Package ‘gCMAPWeb’

October 16, 2019

Type Package

Title A web interface for gene-set enrichment analyses

Version 1.24.0

Date 2017-01-19

Author Thomas Sandmann

Maintainer Thomas Sandmann <`sandmann.t@gmail.com`>

Description The gCMAPWeb R package provides a graphical user interface for the gCMAP package. gCMAPWeb uses the Rook package and can be used either on a local machine, leveraging R's internal web server, or run on a dedicated rApache web server installation. gCMAPWeb allows users to search their own data sources and instructions to generate reference datasets from public repositories are included with the package. The package supports three common types of analyses, specifically queries with 1. one or two sets of query gene identifiers, whose members are expected to show changes in gene expression in a consistent direction. For example, an up-regulated gene set might contain genes activated by a transcription factor, a down-regulated geneset targets repressed by the same factor. 2. a single set of query gene identifiers, whose members are expected to show divergent differential expression (non-directional query). For example, members of a particular signaling pathway, some of which may be up- some down-regulated in response to a stimulus. 3. a query with the complete results of a differential expression profiling experiment. For example, gene identifiers and z-scores from a previous perturbation experiment. gCMAPWeb accepts three types of identifiers: EntreIds, gene Symbols and microarray probe ids and can be configured to work with any species supported by Bioconductor. For each query submission, significantly similar reference datasets will be identified and reported in graphical and tabular form.

Note gCMAPWeb is not compatible with the RStudio IDE, but can be used from the command line or via R's standard GUI.

License Artistic-2.0

Depends Biobase, gCMAP (>= 1.3.0), methods, R (>= 3.4), Rook

Imports brew, BiocGenerics, annotate, AnnotationDbi, graphics, grDevices, GSEABase, hwriter, parallel, stats, utils, yaml

Suggests affy, ArrayExpress, hgfocus.db, hgu133a.db, mgug4104a.db, org.Hs.eg.db, org.Mm.eg.db, RUnit

Enhances bigmemory, bigmemoryExtras

biocViews GUI, GeneSetEnrichment, Visualization, GeneExpression, Transcription, Microarray, DifferentialExpression

LazyLoad yes

Collate 'gCMAPWeb-package.R' 'AllGenerics.R' 'cmapRun-methods.R' 'cmap.R' 'cmapAnalysis.R' 'cmapHeatmap.R' 'createObject.R' 'example_objects.R' 'file_io.R' 'html_elements.R' 'plotting_functions.R' 'reporting.R'

RoxygenNote 5.0.1

OS_type unix

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

git_branch RELEASE_3_9

git_last_commit 1241acf

git_last_commit_date 2019-05-02

Date/Publication 2019-10-15

R topics documented:

`gCMAPWeb-package` *A web-interface for gene-set enrichment analysis*

Description

A web-interface for gene-set enrichment analysis

Details

This package provides a stand-alone web interface for gene-set enrichment analysis. Run either through R's internal Rhttpd server of deployed through an Apache2 webserver, gCMAPWeb allows users to search reference datasets stored in `NChannelSet` or `CMAPCollection` objects via three different methods:

- Directional queries performed by calculating the parametric JG score from the reference datasets
- Non-directional queries performed via Fisher's exact test

- Profile queires performed by calculating the parametric JG score from the query data

A web interace is populated based on a configuration file in YAML format and additional information from the reference dataset objects (if available). For each query, a report is generated in html format, including graphs and tabular output. Extensive customization of the graphical user interface can easily be performed through global options.

Author(s)

Thomas Sandmann

References

Jiang Z, Gentleman R., Extensions to gene set enrichment. Bioinformatics. 2007 Feb 1;23(3):306-13 `http://www.ncbi.nlm.nih.gov/pubmed/17127676`

Examples

```
example( gCMAPWeb )
```

`addLinks` *Adding links to columns of a data.frame*

Description

This function identifies column pairs from a data.frame that differ only in a string suffix (default: .url). The column with the suffix is used to add http href tags to the other column and is then removed.

Usage

```
addLinks(df, pattern = ".url$", swap.colnames = getOption("swap.colnames",
  default = list(padj = "FDR", nFound = "Genes")))
```

Arguments

`df`	data.frame, must contain both target and url columns
`pattern`	character, the suffix linking target and url columns. Default=c('.url')
`swap.colnames`	list, containing alternative names for CMAPResults columns.

Details

To annotate columns with links to other urls, the urls must be included in the data.frame in a separate column matching the target column name but carrying the additional suffix in the column name. For example, to add links to the 'id' column, an additional id.url column can be used. Multiple columns can be processed simultaneously, if each of them has a matching url column.

Value

Data frame with href html tags in the target column(s)

Author(s)

Thomas Sandmann

`add_geneScores_to_df` *CMAPResults data.frame with gene-level scores*

Description

This function converts a CMAPResults object into a data.frame and adds the gene.level scores as an additional column

Usage

```
add_geneScores_to_df(cmap.result)
```

Arguments

`cmap.result`	CMAPResults object

Value

A data.frame with gene-level scores in column 'Scores'

Author(s)

Thomas Sandmann

`body_first` *Definition of html navigation bar elements*

Description

Shared html code, constituting the first part of the body section

Usage

```
body_first(url.base = NULL)
```

Arguments

`url.base`	character, path to htdocs directory (optional)

Value

Html code as a character string

Author(s)

Thomas Sandmann

`cmap-data` *Example reference datasets*

Description

Five example reference datasets containing simulated data to demonstrate the functionality of the gCMAPWeb package. These objects were generated with the `exampleCMAP` function using different seeds.

Format

cmap1 to 4 are `NChannelSet` objects each with 1000 simulated z-scores for 10 samples. cmap5 is a `CMAPCollection` with membership information about 1000 genes and 10 gene sets.

Details

cmap1, cmap2, cmap3 and cmap5 contain data for 1000 human Entrez identifiers. cmap1, cmap2 and cmap3 are `NChannelSet` objects with one channel, the 'z' AssayDataElement. cmap4 is an `NChannelSet` with data for 1000 mouse Entrez identifiers in the 'z' and 'log_fc' channels. cmap5 is a codeCMAPCollection.

`cmapAnalysis` *Function controlling the gCMAP analysis workflow*

Description

This function controls the gCMAPWeb analysis workflow

Usage

```
cmapAnalysis(req, conf_data, reference.cmaps, element = getOption("element",
  default = "z"), save.intermediates = getOption("save.intermediates", default
  = FALSE))
```

Arguments

`req`	a Rook Request object
`conf_data`	list, the configuration data as returned by the read_config_file function
`reference.cmaps`	list, containing the reference database eSet objects
`element`	character, identifying the elementName of the channel extracted from NChannelSet objects
`save.intermediates`	logical, if TRUE rdata files of intermediate results will be stored in the temporary directory for debugging

Value

a list with three elements: conversion, reports and tmp_filename

Author(s)

Thomas Sandmann

`cmapHeatmap` *Function to create an annotated heatmap of gene scores*

Description

This function takes a numerical matrix (with samples in columns and genes in rows), preprocesses the data (if desired), determines the optimal height for the heatmap and calls the .ImagePlot function to create the final heatmap figure.

Usage

```
cmapHeatmap(x, reference.name, col.anno = NULL, row.anno = NULL,
  file.name = "heatmap", url.base = NULL, main = "Query gene scores",
  ColorRamp = colorRampPalette(c("#044381FF", "grey95", "grey95",
  "firebrick"))(100), col.col = c(down = "black", up = "grey"),
  row.col = c(correlated = "#1B9E77", anticorrelated = "#044381FF", over =
  "#1B9E77", under = "#044381FF"), order.by.score = TRUE,
  cluster.rows = TRUE, score.cap = c(-5, 5),
  ylab = "Significant datasets")
```

Arguments

`x`	numerical matrix with samples in rows and genes in columns.
`reference.name`	character, names of the reference cmap, used to construct the html image reference
`col.anno`	character vector with column annotations to be displayed as (horizontal) annotation bar above the heatmap. If not NULL, must contain one elment for each column of 'x'.
`row.anno`	character vector with row annotations to be displayed as (vertical) annotation to the right of the heatmap. If not NULL, must contain one element for each row of 'x'.
`file.name`	character, the path and filename (without suffix) to save the png file to
`url.base`	character, prefix for the html image reference
`main`	Character, main title of the plot
`ColorRamp`	vector of colors used for the heatmap, e.g. generated by a call to colorRampPalette
`col.col`	named vector with a color for each level of col.anno (e.g. c(up="firebrick", down="blue"))
`row.col`	named vector with a color for each level of row.anno (e.g. c(correlated="firebrick",anticorrelated="#0
`order.by.score`	logical, should gene scores be reordered independently for each sample ?
`cluster.rows`	logical, perform hierarchical clustering on significant gene sets ?
`score.cap`	numerical vector of length two, specifying the limits of the color scale. Scores > max(score.cap) or < min(score.cap) will be set to score.cap. Default: c(-5,5)
`ylab`	character, y-axis label

Value

list with two elements, 1. image.html: a character string with html code with the image tag refering to the output png 2. row.order: integer vector with thw row.order obtained after hierarchical clustering or NULL if no clustering was performed

Author(s)

Thomas Sandmann

`cmapRun` *cmapRun*

Description

This S4 method dispatches a gCMAP analysis and selects a suitable gene-set-enrichment analysis method based on the class of its arguments.

Usage

```
cmapRun(user.input, cmap, ...)

## S4 method for signature 'CMAPCollection,CMAPCollection'
cmapRun(user.input, cmap)

## S4 method for signature 'GeneSet,CMAPCollection'
cmapRun(user.input, cmap, ...)

## S4 method for signature 'GeneSet,eSet'
cmapRun(user.input, cmap, ...)

## S4 method for signature 'CMAPCollection,eSet'
cmapRun(user.input, cmap,
  lower = getOption("lower.threshold", default = -3),
  higher = getOption("higher.threshold", default = 3),
  element = getOption("element", default = "z"),
  min.set.size = getOption("min.set.size", default = 5), keep.scores = TRUE)

## S4 method for signature 'SignedGeneSet,eSet'
cmapRun(user.input, cmap,
  element = getOption("element", default = "z"))

## S4 method for signature 'eSet,CMAPCollection'
cmapRun(user.input, cmap,
  element = getOption("element", default = "z"))

## S4 method for signature 'eSet,eSet'
cmapRun(user.input, cmap,
  lower = getOption("lower.threshold", default = -3),
  higher = getOption("higher.threshold", default = 3),
  element = getOption("element", default = "z"))
```

Arguments

`user.input`	An object of class GeneSet, SignedGeneSet, eSet or CMAPCollection with the user-provided query.
`cmap`	The reference database, a CMAPCollection or eSet object.
`...`	Arguments specific to individual methods
`lower`	Numeric, when reference databases are thresholded, genes with scores less than 'lower' are considered down-regulated. Default: getOption("lower.threshold", default="-3")
`higher`	Numeric, when reference databases are thresholded, genes with scores larger than 'higher' are considered up-regulated. Default: getOption("higher.threshold", default="3")
`element`	Character,specifying which channel / assayDataElement of the reference database to query. Default: getOption("element", default="z")
`min.set.size`	integer, gene sets with less than min.set.size members will be dropped from CMAPCollections. Default: getOption("min.set.size", default=5)
`keep.scores`	Scalar boolean, include scores in the result object ?

Details

The cmapRun methods choose one of three approaches for gene-set-enrichment analyses, depending on the nature of the user-provided query and the data available in the reference databases.

1. for non-directional gene set queries (e.g. a list of gene identifiers) a Fisher exact test is performed (with a call to the fisher_score method from the gCMAP package) to evaluate the overlap with significantly up- or down-regulated gene sets from the reference database. If the reference object is an NChannelSet, it is thresholded on the fly to obtain a set of up- or down-regulated genes. If the reference database is a CMAPCollection, the gene sets are used directly. Please note that the Fisher exact test does not take into account information about the directionality of gene regulation (potentially) available in the reference database.

2. for directional gene set queries (e.g. two separate lists of up- and down-regulated genes of interest) the gene expression scores available in the reference database are summarized as the JG score by running the gsealm_jg_score method from the gCMAP package.

3. for queries with complete differential gene expression profiles, directional gene sets are induced from the reference database. As above, the JG score is used to summarize the expression changes for each gene set, but this time the sets are derived from the database, while the scores are provided by the user.

This method is called by the cmapAnalysis function of the gCMAPWeb package.

Please note: Most of the parameters of the cmapRun methods can be set as global options.

Value

A CMAPResults object

See Also

`gsealm_jg_score` and `fisher_score`

`conversion_html` *Identifier conversion html output*

Description

This function formats a data.frame with the results of the gene identifier conversion to html code, including an interactive dataTable. The html page is saved to disk and the path to the output file is returned as a character string.

Usage

```
conversion_html(df, result.dir, file.name, tmp_filename, url.base = NULL,
  message = NULL)
```

Arguments

`df`	data.frame with old / new gene identifiers
`result.dir`	character, path to result directory
`file.name`	character, output filename
`tmp_filename`	character, name of the session-specific subdirectory
`url.base`	character, path to htdocs directory (optional)
`message`	character, optional message to display above the result table

Value

character, name of the output html file

Author(s)

Thomas Sandmann

`convert_gene_identifiers` *Entrez mapper*

Description

This function maps the submitted gene identifiers to Entrez identifiers and returns a translation table with the original and mapped ids

Usage

```
convert_gene_identifiers(gene.ids, query.IdType, species)
```

Arguments

`gene.ids`	Character vector with gene ids
`query.IdType`	A GeneIdentifierType object
`species`	String identifying the species, will be inserted into a new EntrezIdentifier object and returned with the translated results

Value

A list with the following elements

- gene.ids, a character vector of unique EntrezIds
- query.IdType, an EntrezIdentifier object with the species
- conversion, a list containing the full translation table and a summary message
- original.ids, a character vector of the submitted ids corresponding to the returned EntrezIds

Author(s)

Thomas Sandmann

`create_figure_legend` *This function generates the figure legend, with a separate dom identifier for each output panel*

Description

This function generates the figure legend, with a separate dom identifier for each output panel

Usage

```
create_figure_legend(reference.name)
```

Arguments

`reference.name`	character, name of the reference database used

Value

character string with the html code element required to insert the figure legend into the html report

Author(s)

Thomas Sandmann

`create_GeneIdentifierType` *GeneIdentifierType creator*

Description

This function extracts the species or platform information from the POST request and constructs GeneIdentifierType objects

Usage

```
create_GeneIdentifierType(post, conf_data)
```

Arguments

`post`	list, POST component of the Rook request
`conf_data`	list, the configuration data as returned by the read_config_file function

Value

GeneIdentifierType object

Author(s)

Thomas Sandmann

`create_GeneSet` *GeneSet creator*

Description

This function creates a GeneSet object from user-specified identifiers

Usage

```
create_GeneSet(post, conf_data)
```

Arguments

`post`	list, POST component of the Rook request
`conf_data`	list, the configuration data as returned by the read_config_file function

Value

GeneSet object

Author(s)

Thomas Sandmann

`create_gene_report` *This function compiles a gene-level report from a CMAPResults object.*

Description

Usually called by the generate_report function

Usage

```
create_gene_report(cmap.result, query, result.dir, url.base, reference.cmap,
  figure.dir, element = getOption("element", default = "z"),
  annotation.db = "org.Hs.eg.db",
  gene.level.plot = getOption("gene.level.plot", default = TRUE))
```

Arguments

`cmap.result`	a CMAPResults object
`query`	a GeneSet, SignedGeneSet or ExpressionSet object used as query for the gene-set-enrichment analysis giving rise to 'cmap.result'
`result.dir`	character, path to gene_results folder in the per-session output directory
`url.base`	character, the name of the reference database
`reference.cmap`	eSet, the reference database used for the gCMAP analysis
`figure.dir`	character, path to figure folder in the per-session output directory
`element`	character, the assayDataElementName to extract from NChannelSet objects
`annotation.db`	character, the name of the annotation package used to lookup gene identifiers
`gene.level.plot,`	logical: should gene-level plots be included in the report ?

Value

character string with the relative url to the report html, which is directly written to disk.

Author(s)

Thomas Sandmann

`create_gene_table_legend` *This function reads an eSet object and returns the html code to display a legend for all columns that have more descriptive annotations in the varMetadata slot.*

Description

This function reads an eSet object and returns the html code to display a legend for all columns that have more descriptive annotations in the varMetadata slot.

Usage

```
create_gene_table_legend(col.names, eset)
```

Arguments

`col.names`	character, a vector with column names that are always included in the legend
`eset`	eSet object

Value

character string with the html code element required to insert the legend into the html report

Author(s)

Thomas Sandmann

`create_heatmap_legend` *Legend for overview heatmap*

Description

Content for the legend of the overview heatmap plot displayed on the main results page for directional and non-directional queries

Usage

```
create_heatmap_legend(text)
```

Arguments

`text`	Character, text for the legend of the density plot the main report page. Can be set as the global variabel "gene.set.legend".

Value

Character

Author(s)

Thomas Sandmann

`create_legend` *This function reads a CMAPResults object and returns the html code to display a legend for all columns that have more descriptive annotations in the varMetadata slot. A standard set of columns is selected for display in the legend through the 'keep' parameter. In addition, all columns with annotations in the labelDescription field of the varMetadata slot are included if the field content differs from the column name.*

Description

This function reads a CMAPResults object and returns the html code to display a legend for all columns that have more descriptive annotations in the varMetadata slot. A standard set of columns is selected for display in the legend through the 'keep' parameter. In addition, all columns with annotations in the labelDescription field of the varMetadata slot are included if the field content differs from the column name.

Usage

```
create_legend(res, reference.name, keep = c("set", "trend", "FDR", "effect",
  "nSet", "Genes"), swap.colnames = getOption("swap.colnames", default =
  list(padj = "FDR", nFound = "Genes")))
```

Arguments

`res`	CMAPResults object
`reference.name`	character, ame of the reference database used
`keep`	character, a vector with column names that are always included in the legend
`swap.colnames`	list, containing alternative names for CMAPResults columns.

Value

character string with the html code element required to insert the legend into the html report

Author(s)

Thomas Sandmann

`create_overview_legend` *Legend for overview density plot*

Description

Content for the legend of the overview density plot displayed on every main results page

Usage

```
create_overview_legend(text)
```

Arguments

`text`	Character, text for the legend of the density plot the main report page. Can be set as the global variabel "gene.set.legend".

Value

Character

Author(s)

Thomas Sandmann

`create_overview_plot` *Overview plot*

Description

This function creates an overview density plot summarizing the similarity scores obtained for all instances in the reference database.

Usage

```
create_overview_plot(effect.sample, effect.population, file.name,
  reference.name, main = "Distribution of similarity scores",
  xlab = "Effect size", col.set = "black", col.up = "#1B9E77",
  col.down = "blue", strip.cutoffs = c(-3, 3), strip.bounds = c(-6, 6),
  strip.col = c("#1B9E77", "white", "blue"), url.base = NULL,
  up.label = "Correlated", down.label = "Anti-correlated")
```

Arguments

`effect.sample`	numeric, the scores for all gene sets in the reference database deemed significantly similar
`effect.population`	numeric, the scores for all gene sites in the reference database
`file.name`	character, path and name of the output file
`reference.name`	name of the reference dabatase
`main`	main title of the plot
`xlab`	x-axis label
`col.set`	color used for the significant samples in the density plot
`col.up`	color used to indicate significantly correlated / enriched gene sets in the rug plot
`col.down`	color used to indicate significantly anti-correlated / depleted gene sets in the rug plot
`strip.cutoffs`	numeric vector, indicating the area of the heat-strip set to the intermediat strip.col. Default=c(-3,3)
`strip.bounds`	numeric vector, indicating the cutoffs above / below which the color of the heat-strip is set to maximum. Default=c(-6,6)
`strip.col`	character vector with three elements, indicating high, neutral and low scores, respectively. Default=c("#1B9E77", "white", "blue")
`url.base`	path the location of the output files relative to the gCMAPWeb home directory
`up.label`	character, legend label for positive scores
`down.label`	character, legend label for negative scores

Value

None. Pdf and png versions of the overview file are written to file.

Author(s)

Thomas Sandmann

`create_profile_ExpressionSet` *ExpressionSet creator*

Description

This function creates an ExpressionSet object from user-specified identifiers and scores.

Usage

```
create_profile_ExpressionSet(post, conf_data)
```

Arguments

`post`	list, POST component of the Rook request
`conf_data`	list, the configuration data as returned by the read_config_file function

Value

ExpressionSet object

Author(s)

Thomas Sandmann

`create_query_objects` *Query object creator*

Description

This function generates an appropriate R data object from the user query

Usage

```
create_query_objects(post, conf_data)
```

Arguments

`post`	list, POST component of the Rook request
`conf_data`	list, the configuration data as returned by the read_config_file function

Value

one of GeneSet, SignedGeneSet or ExpressionSet

Author(s)

Thomas Sandmann

`create_SignedGeneSet` *SignedGeneSet creator*

Description

This function creates a SignedGeneSet object from user-specified identifiers.

Usage

```
create_SignedGeneSet(post, conf_data)
```

Arguments

`post`	list, POST component of the Rook request
`conf_data`	list, the configuration data as returned by the read_config_file function

Value

SignedGeneSet object

Author(s)

Thomas Sandmann

`create_tab` *Create tab-delimited output*

Description

This function exports results in tab-delimited format

Usage

```
create_tab(df, result.dir, url.base = NULL, file.name)
```

Arguments

`df`	data.frame
`result.dir`	character, path to the output directory
`url.base`	character, the name of the reference database
`file.name`	character, name of the report file

Value

character string with the html code pointing to the download URL

Author(s)

Thomas Sandmann

`create_zip` *Export gCMAPWeb report as zip archive*

Description

This function exports the complete report in a zip file

Usage

```
create_zip(tmp_filename, out.dir)
```

Arguments

`tmp_filename`	character, name of the per-session result directory to be archived
`out.dir`	character, output directory to save zip archive into

Value

Nothing, the zip archive is written to disk.

Author(s)

Thomas Sandmann

`entrez_GeneSets` *GeneSet Entrez mapper*

Description

This function maps the gene identifiers of a GeneSet object to Entrez identifiers and returns an error if none of them could be found.

Usage

```
entrez_GeneSets(gs)
```

Arguments

`gs`	GeneSet or SignedGeneSet

Details

gCMAPWeb uses this function to ensure that all submitted / retrieved Entrez Ids are valid.

Value

GeneSet or SignedGeneSet

Author(s)

Thomas Sandmann

`exampleCMAP` *Generate example NChannelSets*

Description

This function generates NChannelSets from randomly selected gene idenfiers and scores. Scores of the first (simulated) sample will be shifted up by the amount specified by the 'add' parameter for the first 50 genes and down by the same amount for genes 51-100.

Usage

```
exampleCMAP(universe = "org.Hs.eg.db", idType = "ENTREZID", rows = 1000,
  cols = 20, seed = 123, add = 3)
```

Arguments

`universe`	character, the name of an available annotation package. Default=org.Hs.eg.db
`idType`	character, a valid identifier type supported by the selected annotation package. Default=ENTREZID
`rows`	integer, the number of genes to sample. Default=1000
`cols`	integer, the number of samples to generate. Default=20
`seed`	numeric, seed for the random number generator. Default=123
`add`	numeric, score shift for scores in the first sample column. Scores for genes 1-50 will be shifted up, those for genes 51-100 down.

Details

This function is mainly intented for testing purposes.

Value

NChannelSet with simulated scores.

Author(s)

Thomas Sandmann

Examples

```
## generate example NChannelSet with 100 features and 10 columns
## for human EntrezIds
if( require( "org.Hs.eg.db", character.only=TRUE)){
  e <- exampleCMAP( universe="org.Hs.eg.db", rows=100, cols=10)
  dim( e )
  head( featureNames( e ))
  assayDataElementNames( e )
  ## or gene symbols
  s <- exampleCMAP( universe="org.Hs.eg.db", idType="SYMBOL")
  head( featureNames( s ))
  }
```

`examplePost` *Simulate POST request*

Description

This function returns a simulated POST request, similar to those generated from user-input into the web interface.

Usage

```
examplePost(cmap, inputType = "non-directional", species = "human",
  array.platform = NULL, idType = "entrez", rows = 1000, add = 3,
  seed = 123)
```

Arguments

`cmap`	NChannelSet object, e.g. generated by the exampleCMAP function
`inputType`	character, one of directional, non-directional, profile
`species`	character, should match a species supported by the loaded gCMAPWeb configuration file
`array.platform`	character, should match an array platform supported by the loaded gCMAPWeb configuration file
`idType`	character, one of entrez, symbol or probe
`rows`	integer, number of genes to include in a 'profile' query.
`add`	numeric, score shift for scores in the first sample column. Scores for genes 1-50 will be shifted up, those for genes 51-100 down. Only used to simulate 'profile' queries.
`seed`	numeric, seed for the random number generator. Default=123

Details

This function is mainly intented for testing purposes.

Value

list, containg the same elements as a gCMAPWeb POST request

Author(s)

Thomas Sandmann

Examples

```
if( require( "org.Hs.eg.db", character.only=TRUE)){
  ## generate an example NChannelSet for human EntrezIds
  e <- exampleCMAP( universe="org.Hs.eg", rows=1000, cols=10)
  ## generate a matching example Post, simulating a non-directional query
  examplePost( cmap=e )
  ## or a directional query
  examplePost( cmap=e, inputType="directional")
}
```

`gCMAPWeb` *Start local gCMAPWeb instance*

Description

This function starts a gCMAPWeb instance on the local machine using R's internal web server.

Usage

```
gCMAPWeb(config.file.path = system.file("config", "config.yml", package =
  "gCMAPWeb"), url.root = system.file("htdocs", package = "gCMAPWeb"))
```

Arguments

`config.file.path`	character, path to a gCMAPWeb configuration file in yaml format
`url.root`	character, path to the htdocs folder

Value

an Rhttpd class object

Author(s)

Thomas Sandmann

Examples

```
if (interactive()) {
  ## start a gCMAPWeb instance with the example data and
  ## configuration provided in the package
  gCMAPWeb()
  ## same as above, explicitely specifying the location
  ## of the configuration file
  gCMAPWeb(
    config.file.path = system.file("config", "config.yml",
                                    package = "gCMAPWeb" )
  )
}
```

`generate_report` *gCMAPWeb reporting function*

Description

This function reveives the results of a gene-set enrichment analysis and the original reference databases to generate plots and html code elements for the final report.

Usage

```
generate_report(cmap.result, reference, reference.name, annotation.db, element,
  query, tmp_filename, title = "", max.results = getOption("max.results",
  default = 50), min.found = getOption("min.found", default = 1),
  max.padj = getOption("max.padj", default = 0.1),
  gene.level.report = getOption("gene.level.report", default = TRUE),
  gene.level.plot = getOption("gene.level.plot", default = TRUE),
  reportDirectory = tempdir(), excluded.cols = getOption("excluded.cols",
  default = c("geneScores", "signed", "pval", "UID", "z.shift", "log_fc.shift",
  "mod_fc.shift")), swap.colnames = getOption("swap.colnames", default =
  list(padj = "FDR", nFound = "Genes")))
```

Arguments

`cmap.result`	CMAPResults object
`reference`	eSet object, the full reference database, typically an NChannelSet or CMAPCollection object
`reference.name`	character, the name of the reference database
`annotation.db`	character, the name of the annotation package used to lookup gene identifiers
`element`	character, the assayDataElementName to extract from NChannelSet objects
`query`	a GeneSet, SignedGeneSet or ExpressionSet object used as query for the gene-set-enrichment analysis giving rise to 'cmap.result'
`tmp_filename`	character, the name of the temporary output folder
`title`	character, title of the report
`max.results`	integer, maximum number of results to return
`min.found`	integer, minimum number of gene set members that need to overlap with the user query. Gene sets with matching members less than min.found will not be returned as results.
`max.padj`	numeric, maximum adjusted p-value for a gene set to be considered significantly similar. Results with adjusted p-values larger than max.padj will not be returned.
`gene.level.report`	logical, create gene-level reports ?
`gene.level.plot`	logical, create gene-level plots ?
`reportDirectory`	character, path to temporary output directory for this session. Default=tempdir()
`excluded.cols`	character vector listing columns in the CMAPResults objects that should NOT be included in the report
`swap.colnames`	list, containing alternative names for CMAPResults columns.

Details

This function is called by the cmapAnalysis function

Value

a list of character strings, either containing html code snippets or paths to result files. These elements are used to brew the final result html page.

Author(s)

Thomas Sandmann

`gene_density_chart_legend`

Html code for the legend of pie the pie chart of the gene-level report

---

Description

Html code for the legend of pie the pie chart of the gene-level report

Usage

```
gene_density_chart_legend(text)
```

Arguments

`text`	Character, text for the legend of the density plot on gene-level reports for non-directional and directional queries. Can be set as the global variabel "gene.density.legend".

Value

Character string with html code

Author(s)

Thomas Sandmann

`gene_density_profile_legend`

Html code for the legend of the density charts of the gene-level reports from Profile queries

---

Description

Html code for the legend of the density charts of the gene-level reports from Profile queries

Usage

```
gene_density_profile_legend(text)
```

Arguments

`text`	Character, text for the legend of the density plot on gene-level reports for profile queries. Can be set as the global variabel "gene.profile.legend".

Value

Character string with html code

Author(s)

Thomas Sandmann

`gene_pie_chart_legend` *Html code for the legend of the pie chart of the gene-level report*

Description

Html code for the legend of the pie chart of the gene-level report

Usage

```
gene_pie_chart_legend(text)
```

Arguments

`text`	Character, text for the legend of the pie chart on gene-level reports for non-directional queries. Can be set as the global variabel "gene.pie.legend".

Value

Character string with html code

Author(s)

Thomas Sandmann

`hide_species_js` *Javascript code to hide html elements*

Description

Javascript code to hide html elements for all but the first species listed in the configuration file

Usage

```
hide_species_js(conf_data)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function

Value

javascript call as character string

Author(s)

Thomas Sandmann

`html_body_last` *Html footer*

Description

Closing html code, including references to jquery, bootstrap and dataTables javascript libraries

Usage

```
html_body_last(url.base = NULL)
```

Arguments

`url.base`	character, path to htdocs directory (optional)

Value

Html code as character string

Author(s)

Thomas Sandmann

`html_header` *Html header and bootstrap css stylesheet*

Description

Html header including link to the bootstrap cerulean css stylesheet

Usage

```
html_header(url.base = NULL)
```

Arguments

`url.base`	character, path to htdocs directory (optional)

Value

Html code as a character string

Author(s)

Thomas Sandmann

`html_table` *Format data.frame as html table*

Description

Function to format a data.frame as an html table

Usage

```
html_table(df, table_id)
```

Arguments

`df`	A data frame
`table_id`	Character, specifying the id tag for the html table

Details

This function uses the thead and tbody tags required for the dataTables javascript to create interactive html tables.

Value

Html code as a character string.

Author(s)

Thomas Sandmann

`identifier_radio_html` *Html code for identifier type radio button selectors*

Description

This function generates the html code required to display radio buttons for the supported identifier types specified in the gCMAPWeb configuration file

Usage

```
identifier_radio_html(conf_data, single = FALSE)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function
`single,`	logical, use singular nouns for radiobutton labels ?

Value

Html code as character string.

Author(s)

Thomas Sandmann

`index_message` *Html code for text of the additional message on the index page (warning box)*

Description

Html code for text of the additional message on the index page (warning box)

Usage

```
index_message(text)
```

Arguments

`text`	Character, text for the additional message on the index page (warning box)

Value

text Character string with html code

Author(s)

Thomas Sandmann

`index_quote` *Html code for quote on the index page*

Description

Html code for quote on the index page

Usage

```
index_quote(text)
```

Arguments

`text,`	Character, the content of the blockquote field on the index.rhtml page. Can be set as the global variabel "index.quote".

Value

Character string with html code

Author(s)

Thomas Sandmann

`index_subtitle` *Html code for subtitle of the hero unit of the index.rhtml page*

Description

Html code for subtitle of the hero unit of the index.rhtml page

Usage

```
index_subtitle(text)
```

Arguments

`text`	Character, subtitle of the hero unit on the index.rhtml page. Can be set as the global variabel "index.sub".

Value

Character string with html code

Author(s)

Thomas Sandmann

`index_text` *Html code for text of the hero unit of the index.rhtml page*

Description

Html code for text of the hero unit of the index.rhtml page

Usage

```
index_text(text)
```

Arguments

`text`	Character, text of the hero unit on the index.rhtml page. Can be set as the global variabel "index.text".

Value

Character string with html code

Author(s)

Thomas Sandmann

`index_title` *Html code for title of the hero unit of the index.rhtml page*

Description

Html code for title of the hero unit of the index.rhtml page

Usage

```
index_title(text = getOption("index.main", default = "gConnectivity Map"))
```

Arguments

`text`	Character, main title of the hero unit on the index.rhtml page. Can be set as the global variabel "index.main".

Value

Character string with html code

Author(s)

Thomas Sandmann

`inputType_buttons` *Html code for the query type selection buttons on the gCMAPWeb index page*

Description

This function generates the html code required to generate the query type selection buttons on the main index.rhtml page

Usage

```
inputType_buttons(conf_data)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function

Value

Html code as character string.

Author(s)

Thomas Sandmann

`load_cmaps` *Loading reference datasets*

Description

This function connects to / loads all reference datasets and returns them in a list

Usage

```
load_cmaps(conf_data, min.set.size = getOption("min.set.size", default = 5))
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function
`min.set.size`	integer, gene sets with less than min.set.size members will be dropped from CMAPCollections

Value

list of eSet objects

Author(s)

Thomas Sandmann

Examples

```
library(yaml)
## read the example configuration file
conf_data <- yaml.load_file( system.file("config", "config.yml",
             package="gCMAPWeb") )
## load the example reference NChannelSet objects specified in
## the configuration file
ref.cmaps <- load_cmaps( conf_data )
class( ref.cmaps )
names( ref.cmaps )
ref.cmaps[[1]]
```

`parse_file_input` *Uploaded file parser*

Description

This functions parses the user-uploaded files

Usage

```
parse_file_input(temp.file, regexp = "[+,;\t ]+", n.score.col = 1)
```

Arguments

`temp.file`	character, path to temporary file (usually part of the POST request)
`regexp`	character, a regular expression matching all separators used to separate gene identifiers
`n.score.col`	integer, for profile uploads n.score.col specifies the number of data columns (usually 1)

Value

list of character vectors, one element for each row of temp.file

Author(s)

Thomas Sandmann

`parse_request` *POST request parser*

Description

This function parses the a POST request and decodes the user-provided information

Usage

```
parse_request(request, conf_data,
  cmaps.concatenated.by = getOption("cmaps.concatenated.by", default = ","))
```

Arguments

`request`	Request object as defined by the Rook package
`conf_data`	list, the configuration data as returned by the read_config_file function
`cmaps.concatenated.by`	character, specifying the separator (e.g. semicolon, comma) used to concatenate multiple requested reference databases

Value

list, all elements from POST request, decoded if necessary

Author(s)

Thomas Sandmann

`parse_textarea_input` *Textarea parser*

Description

This function parses user-specified data pasted into the textarea fields of the input form

Usage

```
parse_textarea_input(textarea.input, regexp = "[+,;\t ]+")
```

Arguments

`textarea.input`	character, the user-input extracted from the POST request
`regexp`	character, a regular expression matching all separators used to separate gene identifiers

Value

list of character vectors, one element for each original row of textarea.input

Author(s)

Thomas Sandmann

`platform_radio_html` *Html code for platform radio button selectors*

Description

This function generates the html code required to display radio buttons for all array platforms specified in the gCMAPWeb configuration file

Usage

```
platform_radio_html(conf_data)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function

Value

Html code as character string.

Author(s)

Thomas Sandmann

`process_score_input` *Score parser*

Description

This function reformats score input uploaded or pasted by the user

Usage

```
process_score_input(query_data, n.score.col = 1)
```

Arguments

`query_data`	list of row-vectors, as generated by parse_file_input function
`n.score.col`	integer, specifies the number of data columns (usually 1)

Value

matrix of expression scores

Author(s)

Thomas Sandmann

`profile_input_example` *Html code for generating an example query button on the profile submission page*

Description

Content can be provided / modified through the global parameters profile.input.example.popover (text for popover help) profile.input.example (gene identifiers and scores)

Usage

```
profile_input_example()
```

Value

Character string with html code

Author(s)

Thomas Sandmann

`read_config_file` *Parser for gCMAPWeb configuration file*

Description

This function parses the gCMAPWeb configuration file (in yaml format) and returns the information as a list

Usage

```
read_config_file(config.file.path = system.file("config", "config.yml",
  package = "gCMAPWeb"))
```

Arguments

`config.file.path`	character, full path to gCMAPWeb configuration file

Value

a nested list

Author(s)

Thomas Sandmann

`reference_radio_html` *Html code for reference cmap radio button selectors*

Description

This function generates the html code required to display radio buttons for the reference datasets specified in the gCMAPWeb configuration file

Usage

```
reference_radio_html(conf_data, reference.cmaps)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function
`reference.cmaps`	list containing all reference cmaps as eSet-like objects (e.g. NChannelSet, CMAP-Collection)

Value

Html code as character string.

Author(s)

Thomas Sandmann

`retrieve_annotation` *Gene symbol and name lookup*

Description

This function retrieves the official gene symbol and name for Entrez identifiers

Usage

```
retrieve_annotation(entrez, annotation.db)
```

Arguments

`entrez`	character, Entrez identifier to look up
`annotation.db`	character, name of the annotation package to use for lookup

Value

data.frame with entrez, symbol and name columns

Author(s)

Thomas Sandmann

`signed_input_example` *Html code for generating and example query button on the signed gene set submission page*

Description

Content can be provided / modified through the global parameters signed.input.example.popover (text for popover help) signed.input.example.up (up-regulated gene identifiers) signed.input.example.down (down-regulated gene identifiers)

Usage

```
signed_input_example()
```

Value

Character string with html code

Author(s)

Thomas Sandmann

`single_input_example` *Html code for generating and example query button on the single gene query submission page*

Description

Content can be provided / modified through the global parameters single.gene.example.popover (text for popover help) single.gene.example (gene identifiers)

Usage

```
single_input_example()
```

Value

Character string with html code

Author(s)

Thomas Sandmann

`species_radio_html` *Html code for species radio button selectors*

Description

This function generates the html code required to display radio buttons for all species specified in the gCMAPWeb configuration file

Usage

```
species_radio_html(conf_data)
```

Arguments

`conf_data`	list, the configuration data as returned by the read_config_file function

Value

Html code as character string.

Author(s)

Thomas Sandmann

`stripPrefix` *Strip prefix from gene identifier names*

Description

This function strips a prefix from gene identifier strings

Usage

```
stripPrefix(ids, prefix = c("^GeneID:", "^GeneID"))
```

Arguments

`ids`	character, vector of gene identifiers
`prefix`	character, one or more strings to be removed

Value

character, gene identifiers without prefix

Author(s)

Thomas Sandmann

`unsigned_input_example` *Html code for generating and example query button on the unsigned gene set submission page*

Description

Content can be provided / modified through the global parameters unsigned.input.example.popover (text for popover help) unsigned.input.example (gene identifiers)

Usage

```
unsigned_input_example()
```

Value

Character string with html code

Author(s)

Thomas Sandmann

`validate_config_file` *Validation of gCMAPWeb configuration file*

Description

This function validates the content of the gCMAPWeb configuration file.

Usage

```
validate_config_file(config.file.path)
```

Arguments

`config.file.path`	full path to gCMAPWeb configuration file (in yaml format)

Details

Verifies that 1. all required fields are present 2. at least one species has been defined 3. all supported annotation packages are available 4. all cmaps have unique labels 5. each supported species has at least one associated reference dataset

Value

Information from a valid configuration file is returned as a nested list. Invalid entries in the config file with cause an error.

Author(s)

Thomas Sandmann

Examples

```
## read the example configuration yaml file without validation
library(yaml)
conf1 <- yaml.load_file( system.file("config", "config.yml",
            package="gCMAPWeb") )
## read the example configuration file and validate that
## all required information is provided and valid
conf2 <- validate_config_file( system.file("config", "config.yml",
            package="gCMAPWeb") )
identical( conf1, conf2)
```

`validate_request` *Html request validation*

Description

This function extracts the content from a POST request and validates its content

Usage

```
validate_request(req, conf_data,
  cmaps.concatenated.by = getOption("cmaps.concatenated.by", default = ","))
```

Arguments

`req`	Request object as defined by the Rook package
`conf_data`	list, the configuration data as returned by the read_config_file function
`cmaps.concatenated.by`	character, specifying the separator (e.g. semicolon, comma) used to concatenate multiple requested reference databases

Value

Logical, TRUE if all required fields could be validated, FALSE otherwise

Author(s)

Thomas Sandmann

Index