Package ‘ENCODExplorer’

April 14, 2017

Name ENCODExplorer

Type Package

Title A compilation of ENCODE metadata

Version 2.0.6

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 (>= 3.3), shiny, DT, shinythemes

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

Suggests RUnit,BiocGenerics,knitr, curl, httr

LazyData true

biocViews Infrastructure, DataImport

RoxygenNote 5.0.1

NeedsCompilation no

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

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 extract_table*

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 = NULL, 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 :NULL
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 extract. 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

```
fuzzy_result <- fuzzySearch(searchTerm = "brca", database=encode_df, filterVector ="target")
design_result <- createDesign(input = fuzzy_result,df=encode_df, 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 = NULL, 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", encode_df, filterVector = "accession")
## Not run: downloadEncode(fuzzy_result, format="tsv")
```

`ENCODExplorer` *ENCODExplorer*

Description

ENCODExplorer

`encode_df` *Metadata from the ENCODE database in a R object. The tables were generated with the* `prepare_ENCODEdb` *function.*

Description

Metadata from the ENCODE database in a R object. The tables were generated with the `prepare_ENCODEdb` function.

Usage

```
data(encode_df)
```

Format

A data table

Value

A data table

See Also

`get_encode_types` to get a list of possible types. Note that some of the types are empty tables that are not included in the database created with `prepare_ENCODEdb` function.

export_ENCODEdb_matrix *Extract essential informations from a list of data.table in a `list` of `data.tables`*

Description

Extract essential informations from a list of data.table in a `list` of `data.tables`

Usage

```
export_ENCODEdb_matrix(database_filename)
```

Arguments

`database_filename`	The name of the file to save the database into.

Value

a `list` containing two elements. The first one 'experiment' is a `data.table` containing essential informations for each file part of an experiment ; the second one 'dataset' is a `data.table` containing essential informations for each file part of a dataset.

Examples

```
    database_filename <- system.file("extdata/tables",package = "ENCODExplorer")
    ## Not run:
        export_ENCODEdb_matrix(database_filename = database_filename)

## End(Not run)
```

extract_table *Extract a data.frame corresponding to a table in ENCODE database*

Description

Extract a data.frame corresponding to a table in ENCODE database

Usage

```
extract_table(type)
```

Arguments

`type`	The type of table to extract from ENCODE rest api.

Value

a `data.frame` corresponding to the table asked. If no match is found, returns an empty `data.frame`

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 = NULL, 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=encode_df, filterVector ="target", multipleTerm = TRUE)
```

get_encode_types — *A list of known tables from ENCODE database.*

Description

The type (table) names are extracted from the schema list from ENCODE-DCC github repository: https://github.com/ENCODE-DCC/encoded/tree/master/src/encoded/schemas

Usage

```
get_encode_types()
```

Details

The data is extracted using the github api: https://developer.github.com/guides/getting-started/

Value

a vector of `character` with the names of the known tables in the ENCODE database.

`get_schemas` *Extract the schemas from ENCODE's github*

Description

The JSONs are fetched from: https://github.com/ENCODE-DCC/encoded/tree/master/src/encoded/schemas

Usage

```
get_schemas()
```

Details

The data is extracted using the github api: https://developer.github.com/guides/getting-started/

The data is then downloaded using the `jsonlite` package.

Value

a `list` of schemas.

`prepare_ENCODEdb` *Create the list of data.table for the tables in ENCODE*

Description

Create the list of data.table for the tables in ENCODE

Usage

```
prepare_ENCODEdb(database_filename = "inst/extdata/tables.RDA",
  types = get_encode_types(), overwrite = FALSE)
```

Arguments

`database_filename`
The name of the file to save the database into.

`types` The names of the tables to extract from ENCODE rest api.

`overwrite` Should tables already present in database be overwrited Default: `FALSE`.

Value

is a `list` with selected tables from ENCODE that were used to create the list of `data.table` .

Examples

```
prepare_ENCODEdb(database_filename = "tables.RDA", types = "platform")
file.remove("platform.RDA")
    ## Not run:
        prepare_ENCODEdb("ENCODEdb.RDA")

## End(Not run)
```

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 = NULL, 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)
```

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.

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

```
    ## Not run:
    queryEncode(biosample_name = "A549", file_format = "bam")

## End(Not run)
```

---

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 = NULL, 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