Package ‘hpoPlot’

October 13, 2022

Type Package

Title Functions for Plotting HPO Terms

Version 2.4

Date 2015-12-10

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

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

Description Collection of functions for manipulating sets of HPO terms and plotting them with a various options.

License GPL (>= 2)

Depends R (>= 3.0.0)

Imports methods, Rgraphviz, functional, magrittr

Suggests knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-14 12:07:27

R topics documented:

`apply.term.filters` *Apply a list of term filters to a given plotting context*

Description

Apply a list of term filters to a given plotting context

Usage

```
apply.term.filters(hpo.terms, plotting.context, term.filters,
  starting.terms = NULL)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`term.filters`	List of term filtering functions
`starting.terms`	Character vector of HPO term codes to filter. Defaults to all terms in the 'hpo.phenotypes' element of plotting.context, if it is present

Value

Character vector of terms

Examples

```
data(hpo.terms)
apply.term.filters(hpo.terms=hpo.terms, plotting.context=list(
hpo.phenotypes=list(Case1="HP:0001873")), term.filters=list(remove.links))
```

`calibrate.sizes` *Function to scale sizes of terms between two given limits*

Description

Function to scale sizes of terms between two given limits

Usage

```
calibrate.sizes(x, high, low)
```

Arguments

`x`	Numeric vector of term relative sizes named by term
`high`	Numeric vector of largest size
`low`	Numeric vector of smallest size

Value

Numeric vector

Examples

```
calibrate.sizes(c("HP:0000001"=10, "HP:0000006"=5), high=3, low=1)
```

`clean.terms` *Remove redundant/implied terms*

Description

Remove redundant/implied terms

Usage

```
clean.terms(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

Character vector of HPO terms

Examples

```
data(hpo.terms)
clean.terms(hpo.terms, c("HP:0001873", "HP:0001872"))
```

`exclude.branch` *Exclude terms descending from particular term from a character vector of terms*

Description

Exclude terms descending from particular term from a character vector of terms

Usage

```
exclude.branch(hpo.terms, branch.root, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`branch.root`	HPO term whose descendants should be excluded
`terms`	Character vector of HPO terms

Value

Character vector of terms

`get.ancestors` *Get set of all ancestors of set of terms*

Description

Get set of all ancestors of set of terms

Usage

```
get.ancestors(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

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

See Also

`link{get.descendants}`

Examples

```
data(hpo.terms)
get.ancestors(hpo.terms, c("HP:0001873", "HP:0011877"))
```

`get.case.based.colours` *Function to set colours of HPO nodes in plot to distinguish terms belonging to different sets of phenotypes*

Description

Function to set colours of HPO nodes in plot to distinguish terms belonging to different sets of phenotypes

Usage

```
get.case.based.colours(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of colours, named by term

`get.case.based.labels` *Function to label HPO nodes in plot to indicate to which phenotypes each of the terms belong*

Description

Function to label HPO nodes in plot to indicate to which phenotypes each of the terms belong

Usage

```
get.case.based.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of colours, named by term

`get.case.term.matrix` *Get a matrix with columns of hpo terms and rows of patients,*

Description

Get a matrix with columns of hpo terms and rows of patients,

Usage

```
get.case.term.matrix(hpo.phenotypes, columns = NULL)
```

Arguments

`hpo.phenotypes`	List of character vectors of HPO terms. Result includes only terms which are explicitly present in the list items, so if you wish the result to include even terms which are implicitly present, lapply `get.ancestors` to the argument before passing it to this function
`columns`	Force result to have these exact columns, entering F for terms which aren't present

Value

Logical matrix - entry for a patient/hpo term = T if the patient has the term and F otherwise.

Examples

```
get.case.term.matrix(list(Patient1=c("HP:0001873")))
```

`get.code.node.labels` *Function to label HPO nodes in plot with just HPO code*

Description

Function to label HPO nodes in plot with just HPO code

Usage

```
get.code.node.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of labels, named by term

`get.descendants` *Get set of all descendants of single term*

Description

Get set of all descendants of single term

Usage

```
get.descendants(hpo.terms, ancestor, remove.ancestor = FALSE)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`ancestor`	Character vector of length 1 - the HPO code of the term whose descendants you wish to retrieve
`remove.ancestor`	Boolean indicating whether to remove the given ancestor or not

Value

Character vector of terms

See Also

`link{get.ancestors}`

Examples

```
data(hpo.terms)
get.descendants(hpo.terms, ancestor=c("HP:0001873"))
```

`get.frequency.based.colours` *Function to colour HPO nodes in plot with colours based on frequency with which terms appear in phenotypes*

Description

Function to colour HPO nodes in plot with colours based on frequency with which terms appear in phenotypes

Usage

```
get.frequency.based.colours(hpo.terms, terms, plotting.context,
  colour.func = NULL)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`colour.func`	Function capable of returning a set of colours, given the number of colours it needs to return

Value

Character vector of colours, named by term

`get.frequency.based.labels`

Function to label HPO nodes in plot based on frequency of occurrence in phenotypes

---

Description

Function to label HPO nodes in plot based on frequency of occurrence in phenotypes

Usage

```
get.frequency.based.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of labels, named by term

`get.frequency.based.sizes` *Function to size HPO nodes in plot based on frequency of occurrence in phenotypes*

Description

Function to size HPO nodes in plot based on frequency of occurrence in phenotypes

Usage

```
get.frequency.based.sizes(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of sizes, named by term

`get.full.labels` *Function to label HPO nodes in plot with full labels*

Description

Function to label HPO nodes in plot with full labels

Usage

```
get.full.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of labels, named by term

`get.hpo.graph` *Get HPO graph object*

Description

Get HPO graph object

Usage

```
get.hpo.graph(hpo.terms, terms = apply.term.filters(hpo.terms = hpo.terms,
  plotting.context = plotting.context, term.filters = list()),
  plotting.context = NULL, colours = "white",
  labels = get.simple.node.labels, borders = "#FFFFFF00", sizes = 0.75,
  font.sizes = rep(30, length(terms)), shapes = rep("circle",
  length(terms)), nodeAttrs = NULL, arrowAttrs = list(color = "#000000"))
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`colours`	Function to set the colours of the HPO nodes in the graph based on the plotting context, or a character vector of colours
`labels`	Function to set the labels of the HPO nodes in the graph based on the plotting context, or a character vector of node labels
`borders`	Function to set the borders of the HPO nodes in the graph based on the plotting context, or a character vector of border colours
`sizes`	Function to set the sizes of the HPO nodes in the graph based on the plotting context, or a numeric vector of node sizes
`font.sizes`	Function to set the font sizes of the text to be placed in the HPO nodes in the graph based on the plotting context, or an integer vector of font sizes
`shapes`	Function to set the shapes of the HPO nodes in the graph based on the plotting context, or a character vector of shape names (defaults to 'circle')
`nodeAttrs`	Pass nodeAttrs directly to rgraphviz plotting function
`arrowAttrs`	List of properties to set for arrows (note, these properties will be used for all arrow)

Value

graphAM S4 object

See Also

`hpo.plot`

Examples

```
data(hpo.terms)
phenotype.strings <- c(
A="HP:0001382,HP:0004272,HP:0007917,HP:0004912,HP:0001596",
B="HP:0001382,HP:0004272,HP:0002165,HP:0004800,HP:0004912",
C="HP:0004800,HP:0001382,HP:0004912,HP:0007917,HP:0008743",
D="HP:0001257,HP:0001382,HP:0007917,HP:0012623,HP:0002165",
E="HP:0007917,HP:0004800,HP:0004272,HP:0001596,HP:0002165"
)

hpo.phenotypes <- term.set.list.from.character(phenotype.strings)

get.hpo.graph(
hpo.terms=hpo.terms,
plotting.context=list(hpo.phenotypes=hpo.phenotypes)
)
```

`get.informative.node.labels` *Function to label HPO nodes in plot with node description and information content*

Description

Function to label HPO nodes in plot with node description and information content

Usage

```
get.informative.node.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of labels, named by term

`get.mpo.to.hpo` *Get MPO to HPO R-Object*

Description

Get MPO to HPO R-Object

Usage

```
get.mpo.to.hpo(hpo.terms, cross.species.file)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`cross.species.file`	cross species obo file, downloadable from http://compbio.charite.de/hudson/? website

Value

List of HPO terms per MPO term

`get.node.friendly.long.names` *Split up the HPO term descriptions so they fit in nodes for plot*

Description

Split up the HPO term descriptions so they fit in nodes for plot

Usage

```
get.node.friendly.long.names(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

Character vector

Examples

```
data(hpo.terms)
get.node.friendly.long.names(hpo.terms, c("HP:0001873", "HP:0011877"))
```

`get.ontology` *Get R-Object representation of ontology from obo file*

Description

Get R-Object representation of ontology from obo file

Usage

```
get.ontology(file, qualifier = "HP")
```

Arguments

`file`	File path of obo file
`qualifier`	Character vector - "HP" for HPO, "MP" for MPO, etc.

Value

R-Object (list) representing ontology

`get.pop.frequency.based.colours` *Function to colour HPO nodes in plot with colours based on information content/frequency of terms with respect to population*

Description

Function to colour HPO nodes in plot with colours based on information content/frequency of terms with respect to population

Usage

```
get.pop.frequency.based.colours(hpo.terms, terms, plotting.context,
  colourPalette = colorRampPalette(c("Yellow", "Green", "#0099FF"))(10),
  terms.freq = if (is.null(plotting.context$frequency))
  exp(-plotting.context$information[terms]) else plotting.context$frequency,
  max.colour.freq = max(terms.freq), min.colour.freq = min(terms.freq))
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`colourPalette`	Character vector of colours for the different information contents of the terms to be plotted, going from rare to common
`terms.freq`	Numeric vector of frequencies of terms in plot, named by term
`max.colour.freq`	Numeric value in [0, 1] giving the maximum frequency (to which the dullest color will be assigned)
`min.colour.freq`	Numeric value in [0, 1] giving the minimum frequency (to which the brightest color will be assigned)

Value

Character vector of colours, named by term

`get.shortened.names` *Get human readable, shortened (where possible) HPO term names*

Description

Get human readable, shortened (where possible) HPO term names

Usage

```
get.shortened.names(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

Character vector

Examples

```
data(hpo.terms)
get.shortened.names(hpo.terms, c("HP:0001873", "HP:0011877"))
```

`get.significance.based.sizes`

Function to size HPO nodes in plot with colours based on significance of seeing this many of each term in phenotypes

---

Description

Function to size HPO nodes in plot with colours based on significance of seeing this many of each term in phenotypes

Usage

```
get.significance.based.sizes(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of sizes, named by term

`get.simple.node.labels`

Function to label HPO nodes in plot with just node description

---

Description

Function to label HPO nodes in plot with just node description

Usage

```
get.simple.node.labels(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector of labels, named by term

`get.term.adjacency.matrix`

Get an adjacency for set of HPO terms

Description

Get an adjacency for set of HPO terms

Usage

```
get.term.adjacency.matrix(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

A logical matrix representing the directed adjacency matrix of `terms` based on DAG structure of HPO, whereby a TRUE entry signifies that the term corresponding to the column is a parent term of the term correspnding to the row.

See Also

`get.term.pseudo.adjacency.matrix`

Examples

```
data(hpo.terms)
get.term.adjacency.matrix(hpo.terms, c("HP:0000118", "HP:0001873", "HP:0011877"))
```

`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(hpo.terms, terms = NULL, rows = terms,
  cols = terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`rows`	Rows for resultant matrix (defaults to `terms`)
`cols`	Cols for resultant matrix (defaults to `terms`)

Value

A logical descendancy matrix of `terms` by `terms` based on DAG structure of HPO, where by the row term is an ancestor of the column term if result[row.term,col.term] == TRUE

Examples

```
data(hpo.terms)
get.term.descendancy.matrix(hpo.terms, 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(hpo.terms, hpo.phenotypes, patch.missing = FALSE)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`hpo.phenotypes`	List of HPO term character vectors
`patch.missing`	Logical indicating whether to include all HPO even if they're not present in the `hpo.phenotypes` as if they had occurred once

Value

Numeric vector of information contents, named by corresponding HPO 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.terms)
get.term.frequencies(hpo.terms, 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(hpo.terms, hpo.phenotypes, patch.missing = FALSE)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`hpo.phenotypes`	List of HPO term character vectors
`patch.missing`	Logical indicating whether to include all HPO even if they're not present in the `hpo.phenotypes` as if they had occurred once

Value

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

Examples

```
data(hpo.terms)
get.term.info.content(hpo.terms, list("HP:0001873"))
```

`get.term.pseudo.adjacency.matrix` *Get an adjacency to MRCA matrix for set of HPO terms*

Description

Get an adjacency to MRCA matrix for set of HPO terms

Usage

```
get.term.pseudo.adjacency.matrix(hpo.terms, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms

Value

A logical matrix represenging the directed adjacency matrix of `terms` based on DAG structure of HPO, whereby a TRUE entry signifies the term correspnding to the column is MRCA of the row term in `terms`

See Also

`get.term.adjacency.matrix`

Examples

```
data(hpo.terms)
get.term.pseudo.adjacency.matrix(hpo.terms, c("HP:0000118", "HP:0001873", "HP:0011877"))
```

`hpo.plot` *Plot HPO graph object*

Description

Plot HPO graph object

Usage

```
hpo.plot(hpo.terms, terms = apply.term.filters(hpo.terms = hpo.terms,
  plotting.context = plotting.context, term.filters = list()),
  plotting.context = NULL, hpo.phenotypes = NULL, term.frequencies = NULL,
  colours = "cyan", labels = get.simple.node.labels,
  borders = "#FFFFFF00", sizes = 0.75, font.sizes = rep(30,
  length(terms)), shapes = rep("circle", length(terms)), nodeAttrs = NULL,
  arrowAttrs = list(color = "#000000"), ...)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`hpo.phenotypes`	List of HPO term character vectors
`term.frequencies`	Numeric vector of population frequencies of terms (named by term codes)
`colours`	Function to set the colours of the HPO nodes in the graph based on the plotting context, or a character vector of colours
`labels`	Function to set the labels of the HPO nodes in the graph based on the plotting context, or a character vector of node labels
`borders`	Function to set the borders of the HPO nodes in the graph based on the plotting context, or a character vector of border colours

`sizes`	Function to set the sizes of the HPO nodes in the graph based on the plotting context, or a numeric vector of node sizes
`font.sizes`	Function to set the font sizes of the text to be placed in the HPO nodes in the graph based on the plotting context, or an integer vector of font sizes
`shapes`	Function to set the shapes of the HPO nodes in the graph based on the plotting context, or a character vector of shape names (defaults to 'circle')
`nodeAttrs`	Pass nodeAttrs directly to rgraphviz plotting function
`arrowAttrs`	List of properties to set for arrows (note, these properties will be used for all arrow)
`...`	Extra arguments to pass to plot

Value

Plots graph

See Also

`get.hpo.graph`

Examples

```
data(hpo.terms)
hpo.plot(
hpo.terms=hpo.terms,
terms=get.ancestors(hpo.terms,
c("HP:0001382","HP:0004272","HP:0007917","HP:0004912","HP:0001596"))
)
```

`hpo.terms` *HPO Terms object (based on version 887 of the HPO)*

Description

Object comprising `list` of properties of the HPO, indexed by term ID

Format

List of indices containing metadata and structure of HPO

`hpoPlot` *Functions for Plotting HPO Terms*

Description

Functions for performaing operations on sets of HPO terms (character vectors of HPO term IDs) in the context of the HPO structure, and plotting them with a various options

Details

Package:	hpoPlot
Type:	Package
Version:	2.3
Date:	2015-01-7
License:	GPL (>= 2)

The key function is `hpo.plot`, which plots a set of phenotype terms given their HPO IDs and their ontological relations given by the HPO.

Author(s)

Daniel Greene <dg333@cam.ac.uk>

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

References

‘The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data’, Nucl. Acids Res. (1 January 2014) 42 (D1): D966-D974 doi:10.1093/nar/gkt1026 Westbury, S. K. et al. (2015). Human Phenotype Ontology annotation and cluster analysis to un- ravel genetic defects in 707 cases with unexplained bleeding and platelet disorders. Genome Medicine. 7 (2015)

`intersection.with.branches` *Intersect set of terms with branches of HPO*

Description

Intersect set of terms with branches of HPO

Usage

```
intersection.with.branches(hpo.terms, branch.roots, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`branch.roots`	Character vector of roots of branches you wish to intersect with
`terms`	Character vector of HPO terms

Value

Character vector of terms

Examples

```
data(hpo.terms)
intersection.with.branches(hpo.terms, "HP:0001872", c("HP:0001873", "HP:0011877"))
```

`mpo.terms` *MPO Terms object*

Description

Object comprising `list` of properties of the MPO, indexed by term ID

Format

List of indices containing metadata and structure of MPO

`mpo.to.hpo` *Object containing data for mapping between MPO and HPO*

Description

List containing cross-species ontology (MPO to HPO) information - character vectors of HPO terms indexed by associated MPO term IDs

Format

List of HPO terms per MPO term

`n.most.frequent.terms` *Select most frequently annotated terms from a set of phenotypes*

Description

Select most frequently annotated terms from a set of phenotypes

Usage

```
n.most.frequent.terms(hpo.terms, terms, plotting.context, n)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`n`	Integer

Value

Character vector of length at most `n`

See Also

`remove.terms.with.less.than.n.occurrences`, `remove.uninformative.for.plot`

Examples

```
data(hpo.terms)
n.most.frequent.terms(hpo.terms, c("HP:0001873"),
list(hpo.phenotypes=list("HP:0001873", "HP:0001902")), n=2)
```

`p.values.for.occurrence.of.term.in.group`

Get p-values for observing at least as many of each term as have been in phenotypes given information content

Description

Get p-values for observing at least as many of each term as have been in phenotypes given information content

Usage

```
p.values.for.occurrence.of.term.in.group(hpo.terms, hpo.phenotypes, terms.freq)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`hpo.phenotypes`	List of HPO term character vectors
`terms.freq`	Numeric vector of population frequencies of terms

Value

Numeric vector of log p-values named by correspnding term

`prune.branch` *Prune all terms descending from given term down to that term and ensure no degeneracy*

Description

Prune all terms descending from given term down to that term and ensure no degeneracy

Usage

```
prune.branch(hpo.terms, prune.to.point, terms)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`prune.to.point`	HPO term which can be included, but whose descendants should be excluded
`terms`	Character vector of HPO terms

Value

Character vector of terms

`remove.links` *Remove terms with exactly one parent and child from plot*

Description

Remove terms with exactly one parent and child from plot

Usage

```
remove.links(hpo.terms, terms, plotting.context = NULL)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector

See Also

`remove.terms.with.less.than.n.occurrences`, `n.most.frequent.terms`

Examples

```
data(hpo.terms)
remove.links(hpo.terms, c("HP:0001873"), list(hpo.phenotypes=list("HP:0001873", "HP:0001902")))
```

`remove.non.pa.terms` *Remove terms not descending from phenotypic abnormality*

Description

Remove terms not descending from phenotypic abnormality

Usage

```
remove.non.pa.terms(hpo.terms, terms, plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector

See Also

`remove.terms.with.less.than.n.occurrences`, `n.most.frequent.terms`

`remove.terms.with.less.than.n.occurrences`

Remove terms with less than certain number of occurrences

Description

Remove terms with less than certain number of occurrences

Usage

```
remove.terms.with.less.than.n.occurrences(hpo.terms, terms, plotting.context, n)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms
`n`	Integer

Value

Character vector

See Also

`n.most.frequent.terms`, `remove.uninformative.for.plot`

Examples

```
data(hpo.terms)
remove.terms.with.less.than.n.occurrences(hpo.terms,
c("HP:0001873"), list(hpo.phenotypes=list("HP:0001873", "HP:0001902")), 2)
```

`remove.uninformative.for.plot`

Remove uninformative terms (fitting plotting filter format)

Description

Remove uninformative terms (fitting plotting filter format)

Usage

```
remove.uninformative.for.plot(hpo.terms, terms = get.ancestors(hpo.terms,
  unlist(plotting.context$hpo.phenotypes)), plotting.context)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`plotting.context`	List object with hpo.phenotypes slot for list of character vectors of terms

Value

Character vector

See Also

`remove.terms.with.less.than.n.occurrences`, `n.most.frequent.terms`

Examples

```
data(hpo.terms)
remove.uninformative.for.plot(hpo.terms,
c("HP:0001873"), list(hpo.phenotypes=list("HP:0001873", "HP:0001902")))
```

`remove.uninformative.terms` *Get a minimal set of terms which can be used to partition a set of phenotypes*

Description

Get a minimal set of terms which can be used to partition a set of phenotypes

Usage

```
remove.uninformative.terms(hpo.terms, hpo.phenotypes)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`hpo.phenotypes`	List of HPO term character vectors

Value

Character vector of set of terms, excluding terms for which the presence of their descendants all partition the set of terms in the same way

Examples

```
data(hpo.terms)
remove.uninformative.terms(hpo.terms, list(Patient1=c("HP:0001873")))
```

`setDimNames` *setNames for arrays...*

Description

setNames for arrays...

Usage

```
setDimNames(array.object, list.of.dimension.names)
```

Arguments

`array.object`	Array
`list.of.dimension.names`	List of character vectors with which to name each dimension of the array

Value

Named array

Examples

```
setDimNames(matrix(1:4,2,2), list(c("Cat", "Dog"), c("Name", "Weight")))
```

`simpleCap` *Capitalise words in character vector*

Description

Capitalise words in character vector

Usage

```
simpleCap(x)
```

Arguments

`x`	Character vector

Value

Character vector

Examples

```
simpleCap(c("a simple test", "Another-test"))
```

`swap.out.alt.ids` *Remove alternate/deprecated HPO term IDs and swap for new ones*

Description

Remove alternate/deprecated HPO term IDs and swap for new ones

Usage

```
swap.out.alt.ids(hpo.terms, terms, remove.dead = FALSE)
```

Arguments

`hpo.terms`	R-Object representation of HPO
`terms`	Character vector of HPO terms
`remove.dead`	Boolean to indicate whether to strip out terms which can't be found in the given hpo.terms database argument

Value

A directed adjacency matrix of `terms` based on DAG structure of HPO, whereby each term is considered adjacent to it's MRCA in `terms`

Examples

```
data(hpo.terms)
swap.out.alt.ids(hpo.terms, c("HP:0001873"))
```

`term.set.list.from.character` *Get list of character vector of HPO terms, given character vector of comma separated terms*

Description

Get list of character vector of HPO terms, given character vector of comma separated terms

Usage

```
term.set.list.from.character(character.vector)
```

Arguments

`character.vector`	Character vector of comma separated terms

Value

List of character vectors of HPO terms

Examples

```
term.set.list.from.character(c("HP:0001873", "HP:0001902"))
```

Index