Package ‘AtlasRDF’

Type Package

Title Gene Expression Atlas query and gene set enrichment package.

Version 1.10.0

Date 2015-01-26

Author James Malone, Simon Jupp, Maryam Soleimani

Maintainer Simon Jupp <jupp@ebi.ac.uk>

Depends R (>= 2.10), 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 License 2.0

biocViews Microarray, DataImport, GeneSetEnrichment, GeneExpression, DifferentialExpression, DataRepresentation

NeedsCompilation no

R topics documented:

AtlasRDF-package ................................. 2
doFishersEnrichment .............................. 4
doFishersEnrichmentForEnsemblIds .................. 5
doFishersEnrichmentForGeneNames .................. 6
drawHeatMapForAtlasExperiment .................... 8
enrichmentresult-class .......................... 9
excludeSubclasses ............................... 10
factorbackground-class ........................ 10
generef-class ..................................... 12
getAllEnsemblGenesForExFactor ..................... 13
getExFactorURIFromLabel .......................... 14
getExperimentIdsForGeneURI ....................... 14
getExperimentsByDescription ...................... 15
getExperimentURIsForGeneId ....................... 16
getGeneListFromPubmedid ........................ 16
getGenesForExperimentID .......................... 17
AtlasRDF-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. Access the backgroun data sets for enrichment from https://github.com/jamesmalone/AtlasRDF-R

Details

Package: AtlasRDF
Type: Package
Version: 1.3
Date: 2015-01-26
License: Apache 2.0

Note

Linux users may need to perform the following:

```
sudo apt-get install libxml2-dev
from R environment: install.packages("XML")
sudo apt-get install libcurl4-gnutls-dev
from R environment: install.packages("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


Examples

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

###get gene list for given experiment ID
#genelist <- getGenesForExperimentID("E-MEXP-3739")

###get list of pathways associated with a gene for which the genes are differentially expressed for a given condition (Type I diabetes in this case)
pathwayinfo <- getPathwaysFromGenesAndCondition("efo:EFO_0001359")
pathwayinfo$pathwayname

###perform gene set enrichment using Atlas as background to find factors genes are enriched for
#load the background files for the species of interest (requires downloading)
#find background sets at https://github.com/jamesmalone/AtlasRDF-R
#load("human_gene_list.RData")  #human_genelist_bg
#load("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", "TNP01", "CREB1", "NOX")

###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. Access the background data sets for enrichment from https://github.com/jamesmalone/AtlasRDF-R

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
Examples

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

###create 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/ENSG00000145817>",
"<http://identifiers.org/ensembl/ENSG00000003756>"
"
""
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"
"`
```
doFishersEnrichmentForGeneNames

perform gene set enrichment using Atlas as background to find factors genes are enriched for

###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. Brca1). The results are a list of experimental factors for which these genes are enriched. Access the backgroun data sets for enrichment from https://github.com/jamesmalone/AtlasRDF-R

###Usage

doFishersEnrichmentForGeneNames(genenames, taxon, genelist_bg, genecounts, endpoint="http://www.ebi.ac.uk/rdf/services/atlas/sparql")
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>genenames</td>
<td>Variable genenames a vector of genes using common gene names</td>
</tr>
<tr>
<td>taxon</td>
<td>The taxon is an ontology URI which can be found using the function getTaxonURI()</td>
</tr>
<tr>
<td>genelist_bg</td>
<td>genelist_bg background list appropriate for the species of interest should be obtained from svn</td>
</tr>
<tr>
<td>genecounts</td>
<td>genecounts counts list appropriate for the species of interest should be obtained from svn</td>
</tr>
<tr>
<td>endpoint</td>
<td>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</td>
</tr>
</tbody>
</table>

Value

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>factoruri</td>
<td>the URI of the experimental factor</td>
</tr>
<tr>
<td>label</td>
<td>the human readable label for the experimental factor</td>
</tr>
<tr>
<td>p.value</td>
<td>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</td>
</tr>
<tr>
<td>estimate</td>
<td>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.</td>
</tr>
<tr>
<td>alternative</td>
<td>a character string describing the alternative hypothesis.</td>
</tr>
<tr>
<td>null.value</td>
<td>the odds ratio under the null, or. Only present in the 2 by 2 case.</td>
</tr>
<tr>
<td>method</td>
<td>the character string &quot;Fisher’s Exact Test for Count Data&quot;.</td>
</tr>
<tr>
<td>enrichedgenes</td>
<td>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</td>
</tr>
</tbody>
</table>

Author(s)

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

References


Examples

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

###create or load your genelist of interest
genelist <- c("GSTT1", "PLCE1", "XIST", "ENOSF1", "AHR", "FSTL1", "YIPF5", "RBM5", "TNPO1", "CREB1", "NOX")

###get id for species 'human'
```
drawHeatMapForAtlasExperiment

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 = 5, numoffactorsdiffexpressedacross = 1, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")

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.
- numoffactorsdiffexpressedacross: Draw only genes which are differentially expressed across this number of specified factors (default is 1)
- endpoint: SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overidden.

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)
**enrichmentresult-class**

Class "enrichmentresult"

---

**Description**

Class to store the enrichment for a single experimental factor for a given gene set.

**Usage**

```r
#enrichmentresult()
```

**Objects from the Class**

Objects can be created by calls of the form `enrichmentresult()`.

**Fields**

- `factoruri`: Object of class "character" the URI of the experimental factor
- `label`: Object of class "character" human readable label of the experimental factor
- `p.value`: Object of class "numeric" p.value for enrichment for this factor
- `estimate`: Object of class "numeric" see Fishers exact test
- `alternative`: Object of class "character" see Fishers exact test
- `null.value`: Object of class "numeric" see Fishers exact test
- `method`: Object of class "character" Fishers exact test
- `enrichedgenes`: Object of class "vector" Vector of the genes enriched for this factor from the gene list

**Methods**

No methods defined with class "enrichmentresult" in the signature.

**Author(s)**

James Malone

**Examples**

```r
showClass("enrichmentresult")
```
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 from the enrichment results or all subclasses of cancer use this function.

Usage

```
excludeSubclasses(filterparentclass, resultset, endpoint="http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

Arguments

- `filterparentclass`<br>The class from which all subclasses will be excluded - including this class.
- `resultset`<br>Result set from an Atlas gene set enrichment.
- `endpoint`<br>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 package
# filteredresults <- excludeSubclasses("obo:CHEBI_37577", enrichmentresults)
# vizPvalues(filteredresults, 0.01)
```

factorbackground-class

Class "factorbackground"

Description

Used in enrichment to represent an experimental factor and any associated genes for that factor as well as additional ontological information about the factor.

Extends

All reference classes extend and inherit methods from "envRefClass".
Fields

uri: Object of class character URI of the experimental factor.
label: Object of class character name of the experimental factor.
species: Object of class character Species ID of the experimental factor.
geneuris: Object of class vector URIs of the genes for this experimental factor.
numgenesexpressed: Object of class integer Count of the genes expressed for this factor.
numgenesnotexpressed: Object of class integer Count of the genes not expressed for this factor from the total genes for this species.
subclasses: Object of class vector Subclass of this factor from ontology
superclasses: Object of class vector Superclasses of this factor from ontology

Methods

mergegeneuris(value): Merge gene URIs into the current vector
getsuperclasses(): Get superclasses for this factor from the ontology
setsuperclasses(value): Set superclasses for this factor from the ontology
getsubclasses(): Get subclasses for this factor from the ontology
setsubclasses(value): Set ontology subclasses for this factor
getnumgenesnotexpressed(): Get the number of genes not expressed for this factor as a total from total number of genes for this species
setnumgenesnotexpressed(value): Set the number of genes not expressed for this factor as a total from total number of genes for this species
getnumgenesexpressed(): Get the number of genes expressed for this factor.
setnumgenesexpressed(value): Set the number of genes expressed for this factor.
getgeneuris(): Get URIs for genes differentially expressed for this factor.
setgeneuris(value): Set URIs for genes differentially expressed for this factor.
getspecies(): Get species ID for this factor.
setspecies(value): Set species ID for this factor.
getlabel(): Get name for this factor.
setlabel(value): Set name for this factor.
geturi(): Get the URI for this factor.
seturi(value): Set the URI for this factor.

Author(s)
James Malone, Simon Jupp

Examples

showClass("factorbackground")
Description

Class to store gene references, including the URI, name and experimental factors associated with the gene.

Extends

All reference classes extend and inherit methods from "envRefClass".

Fields

geneuri: Object of class character the URI of the gene.
geneensemblid: Object of class character ENSEMBL ID of the gene, if available.
geneensemblid: Object of class character ENSEMBL ID of the gene, if available.
geneensemblid: Object of class character ENSEMBL ID of the gene, if available.

Methods

mergeexfactoruris(value): merge an experimental factor(s) into the list
getexfactoruris(): get the list of experimental factors for this gene
getspecies(): Get the species ID for this gene
setspecies(value): Set the species ID for this gene
getensembled(): Get the ENSEMBL ID for this gene.
setensembled(value): Set the ENSEMBL ID for this gene.
getgenelabel(): Get gene name.
setgenelabel(value): Set gene name.
getgeneuri(): Get the URI for the gene.
setgeneuri(value): Set the URI for the gene.

Author(s)

James Malone, Simon Jupp

Examples

showClass("generef")
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

```r
###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/atlas/sparql")
```

**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>
```

getExperimentIdsForGeneURI

*Get ArrayExpress experiment ID which contain a gene specified by URI.*

**Description**
Get all of the ArrayExpress experiment IDs which contain a gene, specified by a URI.

**Usage**
```
getExperimentIdsForGeneURI(geneuri, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

**Arguments**
- **geneuri**: Gene URI
- **endpoint**: The location of the SPARQL endpoint - default value is live Atlas endpoint
getExperimentsByDescription

Value

List of ArrayExpress experiment IDs for the specified gene.

Author(s)

James Malone, Simon Jupp

Examples

### get the experiment IDs for a gene ENSG00000142556 (ENSMBL ID)

```
getExperimentIdsForGeneURI("http://identifiers.org/ensembl/ENSG00000142556")
```

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/rdf/services/atlas/sparql")
```

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")
```
getExperimentURIsForGeneId

*Get experiment URIs which contain a gene specified by ENSEMBL ID.*

**Description**

Get all of the experiment URIs which contain a gene, specified by an ENSEMBL ID, which has been reported as differentially expressed with said experiment.

**Usage**

```r
generateExperimentURIsForGeneId(geneid, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

**Arguments**

- **geneid**: ENSEMBL gene ID
- **endpoint**: The location of the SPARQL endpoint - default value is live Atlas endpoint

**Value**

List of experiment URIs for the specified gene. Note, the last part of the URI is the accession number used in Arrayexpress to identify an experiment.

**Author(s)**

James Malone, Simon Jupp

**Examples**

```r
# get the experiment IDs for a gene ENSMUSG000000031314 (ENSEMBL ID)
# generateExperimentURIsForGeneId("ENSMUSG000000031314")
```

---

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**

```r
generateGeneListFromPubmedid(searchid, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```
getGenesForExperimentID

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchid</td>
<td>PubMed ID to find data for</td>
</tr>
<tr>
<td>endpoint</td>
<td>SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.</td>
</tr>
</tbody>
</table>

Value

List of genes for experiment (via PubMed ID)

Author(s)

James Malone, Simon Jupp

Examples

```r
getGeneListFromPubmedid("19668377")
```

Description

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

Usage

```r
genes <- getGenesForExperimentID(experiment, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>experiment</td>
<td>Atlas experiment ID.</td>
</tr>
<tr>
<td>endpoint</td>
<td>The location of the SPARQL endpoint - default value is live Atlas endpoint</td>
</tr>
</tbody>
</table>

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 differentially expression for this gene
  - propertyValue: The property value for which this gene was reported as differentially expressed

Author(s)

James Malone, Simon Jupp
getGenesForExperimentURI

Get all of the genes reported in an experiment, specified 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/services/atlas/sparql")

Arguments

- `experiment`: Atlas experiment URI.
- `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 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

```r
###get genes for experiment ID E-MEXP-3739
#Can take several minutes to complete
#res <- getGenesForExperimentID("E-MEXP-3739")
###list first gene in the experiment
#res[1,]
```
getGenesForPathwayURI  \hspace{1em} \textit{Get genes associated with a signalling pathway from Reactome.}

**Description**

Get genes that are associated with proteins which are in turn associated to a specific signalling pathway in Reactome.

**Usage**

```r
getGenesForPathwayURI(pathwayuri, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

**Arguments**

- `pathwayuri`  \hspace{1em} The URI of a reactome pathway.
- `endpoint`  \hspace{1em} SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

**Value**

List of gene URIs for the pathway

**Author(s)**

James Malone, Simon Jupp

**Examples**

```r
### get gene list for the pathway AKT Signaling in Cancer
# getGenesForPathwayURI("<http://identifiers.org/reactome/REACT_147727.2>")
```

---

getGeneUriFromEnsemblId  \hspace{1em} \textit{Get gene URI from an ENSEMBL ID.}

**Description**

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

**Usage**

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

**Arguments**

- `id`  \hspace{1em} ENSEMBL gene ID
- `endpoint`  \hspace{1em} The location of the SPARQL endpoint - default value is live Atlas endpoint
getGeneUriFromName

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
#
geneUriFromEnsemblId("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/services/atlas/sparql")

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**

*Get label of an entity based on the URI.*

**Description**

Every ontology class has a human readable label associated with it. This function will enable you to get that readable label based on the URI (the identifier) for that entity. For instance, the URI http://www.ebi.ac.uk/efo/EFO_0001185 has label HeLa.

**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**

```r
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. This function is useful for querying the Atlas RDF using your own ontology. For instance, if you are using ICD-10 or Snomed this function will enable you to extract any mappings to EFO, used to annotate the Atlas data, thereby enabling querying by your ontology of interest. The returned EFO class should then be used to query Atlas using the other query functions.

**Usage**

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

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


Examples

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

---

getPathwayForGeneId: Get pathways associated to a gene.

Description

Get signalling pathways from Reactome that are associated to a gene, via associated protein(s) which are in turn connected to said signalling pathway(s) in Reactome.

Usage

```python
getPathwayForGeneId(geneid, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

Arguments

- geneid: The ENSEMBL ID for a gene.
- endpoint: SPARQL endpoint to query data from. Default value is the live Atlas SPARQL endpoint but this can be overridden.

Value

List of pathway URIs.

Author(s)

James Malone, Simon Jupp

Examples

```python
# get pathways for the gene SNHG5
#getPathwayForGeneId("http://identifiers.org/ensembl/ENSG00000203875")
```
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

gETCHWALKSFromGenesAndCondition(condition, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")

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

```r
# get pathways connected to genes which are differentially expressed for HeLa (efo:EFO_00011185)
getPathwaysFromGenesAndCondition("efo:EFO_00011185")
```
getRankedPathwaysForGeneIds Get pathway names and URIs for given gene list, sorted into list with most common pathways first.

Description
Given a list of ENSEMBL gene IDs, the function will find pathways which these genes are involved in. These pathways will be grouped and ranked according to those with highest number of genes in common.

Usage
getRankedPathwaysForGeneIds(genelist, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")

Arguments
- genelist: The list of ENSEMBL gene IDs to find pathways for.
- endpoint: The location of the SPARQL endpoint - default value is live Atlas endpoint

getPathwayUriFromName Get pathway URI given a pathway name

Description
Given the name of a pathway find URI(s) that match this name

Usage
getPathwayUriFromName(name, endpoint = "http://www.ebi.ac.uk/rdf/services/atlas/sparql")

Arguments
- name: The textual name of the pathway to search for
- endpoint: The location of the SPARQL endpoint - default value is live Atlas endpoint

Value
data frame containing slots
- pathwayuri: The URI of the corresponding pathway
- label: The textual human readable name of the pathway

Author(s)
James Malone, Simon Jupp

Examples

```r
# get pathways which have the phrase 'ligand binding' included in their name
# getPathwayUriFromName("ligand binding")
```
getSpeciesSpecificEnsemblGenesForExFactor

Description

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

Usage

getSpeciesSpecificEnsemblGenesForExFactor(exfactor, taxon, limit = 0, endpoint="http://www.ebi.ac.uk/rdf/services/atlas/sparql")

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
ensembiid ENSEMBL ID of gene

Author(s)

James Malone, Simon Jupp
includeOnlySubclasses

Examples

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

`getSpeciesSpecificEnsemblGenesForExFactor("efo:EFO_0001185", "obo:NCBITaxon_9606")`

---

getTaxonURI

Get URI of a taxon class based on the common or taxonomic name.

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

Filter enrichment results to only include experimental factors of a given parent class (e.g. all cancers)

Description

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

Usage

`includeOnlySubclasses(includeparentclass, resultset, endpoint = "http://www.ebi.ac.uk/rdf/services/...)"`
**orderEnrichmentResults**

**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**

```r
###include only experimental factors which are diseases in the enrichment results
#filteredgenes <- includeOnlySubclasses("efo:EFO_0000408", mouseresults, endpoint="http://wwwdev.ebi.ac.uk/rdf/services/atlas/sparql")
###view results
#vizPvalues(filteredgenes)
```

**Description**

Following a gene set enrichment, order the results so that the most significantly enriched factors are first, in decreasing order of significance (i.e. where lowest p-value is most significant)

**Usage**

```r
orderEnrichmentResults(resultset)
```

**Arguments**

- `resultset`  
  A set of results following an AtlasRDF gene set enrichment.

**Value**

Ordered set of results with most significant factor first, least significant last.

**Author(s)**

James Malone
Examples

### order the results of an enrichment
# orderedresults <- orderEnrichmentResults(enrichmentresults)
### view the top 10 most significant
# vizPValues(orderedresults[1:20], cutoff = "1")

---

**pathwayresult-class  Class** "pathwayresult"

---

**Description**

Class to store information about a pathway and associated genes.

**Usage**

```
#pathwayresult()
```

**Objects from the Class**

Objects can be created by calls of the form `pathwayresult()`.

**Fields**

- `pathwayuri`: Object of class "character" the URI of the pathway from Reactome.
- `label`: Object of class "character" The human readable label of the pathway.
- `numgenes`: Object of class "numeric" A count of the number of genes for this pathway.
- `genes`: Object of class "vector" List of the genes associated with this pathway.

**Methods**

No methods defined with class "pathwayresult" in the signature.

**Author(s)**

James Malone

**Examples**

```
showClass("pathwayresult")
```
**searchForEFOTerms**

*Search for EFO URIs from a given label text*

**Description**
Finds any EFO ontology class whose label matches the given search label or part of it.

**Usage**
```
searchForEFOTerms(label, endpoint="http://www.ebi.ac.uk/rdf/services/atlas/sparql")
```

**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**
Data frame of EFO class URIs and their corresponding labels

**Author(s)**
James Malone, Simon Jupp

**Examples**
```
### find the efo classes which contain the word 'cancer' in their label
#searchForEFOTerms("cancer")
```

---

**transcription_pathway_enrichment**

*Expression Gene Set Enrichment result for*

**Description**
A result set example from running the Atlas Expression gene set enrichment function. The gene list was originally extracted from a transcription pathways in Reactome and enrichment performed to see which experimental factors were enriched for this pathway.

**Usage**
```
transcription_pathway_enrichment
```

**Format**
A list of enrichmentresult objects

**Source**
Gene Expression Atlas RDF at European Bioinformatics Institute
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 classes
  enrichmentresult-class, 9
  factorbackground-class, 10
  generef-class, 12
  pathwayresult-class, 28

*Topic datasets
  transcription_pathway_enrichment, 29

*Topic package
  AtlasRDF-package, 2

AtlasRDF (AtlasRDF-package), 2

class:enrichmentresult
  (enrichmentresult-class), 9

class:factorbackground
  (factorbackground-class), 10

class:generef (generef-class), 12

class:pathwayresult
  (pathwayresult-class), 28

doFishersEnrichment, 4
doFishersEnrichmentForEnsemblIds, 5
doFishersEnrichmentForGeneNames, 6
drawHeatMapForAtlasExperiment, 8

enrichmentresult
  (enrichmentresult-class), 9

enrichmentresult-class, 9

envRefClass, 10, 12

excludeSubclasses, 10

factorbackground
  (factorbackground-class), 10

factorbackground-class, 10

generef (generef-class), 12

getAllEnsemblGenesForExFactor, 13

getExFactorURIFromLabel, 14

getExperimentIdsForGeneURI, 14

getExperimentsByDescription, 15

getExperimentURIsForGeneId, 16

geneListFromPubmedid, 16

geneURIFromEnsemblId, 19

geneURIFromName, 20

getLabel, 21

getOntologyMappings, 21

getPathwayForGeneId, 22

getPathwaysFromGenesAndCondition, 23

getPathwayURIFromName, 24

getRankedPathwaysForGeneIds, 24

getSpeciesSpecificEnsemblGenesForExFactor, 25

getTaxonURI, 26

includeOnlySubclasses, 26

orderEnrichmentResults, 27

pathwayresult (pathwayresult-class), 28

pathwayresult-class, 28

searchForEFOTerms, 29

transcription_pathway_enrichment, 29

vizPvalues, 30