

# Package ‘hpoPlot’

July 2, 2014

**Type** Package

**Title** Functions to plot graphviz style graphs of sets of HPO terms

**Version** 1.0

**Date** 2014-04-14

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

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

**Description** Functions for visualising a set of HPO-encoded phenotypes in a variety of ways

**License** GPL (>= 2)

**Depends** R (>= 2.10), Rgraphviz, functional, plotrix, Hmisc

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-10 18:28:49

## R topics documented:

hpoPlot-package	2
border.functions	3
clean.terms	3
colouring.functions	4
get.ancestors	4
get.cohort.information	5
get.descendants	6
get.hpo.terms	6
get.node.friendly.long.names	7
get.term.adjacency.matrix	8
get.term.patient.matrix	9
hpo.graph	10
hpo.terms	12

labelling.functions . . . . .	13
remove.duplicate.term.sets . . . . .	13
remove.uninformative.terms . . . . .	14
setDimNames . . . . .	15
size.functions . . . . .	15
suggest.aspect.ratio.width.to.height . . . . .	16
swap.out.alt.ids . . . . .	17
term.set.list.from.character . . . . .	18

<b>Index</b>	<b>19</b>
--------------	-----------

---

hpoPlot-package	<i>Functions to plot graphviz style graphs of sets of HPO terms</i>
-----------------	---

---

## Description

Functions for visualising a set of HPO-encoded phenotypes in a variety of ways. The most important function in the package is `hpo.graph`, which expects to be given a list of HPO term character vectors (each typically the HPO encoded phenotype of a single patient), and some functions telling it how to display the terms in the resulting plot.

## Details

Package: hpoPlot  
 Type: Package  
 Version: 1.0  
 Date: 2014-04-14  
 License: Public Domain

## Author(s)

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

## Examples

```
data(hpo)

patients.character.vector <- 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"
```

```

)

patients.of.interest <- term.set.list.from.character(hpo.terms, patients.character.vector)

#simple example
hpo.graph(
  hpo.terms=hpo.terms,
  patients=patients.of.interest,
  main.title="Simple Example",
  filter.out.uninformative=TRUE
)

#complex example
hpo.graph(
  hpo.terms=hpo.terms,
  patients=patients.of.interest,
  main.title="Complex Example",
  filter.out.uninformative=FALSE,
  colouring.function=colouring.functions$get.patient.based.colours,
  labelling.function=labelling.functions$get.patient.based.labels,
  size.function=size.functions$get.frequency.based.sizes,
  border.function=border.functions$get.no.borders
)

```

---

border.functions	<i>List Of Border Functions</i>
------------------	---------------------------------

---

### Description

A list of functions which automatically allocate borders to nodes representing HPO terms in plots generated by [hpo.graph](#).

### Format

The format is: List of 2 \$ get.grey.borders:function (hpo.terms, plotting.context) \$ get.no.borders :function (hpo.terms, plotting.context)

---

clean.terms	<i>Get Leaves Of HPO Term Set</i>
-------------	-----------------------------------

---

### Description

Get a set of leaf nodes which represent a given set of terms.

### Usage

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

**Arguments**

hpo.terms      An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
 terms          Character vector of terms to ‘clean’.

**Value**

Character vector of leaf terms

**Examples**

```
data(hpo)
headache.terms <- hpo.terms$id[
hpo.terms$name
]
leaf.terms <- clean.terms(hpo.terms, headache.terms)
```

---

colouring.functions      *List Of Colouring Functions*

---

**Description**

A list of functions which automatically generate colours for nodes representing HPO terms in plots generated by [hpo.graph](#).

**Format**

The format is: List of 4 \$get.patient.based.colours :function (hpo.terms, plotting.context) \$get.information.based.colours:fun  
 (hpo.terms, plotting.context, colourPalette = colorRampPalette(c("Yellow", "Green", "#0099FF"))(10))  
 \$get.frequency.based.colours :function (hpo.terms, plotting.context, colour.func = NULL) \$get.white.nodes  
 :function (hpo.terms, plotting.context)

---

get.ancestors              *Get ancestor terms of term set*

---

**Description**

Recursive function to find all the terms from which a term in the given set descends.

**Usage**

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

**Arguments**

hpo.terms      An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
 target.terms    Character vector of HPO terms.

**Value**

Character vector of HPO terms which are ancestors to at least one of the ‘target.terms’

**Examples**

```
data(hpo)
example.ancestors <- get.ancestors(hpo.terms, "HP:0002076")
print(hpo.terms$name[example.ancestors])
```

---

```
get.cohort.information
```

*Get Information Content Of Terms*

---

**Description**

Get information content (i.e. - log frequency) of term with respect to a given cohort of patients/term-sets.

**Usage**

```
get.cohort.information(hpo.terms, patient.hpo.terms, patch.missing = FALSE)
```

**Arguments**

`hpo.terms` An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
`patient.hpo.terms` List of HPO term sets.  
`patch.missing` Return an information content for all HPO terms (i.e. including those not present in the cohort) and set it to the highest information content.

**Value**

Numeric vector of information contents, named by term.

**Examples**

```
data(hpo)

patients.character.vector <- 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"
)

patient.hpo.terms <- term.set.list.from.character(hpo.terms, patients.character.vector)

cohort.information <- get.cohort.information(hpo.terms, patient.hpo.terms)
```

get.descendants      *Get Descendants*

---

**Description**

Get all terms which descend from a given term in the HPO.

**Usage**

```
get.descendants(hpo.terms, ancestor)
```

**Arguments**

hpo.terms	An R-Object representing the HPO, as returned by <a href="#">get.hpo.terms</a> .
ancestor	Single HPO term

**Value**

Character vector of HPO terms which descend from 'ancestor'.

**Examples**

```
data(hpo)

types.of.headache <- get.descendants(
  hpo.terms,
  hpo.terms$id[hpo.terms$name == "Headache"]
)

print(hpo.terms$name[types.of.headache])
```

---

get.hpo.terms      *Get HPO As R-Object*

---

**Description**

Parses a .obo file containing the HPO and returns an R-Object representing the HPO.

**Usage**

```
get.hpo.terms( file )
```

**Arguments**

file	The location of the .obo file (leave blank to download the current official version from " <a href="http://compbio.charite.de/hudson/job/hpo/lastStableBuild/artifact/ontology/release/hp.obo">http://compbio.charite.de/hudson/job/hpo/lastStableBuild/artifact/ontology/release/hp.obo</a> ").
------	--

**Value**

id	Character vector of HPO term IDs
name	Character vector of HPO clinical descriptions names
children	Per-term list of character vectors of direct descendants
parents	Per-term list of character vectors of parent terms
ancestors	Per-term list of character vectors of ancestors
siblings	Per-term list of character vectors of sibling nodes
alt.id	Per-term list of character vectors of alternate/previous IDs
date.downloaded	Date that the object was constructed

**References**

Sebastian Kohler,Sandra C Doelken,Christopher J. Mungall,Sebastian Bauer,Helen V. Firth,Isabelle Bailleul-Forestier,Graeme C. M. Black,Danielle L. Brown,Michael Brudno,Jennifer Campbell,David R. FitzPatrick,Janan T. Eppig,Andrew P. Jackson,Kathleen Freson,Marta Girdea,Ingo Helbig,Jane A. Hurst,Johanna Jahn,Laird G. Jackson,Anne M. Kelly,David H. Ledbetter,Sahar Mansour,Christa L. Martin,Celia Moss,Andrew Mumford,Willem H. Ouwehand,Soo-Mi Park,Erin Rooney Riggs,Richard H. Scott,Sanjay Sisodiya,Steven Van Vooren,Ronald J. Wapner,Andrew O. M. Wilkie,Caroline F. Wright,Anneke T. Vulto-van Silfhout,Nicole de Leeuw,Bert B. A. de Vries,Nicole L. Washington,Cynthia L. Smith,Monte Westerfield,Paul Schofield,Barbara J. Ruef,Georgios V. Gkoutos,Melissa Haendel,Damian Smedley,Suzanna E. Lewis,and Peter N. Robinson - 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

**Examples**

```
#to create ontology object from official .obo file...
#hpo.terms <- get.hpo.terms()
```

---

```
get.node.friendly.long.names
      Get Node-Friendly Long Names
```

---

**Description**

Get the clinical names of HPO terms, shortened, where possible, to exclude phrases like ‘Abnormality of’ and split onto several lines where necessary so they fit neatly onto nodes

**Usage**

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

**Arguments**

hpo.terms      An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
terms          Character vector of HPO terms.

**Value**

Character vector of descriptions

**Examples**

```
data(hpo)

body.height <- hpo.terms$id[hpo.terms$name == "Abnormality of body height"]

print(
  paste(
    hpo.terms$name[hpo.terms$id == body.height],
    " -> ",
    get.node.friendly.long.names(
      hpo.terms,
      body.height
    ),
    sep=""
  )
)
```

---

get.term.adjacency.matrix  
*Get HPO Term Adjacency Matrix*

---

**Description**

Get adjacency matrix based on the HPO for a set of terms.

**Usage**

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

**Arguments**

hpo.terms      An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
terms          Character vector of hpo terms.

**Details**

get.term.adjacency.matrix only creates edges between parents and children, whereas get.term.pseudo.adjacency.matrix also creates edges from a node to its most recent ancestors in the set of terms if its parent is absent.



**Value**

Matrix of logicals x, where x[row, col] is TRUE if row is a parent of column and FALSE otherwise.

**Examples**

```
data(hpo)

migraine.terms <- get.ancestors(hpo.terms, hpo.terms$id[hpo.terms$name == "Migraine"])

migraine.adj.mat <- get.term.adjacency.matrix(hpo.terms, migraine.terms)

#plot simple graph of migraine terms
plot(
  new(
    "graphAM",
    adjMat=migraine.adj.mat,
    edgemode="directed"
  )
)
```

---

get.term.patient.matrix

*Get Term By Patient Matrix*

---

**Description**

Create a matrix of logicals where the patients are represented by the rows, and the HPO terms present in at least 1 patient are represented by the columns. The values in the cells are then TRUE if the patient of the row has the phenotype of the column and FALSE otherwise.

**Usage**

```
get.term.patient.matrix(patient.hpo.terms.with.ancs)
```

**Arguments**

patient.hpo.terms.with.ancs

List of HPO term character vectors, most usually containing ancestors. Such a list can be obtained from a list containing only the leaf terms using 'lapply(list.of.term.sets, function(x) get.ancestors(hpo.terms, x))'.

**Value**

Logical matrix

**Examples**

```

data(hpo)

patients.character.vector <- 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"
)

patient.hpo.terms <- term.set.list.from.character(hpo.terms, patients.character.vector)

term.patient.matrix <- get.term.patient.matrix(
  lapply(
    patient.hpo.terms,
    function(patient.terms) get.ancestors(hpo.terms, patient.terms)
  )
)

```

hpo.graph

*Plot HPO Terms***Description**

Function to display, or save to a PDF, a plot of sets of HPO terms.

**Usage**

```

hpo.graph(
  hpo.terms,
  patients,
  term.information.content = NULL,
  term.population.frequency = NULL,
  terms=NULL,
  colouring.function = colouring.functions$get.frequency.based.colours,
  labelling.function = labelling.functions$get.frequency.based.labels,
  border.function = border.functions$get.no.borders,
  size.function = size.functions$get.standard.sizes,
  main.title = NULL,
  draw.legend = FALSE,
  filter.out.uninformative = FALSE,
  filter.out.non.pa.terms = FALSE,
  min.occs = 1,
  nodeAttrs = NULL,
  pdf.file.name = NULL,
  pdf.height = 10,
  pdf.width = NULL
)

```

**Arguments**

hpo.terms	An R-Object representing the HPO, as returned by <a href="#">get.hpo.terms</a> .
patients	A list of character vectors of HPO terms, where typically each list item represents a patient. e.g. patients=list(A="HP:00000118")
term.information.content	The ‘Information-Content’ (i.e. -log frequency) of the terms relevant to the plot as a named (by the HPO ID) vector in the context of a particular cohort of patients (as returned by <a href="#">get.cohort.information</a> ).
term.population.frequency	An alternative to the ‘term.information.content’, the frequency of the terms relevant to the plot in the population, as opposed to in the sample to be plotted, as a named vector of values between 0 and 1. If both this and ‘term.information.content’ are given, ‘term.information.content’ is used.
terms	Character vector of HPO term codes for terms which are to appear in the plot. Defaults to NULL whereafter the terms in the plot are chosen automatically depending on the filtering options (for example ‘filter.out.uninformative’).
colouring.function	A named vector of colours for the terms in the plot or a function which returns one (see <a href="#">colouring.functions</a> ).
labelling.function	A named vector of labels for the terms in the plot or a function which returns one (see <a href="#">labelling.functions</a> ).
border.function	A named vector of border colours for the terms in the plot or a function which returns one (see <a href="#">border.functions</a> ).
size.function	A named vector of node radii (in inches) for the terms in the plot or a function which returns one (see <a href="#">size.functions</a> ).
main.title	Title for the plot
draw.legend	Boolean to decide whether legend is drawn. NOTE: The legend drawn may intersect the plot.
filter.out.uninformative	Filter out uninformative nodes (i.e. nodes add no new information to the plot, given the presence of their children).
filter.out.non.pa.terms	Filter out nodes which are not descendants of the ‘Phenotypic abnormality’ HPO-term.
min.occs	Filter out nodes which occur less than ‘min.occs’ times in ‘patients’.
nodeAttrs	A custom Rgraphviz node attributes object which overrides any other graphical parameters passed to this function. Use this to set font size, font colour and more (for a full list see the Rgraphviz documentation).
pdf.file.name	File name. If null will plot to the current graphical device.
pdf.height	Height in inches of PDF created.
pdf.width	Width in inches of PDF created (if left NULL, the function will try to set this based on the pdf.height parameter and the number of leaves in the plot, so that the graph looks legible).

**Examples**

```
data(hpo)

patients.character.vector <- 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"
)

patients.of.interest <- term.set.list.from.character(hpo.terms, patients.character.vector)

hpo.graph(
  hpo.terms=hpo.terms,
  patients=patients.of.interest,
  main.title="Test Plot",
  filter.out.uninformative=TRUE
)
```

---

hpo.terms

*HPO R-Object*

---

**Description**

An R-object representing the HPO, and a required argument to many functions in this package.

**Usage**

```
data(hpo)
```

**Format**

See [get.hpo.terms](#).

**Examples**

```
data(hpo)
```

---

labelling.functions    *List Of Labelling Functions*

---

**Description**

A list of functions which automatically generate labels for nodes representing HPO terms in plots generated by [hpo.graph](#).

**Format**

The format is: List of 5 \$ get.frequency.based.labels :function (hpo.terms, plotting.context) \$ get.simple.node.labels :function (hpo.terms, plotting.context) \$ get.informative.node.labels: function (hpo.terms, plotting.context) \$ get.patient.based.labels :function (hpo.terms, plotting.context) \$ get.code.node.labels :function (hpo.terms, plotting.context)

---

remove.duplicate.term.sets  
*Remove Duplicate Term Sets*

---

**Description**

Filters a list of character vectors of HPO terms so that there are no two equivalent.

**Usage**

```
remove.duplicate.term.sets(hpo.terms, hpo.terms.set)
```

**Arguments**

hpo.terms        An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
hpo.terms.set   List of character vectors of HPO terms.

**Value**

List of character vectors of HPO terms.

**Examples**

```
data(hpo)

patients.character.vector <- c(
  Term.Set.A=
  "HP:0001382,HP:0004272,HP:0004912,HP:0007917",
  Term.Set.B=
  "HP:0001382",
  Duplicate.Of.Term.Set.A=
```

```
"HP:0001382,HP:0004272,HP:0007917,HP:0004912"
)

patient.hpo.terms <- term.set.list.from.character(hpo.terms, patients.character.vector)

no.dupes <- remove.duplicate.term.sets(hpo.terms, patient.hpo.terms)
```

---

```
remove.uninformative.terms
```

*Remove Uninformative Terms From Set*

---

### Description

Removes terms which don't add information about which terms people represented in the plot are annotated with.

### Usage

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

### Arguments

`hpo.terms` An R-Object representing the HPO, as returned by [get.hpo.terms](#).

`patient.hpo.terms` A named list of HPO term character vectors representing the phenotypes of a set of patients.

### Value

A character vector of the HPO terms which are informative in the context of the patients' phenotypes, in the sense that the frequency can't be deduced from the frequencies of its child terms alone.

### Examples

```
data(hpo)

patients.character.vector <- 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"
)

patients <- term.set.list.from.character(hpo.terms, patients.character.vector)

informative.terms <- remove.uninformative.terms(hpo.terms, patients)
```

---

setDimNames	<i>Set Names For Dimensions Of Array</i>
-------------	--

---

**Description**

A higher dimensional version of [setNames](#).

**Usage**

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

**Arguments**

`array.object`    Array to be named  
`list.of.dimension.names`  
                  A list of character vectors whose lengths are the same as `dim(array.object)`.

**Value**

Named array

**Examples**

```
setDimNames(  
  matrix(1,2,2),  
  list(  
    c("Row1", "Row2"),  
    c("Col1", "Col2")  
  )  
)
```

---

size.functions	<i>List Of Node-Sizing Functions</i>
----------------	--------------------------------------

---

**Description**

A list of functions which set sizes of nodes representing HPO terms in plots generated by [hpo.graph](#).

**Format**

The format is: List of 3 \$ `get.significance.based.sizes:function (hpo.terms, plotting.context) $ get.frequency.based.sizes :function (hpo.terms, plotting.context) $ get.standard.sizes :function (hpo.terms, plotting.context)`

---

```
suggest.aspect.ratio.width.to.height
```

*Suggest Aspect Ratio For Plot*

---

### Description

Suggests a width to height aspect ratio for an HPO term plot based on the number of terms which are to be plotted.

### Usage

```
suggest.aspect.ratio.width.to.height(hpo.terms, list.of.patients.or.vector.of.terms)
```

### Arguments

`hpo.terms`            An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
`list.of.patients.or.vector.of.terms`  
                          List of vectors of terms or vector of terms which are to be plotted.

### Value

Scalar value of suggested aspect ratio.

### Examples

```
data(hpo)

patients.character.vector <- 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"
)

patients.of.interest <- term.set.list.from.character(hpo.terms, patients.character.vector)

height=10

# pdf(
# "test.pdf",
# height=height,
# width=height * suggest.aspect.ratio.width.to.height(
# hpo.terms,
# patients.of.interest
# )
# )
#
# hpo.graph(
```



```
# hpo.terms=hpo.terms,  
# patients=patients.of.interest,  
# main.title="Simple Example",  
# filter.out.uninformative=TRUE  
# )  
#  
# dev.off()
```

---

swap.out.alt.ids	<i>Swap Out Alternative/Previous IDs</i>
------------------	--

---

### Description

Function to replace the terms in a set with their updated/replacement IDs (if they exist) in the given HPO object.

### Usage

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

### Arguments

hpo.terms	An R-Object representing the HPO, as returned by <a href="#">get.hpo.terms</a> .
hpo.term.set	Character vector of HPO terms.
remove.dead	Boolean to dictate whether to remove terms which are not in the HPO object given. Defaults to FALSE.

### Value

Character vector of HPO terms.

### Examples

```
data(hpo)  
old.code.for.migraine <- "HP:0007194"  
current.code.for.migraine <- swap.out.alt.ids(hpo.terms, old.code.for.migraine)
```

---

`term.set.list.from.character`*Get Term Set List From Character Vector*

---

**Description**

Function which returns a list of HPO term sets (typically to represent patients and used by functions in this package) given a character vector of comma separated HPO terms.

**Usage**

```
term.set.list.from.character(hpo.terms, character.vector, remove.empty = FALSE)
```

**Arguments**

`hpo.terms` An R-Object representing the HPO, as returned by [get.hpo.terms](#).  
`character.vector` A character vector of comma separated lists of HPO terms  
`remove.empty` Boolean to dictate whether to remove sets with no HPO terms.

**Value**

List of character vectors of HPO terms.

**Examples**

```
data(hpo)

patients.character.vector <- 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"
)

patients.of.interest <- term.set.list.from.character(hpo.terms, patients.character.vector)
```

# Index

- \*Topic **HPO**
  - get.hpo.terms, 6
  - hpo.terms, 12
- \*Topic **adjacency**
  - get.term.adjacency.matrix, 8
- \*Topic **ancestors**
  - get.ancestors, 4
- \*Topic **arrays**
  - setDimNames, 15
- \*Topic **aspect-ratio**
  - suggest.aspect.ratio.width.to.height, 16
- \*Topic **border**
  - border.functions, 3
- \*Topic **colour**
  - colouring.functions, 4
- \*Topic **datasets**
  - border.functions, 3
  - colouring.functions, 4
  - hpo.terms, 12
  - labelling.functions, 13
  - size.functions, 15
- \*Topic **descendants**
  - get.descendants, 6
- \*Topic **duplicates**
  - remove.duplicate.term.sets, 13
- \*Topic **frequency**
  - get.cohort.information, 5
- \*Topic **information**
  - get.cohort.information, 5
- \*Topic **labelling**
  - labelling.functions, 13
- \*Topic **ontology**
  - get.hpo.terms, 6
- \*Topic **package**
  - hpoPlot-package, 2
- \*Topic **redundancy**
  - clean.terms, 3
- \*Topic **size**
  - size.functions, 15
- \*Topic **uninformative**
  - remove.uninformative.terms, 14
- border.functions, 3, 11
- clean.terms, 3
- colouring.functions, 4, 11
- get.ancestors, 4
- get.cohort.information, 5, 11
- get.descendants, 6
- get.hpo.terms, 4-6, 6, 8, 11-14, 16-18
- get.node.friendly.long.names, 7
- get.term.adjacency.matrix, 8
- get.term.patient.matrix, 9
- get.term.pseudo.adjacency.matrix (get.term.adjacency.matrix), 8
- hpo.graph, 3, 4, 10, 13, 15
- hpo.terms, 12
- hpoPlot (hpoPlot-package), 2
- hpoPlot-package, 2
- labelling.functions, 11, 13
- remove.duplicate.term.sets, 13
- remove.uninformative.terms, 14
- setDimNames, 15
- setNames, 15
- size.functions, 11, 15
- suggest.aspect.ratio.width.to.height, 16
- swap.out.alt.ids, 17
- term.set.list.from.character, 18