

# Package ‘ontologyIndex’

January 8, 2019

**Type** Package

**Title** Functions for Reading Ontologies into R

**Version** 2.5

**Date** 2019-01-08

**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 (>= 3.1.0)

**Suggests** knitr, rmarkdown, ontologySimilarity, ontologyPlot

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-01-08 16:30:06 UTC

## R topics documented:

ontologyIndex-package . . . . .	2
check . . . . .	3
exclude_descendants . . . . .	3
get_ancestors . . . . .	4
get_descendants . . . . .	5
get_ontology . . . . .	5
get_relation_names . . . . .	6
get_term_descendancy_matrix . . . . .	7
get_term_frequencies . . . . .	7
get_term_info_content . . . . .	8
get_term_property . . . . .	9

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

## Index 14

ontologyIndex-package *Functions for Reading Ontologies into R*

## Description

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

## Details

Package: ontologyIndex  
 Type: Package  
 Version: 2.0  
 Date: 2016-07-20  
 License: GPL (>= 2)

The key functions include [get\\_ontology](#) for creating the index, [get\\_ancestors](#) for computing the set of all terms which are ancestors of at least one from set of terms and [get\\_descendants](#) for getting all terms which descend from a term. The package comes with R objects hpo, mpo and go which index the Human Phenotype Ontology, Mammalian Phenotype Ontology and Gene Ontology respectively.

## Author(s)

Daniel Greene <dg333@cam.ac.uk>

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

## References

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, Washington NL, Smith CL, Westfield 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.

---

check

*Perform simple consistency checks on 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	<i>Get set of terms containing all ancestors of terms in a given set</i>
---------------	--

---

**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	<i>Get set of terms containing all descendants of terms in a given set</i>
-----------------	--

---

**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	<i>Read ontology from OBO file into R</i>
--------------	---

---

**Description**

Read ontology from OBO file into R

**Usage**

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

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

**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. Defaults to "minimal".

**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**

<code>ontology</code>	ontology_index object.
<code>terms</code>	Character vector of ontological terms.
<code>rows</code>	Rows for resultant matrix (defaults to terms).
<code>cols</code>	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

**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

**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	<i>MPO index</i>
-----	------------------

---

**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

**Format**

List of lists and vectors

---

ontology_index	<i>Create ontology_index object from vectors and lists of term properties</i>
----------------	---

---

**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	<i>Exclude terms descending from a given set of roots but include those roots which were originally implicitly present.</i>
-------------------	---

---

**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

check, [3](#)

exclude\_descendants, [3](#), [10](#), [13](#)

get\_ancestors, [2](#), [4](#), [13](#)

get\_descendants, [2](#), [5](#), [13](#)

get\_OBO (get\_ontology), [5](#)

get\_ontology, [2](#), [5](#), [6](#)

get\_relation\_names, [6](#), [6](#)

get\_term\_descendancy\_matrix, [7](#)

get\_term\_frequencies, [7](#)

get\_term\_info\_content, [8](#), [8](#)

get\_term\_property, [9](#)

go, [9](#)

hpo, [9](#)

intersection\_with\_descendants, [4](#), [10](#), [13](#)

minimal\_set, [10](#)

mpo, [11](#)

ontology\_index, [11](#)

ontologyIndex (ontologyIndex-package), [2](#)

ontologyIndex-package, [2](#)

print.ontology\_index, [12](#)

propagate\_relations, [12](#)

prune\_descendants, [4](#), [10](#), [13](#)