Package 'hoplon'

July 13, 2023

Title tidy human phenotype ontology

Version 0.0.0.9000

Description Tidy and lightweight handling of human phenotype ontology (HPO) terms.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 2.10)

Imports dplyr,
magrittr,
purrr,
stringr,
tibble,
tidyr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

ancestors 2
children 2
descendants 3
describe 3
parents 4
parse_hpo 4
prune 5
snomed 5
umls 6

Index **7**

ancestors *Return the ancestors of an HPO term*

Description

Return the ancestors of an HPO term

Usage

```
ancestors(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector where each element is a ancestor term

Examples

```
ancestors("HP:0000001")
## Not run:
data %>% mutate(ancestors = ancestors(term))

## End(Not run)
```

children *Return the children of an HPO term*

Description

Return the children of an HPO term

Usage

```
children(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector where each element is a child term

Examples

```
children("HP:0000001")
## Not run:
data %>% mutate(children = children(term))

## End(Not run)
```

descendants *Return the descendants of an HPO term*

Description

Return the descendants of an HPO term

Usage

```
descendants(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector where each element is a descendant term

Examples

```
descendants("HP:0000001")
## Not run:
data %>% mutate(descendants = descendants(term))

## End(Not run)
```

describe *Return the description of an HPO term*

Description

Return the description of an HPO term

Usage

```
describe(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector of the description (name) of the HPO ID

Examples

```
describe("HP:0000001")
```

parents *Return the parents of an HPO term*

Description

Return the parents of an HPO term

Usage

```
parents(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector where each element is a parent term

Examples

```
parents("HP:0000001")
## Not run:
data %>% mutate(parents = parents(term))

## End(Not run)
```

parse_hpo *Parse an HPO OBO file*

Description

Parse an HPO OBO file

Usage

```
parse_hpo(path)
```

Arguments

path	Character vector; filepath to your hp.obo file.

Value

Tibble of HPO terms, term descriptions, and relations

Examples

```
## Not run:
hpo <- parse_hpo("hp.obo")

## End(Not run)
```

prune *Exclude descendants of a set of terms (roots) that are not ancestors of another set of terms. In other words, this takes a set of terms and returns the subtrees rooted at each node, pruned to nodes that are ancestors of the second set of terms.*

Description

Exclude descendants of a set of terms (roots) that are not ancestors of another set of terms. In other words, this takes a set of terms and returns the subtrees rooted at each node, pruned to nodes that are ancestors of the second set of terms.

Usage

```
prune(roots, terms)
```

Arguments

roots	Character vector, a set of valid HPO IDs; defines descendants
terms	Character vector, a set of valid HPO IDs; defines ancestors

Value

a character vector where each element is a HPO term

Examples

```
prune(roots = c("HP:0001250", "HP:0001943"), terms = "HP:0002173")
## Not run:
data %>% mutate(test = prune(roots, terms = "HP:0002173"))

## End(Not run)
```

snomed *Maps an HPO term to SNOMED terms*

Description

Maps an HPO term to SNOMED terms

Usage

```
snomed(term)
```

Arguments

term	Character vector, a valid HPO ID

Value

a character vector where each element is a SNOMED term

Examples

```
snomed("HP:0000003")
## Not run:
data %>% mutate(snomed = snomed(term))

## End(Not run)
```

`umls` *Maps an HPO term to UMLS concepts*

Description

Maps an HPO term to UMLS concepts

Usage

```
umls(term)
```

Arguments

`term`	Character vector, a valid HPO ID

Value

a character vector where each element is a UMLS concept

Examples

```
umls("HP:0000001")
## Not run:
data %>% mutate(umls = umls(term))

## End(Not run)
```

Index

ancestors, 2

children, 2

descendants, 3
describe, 3

parents, 4
parse_hpo, 4
prune, 5

snomed, 5

umls, 6