

# Package ‘ENCODEExplorer’

September 21, 2017

**Name** ENCODEExplorer

**Type** Package

**Title** A compilation of ENCODE metadata

**Version** 2.2.0

**Date** 2015-02-25

**Author** Charles Joly Beauparlant <charles.joly-beauparlant@crchul.ulaval.ca>, Audrey Lemacon <lemacon.audrey@ulaval.ca>, Louis Gendron <louisg.212@gmail.com> Astrid-Louise Deschenes, <astrid-louise.deschenes@crchudequebec.ulaval.ca> and Arnaud Droit <arnaud.droit@crchudequebec.ulaval.ca >

**Author@R** c(person(`Charles`,`Joly Beauparlant`, email = `charles.joly-beauparlant@crchul.ulaval.ca`, role = `aut,cre`), person(`Audrey`,`Lemacon`, email = `lemacon.audrey@ulaval.ca`, role = `aut`), person(`Arnaud`,`Droit`, email = `arnaud.droit@crchudequebec.ulaval.ca`, role = `aut`), person(`Louis`,`Gendron`, email = `louisg.212@gmail.com`, role = `ctb`), person(`Astrid-Louise`,`Deschenes`, email = `astrid-louise.deschenes@crchudequebec.ulaval.ca`, role = `ctb`), person(`Arnaud`,`Droit`, email = `arnaud.droit@crchudequebec.ulaval.ca`))

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

**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/ENCODEExplorer/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

## R topics documented:

clean_column . . . . .	2
clean_table . . . . .	3
createDesign . . . . .	3
downloadEncode . . . . .	4
ENCODEexplorer . . . . .	5
encode_df . . . . .	5
export_ENCODEdb_matrix . . . . .	6
extract_table . . . . .	6
fuzzySearch . . . . .	7
get_encode_types . . . . .	7
get_schemas . . . . .	8
prepare_ENCODEdb . . . . .	8
queryEncode . . . . .	9
searchEncode . . . . .	10
searchToquery . . . . .	11
shinyEncode . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

clean_column	<i>Clean a single column of the data.frame</i>
--------------	--

---

### 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	<i>Clean a data.frame that was produced by extract_table</i>
-------------	--

---

**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	<i>Create a design for the files associated with the result of a queryEncode, fuzzySearch research or a data.table from createDesign.</i>
--------------	---

---

**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 correspond to the type of dataset that will be extracted. 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	<i>downloadEncode is used to download a serie of files or datasets using their accession.</i>
----------------	---

---

### 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 ENCODEExplorer 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	<i>ENCODExplorer</i>
---------------	----------------------

---

**Description**

ENCODExplorer

---

encode_df	<i>Metadata from the ENCODE database in a R object. The tables were generated with the prepare_ENCODEdb function.</i>
-----------	---

---

**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 = "ENCODEExplorer")
## 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	<i>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.</i>
-------------	--

---

### 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 separate 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	<i>A list of known tables from ENCODE database.</i>
------------------	---

---

### 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	<i>Extract the schemas from ENCODE's github</i>
-------------	---

---

**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	<i>Create the list of data.table for the tables in ENCODE</i>
------------------	---

---

**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 overwritten 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 encoding 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 accession
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.frame`s containing data about ENCODE experiments and datasets

**Examples**

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

## End(Not run)
```

---

searchEncode	<i>Simulate a query on ENCODE website and return the result as a data.frame</i>
--------------	---

---

**Description**

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

**Usage**

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

**Arguments**

<code>searchTerm</code>	a search term
<code>limit</code>	the maximum number of return entries, default 10.
<code>quiet</code>	logical value enables to switch off the result summary information when setting at <code>TRUE</code> . 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	<i>Convert searchEncode output in queryEncode output.</i>
---------------	---

---

**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 meta-data.
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	<i>Launch a shiny interface for ENCODEExplorer</i>
-------------	--

---

**Description**

Launch a shiny interface for ENCODEExplorer

**Usage**

```
shinyEncode()
```

**Value**

None

**Examples**

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

# Index

## \*Topic **datasets**

encode\_df, [5](#)

clean\_column, [2](#)

clean\_table, [3](#)

createDesign, [3](#)

downloadEncode, [4](#)

encode\_df, [5](#)

ENCODEExplorer, [5](#)

ENCODEExplorer-package (ENCODEExplorer), [5](#)

export\_ENCODEdb\_matrix, [6](#)

extract\_table, [6](#)

fuzzySearch, [7](#)

get\_encode\_types, [5](#), [7](#)

get\_schemas, [8](#)

prepare\_ENCODEdb, [5](#), [8](#)

queryEncode, [9](#)

searchEncode, [10](#)

searchToquery, [11](#)

shinyEncode, [11](#)