Package ‘TFutils’

December 3, 2019

Title  TFutils

Description  Package to work with TF metadata from various sources.

Version  1.6.0

Depends  R (>= 3.5.0)

Imports  methods, dplyr, magrittr, miniUI, shiny, Rsamtools, GSEABase

Suggests  knitr, data.table, testthat, AnnotationDbi, AnnotationFilter, Biobase, GenomicFeatures, GenomicRanges, Gviz, IRanges, Rsamtools, S4Vectors, org.Hs.eh.db, EnsDb.Hsapiens.v75, BiocParallel, BiocStyle, GO.db, GenomicFiles, GenomeInfoDb, SummarizedExperiment, UpSetR, ggplot2, png, gwascat, MotifDb, motifStack, RColorBrewer

License  Artistic-2.0

LazyLoad  true

ByteCompile  true

LazyData  true

VignetteBuilder  knitr

RoxygenNote  6.1.1

biocViews  Transcriptomics

Roxygen  list(markdown = TRUE)

Encoding  UTF-8

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

git_branch  RELEASE_3_10

git_last_commit  7ae9a5f

git_last_commit_date  2019-10-29

Date/Publication  2019-12-02

Author  Vincent Carey [aut],
         Shweta Gopaulakrishnan [cre, aut]

Maintainer  Shweta Gopaulakrishnan <reshg@channing.harvard.edu>
### anchor_pmids

<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>anchor_pmids</td>
</tr>
<tr>
<td>browse_lambert_gwaslinks</td>
</tr>
<tr>
<td>browse_lambert_main</td>
</tr>
<tr>
<td>cisbpTFcat</td>
</tr>
<tr>
<td>cisbpTFcat_2.0</td>
</tr>
<tr>
<td>defaultCircosParms</td>
</tr>
<tr>
<td>demo_fimo_granges</td>
</tr>
<tr>
<td>directHitsInCISBP</td>
</tr>
<tr>
<td>encode690</td>
</tr>
<tr>
<td>fimo16</td>
</tr>
<tr>
<td>fimoMap</td>
</tr>
<tr>
<td>fimo_granges</td>
</tr>
<tr>
<td>genemodelDF</td>
</tr>
<tr>
<td>genemodForGviz</td>
</tr>
<tr>
<td>get_rslocs_38</td>
</tr>
<tr>
<td>grabTab</td>
</tr>
<tr>
<td>gwascat_hg19_chr17</td>
</tr>
<tr>
<td>HGNCmap</td>
</tr>
<tr>
<td>hocomoco.mono</td>
</tr>
<tr>
<td>hocomoco.mono.sep2018</td>
</tr>
<tr>
<td>importFIMO,TabixFile,GRanges-method</td>
</tr>
<tr>
<td>importFIMO_local_split</td>
</tr>
<tr>
<td>lambert_snps</td>
</tr>
<tr>
<td>metadata_tf</td>
</tr>
<tr>
<td>named_tf</td>
</tr>
<tr>
<td>retrieve_lambert_main</td>
</tr>
<tr>
<td>seqinfo_hg19_chr17</td>
</tr>
<tr>
<td>setupHIZE</td>
</tr>
<tr>
<td>show,TFCatalog-method</td>
</tr>
<tr>
<td>TFCatalog</td>
</tr>
<tr>
<td>TFCatalog-class</td>
</tr>
<tr>
<td>tffamCirc.plot</td>
</tr>
<tr>
<td>tffamCirc.prep</td>
</tr>
<tr>
<td>tfhash</td>
</tr>
<tr>
<td>TFlargs</td>
</tr>
<tr>
<td>tftColl</td>
</tr>
<tr>
<td>tftCollMap</td>
</tr>
<tr>
<td>topTraitsOfTargets</td>
</tr>
<tr>
<td>URL_s3_tf</td>
</tr>
</tbody>
</table>

**Index**

25

---

**anchor_pmids**

check columns of a dataframe for numerical tokens of 7 or 8 digits and create HTML anchors to pubmed.gov constituting a link to a PMID
Usage

anchor_pmids(dataframe)

Arguments

dataframe  a data.frame instance

Value

data.frame with HTML anchors to pubmed.gov inserted where 7- or 8-digit numbers are found

Note

The method of isolating putative PMIDs is peculiar to patterns found in the comment fields of annotated TF table (supplemental table S1 found in https://www.cell.com/cms/10.1016/j.cell.2018.01.029/attachment/88c0eca1-66f9-4068-b02e-bd3d55144f79/mmc2.xlsx of PMID 29425488). When DT::datatable is called on the output of this function with escape=FALSE the PMIDs will render as hyperlinks. Note that column 1 is assumed to be an ENSEMBL ID which could have 7 or 8 digits but is handled differently

Examples

```r
litdf = data.frame(id="ENSG00000116819", a="Binds the same GCCTGAGGC sequence as the other AP-2s (PMID: 24789576)", stringsAsFactors=FALSE)
anchor_pmids(litdf)
```

---

**browse_lambert_gwaslinks**

_browse several hundred disease-TF associations with hyperlinked PMIDs_

---

**Description**

browse several hundred disease-TF associations with hyperlinked PMIDs

**Usage**

browse_lambert_gwaslinks()

**Note**

Based on supplemental table S4 of PMID 29425488

**Examples**

```r
if (interactive()) browse_lambert_gwaslinks()
```
## `browse_lambert_main`

*use DT::datatable to browse the Lambert table S1*

### Description

use DT::datatable to browse the Lambert table S1

### Usage

```r
browse_lambert_main(cache = BiocFileCache::BiocFileCache(ask = FALSE))
```

### Arguments

- **cache**: a `BiocFileCache` instance

### Value

result of DT::datatable

### Note

PMIDs are converted to HTML anchors and DT::datatable is run with escape=FALSE.

### Examples

```r
if (interactive()) browse_lambert_main()
```

## `cisbpTFcat`

*`cisbpTFcat`: data.frame with information on CISBP TFs for human, retained for reproducibility support; see `cisbpTFcat_2.0` for a more recent catalog*

### Description

`cisbpTFcat`: data.frame with information on CISBP TFs for human, retained for reproducibility support; see `cisbpTFcat_2.0` for a more recent catalog

### Usage

```r
cisbpTFcat
```

### Format

`data.frame`

### Note

Extracted March 2018, checked August 2018. The only changes observed are that genes ZUFSP and T are used has HGNC values in the March catalog; these symbols seem to be absent from the `org.Hs.eg.db` of August 2018. The records involved are 1356, 7412 and 7413. These symbols were left in the package image of CISBP in August 2018.
cisbpTFcat_2.0

Source

http://cisbp.ccbr.utoronto.ca/bulk.php select Homo_sapiens

Examples

head(TFutils::cisbpTFcat)

cisbpTFcat_2.0  cisbpTFcat_2.0: data.frame with information on CISBP TFs for human, described in PMID 31133749

Description

cisbpTFcat_2.0: data.frame with information on CISBP TFs for human, described in PMID 31133749

Usage

cisbpTFcat_2.0

Format

data.frame

Note

Extracted August 2019.

Source

http://cisbp.ccbr.utoronto.ca/bulk.php select Homo_sapiens

Examples

head(TFutils::cisbpTFcat_2.0)

defaultCircosParms  basic layout parameters for circos

Description

basic layout parameters for circos

Usage

defaultCircosParms()

Value

a list

Examples

head(defaultCircosParms())
**demo_fimo_granges**

*a list of GRanges instances with TF FIMO scores returned by fimo_granges*

---

**Description**

A list of GRanges instances with TF FIMO scores returned by fimo_granges.

**Usage**

demo_fimo_granges

**Format**

a list of GRanges instances

**Examples**

```r
names(S4Vectors::mcols(demo_fimo_granges$VDR[[1]]))
```

---

**directHitsInCISBP**

demonstrate interoperation of TF catalog with GWAS catalog

---

**Description**

demonstrate interoperation of TF catalog with GWAS catalog

**Usage**

directHitsInCISBP(traitTag, gwascat)

**Arguments**

- **traitTag** character(1) string found in DISEASE/TRAIT field of gwascat instance
- **gwascat** instance of gwaswloc-class

**Value**

data.frame

**Examples**

data(gwascat_hg19_chr17)
directHitsInCISBP("Prostate cancer", gwascat_hg19_chr17)
**encode690**

*encode690: DataFrame extending AnnotationHub metadata about ENCODE cell line x TF ranges*

**Description**

encode690: DataFrame extending AnnotationHub metadata about ENCODE cell line x TF ranges

**Usage**

encode690

**Format**

DataFrame

**Source**

see metadata(encode690)

**Examples**

```
names(TFutils::encode690)
TFutils::encode690[,1:5]
```

---

**fimo16**

*fimo16: GenomicFiles instance to AWS S3-resident FIMO bed for 16 TFs*

**Description**

fimo16: GenomicFiles instance to AWS S3-resident FIMO bed for 16 TFs

**Usage**

fimo16

**Format**

GenomicFiles for a TabixFileList

**Source**

K. Glass FIMO runs, see https://doi.org/10.1016/j.celrep.2017.10.001

**Examples**

```
TFutils::fimo16
```
**fimoMap**

*fimoMap: table with Mnnnn (motif PWM tags) and HGNC symbols for TFs*

---

**Description**

*fimoMap: table with Mnnnn (motif PWM tags) and HGNC symbols for TFs*

**Usage**

*fimoMap*

**Format**

data.frame

**Source**

Kimberly Glass (rekrg@channing.harvard.edu)

**Examples**

```r
head(TFutils::fimoMap)
```

---

**fimo_granges**

*create a list of GRanges for FIMO hits in a GenomicFiles instance, corresponding to a GRanges-based query*

---

**Description**

create a list of GRanges for FIMO hits in a GenomicFiles instance, corresponding to a GRanges-based query

**Usage**

*fimo_granges(gf, query)*

**Arguments**

- **gf**: GenomicFiles instance, like fimo16 in TFutils
- **query**: a GRanges specifying ranges to check for TF binding scores

**Value**

a list of GRanges, produced by GenomicFiles::reduceByRange

**Note**

Be sure to use `register([BPPARAM])` appropriately.
**Examples**

```r
if (interactive()) {  # need internet
  # setup -- annotate fimo16 object and create an informative
  # query
  colnames(fimo16) = fimo16$HGNC
  si = GenomeInfoDb::Seqinfo(genome="hg19")["chr17"] # to fix query genome
  myg = GRanges("chr17", IRanges(38.07e6,38.09e6), seqinfo=si)
  requireNamespace("BiocParallel")
  BiocParallel::register(BiocParallel::SerialParam())
  f1 = fimo_granges(fimo16[, c("VDR", "POU2F1")], myg)
  f1
}
```

---

**genemodelDF**

*use EnsDb to generate an exon-level model of genes identified by symbol*

**Description**

use EnsDb to generate an exon-level model of genes identified by symbol

**Usage**

```
genemodelDF(sym, resource, columnsKept = c("gene_id", "tx_id"), ...)
```

**Arguments**

- `sym` - a character() vector of gene symbols
- `resource` - should be or inherit from EnsDb, answering exons(), with AnnotationFilter::SymbolFilter as filter parameter
- `columnsKept` - character vector used as columns param in exons()
- `...` - passed to exons()

**Value**

data.frame instance with exons in rows

**Note**

There are many approaches available to acquiring 'gene models' in Bioconductor; this one emphasizes the use of the exons method for Ensembl annotation.

**Examples**

```r
if (requireNamespace("EnsDb.Hsapiens.v75")) {
  orm = genemodelDF("ORMDL3", EnsDb.Hsapiens.v75::EnsDb.Hsapiens.v75)
  dim(orm)
}
head(orm)
```
**Description**

create a GeneRegionTrack instance for selected symbols

**Usage**

```r
genemodForGviz(sym = "ORMDL3", id_elem = c("symbol", "tx_id"),
resource = EnsDb.Hsapiens.v75::EnsDb.Hsapiens.v75, ...)
```

**Arguments**

- `sym` character vector of gene symbols, should be neighboring genes
- `id_elem` vector of names of columns generated by genemodelDF to be used to label transcripts
- `resource` should be or inherit from EnsDb, answering exons(), with AnnotationFilter::SymbolFilter as filter parameter
- `...` passed to genemodelDF

**Value**

instance of Gviz GeneRegionTrack

**Note**

This function helps to display the locations of TF binding sites in the context of complex gene models. A complication is that we have nice visualization of quantitative affinity predictions for TFs in the vignette, based on ggplot2, but it is not clear how to use that specific code to work with Gviz.

**Examples**

```r
if (requireNamespace("EnsDb.Hsapiens.v75") &
requireNamespace("Gviz")) {
  orm = genemodForGviz("ORMDL3", resource = EnsDb.Hsapiens.v75::EnsDb.Hsapiens.v75)
  Gviz::plotTracks(orm, showId=TRUE) # change id_elem for shorter id string
}
```
get_rslocs_38

utility to obtain location etc. for rsids of SNPs

Description
utility to obtain location etc. for rsids of SNPs

Usage
get_rslocs_38(rsids = c("rs6060535", "rs56116432"))

Arguments
rsids character vector of dbSNP identifiers

Note
Uses rest.ensembl.org, posting to variant_recorder/homo_sapiens. Parses result minimally, using only the first SPDI to obtain location information, adding 1 as ensembl genomic coordinates are zero-based.

Examples
get_rslocs_38()

grabTab
create table of TF targets and related metadata

Description
create table of TF targets and related metadata

Usage
grabTab(tfstub = "STAT1", gscoll = TFutils::tftColl,
orgdb = org.Hs.eg.db::org.Hs.eg.db,
gwrngs = TFutils::gwascat_hg19_chr17)

Arguments
tfstub character(1) gene-like symbol for TF; will be grepped in names(gscoll)
gscoll a GSEABase GeneSetCollection
orgdb an instance of OrgDb as defined in AnnotationDbi
gwrngs a GRanges representing EBI gwascat, must have DISEASE/TRAIT, MAPPED_GENE

Value
data.frame instance
Note

This function will link together information on targets of a given TF to the GWAS catalog.

Examples

```r
gt = grabTab("VDR", gscoll=TFutils::tftColl,
             orgdb=org.Hs.eg.db::org.Hs.eg.db, gwrngs=TFutils::gwascat_hg19_chr17)
dim(gt)
head(gt)
```

```r
gwascat_hg19_chr17
gwascat_hg19: GRanges of march 21 2018 EBI gwascat, limit to chr17
```

Description

`gwascat_hg19`: GRanges of march 21 2018 EBI gwascat, limit to chr17

Usage

`gwascat_hg19_chr17`

Format

GenomicRanges GRanges instance

Source

gwascat::makeCurrentGwascat, with gwascat:::lo38to19 applied

Examples

```r
TFutils::gwascat_hg19_chr17[,1:5]
```

HGNCmap

simple accessor for HGNCmap component of TFCatalog

Description

simple accessor for HGNCmap component of TFCatalog

Usage

`HGNCmap(x)`

Arguments

x instance of TFCatalog
Value
dataframe instance

Examples
HGNCmap

hocomoco.mono  hocomoco.mono: data.frame with information on HOCOMOCO TFs for human

Description
hocomoco.mono: data.frame with information on HOCOMOCO TFs for human

Usage
hocomoco.mono

Format
data.frame

Note
Extracted March 2018

Source

Examples
head(TFutils::hocomoco.mono)

hocomoco.mono.sep2018  hocomoco.mono.sep2018: data.frame with information on HOCOMOCO TFs for human, Sept 2018 download

Description
hocomoco.mono.sep2018: data.frame with information on HOCOMOCO TFs for human, Sept 2018 download

Usage
hocomoco.mono.sep2018

Format
data.frame
importFIMO, TabixFile, GRanges-method

import a FIMO bed-like file
importFrom utils read.delim

Description

import a FIMO bed-like file
importFrom utils read.delim

Usage

## S4 method for signature 'TabixFile,GRanges'
importFIMO(src, parms, ...)

## S4 method for signature 'character,missing'
importFIMO(src, parms, ...)

Arguments

src 
TabixFile instance

parms 
a GRanges instance delimiting the import; multiple GRanges can be used

... 
passed to GenomicRanges::GRanges

Value

instance of GRanges

Examples

if (requireNamespace("Rsamtools")) {
  tf = Rsamtools::TabixFile(system.file("M5946_1/chr1.bed.gz", package="TFutils"))
  importFIMO(tf, GenomicRanges::GRanges("chr1", IRanges::IRanges(1e6,1le6)))
}
# importFIMO_local_split

**utility to read FIMO outputs from local resource(cluster), assuming bed text split by chromosome**

## Description

utility to read FIMO outputs from local resource(cluster), assuming bed text split by chromosome

## Usage

```r
importFIMO_local_split(tf, chr)
```

## Arguments

- **tf**: character(1) file id
- **chr**: character(1) chromosome name

## Value

data.table instance

## Examples

```r
requireNamespace("GenomicRanges")
requireNamespace("IRanges")
importFIMO_local_split("M5946_1", "chr1")
dim(importFIMO_local_split("M5946_1", "chr17"))
```

---

lambert_snps

**lambert_snps is Table S3 of Lambert et al PMID 29425488**

## Description

lambert_snps is Table S3 of Lambert et al PMID 29425488

## Usage

```r
lambert_snps
```

## Format

data.frame

## Examples

```r
head(lambert_snps)
```
metadata_tf

**Description**

metadata_tf: list with metadata (motif_id and hgnc_symbol) about all the CISBP FIMO scan TF bed files

**Usage**

metadata_tf

**Format**

list

**Source**

K. Glass ran FIMO

**Examples**

TFutils::metadata_tf

---

named_tf

**Description**

named_tf: named list with the names being the hgnc_symbol of the motif_id

**Usage**

named_tf

**Format**

list

**Source**

K. Glass ran FIMO

**Examples**

TFutils::named_tf
	named_tf["VDR"]
**retrieve_lambert_main**

acquire the Excel spreadsheet content for table S1 of Lambert et al. Cell 2018, "The Human Transcription Factors"

**Description**

acquire the Excel spreadsheet content for table S1 of Lambert et al. Cell 2018, "The Human Transcription Factors"

**Usage**

`retrieve_lambert_main(cache = BiocFileCache::BiocFileCache(ask = FALSE))`

**Arguments**

- **cache**
  a BiocFileCache instance

**Value**

a tbl_df

**Note**

This will download the spreadsheet if not found in cache.

**Examples**

```r
if (interactive()) retrieve_lambert_main()
```

**seqinfo_hg19_chr17**

a Seqinfo instance for a chr17 in hg19

**Description**

a Seqinfo instance for a chr17 in hg19

**Usage**

`seqinfo_hg19_chr17`

**Format**

a Seqinfo instance

**Examples**

`seqinfo_hg19_chr17`
setupHIZE process a gene_attribute_matrix.txt file from harmonizeome into a GeneSetCollection

Description
process a gene_attribute_matrix.txt file from harmonizeome into a GeneSetCollection

Usage
setupHIZE(txtfn = "gene_attribute_matrix.txt", tag)

Arguments
txtfn character(1) path to gene_attribute_matrix.txt file from harmonizeome
tag character(1) will be added to shortDescription field of each GeneSet instance

Note
After uncompressing content of http://amp.pharm.mssm.edu/static/hdfs/harmonizome/data/cheappi/gene_attribute_matrix.txt.gz run this on gene_attribute_matrix.txt with tag="CHEA".

show,TFCatalog-method produce a concise report on TFCatalog instance

Description
produce a concise report on TFCatalog instance

Usage
## S4 method for signature 'TFCatalog'
show(object)

Arguments
object instance of TFCatalog

Value
side effect
**TFCatalog**  
*Constructor for TFCatalog*

**Description**

Constructor for TFCatalog

**Usage**

```
TFCatalog(name, nativeIds, HGNCmap, metadata)
```

**Arguments**

- `name` informative character(1) for collection
- `nativeIds` character() vector of identifiers used by collection creators
- `HGNCmap` data.frame with column 1 nativeIds, column 2 HGNC or hgnc.heur for MSigDb and any other columns of use
- `metadata` a list of metadata elements

**Value**

instance of TFCatalog

**Examples**

```r
if (require("GSEABase")) {
  TFs_MSIG = TFCatalog(name="MsigDb.TFT",nativeIds=names(TFutils::tftColl),
                      HGNCmap=data.frame(TFutils::tftCollMap,stringAsFactors=FALSE))
  TFs_MSIG
}
```

**TFCatalog-class**  
*define a structure to hold information about TFs from diverse reference sources*

**Description**

define a structure to hold information about TFs from diverse reference sources

**Slots**

- `name` character
- `nativeIds` character tokens used by the provider to enumerate transcription factors
- `HGNCmap` data.frame with at least two columns, native id as first column and HGNC symbol as second column
- `metadata` ANY
Note

This class respects the notions that 1) a source of information about transcription factors should have a name, 2) each source has its own ‘native’ nomenclature for the factors themselves, 3) it is common to use the gene symbol to refer to the transcription factor, and 4) additional metadata will frequently be required to establish information about provenance of assertions about transcription factors.

tffamCirc.plot

use a radial plot (by default) for motif stack

Description

use a radial plot (by default) for motif stack

Usage

tffamCirc.plot(motiflist, circosParms = defaultCircosParms())

Arguments

motiflist a list of pfm instances from motifStack

circosParms a list of parameter settings for circos plot

Value

side effect to graphics device

Examples

p1 = tffamCirc.prep()
tffamCirc.plot(p1[c(1:8, 10:17, 19)])

tffamCirc.prep

set up list of pfms in motifStack protocol

Description

set up list of pfms in motifStack protocol

Usage

tffamCirc.prep(tffam = "Paired-related HD factors{3.1.3}", trimfac = 0.4)

Arguments

tffam character(1) name of TF family as found in TFutils::hocomoco.mono field TF family

trimfac fraction passed as parameter t to motifStack::trimMotif
**tfhash**

**Value**

a list of pfm instances as defined in motifStack

**Note**

Uses MotifDb, motifStack to create a list of pfms

**Examples**

```r
n1 = tffamCirc.prep()
str(n1)
```

---

**Description**

tfhash: data.frame with MSigDb TFs, TF targets as symbol or ENTREZ

**Usage**

tfhash

**Format**

list

**Source**

MSigDb "c3" (motif gene sets) has been harvested for simple annotation of TFs and targets.

**Examples**

```r
TFutils::tfhash
tfhash[1:3,]
```

---

**TFtargs**

gadget to help sort through tags naming TFs

**Description**

gadget to help sort through tags naming TFs

**Usage**

```r
TFtargs(gscoll = TFutils::tftColl, initTF = "VDR.Q3",
gwcat = TFutils::gwascat_hg19_chr17,
gadtitle = "Search for a TF; its targets will be checked for mapped status in GWAS catalog")
```
**Arguments**

- `gscoll` a GSEABase GeneSetCollection
- `initTF` character(1) initial TF string for app
- `gwcat` GRanges-like structure with GWAS catalog information
- `gadtitle` character(1) a title for the gadget panel

**Value**

on app conclusion a data.frame is returned

**Note**

Will use TFutils::gwascat_hg19_chr17 to look for 'MAPPED_GENE' field entries matching targets, also hardcoded to use org.Hs.eg.db to map symbols

**Examples**

```r
if (interactive()) Tftargs()
```

---

**`tftColl`**

**GSEABase GeneSetCollection for transcription factor targets**

**Description**

tftColl: GSEABase GeneSetCollection for transcription factor targets

**Usage**

tftColl

**Format**

GSEABase GeneSetCollection instance

**Note**

run GSEABase::getGMT() on c3/TFT geneset collection from MSigDb

**Source**

broad institute

**Examples**

TFutils::tftColl
tftCollMap

**tftCollMap**

data.frame with information on MSigDb TFs for human

**Description**

tftCollMap: data.frame with information on MSigDb TFs for human

**Usage**

tftCollMap

**Format**

data.frame

**Note**

Annotation of TFs is ad-hoc. GeneSet names were tokenized, splitting by underscore, and then fragments were matched to SYMBOL and ALIAS elements of org.Hs.eg.db. Extracted March 2018

**Source**

http://software.broadinstitute.org/gsea/msigdb/genesets.jsp?collection=TFT

**Examples**

head(TFutils::tftCollMap)

topTraitsOfTargets

**Description**

Use MSigDB TF targets resource to find targets of input TF and find traits to which these targets have been mapped

**Usage**

topTraitsOfTargets(TFsym, gsc, gwcat, ntraits = 6, force = FALSE, ...)

**Arguments**

- **TFsym**
  character(1) symbol for a TF must be present in tftCollMap[, "hgnc.heur"]

- **gsc**
  an instance of GeneSetCollection-class, intended to enumerate targets of a single transcription factor in each GeneSet, as in TFutils::tftColl

- **gwcat**
  instance of gwaswloc-class

- **ntraits**
  numeric(1) number of traits to report

- **force**
  logical see note, set to true if you want to skip mapping from TFsym to a specific motif or TF identifier used as name of a GeneSet in gsc

- **...**
  character() vector of fields in mcols(gwcat) to include
Note

If tftCollMap[, "hgnc.heur"] does not possess the necessary symbol, set force = TRUE to use a known ’motif’ name among names(gsc)

Examples

suppressPackageStartupMessages({
  library(GSEABase)
  library(TFutils)
}) # more results if you substitute ebicat37 from gwascat below
topTraitsOfTargets("MTF1", tftColl, gwascat_hg19_chr17)

URL_s3_tf

utility to generate link to biocfound bucket for FIMO TFBS scores

Description

utility to generate link to biocfound bucket for FIMO TFBS scores

Usage

URL_s3_tf(tag = "M3433")

Arguments

tag character(1) token identifying TF, can be an HGNC gene name or Mnnnn PWM tag. It must be findable in TFutils::fimoMap table.

Value

character(1) URL

Examples

URL_s3_tf
Index

*Topic datasets
  cisbpTFcat, 4
  cisbpTFcat_2.0, 5
  demo_fimo_granges, 6
  encode690, 7
  fimo16, 7
  fimoMap, 8
  gwascat_hg19_chr17, 12
  hocomoco.mono, 13
  hocomoco.mono.sep2018, 13
  lambert_snps, 15
  metadata_tf, 16
  named_tf, 16
  seqinfo_hg19_chr17, 17
  tfhash, 21
  tftColl, 22
  tftCollMap, 23
anchor_pmids, 2
browse_lambert_gwaslinks, 3
browse_lambert_main, 4
  cisbpTFcat, 4
  cisbpTFcat_2.0, 5
  defaultCircosParms, 5
  demo_fimo_granges, 6
  directHitsInCISBP, 6
  encode690, 7
  fimo16, 7
  fimo_granges, 8
  fimoMap, 8
  genemodelDF, 9
  genemodForGviz, 10
  get_rslocs_38, 11
  grabTab, 11
  gwascat_hg19_chr17, 12
  HGNCmap, 12
  hocomoco.mono, 13
  hocomoco.mono.sep2018, 13
importFIMO
  (importFIMO,TabixFile,GRanges-method), 14
importFIMO,character,missing-method
  (importFIMO,TabixFile,GRanges-method), 14
importFIMO,TabixFile,GRanges-method, 14
importFIMO_local_split, 15
lambert_snps, 15
metadata_tf, 16
named_tf, 16
retrieve_lambert_main, 17
seqinfo_hg19_chr17, 17
setupHIZE, 18
show,TFCatalog-method, 18
TFCatalog, 19
TFCatalog-class, 19
tffamCirc.plot, 20
tffamCirc.prep, 20
tfhash, 21
Tftargs, 21
tftColl, 22
tftCollMap, 23
topTraitsOfTargets, 23
URL_s3_tf, 24