Package 'BgeeDB'

November 11, 2026

Type Package

Title Annotation and gene expression data retrieval from Bgee database. TopAnat, an anatomical entities Enrichment Analysis tool for UBERON ontology

Version 2.38.1

Date 2024-08-20

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

Maintainer Julien Wollbrett <julien.wollbrett@unil.ch>, Julien Roux <julien.roux@unibas.ch>, 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.

Encoding UTF-8

Depends R (>= 3.6.0), topGO, tidyr

Imports R.utils, data.table, curl, RCurl, digest, methods, stats, utils, dplyr, RSQLite, graph, Biobase, zellkonverter, anndata, HDF5Array, bread

License GPL-3 + file LICENSE

Roxygen list(markdown = TRUE)

URL https://github.com/BgeeDB/BgeeDB_R

BugReports https://github.com/BgeeDB/BgeeDB_R/issues

VignetteBuilder knitr

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

Suggests knitr, BiocStyle, testthat, rmarkdown, markdown

LazyLoad yes

RoxygenNote 7.3.2

NeedsCompilation no

git_url https://git.bioconductor.org/packages/BgeeDB

git_branch RELEASE_3_23

git_last_commit a460555

git_last_commit_date 2026-04-29

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

Bgee-class . . . 2
deleteLocalData . . . 3
deleteOldData . . . 4
formatData . . . 5
geneList . . . 6
getAnnotation . . . 7
getCellProcessedData . . . 8
getData . . . 9
getIntegratedCalls . . . 10
getSampleProcessedData . . . 12
listBgeeRelease . . . 14
listBgeeSpecies . . . 14
loadTopAnatData . . . 15
makeTable . . . 17
topAnat . . . 19

Index **21**

`Bgee-class` *Bgee Reference Class*

Description

This is used to specify information at the beginning of a BgeeDB working session, for example, the targeted species and data type. An object of this class is then passed as argument to other functions of the package to provide these informations. See examples in vignette.

Details

Bgee (`https://www.bgee.org`) integrates different expression data types (RNA-seq, Affymetrix microarray, ESTs, and in-situ hybridizations) from 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 atlas of normal gene expression.

Fields

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

- "rna_seq: target only bulk RNA-seq data"
- "sc_full_length: target only full length single-cell RNA-seq data"
- "sc_droplet_based: target only droplet based single-cell RNA-seq data"
- "affymetrix: target only Affymtrix microarray data"
- "est: target only ESTs data"
- "in_situ: target only In Situ data"

By default all data type are included: `c("rna_seq","sc_full_length","sc_droplet_based","affymetrix","est"`
For download of quantitative expression data, a single data type should be chosen among "rna_seq", "sc_full_length", sc_droplet_based" or "affymetrix".

`pathToData` Path to the directory where the data files are stored. By default the working directory is used. If many analyses are launched in parallel, please consider re-using the cached data files instead of redownlaoding them for each analysis.

`release` Bgee release number to download data from, in the form "Release.subrelease" or "Release_subrelease", e.g., "15.2" or 15_2". Work for release >=13.2. By default, the latest relase of Bgee is used.

`sendStats` A field specifying whether monitoring of users is performed for our internal usage statistics. This is useful to improve the settings of our servers and to get reliable usage statistics (e.g., when asking for funding for Bgee). No identification of the users is attempted, nor possible. Default to TRUE. This option can be set to FALSE, notably if all data files are in cache and that users want to be able to work offline.

`quantitativeData` A field specifying if a single type of quantitative expression data ("rna_seq", "sc_full_length", "sc_droplet_based" or "affymetrix") was specified and if it is available for targeted species, helping the package to know if it should proceed with the execution of getAnnotation() and getData() functions.

Examples

```
{
 bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
 bgee <- Bgee$new(species = "Mus_musculus")
}
```

`deleteLocalData` *Delete local data for the species of the reference class Bgee object*

Description

This function delete data present in the local database. It can delete data rather from one datatype or all datatypes of one species.

Usage

```
deleteLocalData(myBgeeObject, allDataTypes = FALSE)
```

Arguments

myBgeeObject	A Reference Class Bgee object, notably specifying the targeted species
allDataTypes	A boolean defining rather data of all datatypes from the selected species should be deleted. If TRUE, the local sqlite database file is deleted. If FALSE, data corresponding to the datatype of the myBgeeObject object will be deleted from the local sqlite database.

Author(s)

Julien Wollbrett

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  data <- getData(bgee, experimentId = "SRP007359")
  deleteLocalData(bgee, allDataTypes = TRUE)
}
```

---

deleteOldData *Delete .rds data of one species coming from an old version of the BgeeDB R package.*

Description

Since Bioconductor 3.11 BgeeDB store data in a local SQLite database allowing to optimize memory usage. This function allows to delete .rds files used to store local data from BgeeDB versions older than Bioconductor 3.11, for the species selected in the reference class Bgee object.

Usage

```
deleteOldData(myBgeeObject)
```

Arguments

myBgeeObject	A Reference Class Bgee object, notably specifying the targeted species

Author(s)

Julien Wollbrett

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  deleteOldData(bgee)
}
```

`formatData` *Format RNA-seq or Affymetrix data downloaded from Bgee.*

Description

This function formats the data downloaded with the getSampleRawData() function into an object of the Bioconductor "expressionSet" Class.

Usage

```
formatData(myBgeeObject, data, stats = NULL, callType = "all")
```

Arguments

`myBgeeObject`	A Reference Class Bgee object, notably specifying the targeted species and data type.
`data`	A list of data frames including data from multiple experiments, or a data frame including data from a single experiment.
`stats`	A character indicating what expression values should be used in the formatted data expressionSet object matrix. • "rpkm" for bulk RNA-seq (Bgee release 13.2 and before) • "fpkm" for bulk RNA-seq (Bgee release 14.0 to 14.2) • "counts" for RNA-seq • "tpm" for bulk RNA-seq or full length single cell RNA-seq (Bgee release 14 and above) • "cpm" for droplet based single cell RNA-seq (Bgee release 15.2 and above) • "intensities" for Affymetrix microarrays
`callType`	A character indicating whether intensities should be displayed only for present (i.e., expressed) genes, present high quality genes, or all genes (default). • "present" • "present high quality" • "all"

Value

If data was a list of data frames from multiple experiments, returns a list of ExpressionSet objects. If data was a data frame from a single experiment, returns an ExpressionSet object.

Author(s)

Andrea Komljenovic and Julien Roux.

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  dataMouseGSE43721 <- getData(bgee, experimentId = "GSE43721")
  dataMouseGSE43721.tpm <- formatData(bgee,
                                      dataMouseGSE43721,
                                      callType = "present",
                                      stats = "tpm")
}
```

`geneList` *Example of gene list object used to run a topAnat enrichment test, created on June 2018. The format of the gene list is the same as the gene list required to build a* `topGOdata` *object in the* `topGO` *package: a vector with background genes as names, and 0 or 1 values depending if a gene is in the foreground or not. In this example the foreground genes are zebrafish genes with an annotated phenotype related to "pectoral fin", and the background is composed of all zebrafish Ensembl genes with an annotated phenotype from ZFIN. The gene list was built using the biomaRt package, and the code used can be found in the vignette of the package.*

Description

Example of gene list object used to run a topAnat enrichment test, created on June 2018. The format of the gene list is the same as the gene list required to build a `topGOdata` object in the `topGO` package: a vector with background genes as names, and 0 or 1 values depending if a gene is in the foreground or not. In this example the foreground genes are zebrafish genes with an annotated phenotype related to "pectoral fin", and the background is composed of all zebrafish Ensembl genes with an annotated phenotype from ZFIN. The gene list was built using the biomaRt package, and the code used can be found in the vignette of the package.

Usage

```
data(geneList)
```

Format

A named vector of factor values with 3005 elements. The factor levels are "0" for the 2858 genes in the background and "1" for the 147 genes in the foreground. Vector names are the Ensembl IDs of the zebrafish genes.

Examples

```
bgee <- Bgee$new(species = "Danio_rerio")
myTopAnatData <- loadTopAnatData(bgee)
data(geneList)
myTopAnatObject <-  topAnat(myTopAnatData, geneList)
```

`getAnnotation` *Retrieve Bgee experiments annotation for targeted species and data type.*

Description

This function loads the annotation of experiments and samples of quantitative expression datasets (rna_seq, affymetrix, sc_full_length, sc_droplet_based) that are available from Bgee.

Usage

```
getAnnotation(myBgeeObject)
```

Arguments

`myBgeeObject`	A Reference Class Bgee object, notably specifying the targeted species and data type.

Value

A list of two elements, including a data frame of the annotation of experiments for chosen species (field "experiment.annotation") and a data frame of the annotation of chips/libraries from these experiments (field "sample.annotation").

Author(s)

Andrea Komljenovic and Julien Roux.

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  myAnnotation <- getAnnotation(bgee)
}
```

`getCellProcessedData` *Retrieve Bgee processed expression values at cell level for single cell data in Bgee.*

Description

This function loads the processed gene count matrix of single cell experiments in Bgee. These data are available for all droplet based single-cell RNA-seq and full length single-cell RNA-seq. These files are stored in the H5AD format.

Usage

```
getCellProcessedData(myBgeeObject, experimentId, package = "zellkonverter")
```

Arguments

`myBgeeObject`	A Reference Class Bgee object, notably specifying the targeted species and data type.
`experimentId`	Filter allowing to specify one ArrayExpress or GEO accession, e.g., GSE43721. Can not be null
`package`	Name of the R package used to load the H5AD cell data. It can either be \"anndata\" or \"zellkonverter\". If zellkonverter is used then its default reader (R) is used.

Author(s)

Julien Wollbrett

Examples

```
{
## Not run:
  bgee <- Bgee$new(species = "Gallus_gallus", dataType = "sc_droplet_based")
  cellProcessedData <- getCellProcessedData(bgee, experimentId = "ERP132576")

## End(Not run)
}
```

getData *Deprecated : Retrieve Bgee RNA-seq or Affymetrix data.*

Description

Deprecated : This function loads the quantitative expression data and presence calls for samples available from Bgee (rna_seq, affymetrix, sc_full_length).

Usage

```
getData(
  myBgeeObject,
  experimentId = NULL,
  sampleId = NULL,
  anatEntityId = NULL,
  stageId = NULL,
  cellTypeId = NULL,
  sex = NULL,
  strain = NULL,
  withDescendantAnatEntities = FALSE,
  withDescendantStages = FALSE,
  withDescendantCellTypes = FALSE
)
```

Arguments

myBgeeObject	A Reference Class Bgee object, notably specifying the targeted species and data type.
experimentId	Filter allowing to specify one or more ArrayExpress or GEO accession, e.g., GSE43721. Default is NULL: takes all available experiments for targeted species and data type.
sampleId	Filter allowing to specify one or more sample ID. Depending on the selected datatype this sample IDs can correspond to Chip IDs (affymetrix) or RNA-Seq library IDs (rna_seq). Default is NULL: takes all available samples for targeted species and data type.
anatEntityId	Filter allowing to specify one or more anatomical entity IDs from the UBERON ontology (http://uberon.github.io/). Default is NULL: takes all available anatomical entities for targeted species and data type.
stageId	Filter allowing to specify one or more developmental stage IDs from Developmental Stage Ontology (https://github.com/obophenotype/developmental-stage-ontologies). Default is NULL: takes all available developmental stages for targeted species and data type.
cellTypeId	Filter specific to single cell datatype (sc_full_length) allowing to specify one or more cell type IDs from the UBERON ontology (http://uberon.github.io/). Default is NULL: takes all available cell types for targeted species and data type. Available for Bgee 15.0 and after

`sex` Filter allowing to specify one or more sexes. Default is NULL: takes all available sexes for targeted species and data type. Available for Bgee 15.0 and after

`strain` Filter allowing to specify one or more strains. Default is NULL: takes all available strains for targeted species and data type. Available for Bgee 15.0 and after

`withDescendantAnatEntities`
Allows to filter on the selected anatEntityId and all its descendants. This functionality is available for Bgee 15.0 release and after

`withDescendantStages`
Allows to filter on the selected stageId and all its descendants. This functionality is available for Bgee 15.0 release and after

`withDescendantCellTypes`
Allows to filter on the selected cellTypeId and all its descendants. This functionality is available for Bgee 15.0 release and after

Value

Return a dataframe containing all Bgee processed expression data from the selected species and datatype using specified filters with operator AND.

Author(s)

Julien Wollbrett, Andrea Komljenovic and Julien Roux.

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  dataMouseGSE43721 <- getData(bgee, experimentId = "GSE43721")
  dataMouseVariousFilters <- getData(bgee, experimentId = c("GSE43721", "GSE36026"),
                    anatEntityId = c("UBERON:0002107", "UBERON:0000956", "UBERON:0002048"))
}
```

---

getIntegratedCalls *Retrieve Bgee calls from one species.*

Description

This function loads the integrated expression calls from one species. These calls are equivalent to the one provided in the gene page of the Bgee website. The calls have been generated using all datatypes present in Bgee (EST, In Situ, Affymetrix, bulk RNA-Seq and single-cell RNA-Seq).

Usage

```
getIntegratedCalls(
  myBgeeObject,
  conditionParameters = "anatEntity",
  advancedColumns = FALSE,
  geneIds = NULL,
  anatEntityIds = NULL
)
```

Arguments

myBgeeObject	A Reference Class Bgee object, notably specifying the targeted species and data type.
conditionParameters	Specify which condition parameter you are interested in. It can be `anatEntity` or `allCondParams`. The `anatEntity` option retrieve calls generated using only anatomical entity and cell-type as a condition parameter. The `allCondParams` option retrieve calls generated taking into consideration all condition parameters present in Bgee (anat, entity, cell-type, developmental stage, sex and strain). By default the `anatEntity` option is selected.
advancedColumns	Boolean allowing to specify if advancend columns are required. The default value is `FALSE` meaning that only gene , condition parameters, call quality, FDR, expression rank and expression score are retrieved. If `TRUE` then a lot more information used to generate the calls are retrieved like the number of self and descendant observation for each datatype, or the rank and score per datatype.
geneIds	List of genes for which expression calls have to be retrieved. It has to be ensembl IDs if the Bgee genome source is Ensembl (e.g ENSG00000244734) or RefSeq IDs if the Bgee genome source is RefSeq (e.g 734881) but not gene names (e.g HBB, Apoc1, etc.).
anatEntityIds	List of anatomical entity IDs for which expression calls have to be retrieved. It has to be IDs (e.g UBERON:0000955) and not names (e.g brain)

Value

Return a data.table containing all calls from the specified species

Author(s)

Julien Wollbrett

Examples

```
{
  bgee <- Bgee$new(species = "Danio_rerio")
  callsHer1 <- getIntegratedCalls(bgee, geneIds="ENSDARG00000014722")
}
```

getSampleProcessedData *Retrieve Bgee processed expression values for bulk RNA-seq, full length single-cell RNA-seq, droplet based single-cell RNA-seq or Affymetrix data.*

Description

This function loads the quantitative expression data and presence calls for samples available from Bgee. These data are available for all droplet based single-cell RNA-seq, full length single-cell RNA-seq, bulk RNA-seq and affymetrix samples present in Bgee. It is possible to filter the processed expression data using experiment IDs, chip IDs (affymetrix) or library IDs (RNA-seq), anatomical entity IDs from the UBERON ontology, developmental stage IDs, cell-type IDs, sex and/or strain. Celltype, anatomical entity and stage IDs are normalized and come from ontologies, allowing to retrieve data annotated with the selected term or with the term and all its descendants. This option allows for instance to retrieve all processed expression values coming from the brain or any subpart of the brain.

Usage

```
getSampleProcessedData(
  myBgeeObject,
  experimentId = NULL,
  sampleId = NULL,
  anatEntityId = NULL,
  stageId = NULL,
  cellTypeId = NULL,
  sex = NULL,
  strain = NULL,
  withDescendantAnatEntities = FALSE,
  withDescendantStages = FALSE,
  withDescendantCellTypes = FALSE
)
```

Arguments

myBgeeObject	A Reference Class Bgee object, notably specifying the targeted species and data type.
experimentId	Filter allowing to specify one or more ArrayExpress or GEO accession, e.g., GSE43721. Default is NULL: takes all available experiments for targeted species and data type.
sampleId	Filter allowing to specify one or more sample ID. Depending on the selected datatype this sample IDs can correspond to Chip IDs (affymetrix) or RNA-Seq library IDs (rna_seq). Default is NULL: takes all available samples for targeted species and data type.

`anatEntityId` Filter allowing to specify one or more anatomical entity IDs from the UBERON ontology (http://uberon.github.io/). Default is NULL: takes all available anatomical entities for targeted species and data type.

`stageId` Filter allowing to specify one or more developmental stage IDs from Developmental Stage Ontology (https://github.com/obophenotype/developmental-stage-ontologies). Default is NULL: takes all available developmental stages for targeted species and data type.

`cellTypeId` Filter specific to single cell datatype (sc_full_length) allowing to specify one or more cell type IDs from the UBERON ontology (http://uberon.github.io/). Default is NULL: takes all available cell types for targeted species and data type. Available for Bgee 15.0 and after

`sex` Filter allowing to specify one or more sexes. Default is NULL: takes all available sexes for targeted species and data type. Available for Bgee 15.0 and after

`strain` Filter allowing to specify one or more strains. Default is NULL: takes all available strains for targeted species and data type. Available for Bgee 15.0 and after

`withDescendantAnatEntities`
Allows to filter on the selected anatEntityId and all its descendants. This functionality is available for Bgee 15.0 release and after

`withDescendantStages`
Allows to filter on the selected stageId and all its descendants. This functionality is available for Bgee 15.0 release and after

`withDescendantCellTypes`
Allows to filter on the selected cellTypeId and all its descendants. This functionality is available for Bgee 15.0 release and after

Value

Return a dataframe containing all Bgee processed expression data from the selected species and datatype using specified filters with operator AND.

Author(s)

Julien Wollbrett, Andrea Komljenovic and Julien Roux.

Examples

```
{
  bgee <- Bgee$new(species = "Mus_musculus", dataType = "rna_seq")
  dataMouseGSE43721 <- getSampleProcessedData(bgee, experimentId = "GSE43721")
 dataMouseVariousFilters <- getSampleProcessedData(bgee, experimentId = c("GSE43721", "GSE36026"),
                   anatEntityId = c("UBERON:0002107", "UBERON:0000956", "UBERON:0002048"))
}
```

`listBgeeRelease` *List Bgee releases available to use with BgeeDB package*

Description

Returns information on available Bgee releases, the access URL for FTP and webservice, and the date of release

Usage

```
listBgeeRelease(release = NULL)
```

Arguments

`release`	A character specifying a targeted release number. In the form "Release.subrelease" or "Release_subrelease", e.g., "13.2" or 13_2". If not specified, all available releases are shown.

Value

A data frame with information on Bgee releases.

Author(s)

Julien Roux, Julien Wollbrett

Examples

```
{
 listBgeeRelease()
}
```

`listBgeeSpecies` *List species in the Bgee database and the available data types for each of them*

Description

Returns information on available species in Bgee

Usage

```
listBgeeSpecies(
  release = NULL,
  ordering = NULL,
  allReleases = NULL,
  removeFile = TRUE
)
```

Arguments

release	A character specifying a targeted release number. In the form "Release.subrelease" or "Release_subrelease", e.g., "13.2" or 13_2". If not specified, the latest release is used.
ordering	A numeric indicating the number of the column which should be used to sort the data frame. Default NULL, returning unsorted data frame.
allReleases	A data frame with information on all releases. Avoid redownloading this information if .getBgeeRelease() already called.
removeFile	Boolean indicating whether the downloaded file should be deleted. Default to TRUE.

Value

A data frame with species Id, genus name, species name, common name and data type availability for targeted Bgee release

Author(s)

Julien Roux

Examples

```
{
 listBgeeSpecies()
 # species present in a specific Bgee release
 listBgeeSpecies(release = "13.2")
 # in order to order species according to theit taxonomical IDs
 listBgeeSpecies(ordering = 1)
}
```

---

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(
  myBgeeObject,
  callType = "presence",
  confidence = NULL,
  stage = NULL,
  timeout = 1800
)
```

Arguments

myBgeeObject	An output object from Bgee$new().
callType	A character of indicating the type of expression calls to be used for enrichment. Only calls for significant detection of expression are implemented so far ("presence"). Differential expression calls, based on differential expression analysis, might be implemented in the future.
confidence	A character indicating if only high quality present calls should be retrieved. For Bgee releases prior to 14, options are "all" (default) or "high_quality". For Bgee release 14 and above, options are "silver" (default) and "gold".
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 is specified, 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 developmental stage displaying expression; this is equivalent to specifying stage="UBERON:0000104" (life cycle, the root of the stage ontology). For information, 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)
timeout	local timeout used when the function is run. It allows to modify the timeout used by default in R to download files. If not provided the timeout will be fixed at 1800 secondes. When the functions exits (either naturally or as the result of an error) the download timeout go back to the original one used in the R session.

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) from 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 atlas 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 might 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 4 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).
- The Bgee class object thta was used to retrieve the data.

Author(s)

Julien Roux, Julien Wollbrett

Examples

```
{
  bgee <- Bgee$new(species = "Danio_rerio", dataType = "rna_seq")
  myTopAnatData <- loadTopAnatData(bgee)
}
```

---

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, ordering = 7)
```

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.
ordering	A numeric indicating which column should be used to sort the data frame. If the column number is preceded by a \"-\" sign, results are displayed in decreasing ordering. Default is "7", returning data frame sorted by p-values in increasing order.

Value

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

Author(s)

Julien Roux

Examples

```
{
bgee <- Bgee$new(species="Bos_taurus", dataType="rna_seq")
myTopAnatData <- loadTopAnatData(bgee, stage="UBERON:0000092")
geneList <- as.factor(c(rep(0, times=85), rep(1, times=15)))
names(geneList) <- c("ENSBTAG00000000011","ENSBTAG00000000014","ENSBTAG00000000016",
                     "ENSBTAG00000000026","ENSBTAG00000000039","ENSBTAG00000000040",
                     "ENSBTAG00000000042","ENSBTAG00000000050","ENSBTAG00000000056",
                     "ENSBTAG00000000064","ENSBTAG00000000067","ENSBTAG00000000071",
                     "ENSBTAG00000000072","ENSBTAG00000000080","ENSBTAG00000000081",
                     "ENSBTAG00000000084","ENSBTAG00000000091","ENSBTAG00000000099",
                     "ENSBTAG00000000111","ENSBTAG00000000123","ENSBTAG00000000132",
                     "ENSBTAG00000000153","ENSBTAG00000000162","ENSBTAG00000000163",
                     "ENSBTAG00000000169","ENSBTAG00000000179","ENSBTAG00000000197",
                     "ENSBTAG00000000199","ENSBTAG00000000202","ENSBTAG00000000203",
                     "ENSBTAG00000000204","ENSBTAG00000000213","ENSBTAG00000000215",
                     "ENSBTAG00000000223","ENSBTAG00000000224","ENSBTAG00000000225",
                     "ENSBTAG00000000236","ENSBTAG00000000250","ENSBTAG00000000251",
                     "ENSBTAG00000000252","ENSBTAG00000000253","ENSBTAG00000000261",
                     "ENSBTAG00000000274","ENSBTAG00000000277","ENSBTAG00000000279",
                     "ENSBTAG00000000285","ENSBTAG00000000286","ENSBTAG00000000287",
                     "ENSBTAG00000000289","ENSBTAG00000000297","ENSBTAG00000000305",
                     "ENSBTAG00000000312","ENSBTAG00000000328","ENSBTAG00000000335",
                     "ENSBTAG00000000341","ENSBTAG00000000343","ENSBTAG00000000354",
                     "ENSBTAG00000000355","ENSBTAG00000000356","ENSBTAG00000000365",
                     "ENSBTAG00000000372","ENSBTAG00000000379","ENSBTAG00000000380",
                     "ENSBTAG00000000382","ENSBTAG00000000396","ENSBTAG00000000404",
                     "ENSBTAG00000000405","ENSBTAG00000000406","ENSBTAG00000000411",
                     "ENSBTAG00000000425","ENSBTAG00000000434","ENSBTAG00000000435",
                     "ENSBTAG00000000438","ENSBTAG00000000448","ENSBTAG00000000451",
```

```
                        "ENSBTAG00000000454","ENSBTAG00000000456","ENSBTAG00000000457",
                        "ENSBTAG00000000459","ENSBTAG00000000462","ENSBTAG00000000469",
                        "ENSBTAG00000000470","ENSBTAG00000000484","ENSBTAG00000000497",
                        "ENSBTAG00000000501","ENSBTAG00000009707","ENSBTAG00000026266",
                        "ENSBTAG00000021992","ENSBTAG00000005353","ENSBTAG00000005333",
                        "ENSBTAG00000006424","ENSBTAG00000026972","ENSBTAG00000010799",
                        "ENSBTAG00000010799","ENSBTAG00000014614","ENSBTAG00000014614",
                        "ENSBTAG00000045757","ENSBTAG00000046332","ENSBTAG00000046332",
                        "ENSBTAG00000008394")
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 topAnatData class object, ready for gene set enrichment testing with topGO.

Author(s)

Julien Roux

Examples

```
{
bgee <- Bgee$new(species="Bos_taurus", dataType="rna_seq")
myTopAnatData <- loadTopAnatData(bgee, stage="UBERON:0000092")
geneList <- as.factor(c(rep(0, times=85), rep(1, times=15)))
names(geneList) <- c("ENSBTAG00000000011","ENSBTAG00000000014","ENSBTAG00000000016",
                     "ENSBTAG00000000026","ENSBTAG00000000039","ENSBTAG00000000040",
                     "ENSBTAG00000000042","ENSBTAG00000000050","ENSBTAG00000000056",
                     "ENSBTAG00000000064","ENSBTAG00000000067","ENSBTAG00000000071",
                     "ENSBTAG00000000072","ENSBTAG00000000080","ENSBTAG00000000081",
                     "ENSBTAG00000000084","ENSBTAG00000000091","ENSBTAG00000000099",
                     "ENSBTAG00000000111","ENSBTAG00000000123","ENSBTAG00000000132",
                     "ENSBTAG00000000153","ENSBTAG00000000162","ENSBTAG00000000163",
                     "ENSBTAG00000000169","ENSBTAG00000000179","ENSBTAG00000000197",
                     "ENSBTAG00000000199","ENSBTAG00000000202","ENSBTAG00000000203",
                     "ENSBTAG00000000204","ENSBTAG00000000213","ENSBTAG00000000215",
                     "ENSBTAG00000000223","ENSBTAG00000000224","ENSBTAG00000000225",
                     "ENSBTAG00000000236","ENSBTAG00000000250","ENSBTAG00000000251",
                     "ENSBTAG00000000252","ENSBTAG00000000253","ENSBTAG00000000261",
                     "ENSBTAG00000000274","ENSBTAG00000000277","ENSBTAG00000000279",
                     "ENSBTAG00000000285","ENSBTAG00000000286","ENSBTAG00000000287",
                     "ENSBTAG00000000289","ENSBTAG00000000297","ENSBTAG00000000305",
                     "ENSBTAG00000000312","ENSBTAG00000000328","ENSBTAG00000000335",
                     "ENSBTAG00000000341","ENSBTAG00000000343","ENSBTAG00000000354",
                     "ENSBTAG00000000355","ENSBTAG00000000356","ENSBTAG00000000365",
                     "ENSBTAG00000000372","ENSBTAG00000000379","ENSBTAG00000000380",
                     "ENSBTAG00000000382","ENSBTAG00000000396","ENSBTAG00000000404",
                     "ENSBTAG00000000405","ENSBTAG00000000406","ENSBTAG00000000411",
                     "ENSBTAG00000000425","ENSBTAG00000000434","ENSBTAG00000000435",
                     "ENSBTAG00000000438","ENSBTAG00000000448","ENSBTAG00000000451",
                     "ENSBTAG00000000454","ENSBTAG00000000456","ENSBTAG00000000457",
                     "ENSBTAG00000000459","ENSBTAG00000000462","ENSBTAG00000000469",
                     "ENSBTAG00000000470","ENSBTAG00000000484","ENSBTAG00000000497",
                     "ENSBTAG00000000501","ENSBTAG00000009707","ENSBTAG00000026266",
                     "ENSBTAG00000021992","ENSBTAG00000005353","ENSBTAG00000005333",
                     "ENSBTAG00000006424","ENSBTAG00000026972","ENSBTAG00000010799",
                     "ENSBTAG00000010799","ENSBTAG00000014614","ENSBTAG00000014614",
                     "ENSBTAG00000045757","ENSBTAG00000046332","ENSBTAG00000046332",
                     "ENSBTAG00000008394")
myTopAnatObject <-  topAnat(myTopAnatData, geneList)
}
```

Index

∗ **datasets**
    geneList, 6

Bgee *(*Bgee-class*)*, 2
Bgee-class, 2

deleteLocalData, 3
deleteOldData, 4

formatData, 5

geneList, 6
getAnnotation, 7
getCellProcessedData, 8
getData, 9
getIntegratedCalls, 10
getSampleProcessedData, 12

listBgeeRelease, 14
listBgeeSpecies, 14
loadTopAnatData, 15

makeTable, 17

topAnat, 19