Package 'BgeeDB'

October 11, 2016

Type Package

Title Annotation and gene expression data from Bgee database

Version 1.0.3

Date 2016-03-15

Author Andrea Komljenovic [aut, cre], Julien Roux [aut, cre]

Maintainer
Andrea Komljenovic <andreakomljenovic@gmail.com>, Frederic Bastian <bgee@sib.swiss>

Description A package for the annotation and gene expression data download from Bgee database, and TopAnat analysis: GO-like enrichment of anatomical terms, mapped to genes by expression patterns.

Depends R (>= 3.3), topGO, tidyr

Imports data.table, RCurl, methods, stats, utils, dplyr, graph, Biobase

License GPL-2

VignetteBuilder knitr

biocViews Software, DataImport, Sequencing, GeneExpression, Microarray, GO

Suggests knitr, BiocStyle, testthat, rmarkdown

LazyLoad yes

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

`Bgee-class` *Retrieving the Bgee database data*

Description

A Reference Class to give annotation available on Bgee for particular species and the requested data (rna_seq, affymetrix)

Details

The expression calls come from Bgee (http://r.bgee.org), that integrates different expression data types (RNA-seq, Affymetrix microarray, ESTs, or in-situ hybridizations) in multiple animal species. Expression patterns are based exclusively on curated "normal", healthy, expression data (e.g., no gene knock-out, no treatment, no disease), to provide a reference of normal gene expression. This Class retrieves annotation of all experiments in Bgee database (get_annotation), downloading the data (get_data), and formating the data into expression matrix (format_data). See examples and vignette.

Value

- `get_annotation()` returns a list of the annotation of experiments for chosen species.
- `get_data()`, if experiment ID is empty, returns a list of experiments. If specified experiment ID, then returns the dataframe of the chosen experiment
- `format_data()`, if experiment ID is empty, returns a list of ExpressionSet objects. If specified experiment ID, then returns an ExpressionSet object

Fields

`species` A character of species name as listed from Bgee. The species are:

- "Anolis_carolinensis"
- "Bos_taurus"
- "Caenorhabditis_elegans"
- "Danio_rerio"
- "Drosophila_melanogaster"
- "Gallus_gallus"
- "Gorilla_gorilla"
- "Homo_sapiens"
- "Macaca_mulatta"
- "Monodelphis_domestica"
- "Mus_musculus"
- "Ornithorhynchus_anatinus"
- "Pan_paniscus"
- "Pan_troglodytes"
- "Rattus_norvegicus"

- "Sus_scrofa"
- "Xenopus_tropicalis"

Homo sapiens is the default species.

`datatype` A character of data platform. Two types of datasets can be downloaded:

- "rna_seq"
- "affymetrix"

By default, RNA-seq data is retrieved.

`experiment.id` An ArrayExpress or GEO accession, e.g., GSE30617 On default is NULL: takes all available experiments for specified species and datatype.

`data` A dataframe of downloaded Bgee data.

`calltype` A character.

- "expressed"
- "expressed high quality"
- "all"

Retrieve intensities only for expressed (present) genes, expressed high quality genes, or all genes. The default is expressed.

`stats` A character. The expression values can be retrieved in RPKMs and raw counts:

- "rpkm"
- "counts"
- "intensities"

The default is RPKMs for RNA-seq and intensities for microarray.

Author(s)

Andrea Komljenovic <andrea.komljenovic at unil.ch>.

Examples

```
{
 bgee <- Bgee$new(species = "Mus_musculus", datatype = "rna_seq")
 annotation_bgee_mouse <- bgee$get_annotation()
 data_bgee_mouse <- bgee$get_data()
 data_bgee_mouse_gse30617 <- bgee$get_data(experiment.id = "GSE30617")
 gene.expression.mouse.rpkm <- bgee$format_data(data_bgee_mouse_gse30617,
 calltype = "expressed", stats = "rpkm")
 }
```

listBgeeSpecies *List species with Affymetrix or RNA-seq data in the Bgee database*

Description

Returns a list of available genomes for different platforms in Bgee.

Usage

```
listBgeeSpecies(...)
```

Arguments

...	an empty parameter

Value

A list of species with available Affymetrix or RNA-seq data in the Bgee

Author(s)

Andrea Komljenovic <andrea.komljenovic at unil.ch>.

Examples

```
{
 listBgeeSpecies()
}
```

loadTopAnatData *Retrieve data from Bgee to perform GO-like enrichment of anatomical terms, mapped to genes by expression patterns.*

Description

This function loads a mapping from genes to anatomical structures based on calls of expression in anatomical structures. It also loads the structure of the anatomical ontology.

Usage

```
loadTopAnatData(species, datatype = c("rna_seq", "affymetrix", "est",
  "in_situ"), calltype = "expressed", confidence = "all", stage = NULL,
  host = "http://r.bgee.org", pathToData = getwd())
```

Arguments

`species` A numeric indicating the NCBI taxonomic ID of the species to be used. The species has to be among species in Bgee v13, which include:

- 6239 (Caenorhabditis elegans)
- 7227 (Drosophila melanogaster)
- 7955 (Danio rerio)
- 8364 (Xenopus tropicalis)
- 9031 (Gallus gallus)
- 9258 (Ornithorhynchus anatinus)
- 9544 (Macaca mulatta)
- 9593 (Gorilla gorilla)
- 9597 (Pan paniscus)
- 9598 (Pan troglodytes)
- 9606 (Homo sapiens)
- 9823 (Sus scrofa)
- 9913 (Bos taurus)
- 10090 (Mus musculus)
- 10116 (Rattus norvegicus)
- 13616 (Monodelphis domestica)
- 28377 (Anolis carolinensis)

See the listBgeeSpecies() function to get an up-to-date list of species.

`datatype` A vector of characters indicating data type(s) to be used. To be chosen among:

- "rna_seq"
- "affymetrix"
- "est"
- "in_situ"

By default all data type are included: `c("rna_seq","affymetrix","est","in_situ")`. Including a data type that is not present in Bgee for a given species has no effect.

`calltype` A character of indicating the type of expression calls to be used for enrichment. Only calls for significant presence of expression are implemented ("expressed"). Over-expression calls, based on differential expression analysis, will be implemented in the future.

`confidence` A character indicating if only high quality expression calls should be retrieved. Options are "all" or "high_quality". Default is "all".

`stage` A character indicating the targeted developmental stages for the analysis. Developmental stages can be chosen from the developmental stage ontology used in Bgee (available at https://github.com/obophenotype/developmental-stage-ontologies). If a stage ID is given, the expression pattern mapped to this stage and all children developmental stages (substages) will be retrieved. Default is NULL, meaning that expression patterns of genes are retrieved regardless of the stage of expression. This is equivalent to specifying stage="UBERON:0000104" (life cycle, the root of the stage ontology). The most useful stages (going no deeper than level 3 of the ontology) include:

- UBERON:0000068 (embryo stage)
  - UBERON:0000106 (zygote stage)
  - UBERON:0000107 (cleavage stage)
  - UBERON:0000108 (blastula stage)
  - UBERON:0000109 (gastrula stage)
  - UBERON:0000110 (neurula stage)
  - UBERON:0000111 (organogenesis stage)
  - UBERON:0007220 (late embryonic stage)
  - UBERON:0004707 (pharyngula stage)
- UBERON:0000092 (post-embryonic stage)
  - UBERON:0000069 (larval stage)
  - UBERON:0000070 (pupal stage)
  - UBERON:0000066 (fully formed stage)

`host` URL to Bgee webservice. Change host to access development or archive versions of Bgee. Default is "`http://r.bgee.org`" to access current Bgee release.

`pathToData` Path to the directory where the data files are stored / will be stored. Default is the working directory.

Details

The expression calls come from Bgee (`http://bgee.org`), that integrates different expression data types (RNA-seq, Affymetrix microarray, ESTs, or in-situ hybridizations) in multiple animal species. Expression patterns are based exclusively on curated "normal", healthy, expression data (e.g., no gene knock-out, no treatment, no disease), to provide a reference of normal gene expression.

Anatomical structures are identified using IDs from the Uberon ontology (browsable at `http://www.ontobee.org/ontology/UBERON`). The mapping from genes to anatomical structures includes only the evidence of expression in these specific structures, and not the expression in their substructures (i.e., expression data are not propagated). The retrieval of propagated expression data will likely be implemented in the future, but meanwhile, it can be obtained using specialized packages such as topGO, see the `topAnat.R` function.

Value

A list of 3 elements:

- A `gene2anatomy` list, mapping genes to anatomical structures based on expression calls.
- A `organ.names` data frame, with the name corresponding to UBERON IDs.
- A `organ.relationships` list, giving the relationships between anatomical structures in the UBERON ontology (based on parent-child "is_a" and "part_of" relationships).

Author(s)

Julien Roux <julien.roux at unil.ch>.

Examples

```
{
  myTopAnatData <- loadTopAnatData(species = "10090", datatype = "rna_seq")
}
```

`makeTable` *Formats results of the enrichment test on anatomical structures.*

Description

This function loads the results from the topGO test and creates an output table with organ names, fold enrichment and FDR. Data are sorted by p-value and only terms below the specified FDR cutoff are included.

Usage

```
makeTable(topAnatData, topAnatObject, results, cutoff = 1)
```

Arguments

`topAnatData`	A list produced by the function loadTopAnatData().
`topAnatObject`	An object produced by the function topAnat().
`results`	A result object, produced by the runtest() function of topGO.
`cutoff`	An FDR cutoff between 0 and 1. Only terms with FDR lower than this cutoff are included. Default is 1, meaning that all terms are included.

Value

A data frame with significantly enriched anatomical structures, sorted by p-value.

Author(s)

Julien Roux <julien.roux@unil.ch>.

Examples

```
{
 ## Launch topGO test on data loaded from Bgee
  myTopAnatData <- loadTopAnatData(species = "10090", datatype = "rna_seq")
  geneList <- as.factor(c(rep(0, times=90), rep(1, times=10)))
  names(geneList) <- c("ENSMUSG00000064370", "ENSMUSG00000064368", "ENSMUSG00000064367",
                       "ENSMUSG00000064363", "ENSMUSG00000065947", "ENSMUSG00000064360",
                       "ENSMUSG00000064358", "ENSMUSG00000064357", "ENSMUSG00000064356",
                       "ENSMUSG00000064354", "ENSMUSG00000064351", "ENSMUSG00000064345",
                       "ENSMUSG00000064341", "ENSMUSG00000029757", "ENSMUSG00000079941",
                       "ENSMUSG00000053367", "ENSMUSG00000016626", "ENSMUSG00000037816",
                       "ENSMUSG00000036781", "ENSMUSG00000022519", "ENSMUSG00000079606",
```

```
                    "ENSMUSG00000068966", "ENSMUSG00000038608", "ENSMUSG00000047473",
                    "ENSMUSG00000038542", "ENSMUSG00000025386", "ENSMUSG00000028145",
                    "ENSMUSG00000024816", "ENSMUSG00000020978", "ENSMUSG00000055373",
                    "ENSMUSG00000038155", "ENSMUSG00000046408", "ENSMUSG00000030032",
                    "ENSMUSG00000042249", "ENSMUSG00000071909", "ENSMUSG00000039670",
                    "ENSMUSG00000032501", "ENSMUSG00000054252", "ENSMUSG00000068071",
                    "ENSMUSG00000067578", "ENSMUSG00000074892", "ENSMUSG00000027905",
                    "ENSMUSG00000058216", "ENSMUSG00000078754", "ENSMUSG00000062101",
                    "ENSMUSG00000043633", "ENSMUSG00000071350", "ENSMUSG00000021639",
                    "ENSMUSG00000059113", "ENSMUSG00000049115", "ENSMUSG00000053310",
                    "ENSMUSG00000043832", "ENSMUSG00000063767", "ENSMUSG00000026775",
                    "ENSMUSG00000038537", "ENSMUSG00000078716", "ENSMUSG00000096820",
                    "ENSMUSG00000075089", "ENSMUSG00000049971", "ENSMUSG00000014303",
                    "ENSMUSG00000056054", "ENSMUSG00000033082", "ENSMUSG00000020801",
                   "ENSMUSG00000030590", "ENSMUSG00000026188", "ENSMUSG00000014301",
                    "ENSMUSG00000073491", "ENSMUSG00000014529", "ENSMUSG00000036960",
                    "ENSMUSG00000058748", "ENSMUSG00000047388", "ENSMUSG00000002204",
                    "ENSMUSG00000034285", "ENSMUSG00000109129", "ENSMUSG00000035275",
                    "ENSMUSG00000051184", "ENSMUSG00000034424", "ENSMUSG00000041828",
                    "ENSMUSG00000029416", "ENSMUSG00000030468", "ENSMUSG00000029911",
                    "ENSMUSG00000055633", "ENSMUSG00000027495", "ENSMUSG00000029624",
                    "ENSMUSG00000045518", "ENSMUSG00000074259", "ENSMUSG00000035228",
                    "ENSMUSG00000038533", "ENSMUSG00000030401", "ENSMUSG00000014602",
                    "ENSMUSG00000041827", "ENSMUSG00000042345", "ENSMUSG00000028530",
                    "ENSMUSG00000038722", "ENSMUSG00000075088", "ENSMUSG00000039629",
                    "ENSMUSG00000067567", "ENSMUSG00000057594", "ENSMUSG00000005907",
                   "ENSMUSG00000027496")
  myTopAnatObject <- topAnat(myTopAnatData, geneList)
  resFis <- runTest(myTopAnatObject, algorithm = 'elim', statistic = 'fisher')
 ## Format results
  tableOver <- makeTable(myTopAnatData, myTopAnatObject, resFis, 0.1)
}
```

---

topAnat *Produces an object allowing to perform GO-like enrichment of anatomical terms using the topGO package*

Description

This function produces a topAnatObject, ready to use for gene set enrichment testing using functions from the topGO package. This object uses the Uberon ontology instead of the GO ontology.

Usage

```
topAnat(topAnatData, geneList, nodeSize = 10, ...)
```

Arguments

topAnatData	a list including a gene2anatomy list, an organ.relationships list and an organ.names data.frame, produced by the function loadTopAnatData().

geneList	Vector indicating foreground and background genes. Names of the vector indicate the background genes. Values are 1 (gene in foreground) or 0 (gene not in foreground).
nodeSize	Minimum number of genes mapped to a node for it to be tested. Default is 10.
...	Additional parameters as passed to build topGOdata object in topGO package.

Details

To perform the enrichment test for expression in anatomical structures for each term of Uberon ontology (browsable at http://www.ontobee.org/ontology/UBERON), the data are formatted to use the topGO package for testing. This package is interesting because it propagates the mapping of gene to terms to parent terms, and it possesses a pannel of enrichment tests and decorrelation methods. Expert users should be able to use information from the topAnatObject to test enrichment with other packages than topGO.

Value

topAnatObject, a topGO-compatible object ready for gene set enrichment testing.

Author(s)

Julien Roux <julien.roux@unil.ch>.

Examples

```
{
  myTopAnatData <- loadTopAnatData(species = "10090", datatype = "rna_seq")
  geneList <- as.factor(c(rep(0, times=90), rep(1, times=10)))
  names(geneList) <- c("ENSMUSG00000064370", "ENSMUSG00000064368", "ENSMUSG00000064367",
                       "ENSMUSG00000064363", "ENSMUSG00000065947", "ENSMUSG00000064360",
                       "ENSMUSG00000064358", "ENSMUSG00000064357", "ENSMUSG00000064356",
                       "ENSMUSG00000064354", "ENSMUSG00000064351", "ENSMUSG00000064345",
                       "ENSMUSG00000064341", "ENSMUSG00000029757", "ENSMUSG00000079941",
                       "ENSMUSG00000053367", "ENSMUSG00000016626", "ENSMUSG00000037816",
                       "ENSMUSG00000036781", "ENSMUSG00000022519", "ENSMUSG00000079606",
                       "ENSMUSG00000068966", "ENSMUSG00000038608", "ENSMUSG00000047473",
                       "ENSMUSG00000038542", "ENSMUSG00000025386", "ENSMUSG00000028145",
                       "ENSMUSG00000024816", "ENSMUSG00000020978", "ENSMUSG00000055373",
                       "ENSMUSG00000038155", "ENSMUSG00000046408", "ENSMUSG00000030032",
                       "ENSMUSG00000042249", "ENSMUSG00000071909", "ENSMUSG00000039670",
                       "ENSMUSG00000032501", "ENSMUSG00000054252", "ENSMUSG00000068071",
                       "ENSMUSG00000067578", "ENSMUSG00000074892", "ENSMUSG00000027905",
                       "ENSMUSG00000058216", "ENSMUSG00000078754", "ENSMUSG00000062101",
                       "ENSMUSG00000043633", "ENSMUSG00000071350", "ENSMUSG00000021639",
                       "ENSMUSG00000059113", "ENSMUSG00000049115", "ENSMUSG00000053310",
                       "ENSMUSG00000043832", "ENSMUSG00000063767", "ENSMUSG00000026775",
                       "ENSMUSG00000038537", "ENSMUSG00000078716", "ENSMUSG00000096820",
                       "ENSMUSG00000075089", "ENSMUSG00000049971", "ENSMUSG00000014303",
                       "ENSMUSG00000056054", "ENSMUSG00000033082", "ENSMUSG00000020801",
                       "ENSMUSG00000030590", "ENSMUSG00000026188", "ENSMUSG00000014301",
                       "ENSMUSG00000073491", "ENSMUSG00000014529", "ENSMUSG00000036960",
```

```
                      "ENSMUSG00000058748", "ENSMUSG00000047388", "ENSMUSG00000002204",
                      "ENSMUSG00000034285", "ENSMUSG00000109129", "ENSMUSG00000035275",
                      "ENSMUSG00000051184", "ENSMUSG00000034424", "ENSMUSG00000041828",
                      "ENSMUSG00000029416", "ENSMUSG00000030468", "ENSMUSG00000029911",
                      "ENSMUSG00000055633", "ENSMUSG00000027495", "ENSMUSG00000029624",
                      "ENSMUSG00000045518", "ENSMUSG00000074259", "ENSMUSG00000035228",
                      "ENSMUSG00000038533", "ENSMUSG00000030401", "ENSMUSG00000014602",
                      "ENSMUSG00000041827", "ENSMUSG00000042345", "ENSMUSG00000028530",
                      "ENSMUSG00000038722", "ENSMUSG00000075088", "ENSMUSG00000039629",
                      "ENSMUSG00000067567", "ENSMUSG00000057594", "ENSMUSG00000005907",
                     "ENSMUSG00000027496")
  myTopAnatObject <- topAnat(myTopAnatData, geneList, nodeSize=1)
}
```

Index