

# Package ‘TFutils’

May 26, 2026

**Title** TFutils

**Description** This package helps users to work with TF metadata from various sources. Significant catalogs of TFs and classifications thereof are made available. Tools for working with motif scans are also provided.

**Version** 1.32.0

**Depends** R (>= 4.1.0)

**Imports** methods, dplyr, magrittr, miniUI, shiny, Rsamtools, GSEABase, rjson, BiocFileCache, DT, httr, readxl, AnnotationDbi, org.Hs.eg.db, utils, GenomicFiles, SummarizedExperiment

**Suggests** knitr, data.table, testthat, AnnotationFilter, Biobase, GenomicFeatures, GenomicRanges, Gviz, IRanges, S4Vectors, EnsDb.Hsapiens.v75, BiocParallel, BiocStyle, GO.db, Seqinfo, UpSetR, ggplot2, png, gwascat, MotifDb, motifStack, RColorBrewer, rmarkdown

**License** Artistic-2.0

**LazyLoad** true

**ByteCompile** true

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**biocViews** Transcriptomics

**Roxygen** list(markdown = TRUE)

**Encoding** UTF-8

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

**git\_branch** RELEASE\_3\_23

**git\_last\_commit** ac50301

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-05-25

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

**Maintainer** Vincent Carey <stvjc@channing.harvard.edu>

## Contents

anchor_pmids . . . . .	3
browse_gotf_main . . . . .	4
browse_humantfs_main . . . . .	4
browse_lambert_gwaslinks . . . . .	5
browse_lambert_main . . . . .	5
cabr_cell_url . . . . .	6
cisbpTFcat . . . . .	6
cisbpTFcat_2.0 . . . . .	7
defaultCircosParms . . . . .	7
demo_fimo_granges . . . . .	8
directHitsInCISBP . . . . .	8
encode690 . . . . .	9
fimo16 . . . . .	9
fimoMap . . . . .	10
fimo_granges . . . . .	10
genemodelIDF . . . . .	11
genemodForGviz . . . . .	12
get_rslocs_38 . . . . .	13
gotf_url . . . . .	13
grabTab . . . . .	14
gwascat_hg19_chr17 . . . . .	15
HGNcmap . . . . .	15
hocomoco.mono . . . . .	16
hocomoco.mono.sep2018 . . . . .	16
importFIMO,TabixFile,GRanges-method . . . . .	17
importFIMO_local_split . . . . .	18
lambert_snps . . . . .	18
metadata_tf . . . . .	19
named_tf . . . . .	19
retrieve_gotf_main . . . . .	20
retrieve_humantfs_main . . . . .	20
retrieve_lambert_main . . . . .	21
seqinfo_hg19_chr17 . . . . .	22
setupHIZE . . . . .	22
show,TFCatalog-method . . . . .	23
TFCatalog . . . . .	23
TFCatalog-class . . . . .	24
tffamCirc.plot . . . . .	24
tffamCirc.prep . . . . .	25
tffhash . . . . .	26
Tftargs . . . . .	26

<i>anchor_pmids</i>	3
tftColl . . . . .	27
tftCollMap . . . . .	28
topTraitsOfTargets . . . . .	28
URL_s3_tf . . . . .	29
<b>Index</b>	<b>30</b>

---

<i>anchor_pmids</i>	<i>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</i>
---------------------	---

---

### Description

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

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

---

browse\_gotf\_main      *use DT::datatable to browse the GO catalogue of human DNA-binding transcription factors in Table S1.A of Lovering et al.*

---

**Description**

use DT::datatable to browse the GO catalogue of human DNA-binding transcription factors in Table S1.A of Lovering et al.

**Usage**

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

**Arguments**

cache                  a BiocFileCache instance

**Value**

result of DT::datatable

**Examples**

```
if (interactive()) browse_gotf_main()
```

---

browse\_humantfs\_main      *use DT::datatable to browse the Lambert's Human Transcription Factors repository*

---

**Description**

use DT::datatable to browse the Lambert's Human Transcription Factors repository

**Usage**

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

**Arguments**

cache                  a BiocFileCache instance

**Value**

result of DT::datatable

**Examples**

```
if (interactive()) browse_humantfs_main()
```

---

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()
```

**Value**

DT::datatable

**Note**

Based on supplemental table S4 of PMID 29425488

**Examples**

```
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**

```
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**

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

---

```
ccbr_cell_url
```

```
Update Oct 18 2021
```

---

**Description**

```
Update Oct 18 2021
```

**Usage**

```
ccbr_cell_url()
```

---

```
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**

```
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.
```

**Source**

```
http://cisbp.ccb.utoronto.ca/bulk.php select Homo_sapiens
```

**Examples**

```
head(TFutils::cisbpTFcat)
```

---

cisbpTFcat_2.0	<i>cisbpTFcat_2.0: data.frame with information on CISBP TFs for human, described in PMID 31133749</i>
----------------	---

---

**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.ccbbr.utoronto.ca/bulk.php> select Homo\_sapiens

**Examples**

```
head(TFutils::cisbpTFcat_2.0)
```

---

defaultCircosParms	<i>basic layout parameters for circos</i>
--------------------	---

---

**Description**

basic layout parameters for circos

**Usage**

```
defaultCircosParms()
```

**Value**

a list

**Examples**

```
head(defaultCircosParms())
```

---

demo_fimo_granges	<i>a list of GRanges instances with TF FIMO scores returned by fimo_granges</i>
-------------------	---

---

**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**

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

---

directHitsInCISBP	<i>demonstrate interoperation of TF catalog with GWAS catalog</i>
-------------------	---

---

**Description**

demonstrate interoperation of TF catalog with GWAS catalog

**Usage**

```
directHitsInCISBP(traitTag, gwascats)
```

**Arguments**

traitTag	character(1) string found in DISEASE/TRAIT field of gwascats instance
gwascats	instance of <a href="#">gwaswloc-class</a>

**Value**

data.frame

**Examples**

```
data(gwascats_hg19_chr17)
directHitsInCISBP("Prostate cancer" , gwascats_hg19_chr17)
```

---

encode690	<i>encode690: DataFrame extending AnnotationHub metadata about ENCODE cell line x TF ranges</i>
-----------	---

---

**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	<i>fimo16: GenomicFiles instance to AWS S3-resident FIMO bed for 16 TFs</i>
--------	---

---

**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	<i>fimoMap: table with Mnnnn (motif PWM tags) and HGNC symbols for TFs</i>
---------	--

---

**Description**

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

**Usage**

```
fimoMap
```

**Format**

```
data.frame
```

**Source**

Kimberly Glass (rekr@channing.harvard.edu)

**Examples**

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

---

fimo_granges	<i>create a list of GRanges for FIMO hits in a GenomicFiles instance, corresponding to a GRanges-based query</i>
--------------	--

---

**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**

```
if (interactive()) { # need internet
  # setup -- annotate fimo16 object and create an informative
  # query
  colnames(fimo16) = fimo16$HGNC
  si = Seqinfo::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	<i>use EnsDb to generate an exon-level model of genes identified by symbol</i>
-------------	--

---

**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**

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

---

genemodForGviz	<i>create a GeneRegionTrack instance for selected symbols</i>
----------------	---

---

**Description**

create a GeneRegionTrack instance for selected symbols

**Usage**

```
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**

```
if (requireNamespace("EnsDb.Hsapiens.v75") &
    requireNamespace("Gviz")) {
  orm = genemodForGviz("ORMDL3", resource= EnsDb.Hsapiens.v75::EnsDb.Hsapiens.v75)
  orm
  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

**Value**

GRanges instance

**Note**

Uses [rest.ensembl.org](https://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**

```
if (interactive()) get_rslocs_38() # see https://stat.ethz.ch/pipermail/bioc-devel/2020-October/017263.html
```

---

gotf\_url                      *Update Oct 18 2021*

---

**Description**

Update Oct 18 2021

**Usage**

```
gotf_url()
```

---

grabTab	<i>create table of TF targets and related metadata</i>
---------	--

---

### 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

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

---

gwascat_hg19_chr17	<i>gwascat_hg19: GRanges of march 21 2018 EBI gwascat, limit to chr17</i>
--------------------	---

---

**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**

TFutils::gwascat\_hg19\_chr17[,1:5]

---

HGNCmap	<i>simple accessor for HGNCmap component of TFCatalog</i>
---------	---

---

**Description**

simple accessor for HGNCmap component of TFCatalog

**Usage**

HGNCmap(x)

**Arguments**

x                   instance of TFCatalog

**Value**

dataframe instance

**Examples**

HGNCmap

---

hocomoco.mono	<i>hocomoco.mono: data.frame with information on HOCOMOCO TFs for human</i>
---------------	---

---

**Description**

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

**Usage**

```
hocomoco.mono
```

**Format**

```
data.frame
```

**Note**

Extracted March 2018

**Source**

<http://hocomoco11.autosome.ru/human/mono?full=true>

**Examples**

```
head(TFutils::hocomoco.mono)
```

---

hocomoco.mono.sep2018	<i>hocomoco.mono.sep2018: data.frame with information on HOCOMOCO TFs for human, Sept 2018 download</i>
-----------------------	---

---

**Description**

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

**Usage**

```
hocomoco.mono.sep2018
```

**Format**

```
data.frame
```

**Note**

Extracted September 2018

**Source**

<http://hocomoco11.autosome.ru/human/mono?full=true>

**Examples**

```
head(TFutils::hocomoco.mono.sep2018)
```

---

```
importFIMO, TabixFile, GRanges-method
  import a FIMO bed-like file
```

---

**Description**

import a FIMO bed-like file

**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, 11e6)))
}
```

```
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**

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

**Arguments**

tf	character(1) file id
chr	character(1) chromosome name

**Value**

data.table instance

**Examples**

```
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**

```
lambert_snps
```

**Format**

data.frame

**Examples**

```
head(lambert_snps)
```

---

metadata_tf	<i>metadata_tf: list with metadata (motif_if and hgnc_symbol) about all the CISBP FIMO scan TF bed files</i>
-------------	--

---

**Description**

metadata\_tf: list with metadata (motif\_if 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	<i>named_tf: named list with the names being the hgnc_symbol of the motif_id</i>
----------	--

---

**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_gotf_main    acquire the content of Table S1.A from Lovering et al., A
                       GO catalogue of human DNA-binding transcription factors, DOI:
                       https://doi.org/10.1101/2020.10.28.359232
```

---

**Description**

acquire the content of Table S1.A from Lovering et al., A GO catalogue of human DNA-binding transcription factors, DOI: <https://doi.org/10.1101/2020.10.28.359232>

**Usage**

```
retrieve_gotf_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**

```
if (interactive()) retrieve_gotf_main()
```

---

```
retrieve_humantfs_main    acquire the CSV content for table S1 of Lambert et al. Cell 2018 from
                           the Human TFS repository at http://humantfs.cabr.utoronto.ca
```

---

**Description**

acquire the CSV content for table S1 of Lambert et al. Cell 2018 from the Human TFS repository at <http://humantfs.cabr.utoronto.ca>

**Usage**

```
retrieve_humantfs_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**

```
if (interactive()) retrieve_humantfs_main()
```

---

```
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**

```
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

**Value**

GSEABase::GeneSetCollection

**Note**

After uncompressing content of [http://amp.pharm.mssm.edu/static/hdfs/harmonizome/data/cheappi/gene\\_attribute\\_matrix.txt.gz](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**

```

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

```

---

TFCatalog-class	<i>define a structure to hold information about TFs from diverse reference sources</i>
-----------------	--

---

**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 atleast 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 transctiption factor, and 4) additional metadata will frequently be required to establish information about provenance of assertions about transcription factors.

---

tffamCirc.plot	<i>use a radial plot (by default) for motif stack</i>
----------------	---

---

**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	<i>set up list of pfms in motifStack protocol</i>
----------------	---

---

**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

**Value**

a list of pfm instances as defined in motifStack

**Note**

Uses MotifDb, motifStack to create a list of pfms

**Examples**

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

---

tfhash	<i>tfhash</i> : data.frame with MSigDb TFs, TF targets as symbol or ENTREZ
--------	--

---

**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**

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

---

TFtargs	<i>gadget to help sort through tags naming TFs</i>
---------	--

---

**Description**

gadget to help sort through tags naming TFs

**Usage**

```
TFtargs(
  gscoll = TFutils::tftColl,
  initTF = "VDR_Q3",
  gwcat = TFutils::gwascat_hg19_chr17,
  gadttitle =
    "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
gwcats	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**

```
if (interactive()) TFtargs()
```

---

tftColl	<i>tftColl: GSEABase GeneSetCollection for transcription factor targets</i>
---------	---

---

**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	<i>tftCollMap: data.frame with information on MSigDb TFs for human</i>
------------	--

---

**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	<i>Use MSigDB TF targets resource to find targets of input TF and find traits to which these targets have been mapped</i>
--------------------	---

---

**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 <a href="#">GeneSetCollection-class</a> , intended to enumerate targets of a single transcription factor in each GeneSet, as in TFutils::tftColl
gwcatt	instance of <a href="#">gwaswloc-class</a>
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(gwcatt) to include

**Value**

data.frame symbol, set force = TRUE to use a known 'motif' name among names(gsc)

**Note**

If tftCollMap[, "hgnc.heur"] does not possess the necessary

**Examples**

```
suppressPackageStartupMessages({
  library(GSEABase)
}) # 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

## \* datasets

- cisbpTFcat, [6](#)
- cisbpTFcat\_2.0, [7](#)
- demo\_fimo\_granges, [8](#)
- encode690, [9](#)
- fimo16, [9](#)
- fimoMap, [10](#)
- gwascat\_hg19\_chr17, [15](#)
- hocomoco.mono, [16](#)
- hocomoco.mono.sep2018, [16](#)
- lambert\_snps, [18](#)
- metadata\_tf, [19](#)
- named\_tf, [19](#)
- seqinfo\_hg19\_chr17, [22](#)
- tffhash, [26](#)
- tftColl, [27](#)
- tftCollMap, [28](#)

[anchor\\_pmids, 3](#)

[browse\\_gotf\\_main, 4](#)  
[browse\\_humantfs\\_main, 4](#)  
[browse\\_lambert\\_gwaslinks, 5](#)  
[browse\\_lambert\\_main, 5](#)

[ccbr\\_cell\\_url, 6](#)  
[cisbpTFcat, 6](#)  
[cisbpTFcat\\_2.0, 7](#)

[defaultCircosParams, 7](#)  
[demo\\_fimo\\_granges, 8](#)  
[directHitsInCISBP, 8](#)

[encode690, 9](#)

[fimo16, 9](#)  
[fimo\\_granges, 10](#)  
[fimoMap, 10](#)

[genemodelDF, 11](#)  
[genemodForGviz, 12](#)

[get\\_rslocs\\_38, 13](#)  
[gotf\\_url, 13](#)  
[grabTab, 14](#)  
[gwascat\\_hg19\\_chr17, 15](#)

[HGNCmap, 15](#)  
[hocomoco.mono, 16](#)  
[hocomoco.mono.sep2018, 16](#)

[importFIMO](#)  
    ([importFIMO, TabixFile, GRanges-method](#)),  
    [17](#)

[importFIMO, character, missing-method](#)  
    ([importFIMO, TabixFile, GRanges-method](#)),  
    [17](#)

[importFIMO, TabixFile, GRanges-method](#),  
    [17](#)

[importFIMO\\_local\\_split, 18](#)

[lambert\\_snps, 18](#)

[metadata\\_tf, 19](#)

[named\\_tf, 19](#)

[retrieve\\_gotf\\_main, 20](#)  
[retrieve\\_humantfs\\_main, 20](#)  
[retrieve\\_lambert\\_main, 21](#)

[seqinfo\\_hg19\\_chr17, 22](#)  
[setupHIZE, 22](#)  
[show, TFCatalog-method, 23](#)

[TFCatalog, 23](#)  
[TFCatalog-class, 24](#)  
[tffamCirc.plot, 24](#)  
[tffamCirc.prep, 25](#)  
[tffhash, 26](#)  
[TFtargs, 26](#)  
[tftColl, 27](#)  
[tftCollMap, 28](#)

topTraitsOfTargets, [28](#)

URL\_s3\_tf, [29](#)