

Package ‘ENCODExplorerData’

December 10, 2025

Name ENCODExplorerData

Type Package

Title A compilation of ENCODE metadata

Version 0.99.5

Date 2019-04-11

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/ArnaudDroitLab/ENCODExplorerData/issues>

VignetteBuilder knitr

Depends R (>= 3.6),

Imports tools, jsonlite, RCurl, data.table, methods

Suggests AnnotationHub, RUnit, BiocGenerics, knitr, curl, httr

biocViews AnnotationHub, Homo_sapiens, Mus_musculus, CustomDBSchema

RoxygenNote 7.0.2

git_url <https://git.bioconductor.org/packages/ENCODExplorerData>

git_branch devel

git_last_commit c69d7b9

git_last_commit_date 2019-12-04

Repository Bioconductor 3.23

Date/Publication 2025-12-10

Author Charles Joly Beauparlant [aut],
Eric Fournier [aut, cre],
Arnaud Droit [aut]

Maintainer Eric Fournier <Fournier.Eric.2@crchudequebec.ulaval.ca>

Contents

build_file_db_from_raw_tables	2
clean_column	3
clean_table	3
ENCODEExplorerData	4
encode_df_full	4
encode_df_lite	5
fetch_and_clean_raw_ENCODE_tables	6
fetch_table_from_ENCODE_REST	6
generate_encode_df_full	7
generate_encode_df_lite	8
get_encode_types	8
get_schemas	9
get_schema_urls	9
Index	11

build_file_db_from_raw_tables	<i>Extract file metadata from the full set of ENCODE metadata tables.</i>
-------------------------------	---

Description

Extract file metadata from the full set of ENCODE metadata tables.

Usage

build_file_db_from_raw_tables(tables)

Arguments

tables A list of ENCODE metadata tables as loaded by fetch_and_clean_raw_ENCODE_tables.

Value

A list containing two data.table objects: encode_df, containing the most interesting metadata columns, and encode_df_ext, containing relevant metadata for all ENCODE files.

Examples

```
## Not run:
  tables = fetch_and_clean_raw_ENCODE_tables()
  build_file_db_from_raw_tables(tables = tables)

## End(Not run)
```

clean_column	<i>Clean a single column of a flattened data.frame returned by ENCODE.</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 to be processed.
table	The table produced by the fetch_table_from_ENCODE_REST function.

Details

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

Value

A flat column or columns to be cbind'ed into the final data.frame.

clean_table	<i>Clean a data.frame that was produced by fetch_table_from_ENCODE_REST</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 fetch_table_from_ENCODE_REST 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`.

Examples

```
clean_table(ENCODEExplorerData:::fetch_table_from_ENCODE_REST("award"))
```

ENCODEExplorerData	<i>ENCODEExplorerData</i>
--------------------	---------------------------

Description

This package aims to ease access to ENCODE file metadata by converting them into an easy-to-use `data.table`.

Details

The main feature of ENCODEExplorerData are the two ENCODE file metadata data tables exported through AnnotationHub, [encode_df_lite](#) and [encode_df_full](#)). While these can be accessed directly like any other `data.table`, we recommend using the **ENCODEExplorer** companion package, which contains utility functions for querying them, using the online ENCODE search function, downloading selected files, and retrieving control-treatment experimental designs from ENCODE.

This package also exposes functions for regenerating up-to-date versions of the metadata tables. See the [fetch_and_clean_raw_ENCODE_tables](#), [generate_encode_df_lite](#) and [generate_encode_df_full](#) functions for more details.

See Also

[encode_df_lite](#), [encode_df_full](#), [fetch_and_clean_raw_ENCODE_tables](#), [generate_encode_df_lite](#), [generate_encode_df_full](#)

encode_df_full	<i>ENCODE file metadata, Full version</i>
----------------	---

Description

Metadata for the files made available by ENCODE database as a [data.table](#) object. See `inst/scripts/make-data.R` for the generation process. `encode_df_full` contains all processed metadata columns, including content md5sums, cloud URLs, etc. Operations on `encode_df_full` will take longer than those on [encode_df_lite](#), but may be required if some of the extra metadata columns are necessary for your needs.

Format

A data table

See Also

[generate_encode_df_full](#), [encode_df_lite](#)

Examples

```
# You can use AnnotationHub to retrieve encode_df_full.
library(AnnotationHub)
hub <- AnnotationHub()
myfiles <- subset(hub, title=="ENCODE File Metadata (Full, 2019-04-12 build)")

# You can then have a look at the metadata of the retrieved object.
myfiles

# Finally, you can access the data.table itself by indexing into the
# object returned by subset.
myfiles[[1]]
```

encode_df_lite	<i>ENCODE file metadata, Light version</i>
----------------	--

Description

Metadata for the files made available by ENCODE database as a [data.table](#) object. See `inst/scripts/make-data.R` for the generation process. `encode_df_lite` contains a curated subset of the full metadata and is faster to load and easier to work with than [encode_df_full](#).

Format

A data table

See Also

[generate_encode_df_lite](#), [encode_df_full](#)

Examples

```
# You can use AnnotationHub to retrieve encode_df_lite.
library(AnnotationHub)
hub <- AnnotationHub()
myfiles <- subset(hub, title=="ENCODE File Metadata (Light, 2019-04-12 build)")

# You can then have a look at the metadata of the retrieved object.
myfiles

# Finally, you can access the data.table itself by indexing into the
# object returned by subset.
myfiles[[1]]
```

```
fetch_and_clean_raw_ENCODE_tables
```

Fetches and preprocess the raw metadata tables from ENCODE.

Description

Fetches and preprocess the raw metadata tables from ENCODE.

Usage

```
fetch_and_clean_raw_ENCODE_tables(
  cache_filename = "tables.RDA",
  types = get_encode_types(),
  overwrite = FALSE,
  precache = NULL
)
```

Arguments

cache_filename	A file name for caching the selected tables into.
types	The names of the tables to extract using the ENCODE rest api.
overwrite	If cache_filename already exists, should it be overwritten? Default: FALSE.
precache	A path to cache the raw metadata as returned by ENCODE and parsed using jsonlite. If NULL, no caching is performed. Default: FALSE.

Value

A list with all selected tables from ENCODE.

Examples

```
fetch_and_clean_raw_ENCODE_tables(cache_filename = "platform.RDA", types = "platform")
file.remove("platform.RDA")
```

```
fetch_table_from_ENCODE_REST
```

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

Description

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

Usage

```
fetch_table_from_ENCODE_REST(type)
```

Arguments

`type` The type of table to extract from the ENCODE rest api. Available types can be obtained using [get_encode_types](#).

Value

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

`generate_encode_df_full`

Given the raw ENCODE tables, this generate a data.table with the full set of file metadata columns.

Description

Given the raw ENCODE tables, this generate a `data.table` with the full set of file metadata columns.

Usage

```
generate_encode_df_full(tables)
```

Arguments

`tables` A list of ENCODE metadata tables as loaded by `fetch_and_clean_raw_ENCODE_tables`.

Value

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

Examples

```
## Not run:
  tables = fetch_and_clean_raw_ENCODE_tables()
  export_ENCODEdb_matrix(tables = tables)

## End(Not run)
```

```
generate_encode_df_lite
```

Extract file metadata from the full set of ENCODE metadata tables.

Description

Extract file metadata from the full set of ENCODE metadata tables.

Usage

```
generate_encode_df_lite(tables)
```

Arguments

`tables` A list of ENCODE metadata tables as loaded by `fetch_and_clean_raw_ENCODE_tables`.

Value

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

Examples

```
## Not run:
  tables = fetch_and_clean_raw_ENCODE_tables()
  export_ENCODEdb_matrix(tables = tables)

## End(Not run)
```

```
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.

Examples

```
get_encode_types()
```

```
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.

Examples

```
ENCODEExplorerData::get_schemas()
```

```
get_schema_urls
```

Returns the URLs for downloading the XML schemas from ENCODE's github.

Description

Returns the URLs for downloading the XML schemas from ENCODE's github.

Usage

```
get_schema_urls()
```

Value

a character vector of schema download URLs.

Examples

```
ENCODEExplorerData::get_schema_urls()
```

Index

* **internal**

- build_file_db_from_raw_tables, [2](#)
- clean_column, [3](#)
- fetch_table_from_ENCODE_REST, [6](#)

build_file_db_from_raw_tables, [2](#)

clean_column, [3](#)
clean_table, [3](#)

data.table, [4](#), [5](#)

encode_df_full, [4](#), [4](#), [5](#)
encode_df_lite, [4](#), [5](#), [5](#)
ENCODEExplorerData, [4](#)

fetch_and_clean_raw_ENCODE_tables, [4](#), [6](#)
fetch_table_from_ENCODE_REST, [6](#)

generate_encode_df_full, [4](#), [5](#), [7](#)
generate_encode_df_lite, [4](#), [5](#), [8](#)
get_encode_types, [7](#), [8](#)
get_schema_urls, [9](#)
get_schemas, [9](#)