancestor (the LICA) of all the identified taxa. A taxonomic LICA is analogous to a most recent common ancestor (MRCA) in a phylogenetic tree.

Value

- `taxonomy_lica` returns a list about the taxonomic information relating to the LICA for the ott_ids provided.
- `tax_rank` returns a character vector of the taxonomic rank for the LICA.
- `ott_taxon_name` returns a character vector the Open Tree Taxonomy name for the LICA.
- `ott_id` returns a numeric vector of the ott id for the LICA.

Examples

```
## Not run:
req <- taxonomy_lica(ott_ids=c(515698,590452,409712,643717))
tax_rank(req)
ott_taxon_name(req)
ott_id(req)

## End(Not run)
```

---

taxonomy_subtree *Taxonomy subtree*

Description

Given an ott id, return the inclusive taxonomic subtree descended from the specified taxon.

Usage

```
taxonomy_subtree(ott_id = NULL, output_format = c("taxa", "newick", "phylo",
  "raw"), file, ...)
```

Arguments

`ott_id`	The ott id of the taxon of interest.
`output_format`	the format of the object to be returned. See the 'Return' section.
`file`	the file name where to save the output of the function. Ignored unless `output_format` is set to "`phylo`".
`...`	additional arguments to customize the API request (see `rotl` package documentation).

Details

If the output of this function is exported to a file, the only possible value for the `output_format` argument is "`newick`". If the file provided already exists, it will be silently overwritten.

Value

If the `file` argument is missing:

- "`taxa`" a list of the taxa names (species) in slot `tip_label`, and higher-level taxanomy (e.g., families, genera) in slot `edge_label`, descending from the taxa corresponding to the `ott_id` provided.
- "`newick`" a character vector containing the newick formatted string corresponding to the taxonomic subtree for the `ott_id` provided.
- "`phylo`" an object of the class `phylo` from the ape package.
- "`raw`" the direct output from the API, i.e., a list with an element named 'subtree' that contains the subtree as a newick formatted string.

If a `file` argument is provided (and `output_format` is set to "`phylo`"), a logical indicating whether the file was successfully created.

Examples

```
## Not run:
req <- taxonomy_subtree(ott_id=515698)
plot(taxonomy_subtree(ott_id=515698, output_format="phylo"))

## End(Not run)
```

---

taxonomy_taxon *Taxon information*

Description

Information about taxa.

Usage

```
taxonomy_taxon(ott_ids, ...)

## S3 method for class 'taxon_info'
tax_rank(tax)

## S3 method for class 'taxon_info'
ott_taxon_name(tax)

## S3 method for class 'taxon_info'
synonyms(tax, ...)
```

Arguments

ott_ids	the ott ids of the taxon of interest (numeric or character containing only numbers)
...	additional arguments to customize the API request (see rotl package documentation).
tax	an object generated by the taxonomy_taxon function

Details

Given a vector of ott ids, taxonomy_taxon returns information about the specified taxa.

The functions tax_rank, ott_taxon_name, and synonyms can extract this information from an object created by the taxonomy_taxon function.

Value

taxonomy_taxon returns a list detailing information about the taxa. tax_rank and ott_taxon_name return a vector. synonyms returns a list whose elements are the synonyms for each of the ott_id requested.

See Also

tnrs_match_names to obtain ott_id from a taxonomic name.

Examples

```
## Not run:
req <- taxonomy_taxon(ott_id=515698)
tax_rank(req)
ott_taxon_name(req)
synonyms(req)

## End(Not run)
```

---

tax_rank *Methods for Taxonomy*

Description

Methods for dealing with objects returned by functions dealing with the Taxonomy and the Taxonomic Name Resolution Services APIs.

Usage

```
tax_rank(tax)

ott_taxon_name(tax)

ott_id(tax, ...)

synonyms(tax, ...)
```

Arguments

tax	an object returned by taxonomy_taxon, taxonomy_lica, or tnrs_match_names
...	additional arguments (see tnrs_match_names)

Details

This is the page for the generic methods. See the help pages for taxonomy_taxon, taxonomy_lica, and tnrs_match_names for more information.

---

tnrs_contexts *TNRS contexts*

Description

This function returns a list of pre-defined taxonomic contexts (i.e. clades) which can be used to limit the scope of tnrs queries.

Usage

```
tnrs_contexts(...)
```

Arguments

...	additional arguments to customize the API request (see rotl package documentation).

Details

Taxonomic contexts are available to limit the scope of TNRS searches. These contexts correspond to uncontested higher taxa such as 'Animals' or 'Land plants'. This service returns a list containing all available taxonomic context names, which may be used as input (via the context_name argument in other functions) to limit the search scope of other services including tnrs_match_names.

Value

Returns invisibly a list for each major clades (e.g., animals, microbes, plants, fungi, life) whose elements contains the possible contexts.

tnrs_infer_context *Infer the taxonomic context from a list of names*

Description

Return a taxonomic context given a list of taxonomic names

Usage

```
tnrs_infer_context(names = NULL, ...)
```

Arguments

names	Vector of taxon names.
...	additional arguments to customize the API request (see rotl package documentation).

Details

Find the least inclusive taxonomic context that includes all the unambiguous names in the input set. Unambiguous names are names with exact matches to non-homonym taxa. Ambiguous names (those without exact matches to non-homonym taxa) are indicated in results.

Value

A list including the context name, the context ott id and possibly the names in the query that have an ambiguous taxonomic meaning in the query.

Examples

```
## Not run:
res <- tnrs_infer_context(names=c("Stellula calliope", "Struthio camelus"))

## End(Not run)
```

tnrs_match_names *Match names to the Open Tree Taxonomy*

Description

Match taxonomic names to the Open Tree Taxonomy.

Usage

```
tnrs_match_names(names = NULL, context_name = NULL,
  do_approximate_matching = TRUE, ids = NULL, include_deprecated = FALSE,
  include_dubious = FALSE, ...)
```

Arguments

names	taxon names to be queried (character vector)
context_name	name of the taxonomic context to be searched (length-one character vector). Must match (case sensitive) one of the values returned by tnrs_contexts.
do_approximate_matching	A logical indicating whether or not to perform approximate string (a.k.a. "fuzzy") matching. Using FALSE will greatly improve speed. Default, however, is TRUE.
ids	An array of ids to use for identifying names. These will be assigned to each name in the names array. If ids is provided, then ids and names must be identical in length.
include_deprecated	A boolean indicating whether or not to include deprecated taxa in the search.
include_dubious	Whether to include so-called 'dubious' taxa–those which are not accepted by OTT.
...	additional arguments to customize the API request (see rotl package documentation).

Details

Accepts one or more taxonomic names and returns information about potential matches for these names to known taxa in the Open Tree Taxononmy.

This service uses taxonomic contexts to disambiguate homonyms and misspelled names; a context may be specified using the context_name argument. If no context is specified, then the context will be inferred (i.e., the shallowest taxonomic context that contains all unambiguous names in the input). Taxonomic contexts are uncontested higher taxa that have been selected to allow limits to be applied to the scope of TNRS searches (e.g. 'match names only within flowering plants'). Once a context has been identified (either user-specified or inferred), all taxon name matches will performed only against taxa within that context. For a list of available taxonomic contexts, see tnrs_contexts.

A name is considered unambiguous if it is not a synonym and has only one exact match to any taxon name in the entire taxonomy.

Several functions listed in the 'See also' section can be used to inspect and manipulate the object generated by this function.

Value

A data frame summarizing the results of the query. The original query output is appended as an attribute to the returned object (and can be obtained using attr(object, "original_response")).

See Also

inspect.match_names, update.match_names, synonyms.match_names.

Examples

```
## Not run:
 deuterostomes <- tnrs_match_names(names=c("echinodermata", "xenacoelomorpha",
                                           "chordata", "hemichordata"))

## End(Not run)
```

`tol_about` *Information about the Tree of Life*

Description

Basic information about the Open Tree of Life (the synthetic tree)

Usage

```
tol_about(study_list = FALSE, ...)
```

Arguments

`study_list`	Whether to return the list of source studies. (Logical, default = FALSE).
`...`	additional arguments to customize the API call (see `rotl` for more information).

Details

Summary information about the current draft tree of life, including information about the list of trees and the taxonomy used to build it.

Value

An invisible list of synthetic tree summary statistics:

- tree_id The name identifier of the synthetic tree.
- date The date that the synthetic tree was constructed.
- taxonomy_version The version of the taxonomy used to initialize the graph.
- num_source_studies The number of unique source trees used in the synthetic tree.
- num_tips The number of terminal (tip) taxa in the synthetic tree.
- root_taxon_name The taxonomic name of the root node of the synthetic tree.
- root_node_id The node ID of the root node of the synthetic tree.
- root_ott_id The OpenTree Taxonomy ID (ottID) of the root node of the synthetic tree.

See Also

`study_list` to explore the list of studies used in the synthetic tree.

Examples

```
## Not run:
res <- tol_about()
studies <- study_list(tol_about(study_list=TRUE))

## End(Not run)
```

`tol_induced_subtree` *induced subtree*

Description

Extract a subtree based on a vector of ott ids.

Usage

```
tol_induced_subtree(ott_ids = NULL, file, ...)
```

Arguments

`ott_ids`	OTT ids indicating nodes to be used as tips in the induced tree
`file`	if specified, the function will write the subtree to a file in newick format.
`...`	additional arguments to customize the API call (see `rotl` for more information).

Details

Return a tree with tips corresponding to the nodes identified in the input set that is consistent with the topology of the current synthetic tree. This tree is equivalent to the minimal subtree induced on the draft tree by the set of identified nodes. Ott ids that do not correspond to any nodes found in the graph, or which are in the graph but are absent from the synthetic tree, will be identified in the output (but obvisouly will be absent from the resulting induced tree). Branch lengths in the result may be arbitrary, and the tip labels of the tree may either be taxonomic names or (for nodes not corresponding directly to named taxa) node ids.

Value

If no value is specified to the `file` argument (default), a phyogenetic tree of class `phylo`.

Otherwise, the function returns invisibly a logical indicating whether the file was successfully created.

Examples

```
## Not run:
res <- tol_induced_subtree(ott_ids=c(292466, 501678, 267845, 666104, 316878, 102710, 176458))
tree_file <- tempfile(fileext=".tre")
tol_induced_subtree(ott_ids=c(292466, 501678, 267845, 666104, 316878, 102710, 176458),
                    file=tree_file)

## End(Not run)
```

`tol_mrca` *MRCA of taxa from the synthetic tree*

Description

Most Recent Common Ancestor for a set of nodes

Usage

```
tol_mrca(ott_ids = NULL, ...)
```

Arguments

`ott_ids`	the ott ids for which the MRCA is desired (character or numeric vector)
`...`	additional arguments to customize the API call (see `rotl` for more information).

Details

Get the MRCA of a set of nodes on the current synthetic tree. Accepts any combination of node ids and ott ids as input. Returns information about the most recent common ancestor (MRCA) node as well as the most recent taxonomic ancestor (MRTA) node (the closest taxonomic node to the MRCA node in the synthetic tree; the MRCA and MRTA may be the same node). Node ids that are not in the synthetic tree are dropped from the MRCA calculation. For a valid ott id that is not in the synthetic tree (i.e. it is not recovered as monophyletic from the source tree information), the taxonomic descendants of the node are used in the MRCA calculation. Returns any unmatched node ids / ott ids.

Return the most recent common ancestor of a set of nodes in the synthetic tree.

Value

A list

Examples

```
## Not run:
birds_mrca <- tol_mrca(ott_ids=c(412129, 536234))

## End(Not run)
```

`tol_subtree` *Extract a subtree from the synthetic tree*

Description

Extract a subtree from the synthetic tree from an ott id.

Usage

```
tol_subtree(ott_id = NULL, tree_id = NULL, file, ...)
```

Arguments

`ott_id`	the ott id of the node in the tree that should serve as the root of the tree returned.
`tree_id`	the identifier for the synthesis tree. Currently a single draft tree is supported, so this argument is superfluous and may be safely ignored.
`file`	if specified, the function will write the subtree to a file in newick format.
`...`	additional arguments to customize the API call (see `rotl` for more information).

Details

Return a complete subtree of the draft tree descended from some specified node. The node to use as the start node may be specified using an ott id. If the specified node is not in the synthetic tree (or is entirely absent from the graph), an error will be returned.

Value

If no value is specified to the `file` argument (default), a phyogenetic tree of class `phylo`.

Otherwise, the function returns invisibly a logical indicating whether the file was successfully created.

Examples

```
## Not run:
      res <- tol_subtree(ott_id=81461)

## End(Not run)
```

Index