

Package ‘`ontoProc`’

April 12, 2018

Title processing of ontologies of anatomy, cell lines, and so on

Description Support harvesting of diverse bioinformatic ontologies, making particular use of the `ontologyIndex` package on CRAN. We provide snapshots of key ontologies for terms about cells, cell lines, chemical compounds, and anatomy, to help analyze genome-scale experiments, particularly cell x compound screens. Another purpose is to strengthen development of compelling use cases for richer interfaces to emerging ontologies.

Version 1.0.0

Author Vince Carey <stvjc@channing.harvard.edu>

Imports Biobase, S4Vectors, methods, AnnotationDbi, stats, utils, shiny

Suggests knitr, org.Hs.eg.db, org.Mm.eg.db, testthat

Depends R (>= 3.4), ontologyIndex

Maintainer VJ Carey <stvjc@channing.harvard.edu>

License Artistic-2.0

LazyLoad yes

biocViews Infrastructure, GO

RoxygenNote 6.0.1.9000

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

allGOterms	2
c,TermSet-method	2
cellTypeToGO	3
demoApp	4
fastGrep	4
getCellOnto	5
makeSelectInput	6
secLevGen	6
siblings_TAG	7
TermSet-class	8
Index	9

allGOterms	<i>allGOterms: data.frame with ids and terms</i>
------------	--

Description

allGOterms: data.frame with ids and terms

Usage

```
allGOterms
```

Format

data.frame instance

Source

This is a snapshot of all the terms available from GO.db (3.4.2), August 2017, using keys(GO.db, keytype="TERM").

Examples

```
data(allGOterms)
head(allGOterms)
```

c, TermSet-method	<i>combine TermSet instances</i>
-------------------	----------------------------------

Description

combine TermSet instances

Usage

```
## S4 method for signature 'TermSet'
c(x, ...)
```

Arguments

x	TermSet instance
...	additional instances

Value

TermSet instance

cellTypeToGO	<i>utilities for approximate matching of cell type terms to GO categories and annotations</i>
--------------	---

Description

utilities for approximate matching of cell type terms to GO categories and annotations

Usage

```
cellTypeToGO(celltypeString, gotab, ...)  
  
cellTypeToGenes(celltypeString, gotab, orgDb, cols = c("ENSEMBL", "SYMBOL"),  
  ...)
```

Arguments

celltypeString	character atom to be used to search GO terms using
gotab	a data.frame with columns GO (goids) and TERM (term strings) agrep
...	additional arguments to agrep
orgDb	instances of orgDb
cols	columns to be retrieved in select operation

Value

data.frame
data.frame

Note

Very primitive, uses [agrep](#) to try to find relevant terms.

Examples

```
data(allGOterms)  
library(org.Hs.eg.db)  
head(cellTypeToGO("serotonergic neuron", allGOterms))  
head(cellTypeToGenes("serotonergic neuron", allGOterms, org.Hs.eg.db))
```

demoApp	<i>demonstrate the use of makeSelectInput</i>
---------	---

Description

demonstrate the use of makeSelectInput

Usage

```
demoApp()
```

Value

Run only for side effect of starting a shiny app.

Examples

```
if (interactive()) {
  require(shiny)
  print(demoApp())
}
```

fastGrep	<i>some fields of interest are lists, and grep per se should not be used – this function checks and uses grep within vapply when appropriate</i>
----------	--

Description

some fields of interest are lists, and grep per se should not be used – this function checks and uses grep within vapply when appropriate

Usage

```
fastGrep(patt, onto, field, ...)
```

Arguments

patt	a regular expression whose presence in field should be checked
onto	an ontologyIndex instance
field	the ontologyIndex component to be searched
...	passed to grep

Value

logical vector indicating vector or list elements where a match is found

Examples

```
cheb = getChebiOnto()
ind = fastGrep("17-AAG", cheb, "synonym")
cheb$name[ind]
```

getCellOnto	<i>load ontologies that may include non-ascii strings and therefore cannot be in data folder</i>
-------------	--

Description

load ontologies that may include non-ascii strings and therefore cannot be in data folder

Usage

```
getCellOnto()  
  
getCellLineOnto()  
  
getEF0Onto()  
  
getChebiLite()  
  
getCellosaurusOnto()  
  
getUBERON_NE()  
  
getChebiOnto()
```

Value

instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex
instance of ontology_index (S3) from ontologyIndex

Examples

```
co = getCellOnto()  
co  
clo = getCellLineOnto()  
length(clo$id)  
che = getChebiLite()  
length(che$id)  
efo = getEF0Onto()  
length(efo$id)
```

makeSelectInput	<i>generate a selectInput control for an ontologyIndex slice</i>
-----------------	--

Description

generate a selectInput control for an ontologyIndex slice

Usage

```
makeSelectInput(onto, term, type = "siblings", inputId, label,
  multiple = TRUE, ...)
```

Arguments

onto	ontologyIndex instance
term	character(1) term used as basis for term list option set in the control
type	character(1) 'siblings' or 'children', relationship to 'term' that the options will satisfy
inputId	character(1) for use in server
label	character(1) for labeling in ui
multiple	logical(1) passed to selectInput
...	additional parameters passed to selectInput

Value

a [selectInput](#) control

Examples

```
makeSelectInput
```

secLevGen	<i>simple generation of children of 'choices' given as terms, returned as TermSet</i>
-----------	---

Description

simple generation of children of 'choices' given as terms, returned as TermSet

Usage

```
secLevGen(choices, ont)
```

Arguments

choices	vector of terms
ont	instance of ontology_index (S3) from ontologyIndex package

Value

TermSet instance

Examples

```
efoOnto = getEF0Onto()
secLevGen( "disease", efoOnto )
```

siblings_TAG	<i>generate a TermSet with siblings of a given term, excluding that term by default</i>
--------------	---

Description

generate a TermSet with siblings of a given term, excluding that term by default
 acquire the label of an ontology subject tag
 acquire the labels of children of an ontology subject tag

Usage

```
siblings_TAG(Tagstring = "EFO:1001209", ontology, justSibs = TRUE)
label_TAG(Tagstring = "EFO:0000311", ontology)
children_TAG(Tagstring = "EFO:1001209", ontology)
```

Arguments

Tagstring	a character(1) that identifies a term
ontology	instance of ontology_index (S3) from ontologyIndex
justSibs	character(1)

Value

TermSet instance
 character(1)
 TermSet instance

Note

for label_TAG, Tagstring may be a vector

Examples

```
efoOnto = getEF0Onto()
siblings_TAG( "EFO:1001209", efoOnto )
efoOnto = getEF0Onto()
label_TAG( "EFO:0000311", efoOnto )
efoOnto = getEF0Onto()
children_TAG( ontology = efoOnto )
```

TermSet-class	<i>manage ontological data with tags and a DataFrame instance</i>
---------------	---

Description

manage ontological data with tags and a DataFrame instance
abbreviated display for TermSet instances

Usage

```
## S4 method for signature 'TermSet'  
show(object)
```

Arguments

object instance of TermSet class

Value

instance of TermSet

Examples

```
efoOnto = getEFOnto()  
defsibs = siblings_TAG("EFO:1001209", efoOnto)  
class(defsibs)  
defsibs
```


Index

*Topic **datasets**

allGOterms, [2](#)

agrep, [3](#)

allGOterms, [2](#)

c, TermSet-method, [2](#)

cellTypeToGenes (cellTypeToGO), [3](#)

cellTypeToGO, [3](#)

children_TAG (siblings_TAG), [7](#)

demoApp, [4](#)

fastGrep, [4](#)

getCellLineOnto (getCellOnto), [5](#)

getCellOnto, [5](#)

getCellosaurusOnto (getCellOnto), [5](#)

getChebiLite (getCellOnto), [5](#)

getChebiOnto (getCellOnto), [5](#)

getEFOnto (getCellOnto), [5](#)

getUBERON_NE (getCellOnto), [5](#)

label_TAG (siblings_TAG), [7](#)

makeSelectInput, [6](#)

secLevGen, [6](#)

selectInput, [6](#)

show (TermSet-class), [8](#)

show, TermSet-method (TermSet-class), [8](#)

siblings_TAG, [7](#)

TermSet-class, [8](#)