Package ‘CTDquerier’

April 15, 2019

Title Package for CTDbase data query, visualization and downstream analysis

Version 1.2.0

Maintainer Carles Hernandez-Ferrer <carles.hernandez@isglobal.org>

Description Package to retrieve and visualize data from the Comparative Toxicogenomics Database (http://ctdbase.org/). The downloaded data is formated as DataFrames for further downstream analyses.

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports RCurl, stringr, S4Vectors, stringdist, ggplot2, igraph, utils, grid, gridExtra, methods, stats, BiocFileCache, rappdirs

VignetteBuilder knitr

Suggests BiocStyle, knitr

RoxygenNote 6.0.1

biocViews Software, BiomedicalInformatics, Infrastructure, DataImport, DataRepresentation, GeneSetEnrichment, NetworkEnrichment, Pathways, Network, GO, KEGG

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

git_branch RELEASE_3_8

git_last_commit fd2eb0a

git_last_commit_date 2018-10-30

Date/Publication 2019-04-15

Author Carles Hernandez-Ferrer [aut, cre],
Jaun R. Gonzalez [aut]

R topics documented:

`CTDdata` *Class CTDdata*

Description

Class resulting of `query_ctd_gene`, `query_ctd_chem` and `query_ctd_dise`. It is used to encapsulate all the information in *CTDbase* for given set of genes, chemicals or diseases.

Usage

```
## S4 method for signature 'CTDdata'
enrich(x, y, universe, use = "curated", warnings = TRUE,
  ...)

## S4 method for signature 'CTDdata'
get_table(object, index_name, ...)

## S4 method for signature 'CTDdata'
get_terms(object)

## S4 method for signature 'CTDdata,ANY'
plot(x, y, index_name = "base",
  representation = "heatmap", ...)
```

Arguments

`x`	Object of class `CTDdata`.
`y`	NOT USED
`universe`	String vector of genes used as universe. If not provided, all genes in CTDbase are used.
`use`	Select if all or only curated relations are used.
`warnings`	Shows or hiddes warnings.
`...`	NOT USED
`object`	Object of class `CTDdata`.

`index_name` Name of the plot to be draw. `"base"` shows a bar-plot indicating the number of terms lost&found at CTDbase. For *gene* queries `index_name` can take values `"disease"`, `"chemical"`, `"gene-gene interaction"`, `"kegg pathways"` and `"go terms"`. The first two have only `"heatmap"` representation, the last to have only `"network"` representation, and `"gene-gene interaction"` has both. For *chemical* queries `index_name` can take values `"gene"`, `"disease"`, `"kegg pathways"` and `"go terms"`. The first has both `"heatmap"` and `"network"` representation, while the last only `"network"`. The otehr three have only `"heatmap"` representation. For *disease* queries `index_name` can take values `"gene"`, `"chemical"` and `"kegg pathways"`. The first two have `"heatmap"` representation while the last `"network"` representation.

`representation` Can take values `"heatmap"` or `"network"`.

Value

An object of class `CTDdata`

Methods (by generic)

- `enrich`: Method to perform enrichment analysis given two object of class `CTDdata`.
- `get_table`: Method to obtain a specific inner table from an `CTDdata` object.
- `get_terms`: Return a list with the terms found to create the object.
- `plot`: Generates a basic plot showing the number of terms that can be used to query CTDbase.

Slots

`type` Character saving `"GENE"`, `"CHEMICAL"` or `"DISEASE"` depending if it was created using `query_ctd_gene`, `query_ctd_chem` or `query_ctd_dise`

`terms` `DataFrame` with the genes, chemicals or diseases used to create the object.

`losts` Character with the terms used to create the object but that were nor present in CTDbase.

`gene_interactions` (Only for chemicals) Table with a relation of the genes interacting with the given chemicals.

`chemicals_interactions` (Only for genes) Table with a relation of the chemicals interacting with the given genes.

`diseases` Table with a relation of the diseases associated with given genes or chemicals.

`gene_gene_interactions` (Only for genes) Table with a relation of the genes interacting with the given genes.

`kegg` Table with a relation of the KEGG pathways affected by the given chemicals or where the given genes play a role.

`go` Table with a relatio of the GO terms afected by the given chemicals or where the given genes play a role.

See Also

`query_ctd_gene` to create a `CTDdata` from a set of genes, `query_ctd_chem` to create a `CTDdata` from a set of chemicals, `query_ctd_dise` to create a `CTDdata` from a set of diseases, `get_table` to retrive encapsulated data and `plot` to get nice plots from stored data.

`CTDquerier` *Package for CTDbase data query, data visualization and data analysis.*

Description

Package for CTDbase data query, data visualization and data analysis.

Data Download

`CTDquerier` offers two functions to query CTDbase (http://ctdbase.org): `query_ctd_gene` to query CTDbase given a set of genes; and `query_ctd_chem` to query CTDbase given a set of chemicals. Both functions returns `CTDdata` objects. Raw downloaded information can be retrieved from `CTDdata` using method `get_table`.

Data Visualization

`CTDdata` objects offers basic visualization of the downloaded information using standard `plot` method.

`download_ctd_chem` *Function to download checmicals available in CTDbase*

Description

This function download the "Chemical vocabulary" file (`CTD_chemicals.tsv.gz`) from `http://ctdbase.org/downloa`

Usage

```
download_ctd_chem(filename = "CTD_chemicals.tsv.gz", mode = "auto",
  verbose = FALSE)
```

Arguments

`filename`	(default `"CTD_chemicals.tsv.gz"`) Name of the file in the local system.
`mode`	(default `"auto"`) Mode passed to `download.file`.
`verbose`	(default `FALSE`) If set to `TRUE` is shows relevant information of each step.

Details

The field included in the file (`CTD_chemicals.tsv.gz`) are:

1. ChemicalName
2. ChemicalID (MeSH identifier)
3. CasRN (CAS Registry Number, if available)
4. Definition
5. ParentIDs (identifiers of the parent terms; ’|’-delimited list),
6. TreeNumbers (identifiers of the chemical’s nodes; ’|’-delimited list),
7. ParentTreeNumbers (identifiers of the parent nodes; ’|’-delimited list),
8. Synonyms (’|’-delimited list)
9. DrugBankIDs (’|’-delimited list)

Value

Passed name into `filename` argument if it could be download `1` otherwise.

Examples

```
download_ctd_chem()
file.exists( "CTD_chemicals.tsv.gz" )
```

`download_ctd_dise` *Function to download diseases available in CTDbase*

Description

This function download the "Disease vocabulary" file (`CTD_diseases.tsv.gz`) from `http://ctdbase.org/downloads.`

Usage

```
download_ctd_dise(filename = "CTD_diseases.tsv.gz", mode = "auto",
  verbose = FALSE)
```

Arguments

`filename`	(default `"CTD_diseases.tsv.gz"`) Name of the file in the local system.
`mode`	(default `"auto"`) Mode passed to `download.file`.
`verbose`	(default `FALSE`) If set to `TRUE` is shows relevant information of each step.

Details

The field included in the file (`CTD_diseases.tsv.gz`) are:

1. DiseaseName
2. DiseaseID (MeSH or OMIM identifier)
3. Definition
4. AltDiseaseIDs (alternative identifiers; ’|’-delimited list)
5. ParentIDs (identifiers of the parent terms; ’|’-delimited list)
6. TreeNumbers (identifiers of the disease’s nodes; ’|’-delimited list)
7. ParentTreeNumbers (identifiers of the parent nodes; ’|’-delimited list)
8. Synonyms (’|’-delimited list)
9. SlimMappings (MEDIC-Slim mappings; ’|’-delimited list)

Value

Passed name into `filename` argument if it could be download `1` otherwise. download_ctd_dise() file.exists("CTD_diseases.tsv.gz")

download_ctd_genes *Function to download genes available in CTDbase*

Description

This function download the "Gene vocabulary" file (`CTD_genes.tsv.gz`) from `http://ctdbase.org/downloads`.

Usage

```
download_ctd_genes(verbose = FALSE)
```

Arguments

`verbose`	(default `FALSE`) If set to `TRUE` is shows relevant information of each step.

Details

The field included in the file (`CTD_genes.tsv.gz`) are:

1. GeneSymbol
2. GeneName
3. GeneID (NCBI Gene identifier)
4. AltGeneIDs (alternative NCBI Gene identifiers; '|'-delimited list)
5. Synonyms ('|'-delimited list)
6. BioGRIDIDs ('|'-delimited list)
7. PharmGKBIDs ('|'-delimited list)
8. UniprotIDs ('|'-delimited list)

Value

Passed name into `filename` argument if it could be download `1` otherwise.

Examples

```
download_ctd_genes()
```

enrich *Method to perform enrichment analysis using two `CTDdata` objects*

Description

This methods performs a fisher test using the genes in two objects of class `CTDdata`. The object in 'x' is used as source while the object on 'y' is used as universe. When object 'x' corresponds to an object created with `query_ctd_gene`, the used genes are the found terms in CTDbase. In the other cases (chemicals and disease `CTDdata`), the genes from the 'gene interactions' table are used. If `universe` is missing, all genes in CTDbase are used as universe.

Usage

```
enrich(x, y, universe, use = "curated", warnings = TRUE, ...)
```

Arguments

`x` Object of class `CTDdata`.

`y` Object of class `CTDdata`.

`universe` Vector of strings corresponding to the genes to be used as universe.

`use` (default: `"curated"`) It can take values `"curated"` or `"all"` to filter or not filter for curated the genes into the `CTDdata` objects.

`warnings` (default: `TRUE`).

`...` NOT USED

Value

A list with class `htest`. Check `fisher.test` for more information.

`gala` `CTDdata` *for ilustrative purpouses*

Description

`CTDdata` with information of 258 genes downloaded from CTDbase. The object was created from from the genes obtained from the scientific article entitleed Case-control admixture mapping in Latino populations enriches for known asthma-associated genes" (Table E1) by Torgerson et. al. The genes were used to query CTDbase using `query_ctd_genes` function.

Usage

```
data("gala")
```

Format

An object of class `CTDdata` of length 1.

Value

An `CTDdata` object.

Examples

```
data("gala")
gala
```

get_table *Method to obtain a specific inner table from a* `CTDdata` *object.*

Description

Obtain the raw data from a `CTDdata` object, result from a query to CTDbase.

Usage

```
get_table(object, index_name, ...)
```

Arguments

`object`	Object of class `CTDdata`.
`index_name`	String indicating the type of data to obtain.
`...`	NOT USED

Details

Available tables are (`index_name`):

1. `"gene interactions"`: (Only for chemicals) Table with a relation of the genes interacting with the given chemicals.
2. `"chemical interactions"`: (Only for genes) Table with a relation of the chemicals interacting with the given genes.
3. `"diseases"`: Table with a relation of the diseases associated with given genes or chemicals.
4. `"gene-gene interactions"`: (Only for genes) Table with a relation of the genes interacting with the given genes.
5. `"kegg pathways"`: Table with a relation of the KEGG pathways affected by the given chemicals or where the given genes play a role.
6. `"go terms"`: Table with a relatio of the GO terms afected by the given chemicals or where the given genes play a role.

Value

A `DataFrame` containing the raw result from CTDdata.

Examples

```
data("gala")
get_table(gala, "diseases")[1:3, ]
```

get_terms *Getter to obtain the terms used to perform a query into CTDbase*

Description

Getter to obtain the terms used to perform a query into CTDbase

Usage

```
get_terms(object)
```

Arguments

object	Object of class CTDdata.

Value

A list with two accessors: "used" for the terms that exsist in CTDbase, and "lost" with the terms that do not existi in CTDbase.

Examples

```
data("gala")
get_terms(gala)[["lost"]]
```

leaf_plot *Function to create a leaf plot*

Description

This functions taked a data.frame and returns a gtable with three plots. The left-leafes, the axis names and the right-leafes.

Usage

```
leaf_plot(dta, label = "name", valueLeft = "var1", valueRight = "var2",
  titleLeft = NULL, titleRight = NULL, colorLeft = "#FF7F50",
  colorRight = "#20B2AA")
```

Arguments

dta	data.frame with, at last, three columns corresponding to the axis labels, the left values and the right values.
label	(default "name") Name of the column in dta with the labels.
valueLeft	(default "var1") Name of the column with the values for the left plot.
valueRight	(default "var2") Name of the column with the values for the right plot.
titleLeft	(default NULL) Character used as a title for the left plot.
titleRight	(default NULL) Character used as a title for the right plot.
colorLeft	(default "#FF7F50") Color for left plot bars.
colorRight	(default "#20B2AA") Color for right plot bars.

Value

A ggplo2 object.

Examples

```
data <- data.frame(
   labels = LETTERS[1:15],
   right = runif(n = 15) * 11,
   left = runif(n = 15) * 9
)
leaf_plot( data, "labels", "left", "right", "runif09", "runif11")
```

load_ctd_chem *Function to load the* `.tsv.gz` *file for chemicals*

Description

Function to load the `.tsv.gz` file for chemicals

Usage

```
load_ctd_chem(filename = "CTD_chemicals.tsv.gz")
```

Arguments

`filename`	(default "CTD_chemicals.tsv.gz") File with the chemicals downloaded from CTDbase.

Details

The field included in the file (`CTD_chemicals.tsv.gz`) are:

1. ChemicalName
2. ChemicalID (MeSH identifier)
3. CasRN (CAS Registry Number, if available)
4. Definition
5. ParentIDs (identifiers of the parent terms; ’|’-delimited list),
6. TreeNumbers (identifiers of the chemical’s nodes; ’|’-delimited list),
7. ParentTreeNumbers (identifiers of the parent nodes; ’|’-delimited list),
8. Synonyms (’|’-delimited list)
9. DrugBankIDs (’|’-delimited list)

Value

A `data.frame` with the content of the file "CTD_genes.tsv.gz"

Examples

```
download_ctd_chem()
fdl <- load_ctd_chem()
dim( fdl )
```

load_ctd_dise *Function to load the .tsv.gz file for disease*

Description

Function to load the .tsv.gz file for disease

Usage

```
load_ctd_dise(filename = "CTD_diseases.tsv.gz")
```

Arguments

filename	(default "CTD_diseases.tsv.gz") File with the chemicals downloaded from CTDbase.

Details

The field included in the file (CTD_diseases.tsv.gz) are:

1. DiseaseName
2. DiseaseID (MeSH or OMIM identifier)
3. Definition
4. AltDiseaseIDs (alternative identifiers; '|'-delimited list)
5. ParentIDs (identifiers of the parent terms; '|'-delimited list)
6. TreeNumbers (identifiers of the disease's nodes; '|'-delimited list)
7. ParentTreeNumbers (identifiers of the parent nodes; '|'-delimited list)
8. Synonyms ('|'-delimited list)
9. SlimMappings (MEDIC-Slim mappings; '|'-delimited list)

Value

A data.frame with the content of the file "CTD_genes.tsv.gz"

Examples

```
download_ctd_dise()
fdl <- load_ctd_dise()
dim( fdl )
```

load_ctd_gene *Function to load the .tsv.gz file for genes*

Description

This function works in pair with download_ctd_genes. This function loads into the R session the downloaded "CTD_genes.tsv.gz" file.

Usage

```
load_ctd_gene()
```

Details

The field included in the file (CTD_genes.tsv.gz) are:

1. GeneSymbol
2. GeneName
3. GeneID (NCBI Gene identifier)
4. AltGeneIDs (alternative NCBI Gene identifiers; '|'-delimited list)
5. Synonyms ('|'-delimited list)
6. BioGRIDIDs ('|'-delimited list)
7. PharmGKBIDs ('|'-delimited list)
8. UniprotIDs ('|'-delimited list)

Value

A data.frame with the content of the file "CTD_genes.tsv.gz"

Examples

```
download_ctd_genes()
fdl <- load_ctd_gene()
dim( fdl )
```

query_ctd_chem *Function to query CTDbase using chemical terminology (Chemical Name)*

Description

This function cheks for CTDbase gene vocabulary and query CTDbase for each one, downloading chemica-genes interactions, associated diseases, associated KEGG pathways and associated GO terms.

Usage

```
query_ctd_chem(terms, filename = "CTD_chemicals.tsv.gz", mode = "auto",
  max.distance = 10, verbose = FALSE)
```

Arguments

`terms`	Character vector with the chemicals used in the query.
`filename`	(default `"CTD_chemicals.tsv.gz"`) Name of the file to store the CTDbase chemicals vocabilary.
`mode`	(default `"auto"`) Mode passed to `download.file`.
`max.distance`	(default `10`) Maximum distance allowed between a given element in `terms` argument and a possible match in CTDbase.
`verbose`	(default `FALSE`) If set to `TRUE` is shows relevant information of each step.

Value

An object of class `CTDdata`.

Examples

```
rst <- query_ctd_chem( terms = c( "Iron", "Air Pollutants" ), verbose = TRUE )
```

`query_ctd_dise` *Function to query CTDbase using disease terminology*

Description

This function cheks for CTDbase disease vocabulary and query CTDbase for each one, downloading disease-gene interactions, chemicals interactions, associated desease, associated KEGG pathways and associated GO terms.

Usage

```
query_ctd_dise(terms, filename = "CTD_diseases.tsv.gz", mode = "auto",
  verbose = FALSE)
```

Arguments

`terms`	Character vector with the diseases used in the query.
`filename`	(default `"CTD_diseases.tsv.gz"`) Name of the file to store the CTDbase disease vocabilary.
`mode`	(default `"auto"`) Mode passed to `download.file`.
`verbose`	(default `FALSE`) If set to `TRUE` is shows relevant information of each step.

Value

An object of class `CTDdata`.

Examples

```
rst <- query_ctd_dise( terms = "Asthma", verbose = TRUE )
```

query_ctd_gene *Function to query CTDbase using gene terminology (Gene Symbol)*

Description

This function checks for CTDbase gene vocabulary and query CTDbase for each one, downloading gene-gene interactions, chemicals interactions, associated disease, associated KEGG pathways and associated GO terms.

Usage

```
query_ctd_gene(terms, verbose = FALSE)
```

Arguments

terms	Character vector with the genes used in the query.
verbose	(default FALSE) If set to TRUE is shows relevant information of each step.

Value

An object of class CTDdata.

Examples

```
rst <- query_ctd_gene( terms = c( "APP", "HMOX1A", "hmox1" ), verbose = TRUE )
```

Index