Package 'ENCODExplorer'

May 24, 2019

Name ENCODExplorer

Type Package

Title A compilation of ENCODE metadata

Version 2.10.0

Date 2015-02-25

Description This package allows user to quickly access ENCODE project files
metadata and give access to helper functions to query the ENCODE rest api,
download ENCODE datasets and save the database in SQLite format.

License Artistic-2.0

BugReports https://github.com/CharlesJB/ENCODExplorer/issues

VignetteBuilder knitr

Depends R (>= 3.6), shiny, DT, shinythemes

Imports tools, jsonlite, RCurl, tidyr, data.table, dplyr, stringr,
stringi, utils, AnnotationHub

Suggests RUnit,BiocGenerics,knitr, curl, httr

LazyData true

biocViews Infrastructure, DataImport

RoxygenNote 6.1.1

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

git_branch RELEASE_3_9

git_last_commit a360412

git_last_commit_date 2019-05-02

Date/Publication 2019-05-23

Author Charles Joly Beauparlant [aut, cre],
Audrey Lemacon [aut],
Eric Fournier [aut],
Louis Gendron [ctb],
Astrid-Louise Deschenes [ctb],
Arnaud Droit [aut]

Maintainer Charles Joly Beauparlant <charles.joly-beauparlant@crchul.ulaval.ca>

R topics documented:

---

clean_column   *Clean a single column of the data.frame*

Description

The input column can either be a data.frame, a vector of character, a vector of numeric or a list of one the previous type.

Usage

```
clean_column(column_name, table)
```

Arguments

column_name	The name of the column for the table that is been process.
table	The table produced by the extract_table function.

Details

This function will either remove columns that are not relevant and convert columns to a vector or data.frame.

Value

a data.frame corresponding to the cleaned version of the input data.frame.

| clean_table | *Clean a data.frame that was produced by ENCODE REST API* |

Description

data.frames produced when converting JSON to data.frame with the fromJSON function will sometime have columns that are lists and/or columns that are data.frames.

Usage

```
clean_table(table)
```

Arguments

table   The table produced by the extract_table function.

Details

This function will either remove columns that are not relevant and convert columns to a vector or data.frame.

Value

a data.frame corresponding to the cleaned version of the input data.frame.

| createDesign | *Create a design for the files associated with the result of a queryEncode, fuzzySearch research or a* data.table *from createDesign.* |

Description

Create a design for the files associated with the result of a queryEncode, fuzzySearch research or a data.table from createDesign.

Usage

```
createDesign(input = NULL, df = get_encode_df(), split = FALSE,
  fileFormat = "bam", dataset_type = "experiments", format = "long",
  output_type = "data.table", ID = c(1, 2))
```

Arguments

input   The data.table created by a queryEncode or searchEncode research, or a

df   The data.table used to extract the files link. Default :get_encode_df()

split   Allow to the function to return a list of data.table where each data.table contain the files for a single experiment Default: FALSE.

fileFormat   A string that correspond to the type of the files that need to be extracted. Default: bam

dataset_type	A string that correspong to the type of dataset that will be extrated. Default: experiments
format	The format (long or wide) to represent the data. The 'long' format will contain three columns (File, Experiment, Value). The 'wide' format organize the data as an array with the experiments as columns and files as rows. Default: long
output_type	The type of output of the function, can be data.table or a data.table Default: data.table
ID	A two element numeric vector, that first element is the value assign to replicate and the second is the value assign to control. Default: 1 and 2

Value

is a data.table with files for all the experiments or a list of data.table with all the file per experiment when the parameter split is set to TRUE

Examples

```
# You will need to replace get_encode_df_demo() with your own encode_df object,
# the get_encode_df() function or the get_encode_df_full() function.
fuzzy_result <- fuzzySearch(searchTerm = "brca",
database=get_encode_df_demo(), filterVector ="target")
design_result <- createDesign(input = fuzzy_result,df=get_encode_df_demo(),
fileFormat="fastq")
```

---

downloadEncode	*downloadEncode is used to download a serie of files or datasets using their accession.*

Description

downloadEncode is used to download a serie of files or datasets using their accession.

Usage

```
downloadEncode(file_acc = NULL, df = get_encode_df(), format = "all",
  dir = ".", force = TRUE)
```

Arguments

file_acc	A character of ENCODE file or experiment accessions. Can also be a data.table coming from any ENCODExplorer search function.
df	The reference data.table used to find the download. Files that are not available will be searched directly through the current ENCODE database.
format	The specific file format to download. Default : all
dir	The directory to locate the downloaded files
force	boolean to allow downloading a file even if it already exists in the directory. Default : TRUE

Value

A `character` with the downloaded files

Examples

```
fuzzy_result <- fuzzySearch("ENCSR396EAG", get_encode_df_demo(), filterVector = "accession")
## Not run: downloadEncode(fuzzy_result, format="tsv")
```

---

download_dt_file	*Downloads all files inside a data.table.*

Description

Downloads all files inside a data.table.

Usage

```
download_dt_file(input_dt, dir, force, show_experiment = FALSE)
```

Arguments

input_dt	The data.table to be looped over for determining which files should be downloaded.
dir	The path of the directory where the downloaded files should be saved.
force	If TRUE, existing files are downloaded again.
show_experiment	
	If TRUE, the name of the experiment is extracted from the data table and displayed in status messages.

Value

The name of the files which were downloaded.

---

download_single_file	*download_single_file Downloads a single file and checks if md5 checksums match.*

Description

download_single_file Downloads a single file and checks if md5 checksums match.

Usage

```
download_single_file(file_url, file_md5, dir = ".",
  experiment_name = NULL, force = TRUE)
```

Arguments

`file_url`	A character giving the URL of the file to be downloaded.
`file_md5`	A character giving the expected md5 checksum hash of the file to be downloaded.
`dir`	The directory where the downloaded file should be saved. Default: "."
`experiment_name`	
	An optional experiment name to be displayed with the status reports.
`force`	boolean indicating if existing files should be downloaded again. Default : TRUE

Value

A `character` with the name of the downloaded file.

ENCODExplorer	*ENCODExplorer*

Description

ENCODExplorer

---

fuzzySearch	*Fuzzysearch is a searching function for a string or a list of string within the encode_df* `data.table`. *For faster processing, pass encode_df object as database parameter.*

Description

Fuzzysearch is a searching function for a string or a list of string within the encode_df `data.table`. For faster processing, pass encode_df object as database parameter.

Usage

```
fuzzySearch(searchTerm = NULL, database = get_encode_df(),
  filterVector = NULL, multipleTerm = FALSE, ignore_case = TRUE)
```

Arguments

`searchTerm`	The keyword or a `list` of keyword to search.
`database`	A `data.table` with similar format as encode_df database.
`filterVector`	A `character` to apply the search on specific column.
`multipleTerm`	A boolean that indicate if the searchTerm is a list or even multiple searchTerm separete by a comma in a single string.
`ignore_case`	A boolean to enable the case sensitivity.

Value

A `data.table` corresponding the every row of the database that contain at least of one the searchTerm.

Examples

```
fuzz_ex <- fuzzySearch(searchTerm=c("ELAVL1","atf7"),
database=get_encode_df_demo(), filterVector ="target", multipleTerm = TRUE)
```

---

get_encode_df   *Returns a "light" version of ENCODE file metadata.*

Description

Returns a "light" version of ENCODE file metadata.

Usage

```
get_encode_df()
```

Value

a `data.table` containing the most relevant metadata for all ENCODE files.

Examples

```
my_encode_df = get_encode_df()
```

---

get_encode_df_demo   *Get a demo encode_df*

Description

Get a demo encode_df

Usage

```
get_encode_df_demo()
```

Value

A vector of regions filenames

Examples

```
encode_df_demo <- get_encode_df_demo()
```

get_encode_df_full	*Concatenates all available file metadata into a single data table.*

Description

Concatenates all available file metadata into a single data table.

Usage

```
get_encode_df_full()
```

Value

a `data.table` containing relevant metadata for all ENCODE files.

Examples

```
my_full_encode_df = get_encode_df_full()
```

queryEncode	*Produce a subset of data following predefined criteria*

Description

After running the `prepare_ENCODEDb` function, this function will allow you to extract a subset of data encording to the following criteria : accession, assay name, biosample, dataset accession, file accession, file format, laboratory, donor organism, target and treatment.

Usage

```
queryEncode(df = get_encode_df(), set_accession = NULL, assay = NULL,
  biosample_name = NULL, dataset_accession = NULL,
  file_accession = NULL, file_format = NULL, lab = NULL,
  organism = NULL, target = NULL, treatment = NULL, project = NULL,
  biosample_type = NULL, file_status = "released",
  status = "released", fixed = TRUE, quiet = FALSE, fuzzy = FALSE)
```

Arguments

df	data.frame containing ENCODE experiment and dataset metadata
set_accession	character string to select the accession
assay	character string to select the assay type
biosample_name	character string to select the biosample name
dataset_accession	
	character string to select the dataset accession
file_accession	character string to select the file accesion
file_format	character string to select the file format

`lab`	character string to select the laboratory
`organism`	character string to select the donor organism
`target`	character string to select the experimental target
`treatment`	character string to select the treatment
`project`	character string to select the project
`biosample_type`	character string to select the biosample type
`file_status`	character string to select the file status ("released", "revoked", "all"). Default "released"
`status`	character string to select the dataset/experiment status
`fixed`	logical. If TRUE, pattern is a string to be matched as it is.
`quiet`	logical enables to switch off the result summary information when setting at TRUE.
`fuzzy`	Search for substring or alternate hyphenations. Default: TRUE

Details

By default, the query can be made on an exact match term. This behaviour can be modified by setting the `fixed` argument at `TRUE`

Value

a `data.frames` containing data about ENCODE experiments and datasets

Examples

```
queryEncode(df = get_encode_df_demo(), biosample_name = "A549",
            file_format = "bam")
```

---

queryEncodeGeneric   *Produce a subset of data following predefined criteria.*

Description

After running the `prepare_ENCODEDb` function, this function will allow you to extract a subset of the files it describes. Search terms are passed in as named parameters, where the parameter's name indicates the field, and its value the terms to be searched for. Each term may be a vector of values, which are processed using the OR logical operation (the function will return all results matching at least one of the terms). In contrast, separate search fields are subjected to the AND logical operation.

Usage

```
queryEncodeGeneric(df = get_encode_df(), fixed = TRUE, quiet = FALSE,
  fuzzy = FALSE, ...)
```

Arguments

df	data.frame containing ENCODE experiment and dataset metadata
fixed	logical. If TRUE, pattern is a string to be matched as it is. If FALSE, case insensitive perl regular expression matching is used.
quiet	logical enables to switch off the result summary information
fuzzy	logical. If TRUE while fixed is also TRUE, allows searching by substrings and alternate space or hyphenation spellings. For example, "MCF7" will match "MCF-7" or "RNA-Seq" will match "polyA mRNA RNA-Seq".
...	All other named parameters are used as terms to be searched for, with the parameter name naming the field (biosample_name, assay, etc.) and the value being the terms that are searched for.

Details

Possible search fields include the following: accession, assay name, biosample, dataset accession, file accession, file format, laboratory, donor organism, target and treatment.

By default, the query is made using exact matches. Set `fixed` to `FALSE` to use regular expression matching, and `fuzzy` to `TRUE` to search for substring or alternate hyphenations. These options cannot be combined.

Value

a data.frames containing data about ENCODE experiments and datasets

Examples

```
# Will return all bam files from biosample A549.
res = queryEncodeGeneric(biosample_name = "A549", file_format = "bam")

# Will return all bam files from biosamples A549 and HeLA-S3.
res = queryEncodeGeneric(biosample_name = c("A549", "HeLa-S3"), file_format = "bam")

# Will return all fles where the assay contains RNA-Seq or a substrings
# thereof, such as "polyA mRNA RNA-Seq" or "small RNA-Seq".
res = queryEncodeGeneric(assay="RNA-Seq", fuzzy=TRUE)
```

---

searchEncode	*Simulate a query on ENCODE website and return the result as a* data.frame

---

Description

data.frames produced when converting JSON to data.frame with the fromJSON function will sometime have columns that are lists and/or columns that are data.frames.

Usage

```
searchEncode(searchTerm = NULL, limit = 10, quiet = FALSE)
```

Arguments

searchTerm	a search term
limit	the maximum number of return entries, default 10.
quiet	logical value enables to switch off the result summary information when setting at TRUE. will return all the result. It can generate large results set.

Details

This function simulates a basic query on ENCODE website

Value

a data.frame corresponding Every object that matches the search term

Examples

```
searchEncode("ChIP-Seq+H3K4me1")
```

---

searchToquery	*Convert searchEncode output in queryEncode output.*

Description

After processing to a basic search with the searchEncode function you can convert your result in a queryEncode output. Thus you can benefit from the collected metadata.

Usage

```
searchToquery(df = get_encode_df(), searchResults, quiet = TRUE)
```

Arguments

df	list of two data.frame containing ENCODE experiment and dataset metadata.
searchResults	the results set generated from searchEncode
quiet	logical enables to switch off the result summary information when setting at TRUE.

Details

The output is compatible with the dowload function.

Value

a list of two data.frames containing data about ENCODE experiments and datasets

Examples

```
search_res <- searchEncode(searchTerm = "switchgear elavl1", limit = "1")
res <- searchToquery(searchResults = search_res, quiet = TRUE)
```

---

shinyEncode	*Launch a shiny interface for ENCODExplorer*

Description

Launch a shiny interface for ENCODExplorer

Usage

```
shinyEncode()
```

Value

None

Examples

```
## Not run: shinyEncode
```

Index