

# Package ‘ENCODExplorer’

March 8, 2021

**Name** ENCODExplorer

**Type** Package

**Title** A compilation of ENCODE metadata

**Version** 2.16.0

**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)

**Imports** methods, tools, jsonlite, RCurl, tidyr, data.table, dplyr, stringr, stringi, utils, AnnotationHub, GenomicRanges, rtracklayer, S4Vectors, GenomeInfoDb, ENCODExplorerData

**Suggests** RUnit, BiocGenerics, knitr, curl, httr, shiny, shinythemes, DT

**LazyData** true

**biocViews** Infrastructure, DataImport

**RoxygenNote** 7.0.2

**git\_url** <https://git.bioconductor.org/packages/ENCODExplorer>

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** cd5a8bf

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2021-03-07

**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](mailto:charles.joly-beauparlant@crchul.ulaval.ca)>

**R topics documented:**

buildConsensusPeaks . . . . .	2
buildExpressionSummary . . . . .	3
createDesign . . . . .	4
downloadEncode . . . . .	6
download_dt_file . . . . .	6
download_single_file . . . . .	7
ENCODEBindingConsensus-class . . . . .	8
ENCODEExpressionSummary-class . . . . .	9
ENCODESummary-class . . . . .	10
ENCODEExplorer . . . . .	12
fuzzySearch . . . . .	12
get_encode_df . . . . .	13
get_encode_df_demo . . . . .	14
get_encode_df_full . . . . .	14
queryConsensusPeaks . . . . .	15
queryEncode . . . . .	16
queryEncodeGeneric . . . . .	17
queryExpressionGeneric . . . . .	18
queryGeneExpression . . . . .	19
queryTranscriptExpression . . . . .	20
searchEncode . . . . .	21
searchToquery . . . . .	22
shinyEncode . . . . .	23
<b>Index</b>	<b>24</b>

---

buildConsensusPeaks	<i>Calculates the consensus peaks defined by the results of a previously completed ENCODE query.</i>
---------------------	--

---

**Description**

This function takes the result of a previous call to [queryEncode](#), splits the contained peak files by conditions (as specified by the `split_by` argument), then builds consensus peaks for each condition.

**Usage**

```
buildConsensusPeaks(
  query_results,
  split_by = NULL,
  consensus_threshold = 1,
  simplify = FALSE,
  temp_dir = ".",
  force = FALSE,
  consensus_threshold_n = NULL
)
```

**Arguments**

query_results	A data.table returned by <a href="#">queryEncode</a> or <a href="#">queryEncodeGeneric</a> .
split_by	A vector of column names from query_results that will be used to split the consensus binding sites according to condition. If NULL, all elements of query_results are used in the same consensus calculation.
consensus_threshold	A numeric value between 0 and 1, indicating the proportion of peak files in which a peak must appear for it to be included within the consensus.
simplify	If TRUE, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".
temp_dir	The path to a directory where peak files will be downloaded.
force	A logical indicating whether already present files be redownloaded.
consensus_threshold_n	An integer indicating the number of peak files in which a peak must appear for it to be included within the consensus. IF this argument is provided, it supercedes consensus_threshold.

**Value**

An object of class [ENCODEBindingConsensus](#).

**Examples**

```
query_results = queryEncodeGeneric(biosample_name="A549", assembly="GRCh38",
                                  file_format="bed", output_type="peaks",
                                  treatment_duration_unit="minute",
                                  treatment_duration="(^5$|^10$)",
                                  target="NR3C1", fixed=FALSE)
res = buildConsensusPeaks(query_results, split_by=c("treatment_duration"),
                          consensus_threshold=0.5)
```

---

buildExpressionSummary

*Calculates average expression levels of the results of a previously completed ENCODE query.*

---

**Description**

This function takes the result of a previous call to [queryEncode](#), splits the contained expression files by conditions (as specified by the split\_by argument), then calculates average expression levels for each condition.

**Usage**

```
buildExpressionSummary(
  query_results,
  split_by,
  metric = NULL,
  simplify = FALSE,
```

```

    aggregate_function = mean,
    temp_dir = ".",
    force = FALSE
  )

```

### Arguments

**query\_results** A `data.table` returned by [queryEncode](#) or [queryEncodeGeneric](#).

**split\_by** A vector of column names from `query_results` that will be used to split the average expression levels. If `NULL`, all elements of `query_results` are used in the same average expression calculation.

**metric** A regular expression, indicating which column from the ENCODE data must be extracted. If `NULL`, ENCODExplorer data automatically detects and selects one of the TPM, FPKM or featurecount columns.

**simplify** If `TRUE`, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".

**aggregate\_function**  
A function which takes a vector as input and returns a single value summarizing the whole. Used to summarize expression metrics.

**temp\_dir** The path to a directory where peak files will be downloaded.

**force** A logical indicating whether already present files be redownloaded.

### Value

An object of class [ENCODEExpressionSummary](#).

### Examples

```

query_results = queryEncodeGeneric(biosample_name="neural tube",
                                   output_type="gene quantifications",
                                   file_type="tsv",
                                   assay="polyA plus RNA-seq",
                                   assembly="^mm10$",
                                   dataset_biosample_summary="(15.5|13.5)",
                                   fixed=FALSE)

buildExpressionSummary(query_results, split_by="dataset_biosample_summary")

```

---

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

---

download_dt_file	<i>Downloads all files inside a data.table.</i>
------------------	---

---

### 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.

---

ENCODEBindingConsensus-class

*ENCODEBindingConsensus: consensus peaks derived from ENCODE files.*

---

## Description

ENCODEBindingConsensus objects represent the intersection of called peaks across multiple replicates, split by arbitrary metadata columns. They can be constructed using the [queryConsensusPeaks](#) and [buildConsensusPeaks](#) functions.

## Usage

```
## S4 replacement method for signature 'ENCODEBindingConsensus,character'
names(x) <- value

peaks(x)

## S4 method for signature 'ENCODEBindingConsensus'
peaks(x)

consensus(x)

## S4 method for signature 'ENCODEBindingConsensus'
consensus(x)

## S4 method for signature 'ENCODEBindingConsensus'
show(object)
```

## Arguments

x	The <a href="#">ENCODESummary</a> object.
value	The new names for the elements of the <a href="#">ENCODESummary</a> object.
object	The <a href="#">ENCODESummary</a> object.

## Value

For peaks, a list of [GRangesList](#) of the per-condition original peaks used to build the object. For consensus, a [GRangesList](#) of the per-condition consensus peaks.

## Slots

peaks	The per-condition original peaks used to build the consensus.
consensus	The per-condition consensus peaks.
consensus_threshold	The proportion of replicates which must bear a specific peak for it to be added to the set of consensus peaks.



**Methods**

ENCODEBindingConsensus object can be accessed through the methods from the ENCODESummary class, as well as ENCODEBindingConsensus-specific methods:

peaks Returns a list of [GRangesList](#) of the per-condition original peaks used to build the object.

consensus Returns a [GRangesList](#) of the per-condition consensus peaks.

**Examples**

```
res = queryConsensusPeaks("22Rv1", "GRCh38", "CTCF")
peaks(res)
consensus(res)
```

---

ENCODEExpressionSummary-class

*ENCODEExpressionSummary summarize means of expression across ENCODE files.*

---

**Description**

ENCODEExpressionSummary objects represent means (or medians) of expression levels across multiple replicate samples, split by arbitrary metadata columns. They can be constructed using the [queryGeneExpression](#), [queryTranscriptExpression](#) and [buildExpressionSummary](#) functions.

**Usage**

```
## S4 replacement method for signature 'ENCODEExpressionSummary,character'
names(x) <- value

metric_data(x)

## S4 method for signature 'ENCODEExpressionSummary'
metric_data(x)

metric(x)

## S4 method for signature 'ENCODEExpressionSummary'
metric(x)

raw_data(x)

## S4 method for signature 'ENCODEExpressionSummary'
raw_data(x)

## S4 method for signature 'ENCODEExpressionSummary'
show(object)
```

**Arguments**

x	The <a href="#">ENCODESummary</a> object.
value	The new names for the elements of the <a href="#">ENCODESummary</a> object.
object	The <a href="#">ENCODESummary</a> object.

**Value**

For `raw_data`, a list of [GRangesList](#) of the per-condition original expression tables used to build the object. For `metric`, the regular expression used to select the column of metric values from the ENCODE files. For `metric_data`, a [data.frame](#) of the per-condition metric values.

**Slots**

<code>raw_data</code>	A list of data-frames containing the full raw data of each of the downloaded ENCODE files.
<code>metric</code>	A character giving the regular expression used to extract expression metrics from the ENCODE files.
<code>metric_data</code>	A <a href="#">data.frame</a> of the per-condition metric values.
<code>expression_type</code>	The type of expression which is being reported, either gene or transcripts.

**Methods**

ENCODEExpressionSummary object can be accessed through the methods from the ENCODESummary class, as well as ENCODEBindingConsensus-specific methods:

<code>raw_data</code>	Returns a list of <a href="#">GRangesList</a> of the per-condition original expression tables used to build the object.
<code>metric</code>	Returns the regular expression used to select the column of metric values from the ENCODE files.
<code>metric_data</code>	Returns a <a href="#">data.frame</a> of the per-condition metric values.

**Examples**

```
res = queryGeneExpression("bone marrow")
raw_data(res)
metric(res)
metric_data(res)
```

---

ENCODESummary-class     *ENCODESummary objects: summaries of multiple ENCODE files.*

---

**Description**

ENCODESummary objects is the base class of [ENCODEBindingConsensus-class](#) and [ENCODEExpressionSummary-class](#) objects. It provides methods to query which files were used to build the summary, the names of the grouped elements as well as their metadata.

**Usage**

```

## S4 method for signature 'ENCODESummary'
names(x)

## S4 replacement method for signature 'ENCODESummary,character'
names(x) <- value

## S4 method for signature 'ENCODESummary'
length(x)

## S4 method for signature 'ENCODESummary'
metadata(x)

file_metadata(x)

## S4 method for signature 'ENCODESummary'
file_metadata(x)

files(x)

## S4 method for signature 'ENCODESummary'
files(x)

## S4 method for signature 'ENCODESummary'
show(object)

```

**Arguments**

x	The <a href="#">ENCODESummary</a> object.
value	The new names for the elements of the <a href="#">ENCODESummary</a> object.
object	The <a href="#">ENCODESummary</a> object.

**Value**

For names, names<-, a copy of the object. For length, the number of elements. For files, a character vector. For file\_metadata, a list of data-frames with each file's metadata. For metadata, a data-frame with the discriminating metadata of each sample group.

**Slots**

files	The path of the files used in this summary.
file_metadata	A list of data-frames representing the ENCODE metadata of the files used to build the per-condition consensus.
metadata	A data-frame with the metadata of each element in the summary.

**Methods**

ENCODESummary object can be accessed through a variety of methods:

names	Returns the names of the elements.
names<-	Sets the names of the elements.

`length` Returns the number of elements.

`files` Returns a character vector of the ENCODE files used to build this object.

`file_metadata` Returns a list of per-condition metadata of the ENCODE files used to build the object.

`metadata` Returns a data-frame of the common per-condition metadata of the ENCODE files used to build the object.

`show` Print a summary of the object.

### Examples

```
res = queryConsensusPeaks("22Rv1", "GRCh38", "CTCF")
names(res)
files(res)
metadata(res)
print(res)
```

---

ENCODEexplorer

*ENCODEexplorer*

---

### Description

ENCODEexplorer

---

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 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=get_encode_df_demo(), filterVector ="target", multipleTerm = TRUE)
```

---

get_encode_df	<i>Returns a "light" version of ENCODE file metadata.</i>
---------------	---

---

**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()
```

---

queryConsensusPeaks     *Queries ENCODE for consensus peaks.*

---

### Description

Queries the ENCODE metadata to determine which peak files exists for the target protein in the biosample\_name biosample for the assembly genomic assembly, then builds per-condition (as determined by the treatment column and its adjuncts) consensus peaks.

### Usage

```
queryConsensusPeaks(  
  biosample_name,  
  assembly,  
  target,  
  simplify = FALSE,  
  use_interactive = FALSE  
)
```

### Arguments

biosample_name	The cell-line/tissue for which consensus peaks should be queried.
assembly	The target genomic assembly.
target	The target protein.
simplify	If TRUE, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".
use_interactive	If TRUE, the user will be prompted when ENCODEExplorer must choose how to filter the available data.

### Details

If you wish to have more control over the files used to build the consensus, use [buildConsensusPeaks](#).

### Value

An object of class [ENCODEBindingConsensus](#).

### See Also

[buildConsensusPeaks](#)

### Examples

```
queryConsensusPeaks("22Rv1", "GRCh38", "CTCF")
```

---

 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 = 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

<code>df</code>	data.frame containing ENCODE experiment and dataset metadata
<code>set_accession</code>	character string to select the accession
<code>assay</code>	character string to select the assay type
<code>biosample_name</code>	character string to select the biosample name
<code>dataset_accession</code>	character string to select the dataset accession
<code>file_accession</code>	character string to select the file accession
<code>file_format</code>	character string to select the file format
<code>lab</code>	character string to select the laboratory
<code>organism</code>	character string to select the donor organism
<code>target</code>	character string to select the experimental target
<code>treatment</code>	character string to select the treatment
<code>project</code>	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 files 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)
```

---

queryExpressionGeneric

*Queries and returns average expression levels for a given biosample\_name.*

---

**Description**

ENCODE files are automatically split by biosample\_description (which will separate samples from different cell fractions or sequencing methods) and by the treatment columns.

**Usage**

```

queryExpressionGeneric(
  biosample_name,
  level = "gene quantifications",
  assay = NULL,
  assembly = NULL,
  simplify = TRUE,
  use_interactive = FALSE
)

```

**Arguments**

<code>biosample_name</code>	The cell-line/tissue for which average expression levels should be queried.
<code>level</code>	The type of expression level to summarize, either "gene quantifications" or "transcript quantifications".
<code>assay</code>	The assay type to summarize. If NULL, the most generic assay type is automatically selected.
<code>assembly</code>	The target genomic assembly. If NULL, the most recent available assembly is selected.
<code>simplify</code>	If TRUE, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".
<code>use_interactive</code>	If TRUE, the user will be prompted to select preferred metadata values when multiple possibilities are available.

**Value**

An object of class [ENCODEExpressionSummary](#).

**See Also**

[buildExpressionSummary](#), [queryGeneExpression](#)

**Examples**

```
queryExpressionGeneric("bone marrow")
```

---

<code>queryGeneExpression</code>	<i>Queries and returns average gene expression level for a given biosample_name.</i>
----------------------------------	--

---

**Description**

ENCODE files are automatically split by `biosample_description` (which will separate samples from different cell fractions or sequencing methods) and by the treatment columns.

**Usage**

```

queryGeneExpression(
  biosample_name,
  assay = NULL,
  assembly = NULL,
  simplify = TRUE,
  use_interactive = FALSE
)

```

**Arguments**

<code>biosample_name</code>	The cell-line/tissue for which average expression levels should be queried.
<code>assay</code>	The assay type to summarize. If NULL, the most generic assay type is automatically selected.
<code>assembly</code>	The target genomic assembly. If NULL, the most recent available assembly is selected.
<code>simplify</code>	If TRUE, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".
<code>use_interactive</code>	If TRUE, the user will be prompted to select preferred metadata values when multiple possibilities are available. available assembly is selected.

**Value**

An object of class [ENCODEExpressionSummary](#).

**See Also**

[buildExpressionSummary](#), [queryTranscriptExpression](#)

**Examples**

```
queryGeneExpression("bone marrow")
```

---

```
queryTranscriptExpression
```

*Queries and returns average transcript expression level for a given biosample\_name.*

---

**Description**

ENCODE files are automatically split by `biosample_description` (which will separate samples from different cell fractions or sequencing methods) and by the treatment columns.

**Usage**

```
queryTranscriptExpression(
  biosample_name,
  assay = NULL,
  assembly = NULL,
  simplify = TRUE,
  use_interactive = FALSE
)
```

**Arguments**

`biosample_name` The cell-line/tissue for which average expression levels should be queried.

`assay` The assay type to summarize. If `NULL`, the most generic assay type is automatically selected.

`assembly` The target genomic assembly. If `NULL`, the most recent available assembly is selected.

`simplify` If `TRUE`, non-discriminatory columns are removed from the metadata, and if only one sample group is found, it is renamed "All".

`use_interactive` If `TRUE`, the user will be prompted to select preferred metadata values when multiple possibilities are available.

**Value**

An object of class [ENCODEExpressionSummary](#).

**See Also**

[buildExpressionSummary](#), [queryGeneExpression](#)

**Examples**

```
queryTranscriptExpression("bone marrow")
```

---

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

```
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	<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 = get_encode_df(), 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`*Launch a shiny interface for ENCODEExplorer*

---

**Description**

Launch a shiny interface for ENCODEExplorer

**Usage**

```
shinyEncode()
```

**Value**

None

**Examples**

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

# Index

buildConsensusPeaks, [2](#), [8](#), [15](#)  
buildExpressionSummary, [3](#), [9](#), [19–21](#)

consensus  
    (ENCODEBindingConsensus-class),  
    [8](#)  
consensus, ENCODEBindingConsensus, ENCODEBindingConsensus-method  
    (ENCODEBindingConsensus-class),  
    [8](#)  
consensus, ENCODEBindingConsensus-method  
    (ENCODEBindingConsensus-class),  
    [8](#)  
createDesign, [4](#)

data.frame, [10](#)  
download\_dt\_file, [6](#)  
download\_single\_file, [7](#)  
downloadEncode, [6](#)

ENCODEBindingConsensus, [3](#), [15](#)  
ENCODEBindingConsensus-class, [8](#), [10](#)  
ENCODEExpressionSummary, [4](#), [19–21](#)  
ENCODEExpressionSummary-class, [9](#), [10](#)  
ENCODESummary, [8](#), [10](#), [11](#)  
ENCODESummary-class, [10](#)  
ENCODEExplorer, [12](#)

file\_metadata (ENCODESummary-class), [10](#)  
file\_metadata, ENCODESummary, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
file\_metadata, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
files (ENCODESummary-class), [10](#)  
files, ENCODESummary, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
files, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
fuzzySearch, [12](#)

get\_encode\_df, [13](#)  
get\_encode\_df\_demo, [14](#)  
get\_encode\_df\_full, [14](#)  
GRangesList, [8–10](#)

length, ENCODESummary-method  
    (ENCODESummary-class), [10](#)

metadata, ENCODESummary, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
metadata, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
metric (ENCODEExpressionSummary-class),  
    [9](#)  
metric, ENCODEBindingConsensus, ENCODEBindingConsensus-method  
    (ENCODEExpressionSummary-class),  
    [9](#)  
metric, ENCODEExpressionSummary-method  
    (ENCODEExpressionSummary-class),  
    [9](#)  
metric\_data  
    (ENCODEExpressionSummary-class),  
    [9](#)  
metric\_data, ENCODEBindingConsensus, ENCODEBindingConsensus-method  
    (ENCODEExpressionSummary-class),  
    [9](#)  
metric\_data, ENCODEExpressionSummary-method  
    (ENCODEExpressionSummary-class),  
    [9](#)

names, ENCODESummary-method  
    (ENCODESummary-class), [10](#)  
names<-, ENCODEBindingConsensus, character-method  
    (ENCODEBindingConsensus-class),  
    [8](#)  
names<-, ENCODEExpressionSummary, character-method  
    (ENCODEExpressionSummary-class),  
    [9](#)  
names<-, ENCODESummary, character-method  
    (ENCODESummary-class), [10](#)

peaks (ENCODEBindingConsensus-class), [8](#)  
peaks, ENCODEBindingConsensus, ENCODEBindingConsensus-method  
    (ENCODEBindingConsensus-class),  
    [8](#)  
peaks, ENCODEBindingConsensus-method  
    (ENCODEBindingConsensus-class),  
    [8](#)

queryConsensusPeaks, [8](#), [15](#)



queryEncode, [2-4](#), [16](#)  
queryEncodeGeneric, [3](#), [4](#), [17](#)  
queryExpressionGeneric, [18](#)  
queryGeneExpression, [9](#), [19](#), [19](#), [21](#)  
queryTranscriptExpression, [9](#), [20](#), [20](#)

raw\_data  
    (ENCODEExpressionSummary-class),  
    [9](#)

raw\_data, ENCODEBindingConsensus, ENCODEBindingConsensus-method  
    (ENCODEExpressionSummary-class),  
    [9](#)

raw\_data, ENCODEExpressionSummary-method  
    (ENCODEExpressionSummary-class),  
    [9](#)

searchEncode, [21](#)  
searchToquery, [22](#)  
shinyEncode, [23](#)

show, ENCODEBindingConsensus-method  
    (ENCODEBindingConsensus-class),  
    [8](#)

show, ENCODEExpressionSummary-method  
    (ENCODEExpressionSummary-class),  
    [9](#)

show, ENCODESummary-method  
    (ENCODESummary-class), [10](#)