Package 'CALANGO'

April 26, 2023

Type Package

Title Comparative Analysis with Annotation-Based Genomic Components

Version 1.0.16

Date 2023-04-26

Language en-US

Maintainer Felipe Campelo <fcampelo@gmail.com>

Description A first-principle, phylogeny-aware comparative genomics tool for investigating associations between terms used to annotate genomic components (e.g., Pfam IDs, Gene Ontology terms,) with quantitative or rank variables such as number of cell types, genome size, or density of specific genomic elements. See the project website for more information, documentation and examples, and <doi:10.1016/j.patter.2023.100728> for the full paper.

License GPL-2

Depends R (>= 3.6.0)

Imports assertthat (>= 0.2.1), pbmcapply (>= 1.5.0), ape (>= 5.3.0), rmarkdown (>= 2.1.0), nlme (>= 3.1.0), BiocManager (>= 1.30.10), taxize (>= 0.9.92), dendextend (>= 1.15.2), heatmaply (>= 1.1.0), ggplot2 (>= 2.3.2), plotly (>= 4.9.2), DT (>= 0.13), htmltools (>= 0.5.0), htmlwidgets (>= 1.5.1), pkgdown (>= 1.5.1), knitr (>= 1.28)

Suggests AnnotationDbi, KEGGREST, GO.db

Encoding UTF-8

RoxygenNote 7.2.2

URL https://labpackages.github.io/CALANGO/

BugReports https://github.com/fcampelo/CALANGO/issues/

VignetteBuilder knitr

NeedsCompilation no

Author Francisco Lobo [aut],
Felipe Campelo [aut, cre],
Jorge Augusto Hongo [aut],
Giovanni Marques de Castro [aut],
Gabriel Almeida [sad, dnc]

Repository CRAN

Date/Publication 2023-04-26 13:22:33 UTC

R topics documented:

`install_bioc_dependencies` *Install Bioconductor dependencies*

Description

This function installs the latest versions of all Bioconductor packages required for the report generation, namely:

- AnnotationDbi
- KEGGREST
- GO.db

Usage

```
install_bioc_dependencies(bioc.args = list(), force = FALSE)
```

Arguments

`bioc.args`	list containing further arguments to be passed down to 'BiocManager::install()'.
`force`	logical: reinstall already-installed packages?

Details

It is essential that these Bioconductor packages be installed for CALANGO to work properly. It uses 'BiocManager::install()' for installing Bioconductor packages. Further arguments to this function are passed as a list.

Value

No return value, called for side effects.

Examples

```
## Not run:
  install_bioc_dependencies()

## End(Not run)
```

`make_report` *Prepare and render the HTML5 report*

Description

This script generates the HTML5 report based on an enriched 'CALANGO'-type list output by 'run_CALANGO()'.

Usage

```
make_report(defs, render.report = TRUE)
```

Arguments

`defs`	an enriched 'CALANGO'-type list generated by 'run_CALANGO()'
`render.report`	logical: should the HTML5 report be generated (for internal use only)

Value

Updated 'defs' list, containing:

- All input parameters originally passed (see [run_CALANGO()] for details).
- Derived fields calculated for displaying the results, including several statistical summaries of the data (including correlations, contrasts, covariances, p-values).

This function is mainly called for its side effect, namely the generation of an HTML5 report of the analysis, which is saved to the folder indicated in 'defs$output.dir'.

`retrieve_calanguize_genomes` *Retrieve calanguize_genomes script from the Github repository*

Description

This script downloads the *calanguize_genomes.pl* Perl script from the repository, together with associated README instructions for using the script, managing dependencies, etc. It will extract the data into a folder containing everything that is needed for preparing data for using CALANGO.

Usage

```
retrieve_calanguize_genomes(
  target.dir,
  method = "auto",
  unzip = getOption("unzip")
)
```

Arguments

`target.dir`	path to the folder where the files will be saved (accepts relative and absolute paths)
`method`	Method to be used for downloading files. Current download methods are "internal", "wininet" (Windows only) "libcurl", "wget" and "curl", and there is a value "auto": see _Details_ and _Note_ in the documentation of `utils::download.file()`.
`unzip`	The unzip method to be used. See the documentation of `utils::unzip()` for details.

Details

If the ‘target.dir‘ provided does not exist it is created (recursively) by the function.

Value

No return value, called for side effects (see Description).

Examples

```
## Not run:
  CALANGO::retrieve_calanguize_genomes(target.dir = "./data")

## End(Not run)
```

`retrieve_data_files` *Retrieve data files from the Github repository*

Description

This script downloads relevant data files from the CALANGO project repository. It will extract the data into a folder containing directories related to dictionary files, Gene Ontology annotation files, tree files, etc. Note: you may need to edit the file paths in the example scripts contained under the ‘parameters‘ subfolder of ‘target.dir‘, or pass an appropriate base path using parameter ‘basedir‘ in [run_CALANGO()].

Usage

```
retrieve_data_files(
  target.dir,
  method = "auto",
  unzip = getOption("unzip"),
  ...
)
```

Arguments

`target.dir`	path to the folder where the files will be saved (accepts relative and absolute paths)
`method`	Method to be used for downloading files. Current download methods are "internal", "wininet" (Windows only) "libcurl", "wget" and "curl", and there is a value "auto": see _Details_ and _Note_ in the documentation of `utils::download.file()`.
`unzip`	The unzip method to be used. See the documentation of `utils::unzip()` for details.
`...`	additional attributes (currently ignored)

Details

If the ‘target.dir‘ provided does not exist it is created (recursively) by the function.

Value

No return value, called for side effects (see Description).

Examples

```
## Not run:
  CALANGO::retrieve_data_files(target.dir = "./data")

## End(Not run)
```

run_CALANGO *Run the CALANGO pipeline*

Description

This function runs the complete workflow of CALANGO and generates the HTML5 output pages and export files.

Usage

```
run_CALANGO(
  defs,
  type = "correlation",
  cores = NULL,
  render.report = TRUE,
  basedir = ""
)
```

Arguments

defs	either a CALANGO-type list object or a path to a text file containing the required definitions (see Details).
type	type of analysis to perform. Currently only "correlation" is supported.
cores	positive integer, how many CPU cores to use (multicore acceleration does not work in Windows systems). Setting this parameter overrides any 'cores' field from 'defs'. Multicore support is currently implemented using the 'parallel' package, which uses forking (which means that multicore support is not available under Windows)
render.report	logical: should a HTML5 report be generated?
basedir	path to base folder to which all relative paths in 'defs' refer to.

Details

The script expects a 'CALANGO'-type list, passed either as an actual list object or as a file path. In the latter case, notice that the file must be a text file with a 'field = value' format. Blank likes and lines starting with '#' are ignored. The function expects the input list to have the following fields:

- annotation.files.dir (required, string) - Folder where annotation files are located.
- output.dir (required, string) - output folder for results
- dataset.info (required, string) - genome metadata file, it should contain at least:
  - File names. Please notice this information should be the first column in metadata file;
  - Phenotype data (numeric, this is the value CALANGO uses to rank species when searching for associations)

  - Normalization data (numeric, this is the value CALANGO uses as a denominator to compute annotation term frequencies to remove potential biases caused by, for instance, over annotation of model organisms or large differences in the counts of genomic elements). Please notice that CALANGO does not require normalization data for GO, as it computes the total number of GO terms per species and uses it as a normalizing factor.

- `x.column` (required, numeric) - which column in "dataset.info" contains the phenotype data?
- `ontology` (required, string) - which dictionary data type to use? Possible values are "GO" and "other". For GO, CALANGO can compute normalization data.
- `dict.path` (required, string) - file for dictionary file (two-column file containing annotation IDs and their descriptions. Not needed for GO.
- `column` (required, string) - which column in annotation files should be used (column name)
- `denominator.column` (optional, numeric) - which column contains normalization data (column number)
- `tree.path` (required, string) - path for tree file in either newick or nexus format
- `tree.type` (required, string) - tree file type (either "nexus" or "newick")
- `cores` (optional, numeric) - how many cores to use? If not provided the function defaults to 1.
- `linear.model.cutoff` (required, numeric) - parameter that regulates how much graphical output is produced. We configure it to generate plots only for annotation terms with corrected q-values for phylogenetically independent contrasts (standard: smaller than 0.5).
- `MHT.method` (optional, string) - type of multiple hypothesis correction to be used. Accepts all methods listed by ‘stats::p.adjust.methods()‘. If not provided the function defaults to "BH".

Value

Updated ‘defs‘ list, containing:

- All input parameters originally passed or read from a ‘defs‘ file (see **Details**).
- Derived fields loaded and preprocessed from the files indicated in ‘defs‘.
- Several statistical summaries of the data (used to render the report), including correlations, contrasts, covariances, p-values and other summary statistics.

Results are also saved to files under ‘defs$output.dir‘.

Examples

```
## Not run:

## Install any missing BioConductor packages for report generation
## (only needs to be done once)
# CALANGO::install_bioc_dependencies()

# Retrieve example files
basedir <- tempdir()
retrieve_data_files(target.dir = paste0(basedir, "/data"))
defs <- paste0(basedir, "/data/parameters/parameters_domain2GO_count_less_phages.txt")

# Run CALANGO
```

```
res <- run_CALANGO(defs, cores = 2)

## End(Not run)
```

Index