

Package ‘AtlasRDF’

September 17, 2013

Type Package

Title Gene Expression Atlas query and gene set enrichment package.

Version 1.0

Date 2013-09-09

Author James Malone, Simon Jupp, Maryam Soleimani

Maintainer James Malone <malone@ebi.ac.uk>

Depends hash, SPARQL, methods

Description Query the Gene Expression Atlas RDF data at the European Bioinformatics Institute using genes, experimental factors (such as disease, cell type, compound treatments), pathways and proteins. Also contains a function to perform an enrichment of your gene list across Experimental Factor Ontology (EFO) using the Atlas background set. Note: use the R SPARQL package available at <http://cran.r-project.org/web/packages/SPARQL/>

License Apache 2.0

R topics documented:

AtlasRDF-package	2
doFishersEnrichment	3
doFishersEnrichmentForEnsemblIds	5
doFishersEnrichmentForGeneNames	6
drawHeatMapForAtlasExperiment	8
excludeSubclasses	8
getAllEnsemblGenesForExFactor	9
getExFactorURIFromLabel	10
getExperimentsByDescription	11
getGeneListFromPubmedid	11
getGenesForExperimentID	12
getGenesForExperimentURI	13
getGeneUriFromEnsemblId	14
getGeneUriFromName	14
getLabel	15
getOntologyMappings	16
getPathwaysFromGenesAndCondition	16
getPathwayUriFromName	17

getSpeciesSpecificEnsemblGenesForExFactor	18
getTaxonURI	19
includeOnlySubclasses	19
vizPvalues	20

Index	21
--------------	-----------

AtlasRDF-package *Gene Expression Atlas query and gene set enrichment package.*

Description

Query the Gene Expression Atlas RDF data at the European Bioinformatics Institute using genes, experimental factors (such as disease, cell type, compound treatments), pathways and proteins. Also contains a function to perform an enrichment of your gene list across Experimental Factor Ontology (EFO) using the Atlas background set.

Details

Package: AtlasRDF
 Type: Package
 Version: 1.0
 Date: 2013-09-09
 License: Apache 2.0

Note

Linux users may need to perform the following:

```
sudo apt-get install libxml2-dev
from R environment: install.package("XML")
sudo apt-get install libcurl4-gnutls-dev
from R environment: install.package("RCurl")
from R environment: install.package("hash")
from R environment: install.package("SPARQL")
```

Author(s)

James Malone, Simon Jupp, Maryam Soleimani
 Maintainer: James Malone <malone@ebi.ac.uk>

References

Gene Expression Atlas RDF - <http://wwwdev.ebi.ac.uk/rdf/services/atlas/> Gene Expression Atlas - <http://www.ebi.ac.uk/gxa> Experimental Factor Ontology - <http://www.ebi.ac.uk/efo> Kapushesky M, Adamusiak T, Burdett T, Culhane A, Farne A, Filippov A, Holloway E, Klebanov A, Kryvych N, Kurbatova N, Kurnosov P, Malone J, Melnichuk O, Petryszak R, Pultsin N, Rustici G, Tikhonov

A, Travillian RS, Williams E, Zorin A, Parkinson H, Brazma A. (2012) Gene Expression Atlas update—a value-added database of microarray and sequencing-based functional genomics experiments. Nucleic Acids Res. 2012 Jan;40(Database issue):D1077-81.

Examples

```
###find experiments where the description contains a given search term
#experimentlist <- getExperimentsByDescription("breast carcinoma")

###get gene list for given experiment ID
#genelist <- getGenesForExperiment("E-GEOD-1561")

###get list of pathways associated with a gene for which the genes are differentially
#pathwayinfo <- getPathwaysFromGenesAndCondition("efo:EFO_0001359")
#pathwayinfo$pathwayname

###perform gene set enrichment using Atlas as background to find factors genes are enriched
#load the background files for the species of interest (requires downloading)
#find background sets at http://
#load("human/human_gene_list.RData")    #human_genelist_bg
#load("human/human_factor_counts.RData")    #human_factor_counts

###creat or load your genelist of interest
#genelist <- c("GSTT1", "PLCE1", "XIST", "ENOSF-1", "AHR", "FSTL1", "YIPF5", "RBM5",

###get id for species 'human'
#taxon <-getTaxonURI("human")    #obo:NCBITaxon_9606

###do enrichment
#results <- doFishersEnrichmentForGeneNames(genelist, taxon, human_genelist_bg, human_factor_counts)

###visualize results, filtering by p-value
#vizPvalues(results, "0.005")
```

doFishersEnrichment

Perform a gene set enrichment using gene list based on gene URIs.

Description

Perform an enrichment of an input gene list using the Atlas as a background set for a specified species. The genes are given as a set of gene URIs (e.g. <<http://identifiers.org/ensembl/ENSG00000229807>>). The results are a list of experimental factors for which these genes are enriched.

Usage

```
doFishersEnrichment(genelist, genelist_bg, genecounts)
```

Arguments

genelist	Variable genelist a vector of genes using gene URIs
genelist_bg	genelist_bg background list appropriate for the species of interest should be obtained from svn
genecounts	genecounts counts list appropriate for the species of interest should be obtained from svn

Value

Returns a list of enrichmentresult objects which are described as follows:

factoruri	the URI of the experimental factor
label	the human readable label for the experimental factor
p.value	the p.value for the Fisher's exact test for the gene list with this factor. This can be seen as evidence for this factor being highly enriched (low p value) or not (high p value) for the input gene list
estimate	an estimate of the odds ratio. Note that the conditional Maximum Likelihood Estimate (MLE) rather than the unconditional MLE (the sample odds ratio) is used. Only present in the 2 by 2 case.
alternative	a character string describing the alternative hypothesis.
null.value	the odds ratio under the null, or. Only present in the 2 by 2 case.
method	the character string "Fisher's Exact Test for Count Data".
enrichedgenes	list of the genes from the input gene list that were enriched for this factor as a list of Gene URIs. To get the gene names use the function getClassLabel

Author(s)

James Malone, European Bioinformatics Institute Simon Jupp, European Bioinformatics Institute
Maryam Soleimani, European Bioinformatics Institute

References

Agresti, A. (1990) Categorical data analysis. New York: Wiley. Pages 59-66.

Examples

```
###perform gene set enrichment using Atlas as background to find factors genes are enriched
#load the background files for the species of interest (requires downloading)
#load("human/human_gene_list.RData")    #human_genelist_bg
#load("human/human_factor_counts.RData")    #human_factor_counts

###creat or load your genelist of interest
genelist <- c("<http://identifiers.org/ensembl/ENSG00000184674>",
"<http://identifiers.org/ensembl/ENSG00000138193>",
"<http://identifiers.org/ensembl/ENSG00000229807>",
"<http://identifiers.org/ensembl/ENSG00000106546>",
"<http://identifiers.org/ensembl/ENSG00000163430>", "<http://identifiers.org/ensembl/ENSG00000177731>")

###do enrichment
#results <- doFishersEnrichment(genelist, human_genelist_bg, human_factor_counts)
```

doFishersEnrichmentForEnsemblIds

Perform a gene set enrichment using gene list based on common ENSEMBL gene IDs.

Description

Perform an enrichment of an input gene list using the Atlas as a background set for a specified species. The genes are given as a set of ENSEMBL gene IDs (e.g. "ENSG00000188554"). The results are a list of experimental factors for which these genes are enriched.

Usage

```
doFishersEnrichmentForEnsemblIds(geneids, taxon, genelist_bg, genecounts)
```

Arguments

geneids	Variable geneids a vector of genes using ENSEMBL gene IDs
taxon	The taxon is an ontology URI which can be found using the function <code>getTaxonURI()</code>
genelist_bg	genelist_bg background list appropriate for the species of interest should be obtained from svn
genecounts	genecounts counts list appropriate for the species of interest should be obtained from svn

Value

Returns a list of enrichmentresult objects which are described as follows:

factoruri	the URI of the experimental factor
label	the human readable label for the experimental factor
p.value	the p.value for the Fisher's exact test for the gene list with this factor. This can be seen as evidence for this factor being highly enriched (low p value) or not (high p value) for the input gene list
estimate	an estimate of the odds ratio. Note that the conditional Maximum Likelihood Estimate (MLE) rather than the unconditional MLE (the sample odds ratio) is used. Only present in the 2 by 2 case.
alternative	a character string describing the alternative hypothesis.
null.value	the odds ratio under the null, or. Only present in the 2 by 2 case.
method	the character string "Fisher's Exact Test for Count Data".
enrichedgenes	list of the genes from the input gene list that were enriched for this factor as a list of Gene URIs. To get the gene names use the function <code>getClassLabel</code>

Author(s)

James Malone, European Bioinformatics Institute Simon Jupp, European Bioinformatics Institute Maryam Soleimani, European Bioinformatics Institute

References

Agresti, A. (1990) Categorical data analysis. New York: Wiley. Pages 59-66.

Examples

```
###perform gene set enrichment using Atlas as background to find factors genes are enriched
#load the background files for the species of interest (requires downloading)
#load("human_bg/human_gene_list.RData")      #human_genelist_bg
#load("human_bg/human_factor_counts.RData")    #human_factor_counts

###creat or load your genelist of interest
genelist <- c("ENSG00000184674", "ENSG00000138193", "ENSG00000229807", "ENSG00000254235",
              "ENSG00000163430", "ENSG00000118260", "ENSG00000086991")

###get id for species 'human'
#taxon <-getTaxonURI("human")    #obo:NCBITaxon_9606

###do enrichment
#results <- doFishersEnrichmentForEnsemblIds(genelist, taxon, human_genelist_bg, human_factor_counts)
```

doFishersEnrichmentForGeneNames

Perform a gene set enrichment using gene list based on common gene names.

Description

Perform an enrichment of an input gene list using the Atlas as a background set for a specified species. The genes are given as a set of common gene names (e.g. Brcal). The results are a list of experimental factors for which these genes are enriched.

Usage

```
doFishersEnrichmentForGeneNames(genenames, taxon, genelist_bg, genecounts, endpoint)
```

Arguments

genenames	Variable genenames a vector of genes using common gene names
taxon	The taxon is an ontology URI which can be found using the function <code>getTaxonURI()</code>
genelist_bg	genelist_bg background list appropriate for the species of interest should be obtained from svn
genecounts	genecounts counts list appropriate for the species of interest should be obtained from svn
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden. Required to find Gene URIs from gene names

Value

Returns a list of enrichmentresult objects which are described as follows:

factoruri	the URI of the experimental factor
label	the human readable label for the experimental factor
p.value	the p.value for the Fisher's exact test for the gene list with this factor. This can be seen as evidence for this factor being highly enriched (low p value) or not (high p value) for the input gene list
estimate	an estimate of the odds ratio. Note that the conditional Maximum Likelihood Estimate (MLE) rather than the unconditional MLE (the sample odds ratio) is used. Only present in the 2 by 2 case.
alternative	a character string describing the alternative hypothesis.
null.value	the odds ratio under the null, or. Only present in the 2 by 2 case.
method	the character string "Fisher's Exact Test for Count Data".
enrichedgenes	list of the genes from the input gene list that were enriched for this factor as a list of Gene URIs. To get the gene names use the function getClassLabel

Author(s)

James Malone, European Bioinformatics Institute Simon Jupp, European Bioinformatics Institute
Maryam Soleimani, European Bioinformatics Institute

References

Agresti, A. (1990) Categorical data analysis. New York: Wiley. Pages 59-66.

Examples

```
###perform gene set enrichment using Atlas as background to find factors genes are enriched
#load the background files for the species of interest (requires downloading)
#load("human/human_gene_list.RData") #human_genelist_bg
#load("human/human_factor_counts.RData") #human_factor_counts

###creat or load your genelist of interest
genelist <- c("GSTT1", "PLCE1", "XIST", "ENOSF-1", "AHR", "FSTL1", "YIPF5", "RBM5", "

###get id for species 'human'
taxon <-getTaxonURI("human") #obo:NCBITaxon_9606

###do enrichment
#results <- doFishersEnrichmentForGeneNames(genelist, taxon, human_genelist_bg, human
```

```
drawHeatMapForAtlasExperiment
```

Produce a heatmap of differentially expressed genes vs conditions for an Atlas experiment.

Description

Draws a heatmap of genes that are differentially expressed beyond a given t stat significance against conditions they are diff expressed for for a given Atlas experiment.

Usage

```
drawHeatMapForAtlasExperiment(experimentid, tstatsignificance, endpoint)
```

Arguments

`experimentid` Atlas experiment ID which is to be visualized

`tstatsignificance`

The significance beyond which a gene should be included for visualisation. A significance value of '5' will only include genes with a t-stat of ≥ 5 or ≤ -5 .

`endpoint` SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

matrix of gene names with t-statistics for level of differential expression for each factor (one factor per column)

Author(s)

Simon Jupp, James Malone

Examples

```
#data <- drawHeatMapForAtlasExperiment("E-MTAB-1670", 12)
```

```
excludeSubclasses
```

Exclude subclasses of factors from enrichment results

Description

Removes all factors which are subclasses of a specified type from Atlas gene set enrichment results. For instance, if all chemical compounds were to be excluded form the enrichment results or all subclasses of cancer use this function.

Usage

```
excludeSubclasses(filterparentclass, resultset, endpoint="http://www.ebi.ac.uk/r
```


Arguments

filterparentclass	The class from which all subclasses will be excluded - including this class.
resultset	Result set from an Atlas gene set enrichment.
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

new enrichment result set with the specified class and subclasses excluded.

Author(s)

James Malone, Simon Jupp

Examples

```
####filter out all subclasses of chemical compounds from the result set
####enrichment results has been produced using the enrichment functions in AtlasRDF pack
#filteredresults <- excludeSubclasses("obo:CHEBI_37577", enrichmentresults)
#vizPvalues(filteredresults, 0.01)
```

```
getAllEnsemblGenesForExFactor
```

Get all ENSEMBL genes reported as differentially expressed for a given experimental factor.

Description

Get all ENSEMBL genes reported as differentially expressed for a given experimental factor. Returns the gene URI, ENSEMBL Id and common gene name.

Usage

```
getAllEnsemblGenesForExFactor(exfactor, limit, endpoint)
```

Arguments

exfactor	The URI of the experimental factor e.g. "efo:EFO_0001420"
limit	Limit the number of genes returned by the query to this, no limit specified or value of 0 returns all.
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

Data frame describing genes for the specified experimental factor with the following slots:

dbXref	the URI of the gene
geneName	the common name of the gene
ensemblid	the ENSEMBL id of the gene
propertyValue	the property value for this gene corresponding to the experimental factor in the query

Author(s)

James Malone, Simon Jupp

Examples

```
###get the genes for the experimental factor 'sepsis' (efo:EFO_0001420)
#results <- getAllEnsemblGenesForExFactor("efo:EFO_0001420")
#results[1,] #first gene result
```

```
getExFactorURIFromLabel
```

Get the EFO URI from a given label

Description

Find the EFO ontology class URI for a given search label.

Usage

```
getExFactorURIFromLabel(label, endpoint="http://www.ebi.ac.uk/rdf/services/a
```

Arguments

label	Search string to find the EFO class URI by
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

EFO class URI for the given search term

Author(s)

James Malone, Simon Jupp

Examples

```
####find the efo class uri for "asthma"
#getExFactorURIFromLabel("asthma") #<http://www.ebi.ac.uk/efo/EFO_0000270>
```

```
getExperimentsByDescription
```

Search for experiments in Atlas based on a string, e.g. liver, cancer etc.

Description

Find experiments in Atlas based on a search string such as a "liver" or "cancer". Returns the experiment IDs plus the text in which the search string was found.

Usage

```
getExperimentsByDescription(searchterm, limit = 0, endpoint = "http://www.ebi.ac.uk/ontology/atlas/terms")
```

Arguments

searchterm	The search term.
limit	Optional limit on number of returned results (default is everything)
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

\$experiment	Atlas Experiment ID of matched experiment to search term'
\$description	Description of experiment which matched term

Author(s)

James Malone, Simon Jupp

Examples

```
###get experiments which contain the word liver in the description
#getExperimentsByDescription("liver")
```

```
getGeneListFromPubmedid
```

Get gene lists in Atlas for a given pubmed id (assuming the data is in Atlas)

Description

Get differentially expressed genes reported in an experiment in Atlas for which a given pubmed id is linked (assuming the data is in Atlas)

Usage

```
getGeneListFromPubmedid(searchid, endpoint = "http://www.ebi.ac.uk/rdf/services/ontology/atlas/terms")
```

Arguments

searchid	PubMed ID to find data for
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

List of genes for experiment (via PubMed ID)

Author(s)

James Malone, Simon Jupp

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--      or do help(data=index) for the standard data sets.

#getGeneListFromPubmedid("19668377")
```

```
getGenesForExperimentID
```

Get all of the genes reported in an experiment, specified by the Atlas experiment ID.

Description

Get all of the gene as reported within an experiment. The experiment is specified as an Atlas experiment ID e.g. E-GEOD-3865

Usage

```
getGenesForExperimentID(experiment, endpoint = "http://www.ebi.ac.uk/rdf/service")
```

Arguments

experiment	Atlas experiment ID.
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

data frame containing following elements

expressionValue	The summarised expression value for this gene
dbXref	The URI identifier for this gene
pvalue	The p-value representing differential expression for this gene
propertyValue	The property value for which this gene was reported as differentially expressed

Author(s)

James Malone, Simon Jupp

Examples

```
###get genes for experiment ID E-MTAB-921
#
#res <- getGenesForExperimentID("E-MTAB-921")
###list first gene in the experiment
#res[1,]
```

getGenesForExperimentURI

Get all of the genes reported in an experiment, speicified by the Atlas experiment URI.

Description

Get all of the gene as reported within an experiment. The experiment is specified as an Atlas experiment URI e.g. <<http://rdf.ebi.ac.uk/resource/atlas/E-GEOD-13396>>

Usage

```
getGenesForExperimentURI(experiment, endpoint = "http://www.ebi.ac.uk/rdf/service")
```

Arguments

experiment	Atlas experiment URI.
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

data frame containg following elements

expressionValue

The summarised expression value for this gene

dbXref

The URI identifier for this gene

pvalue

The p-value representing differentially expression for this gene

propertyValue

The property value for which this gene was reported as differentially expressed

Author(s)

James Malone, Simon Jupp

Examples

```
###get genes for experiment <http://rdf.ebi.ac.uk/resource/atlas/E-GEOD-13396>
#
#res <- getGenesForExperimentID("<http://rdf.ebi.ac.uk/resource/atlas/E-GEOD-13396>")
###list first gene in the experiment
#res[1,]
```

```
getGeneUriFromEnsemblId
```

Get gene URI from an ENSEMBL ID.

Description

Get the URI for an ENSEMBL gene based on the ENSEMBL ID.

Usage

```
getGeneUriFromEnsemblId(id, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/
```

Arguments

id	ENSEMBL gene ID
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

List of URIs for a given ENSEMBL ID - this is almost always one but it is feasible there are two so be aware.

Author(s)

James Malone, Simon Jupp

Examples

```
###get the URI identifier for a gene based on the ENSEMBL ID
#getGeneUriFromEnsemblId("ENSMUSG00000031314")
```

```
getGeneUriFromName Get URI of a gene based on the common gene name.
```

Description

Get the URI of an ENSEMBL gene based on a search by the common gene name.

Usage

```
getGeneUriFromName(genename, taxon, endpoint = "http://www.ebi.ac.uk/rdf/ser
```

Arguments

genename	The common gene name by which to search
taxon	The URI of the taxon to search genes for, use function getTaxonURI to get the URI
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

list of gene URIs matching the query: note this can be more than one

Author(s)

James Malone, Simon Jupp

Examples

```
###get the gene URI identifier based on the common gene name for "SRGN" in human
#getGeneUriFromName("SRGN", "obo:NCBITaxon_9606")
```

getLabel	<i>Get label of an entity based on the URI.</i>
----------	---

Description

Get label of an entity base on the URI

Usage

```
getLabel(uri, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

Arguments

uri	URI of the class for which the label is required.
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

List of labels for the class URI

Author(s)

James Malone, Simon Jupp

Examples

```
###get the label for the class efo:EFO_0001185 (HeLa)
#getLabel("efo:EFO_0001185")
```

getOntologyMappings

Get mappings for a given ontology class URI to EFO using the NCBO BioPortal

Description

Get mappings for a given class URI from an ontology other than EFO using the NCBO BioPortal. Function is useful for querying the Atlas RDF using your own ontology. For instance, ICD-10 or SnoMed. The function will find mappings between these classes in other ontologies do EFO where they exist. The returned EFO class can be used to query Atlas.

Usage

```
getOntologyMappings(searchuri, endpoint = "http://www.ebi.ac.uk/rdf/services
```

Arguments

searchuri	The ontology class URI for which mappings to EFO are required.
endpoint	SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

List of EFO URI(s) to which the class maps

Author(s)

James Malone, Simon Jupp

References

NCBO BioPortal - <http://bioportal.bioontology.org/>

Examples

```
###find mappings in EFO from <http://purl.bioontology.org/ontology/SNOMEDCT/87163000> (
#getOntologyMappings("<http://purl.bioontology.org/ontology/SNOMEDCT/87163000>")
```

getPathwaysFromGenesAndCondition

Get pathways connected to genes which are differentially expressed for a user specified condition.

Description

Get list of pathway names that are connected to genes (via proteins which are encoded for by those genes) and which are reported as differentially expressed under a specified condition. The condition is specified as an ontology URI.

Usage

```
getPathwaysFromGenesAndCondition(condition, endpoint = "http://www.ebi.ac.uk
```

Arguments

condition	URI of a condition based on the EFO ontology class.
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

Returns a data frame with the following:

pathwayname	The name of the pathway
pathway	The URI identifier of the pathway
expressionvalue	The gene and expression for the pathway
pvalue	p-value of the differential expression of the gene for this pathway in this condition

Author(s)

James Malone, Simon Jupp

Examples

```
###get pathways connected to genes which are differentially expressed for HeLa (efo:E
#getPathwaysFromGenesAndCondition("efo:EFO_0001185")
```

```
getPathwayUriFromName
```

Get pathway URI given a pathway name

Usage

```
getPathwayUriFromName(id, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sp
```

Arguments

id
endpoint

Author(s)

James Malone, Simon Jupp

Examples

```
##---- Should be DIRECTLY executable !! ----
```

```
getSpeciesSpecificEnsemblGenesForExFactor
```

Get ENSEMBL genes which have been differentially expressed for a given condition and species.

Description

Get the ENSEMBL gene IDs and names which are reported as differentially expressed in Atlas for a specified conditions and species.

Usage

```
getSpeciesSpecificEnsemblGenesForExFactor(exfactor, taxon, limit = 0, endpoint="")
```

Arguments

exfactor	URI of the experimental factor search term
taxon	URI of the species to limit search to
limit	Optional limit to number of genes returned (default is all)
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

data frame of results with the following:

dbXref	URI identifier of the gene
genename	Common gene name
ensemblid	ENSEMBL ID of gene

Author(s)

James Malone, Simon Jupp

Examples

```
###get genes differentially expressed for HeLa cells (efo:EFO_0001185) in human (obo:NCBITaxon_9606)
#getSpeciesSpecificEnsemblGenesForExFactor("efo:EFO_0001185", "obo:NCBITaxon_9606")
```

getTaxonURI	<i>Get URI of a taxon class based on the common or taxonomic name.</i>
-------------	--

Description

Convenience method to the ontology URI, used in other queries, based on an input common or taxonomic name such as "human" or "Homo sapiens"

Usage

```
getTaxonURI (taxonName)
```

Arguments

taxonName	Common or taxonomic name
-----------	--------------------------

Value

URI of the taxon

Author(s)

James Malone, Simon Jupp

Examples

```
###get URI for human
#getTaxonURI ("human")
###or for arabidopsis
#getTaxonURI ("arabidopsis")
```

```
includeOnlySubclasses
```

Function which allows filtering of enrichment results to remove experimental factors apart from those factors which are a subclasses of a given parent class. For instance, include only disease factors or only cell types.

Usage

```
includeOnlySubclasses(includeparentclass, resultset, endpoint = "http://www.ebi.
```

Arguments

includeparentclass	The URI of the parent class for which all experimental factor subclasses will be included (removing all others).
resultset	The full enrichment result set
endpoint	The location of the SPARQL endpoint - default value is live Atlas endpoint

Value

New Atlas enrichment result set, filtered to include only experimental factors specified in the function call.

Author(s)

James Malone, Simon Jupp

Examples

```
###include only experimental factors which are diseases in the enrichment results
#filteredgenes <- includeOnlySubclasses("efo:EFO_0000408", mouserresults, endpoint="http://")
###view results
#vizPvalues(filteredgenes)
```

vizPvalues	<i>Visualize the results of an enrichment filtering for a specified p-value.</i>
------------	--

Description

Produce a plot of the most enriched experimental factors following an Atlas gene set enrichment. The cutoff allows a filter to be placed by p-value.

Usage

```
vizPvalues(resultset, cutoff = "0.05")
```

Arguments

resultset	A set of results following an AtlasRDF gene set enrichment.
cutoff	The p-value threshold at which to include a gene for visualization

Value

Plot of results

Author(s)

James Malone

Examples

```
###visualize the results from a gene set enrichment performed using AtlasRDF package
#vizPvalues(enrichmentresults, cutoff = "0.02")
```

Index

***Topic package**

AtlasRDF-package, **2**

AtlasRDF (*AtlasRDF-package*), **2**

AtlasRDF-package, **2**

doFishersEnrichment, **3**

doFishersEnrichmentForEnsemblIds,
5

doFishersEnrichmentForGeneNames,
6

drawHeatMapForAtlasExperiment, **8**

excludeSubclasses, **8**

getAllEnsemblGenesForExFactor, **9**

getExFactorURIFromLabel, **10**

getExperimentsByDescription, **11**

getGeneListFromPubmedid, **11**

getGenesForExperimentID, **12**

getGenesForExperimentURI, **13**

getGeneUriFromEnsemblId, **14**

getGeneUriFromName, **14**

getLabel, **15**

getOntologyMappings, **16**

getPathwaysFromGenesAndCondition,
16

getPathwayUriFromName, **17**

getSpeciesSpecificEnsemblGenesForExFactor,
18

getTaxonURI, **19**

includeOnlySubclasses, **19**

vizPvalues, **20**