Package ‘ontologyIndex’

July 22, 2025

Type Package

Title Reading Ontologies into R

Version 2.12

Encoding UTF-8

Date 2024-02-20

Author Daniel Greene <dg333@cam.ac.uk>

Maintainer Daniel Greene <dg333@cam.ac.uk>

Description Functions for reading ontologies into R as lists and manipulating sets of ontological terms - 'ontologyX: A suite of R packages for working with ontological data', Greene et al 2017 <doi:10.1093/bioinformatics/btw763>.

License GPL (>= 2)

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.1

NeedsCompilation no

Repository CRAN

Date/Publication 2024-02-27 00:00:02 UTC

Contents

ontologyIndex-package 2
as.data.frame.ontology_index 3
check 3
exclude_descendants 4
get_ancestors 4
get_descendants 5
get_ontology 6
get_relation_names 7
get_term_descendancy_matrix 7
get_term_frequencies 8

get_term_info_content 9
get_term_property 9
go 10
hpo 10
intersection_with_descendants 10
minimal_set 11
mpo 11
ontology_index 12
print.ontology_index 12
propagate_relations 13
prune_descendants 14

Index **15**

`ontologyIndex-package` *Functions for Reading Ontologies into R*

Description

Functions for reading ontologies into R and manipulating sets of ontological terms.

Author(s)

Daniel Greene <dg333@cam.ac.uk>

Maintainer: Daniel Greene <dg333@cam.ac.uk>

References

Greene D, Richardson S, Turro E (2017). 'ontologyX: a suite of R packages for working with ontological data. _Bioinformatics_, 33(7), 1104–1106.

Westbury SK, Turro E, Greene D, Lentaigne C, Kelly AM, Bariana TK, Simeoni I, Pillois X, Attwood A, Austin S, Jansen SB, Bakchoul T, Crisp-Hihn A, Erber WN, Favier R, Foad N, Gattens M, Jolley JD, Liesner R, Meacham S, Millar CM, Nurden AT, Peerlinck K, Perry DJ, Poudel P, Schulman S, Schulze H, Stephens JC, Furie B, Robinson PN, Geet Cv, Rendon A, Gomez K, Laffan MA, Lambert MP, Nurden P, Ouwehand WH, Richardson S, Mumford AD and Freson K (2015). 'Human phenotype ontology annotation and cluster analysis to unravel genetic defects in 707 cases with unexplained bleeding and platelet disorders.' _Genome Med_, *7*(1), pp. 36.

Kohler S, Doelken SC, Mungall CJ, Bauer S, Firth HV, Bailleul-Forestier I, Black GC, Brown DL, Brudno M, Campbell J, FitzPatrick DR, Eppig JT, Jackson AP, Freson K, Girdea M, Helbig I, Hurst JA, Jahn J, Jackson LG, Kelly AM, Ledbetter DH, Mansour S, Martin CL, Moss C, Mumford A, Ouwehand WH, Park SM, Riggs ER, Scott RH, Sisodiya S, Van Vooren S, Wapner RJ, Wilkie AO, Wright CF, Vulto-van Silfhout A, de Leeuw N, de Vries B, Washingthon NL, Smith CL, Westerfield M, Schofield P, Ruef BJ, Gkoutos GV, Haendel M, Smedley D, Lewis SE and Robinson PN (2014). 'The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data.' _Nucleic Acids Res._, *42*(Database issue), pp. D966-974.

Ashburner et al. Gene ontology: tool for the unification of biology (2000) Nat Genet 25(1):25-9

Smith, Cynthia L., Carroll-Ann W. Goldsmith, and Janan T. Eppig. 'The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information.' Genome biology 6.1 (2004): 1.

`as.data.frame.ontology_index` *Case* `ontology_index` *object to* `data.frame`

Description

Case `ontology_index` object to `data.frame`

Usage

```
## S3 method for class 'ontology_index'
as.data.frame(x, ...)
```

Arguments

`x`	`ontology_index` object.
`...`	Arguments pass to `as.data.frame`.

Value

`data.frame` containing information about an ontology, rows corresponding to terms and columns to properties.

`check` *Check an ontology index object*

Description

Perform simple consistency checks on `ontology_index` object

Usage

```
check(ontology, stop_if_invalid = FALSE)
```

Arguments

`ontology`	`ontology_index` object.
`stop_if_invalid`	Logical value determining whether the function should call `stop` and print an error message upon finding that the given `ontology_index` is in valid.

exclude_descendants *Exclude terms descending from any in a given set of root terms*

Description

Exclude from set `terms`, any terms that are either in, or descend from one of, the set `roots`.

Usage

```
exclude_descendants(ontology, roots, terms)
```

Arguments

`ontology`	`ontology_index` object.
`roots`	Character vector of IDs for terms with respect to which descendants are to be defined.
`terms`	Character vector of ontological terms.

Value

Character vector of terms

See Also

`intersection_with_descendants`, `prune_descendants`

get_ancestors *Get set of terms containing all ancestors of terms in a given set*

Description

Get set of terms containing all ancestors of terms in a given set

Usage

```
get_ancestors(ontology, terms)
```

Arguments

`ontology`	`ontology_index` object.
`terms`	Character vector of ontological terms.

Value

Character vector of all terms which are an ancestor of at least one term in `terms`, including the terms themselves

See Also

link{get_descendants}

Examples

```
data(hpo)
get_ancestors(hpo, c("HP:0001873", "HP:0011877"))
```

---

get_descendants *Get set of terms containing all descendants of terms in a given set*

Description

Get set of terms containing all descendants of terms in a given set

Usage

```
get_descendants(ontology, roots, exclude_roots = FALSE)
```

Arguments

ontology	ontology_index object.
roots	Character vector of IDs for terms with respect to which descendants are to be defined.
exclude_roots	Boolean determining whether to remove the given roots terms from the result.

Value

Character vector of terms

See Also

link{get_ancestors}

Examples

```
data(hpo)
get_descendants(hpo, roots="HP:0001873")
```

get_ontology *Read ontology from OBO file into R*

Description

Read ontology from OBO file into R

Usage

```
get_ontology(
  file,
  propagate_relationships = "is_a",
  extract_tags = "minimal",
  merge_equivalent_terms = TRUE
)

get_OBO(
  file,
  propagate_relationships = "is_a",
  extract_tags = "minimal",
  merge_equivalent_terms = TRUE
)
```

Arguments

`file` File path of OBO formatted file.

`propagate_relationships` Character vector of relations

`extract_tags` Character value: either "minimal" or "everything", determining whether to extract only the properties of terms which are required to run functions in the package - i.e. `"id"`, `"name"`, `"parents"`, `"children"` and `"ancestors"` - or extract all properties provided in the file. Term properties are named in the resulting `ontology_index` as their corresponding tags in the OBO file (except `"parents"`, `"children"` and `"ancestors"` which are appended with `"_OBO"` to avoid clashing with standard `ontology_index` properties. Defaults to `"minimal"`.

`merge_equivalent_terms` Logical value determining whether terms that are marked `"equivalent_to"` a target term should be merged, retaining properties of the target term when the property should have one value, e.g. the term ID and name. Defaults to `TRUE`.

Value

`ontology_index` object.

See Also

`get_relation_names`

get_relation_names *Get names of relations used in OBO file*

Description

Get names of relations used in OBO file

Usage

```
get_relation_names(file)
```

Arguments

file	File path of OBO formatted file.

See Also

get_ontology

get_term_descendancy_matrix *Get logical descendancy matrix for set of terms*

Description

Get logical descendancy matrix for set of terms

Usage

```
get_term_descendancy_matrix(ontology, terms = NULL, rows = terms, cols = terms)
```

Arguments

ontology	ontology_index object.
terms	Character vector of ontological terms.
rows	Rows for resultant matrix (defaults to terms).
cols	Cols for resultant matrix (defaults to terms).

Value

A logical square matrix of with length(terms) columns and rows. result[row_term,col_term] == TRUE if row_term is an ancestor (and not the same as) of col_term.

A logical matrix.

Examples

```
data(hpo)
get_term_descendancy_matrix(hpo, c("HP:0001873", "HP:0011877"))
```

`get_term_frequencies` *Get frequency of each term in a set of phenotypes*

Description

Get frequency of each term in a set of phenotypes

Usage

```
get_term_frequencies(ontology, term_sets, patch_missing = FALSE)
```

Arguments

`ontology`	`ontology_index` object.
`term_sets`	List of character vectors of ontological term IDs.
`patch_missing`	Logical indicating whether to include whole ontology even if they're not present in the `term_sets` as if they had occurred once

Value

Numeric vector of information contents, named by corresponding terms. Takes into account ancestors, in the sense that all ancestor terms implied by the phenotypes are considered 'on'

See Also

`get_term_info_content`

Examples

```
data(hpo)
get_term_frequencies(hpo, list("HP:0001873"))
```

get_term_info_content *Get information content of each term in a set of phenotypes*

Description

Get information content of each term in a set of phenotypes

Usage

```
get_term_info_content(ontology, term_sets, patch_missing = FALSE)
```

Arguments

ontology	ontology_index object.
term_sets	List of character vectors of ontological term IDs.
patch_missing	Logical indicating whether to include all ontology terms even if they're not present in the term_sets as if they had occurred once

Value

Numeric vector of information contents, named by corresponding terms. Takes into account ancestors, in the sense that all ancestor terms implied by the phenotypes are considered 'on'

Examples

```
data(hpo)
get_term_info_content(hpo, list("HP:0001873"))
```

get_term_property *Get property of individual ontological term*

Description

Get property of individual ontological term

Usage

```
get_term_property(ontology, property_name, term, as_names = FALSE)
```

Arguments

ontology	ontology_index object.
property_name	Name of property.
term	Character value of term ID.
as_names	Logical value determining whether to return character vector of names (defaults to FALSE).

go — *GO index*

Description

`ontology_index` object encapsulating structure of the Gene Ontology (HPO) comprising a `list` of lists/vectors of properties of GO terms indexed by term ID (downloaded from https://current.geneontology.org/ontology/go-basic.obo).

Format

List of lists and vectors

hpo — *HPO index*

Description

`ontology_index` object encapsulating structure of the Human Phenotype Ontology (HPO) comprising a `list` of lists/vectors of properties of HPO terms indexed by term ID (downloaded from https://hpo.jax.org/app/data/ontology).

Format

List of lists and vectors

intersection_with_descendants — *Intersect a set of terms with the descendants of a given set of roots*

Description

Intersect a set of terms with the descendants of a given set of roots

Usage

```
intersection_with_descendants(ontology, roots, terms)
```

Arguments

`ontology`	`ontology_index` object.
`roots`	Character vector of IDs for terms with respect to which descendants are to be defined.
`terms`	Character vector of ontological terms.

Value

Character vector of terms

See Also

exclude_descendants, prune_descendants

Examples

```
data(hpo)
intersection_with_descendants(hpo, c("HP:0001872", "HP:0000707"), c("HP:0001873", "HP:0011877"))
```

---

minimal_set — *Remove redundant/implied terms from a set of terms*

Description

Remove redundant/implied terms from a set of terms

Usage

```
minimal_set(ontology, terms)
```

Arguments

ontology	ontology_index object.
terms	Character vector of ontological terms.

Value

Character vector of terms

Examples

```
data(hpo)
minimal_set(hpo, c("HP:0001873", "HP:0001872"))
```

---

mpo — *MPO index*

Description

ontology_index object encapsulating structure of the Mammalian Phenotype Ontology (MPO) comprising a list of lists/vectors of properties of MPO terms indexed by term ID (downloaded from https://www.informatics.jax.org/downloads/reports/MPheno_OBO.ontology).

Format

List of lists and vectors

ontology_index — *Create* ontology_index *object from vectors and lists of term properties*

Description

Create ontology_index object from vectors and lists of term properties

Usage

```
ontology_index(
  parents,
  id = names(parents),
  name = id,
  obsolete = setNames(nm = id, rep(FALSE, length(id))),
  version = NULL,
  ...
)
```

Arguments

parents	List of character vectors of parents per term.
id	Character vector of term IDs. Defaults to the "names" attribute of the parents argument and must be the same length as parents.
name	Character vector of term labels.
obsolete	Logical vector indicating whether given terms are obsolete.
version	Version information about the ontology.
...	Additional arguments, each of which should be either a vector or list of term properties, each with the same length as id.

Examples

```
animal_superclasses <- list(animal=character(0), mammal="animal", cat="mammal", fish="animal")
animal_ontology <- ontology_index(parents=animal_superclasses)
unclass(animal_ontology)
```

print.ontology_index — *Print* ontology_index *object*

Description

Print ontology_index object

Usage

```
## S3 method for class 'ontology_index'
print(x, ...)
```

Arguments

x	ontology_index object.
...	Unused parameters.

Value

Prints a summary

---

propagate_relations *Select terms by propagating relations from a set of terms*

Description

An ‘ontology_index‘ can contain multiple relations (for example in the case of the Gene Ontology, "is_a" and "part_of" could be stored as separate properties in an ‘ontology_index‘). Transitive relations (i.e. relations such that x related to y and y related to z implies x related to z, for example the relation ’is an ancestor of’) stored by an ‘ontology_index‘ can be propagated using this function. The ’inverse relations’ (i.e. x inversely related to y if y related to x) can also be propagated by setting the use_inverse_relations parameter to TRUE.

Usage

```
propagate_relations(
  ontology,
  roots,
  relations,
  use_inverse_relations = FALSE,
  exclude_roots = FALSE
)
```

Arguments

ontology	ontology_index object.
roots	Character vector of term IDs from which relations will be propagated.
relations	Character vector given names of transitive relations to be propagated.
use_inverse_relations	Boolean vector indicating whether to propagate inverse relations. If use_inverse_relations is the same length as relations, each element determines whether the corresponding relation in relations is inverted.
exclude_roots	Boolean determining whether to remove the given roots terms from the result.

Value

Character vector of terms

See Also

get_ancestors, get_descendants

---

prune_descendants — *Exclude terms descending from a given set of roots but include those roots which were originally implicitly present.*

Description

Given two sets of terms, roots and terms, construct a set of terms containing those in terms which do not descend from any term in roots, and also any terms in roots which are ancestors of any term in terms.

Usage

```
prune_descendants(ontology, roots, terms)
```

Arguments

ontology	ontology_index object.
roots	Character vector of IDs for terms with respect to which descendants are to be defined.
terms	Character vector of ontological terms.

Value

Character vector of terms

See Also

exclude_descendants, intersection_with_descendants

Index

as.data.frame.ontology_index, 3

check, 3

exclude_descendants, 4, *11*, *14*

get_ancestors, 4, *14*
get_descendants, 5, *14*
get_OBO *(*get_ontology*)*, 6
get_ontology, 6, *7*
get_relation_names, *6*, 7
get_term_descendancy_matrix, 7
get_term_frequencies, 8
get_term_info_content, *8*, 9
get_term_property, 9
go, 10

hpo, 10

intersection_with_descendants, *4*, 10, *14*

minimal_set, 11
mpo, 11

ontology_index, 12
ontologyIndex *(*ontologyIndex-package*)*, 2
ontologyIndex-package, 2

print.ontology_index, 12
propagate_relations, 13
prune_descendants, *4*, *11*, 14