Package ‘TFutils’

September 4, 2019

Title TFutils

Description Package to work with TF data.

Version 1.4.0

Depends R (>= 3.5.0)

Imports methods, dplyr, magrittr, miniUI, shiny

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

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_9

git_last_commit a2f8f46

git_last_commit_date 2019-05-02

Date/Publication 2019-09-03

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

Maintainer Shweta Gopaulakrishnan <reshg@channing.harvard.edu>
R topics documented:

- cisbpTFcat: data.frame with information on CISBP TFs for human

Description

cisbpTFcat: data.frame with information on CISBP TFs for human

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.
**directHitsInCISBP**

**Source**


**Examples**

```r
demonstrate interoperation of TF catalog with GWAS catalog
```

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

**Usage**

```r
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**

```r
data(gwascat_hg19_chr17)
directHitsInCISBP("Prostate cancer", gwascat_hg19_chr17)
```

---

**encode690**

**Description**

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

**Usage**

```r
encode690
```

**Format**

DataFrame

**Source**

see metadata(encode690)
Examples

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

---

```r
fimoMap:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fimoMap: table with Mnnn (motif PWM tags) and HGNC symbols for TFs</td>
</tr>
</tbody>
</table>

---

Usage

```r
fimo16
```

---

```r
fimoMap:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fimoMap: table with Mnnn (motif PWM tags) and HGNC symbols for TFs</td>
</tr>
</tbody>
</table>

---

Usage

```r
fimoMap
```

---

```r
fimoMap:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fimoMap: table with Mnnn (motif PWM tags) and HGNC symbols for TFs</td>
</tr>
</tbody>
</table>

---

Usage

```r
fimoMap
```

---

```r
fimoMap:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fimoMap: table with Mnnn (motif PWM tags) and HGNC symbols for TFs</td>
</tr>
</tbody>
</table>

---

Usage

```r
fimoMap
```

---

```r
fimoMap:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fimoMap: table with Mnnn (motif PWM tags) and HGNC symbols for TFs</td>
</tr>
</tbody>
</table>

---

Examples

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

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

**Usage**

```r
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

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
}

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)
```

---

**gwascat_hg19_chr17**

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

**Usage**

```r
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**

```r
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

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, 1e6)))
}
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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tf</td>
<td>character(1) file id</td>
</tr>
<tr>
<td>chr</td>
<td>character(1) chromosome name</td>
</tr>
</tbody>
</table>

Value
data.table instance

Examples

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

metadata_tf

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

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

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

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"]]

setupHIZE

process a gene_attribute_matrix.txt file from harmonizome into a GeneSetCollection

Description

process a gene_attribute_matrix.txt file from harmonizome into a GeneSetCollection

Usage

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

Arguments

txtfn character(1) path to gene_attribute_matrix.txt file from harmonizome
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
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.

---

**tfhash**

*tfhash: 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**

*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**

*tftColl: GSEABase GeneSetCollection for transcription factor targets*

**Description**

tftColl: GSEABase GeneSetCollection for transcription factor targets

**Usage**

tftColl

**Format**

GSEABase GeneSetCollection instance
**tftCollMap**

**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  

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

**Description**

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

**Usage**

```r
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**

```r
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**

```r
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, 2
- encode690, 3
- fimo16, 4
- fimoMap, 4
- gwascat_hg19_chr17, 7
- hocomoco.mono, 8
- hocomoco.mono.sep2018, 8
- metadata_tf, 10
- named_tf, 11
- tftColl, 14
- tftCollMap, 15

- cisbpTFcat, 2
- directHitsInCISBP, 3
- encode690, 3
- fimo16, 4
- fimoMap, 4
- genemodelDF, 5
- genemodForGviz, 5
- grabTab, 6
- gwascat_hg19_chr17, 7

- HGNCmap, 7
- hocomoco.mono, 8
- hocomoco.mono.sep2018, 8

- importFIMO
  - (importFIMO,TabixFile,GRanges-method), 9
- importFIMO,character,missing-method
  - (importFIMO,TabixFile,GRanges-method), 9
- importFIMO,TabixFile,GRanges-method, 9
- importFIMO_local_split, 10

- metadata_tf, 10
- named_tf, 11
- setupHIZE, 11

show,TFCatalog-method, 12

- TFCatalog, 12
- TFCatalog-class, 13
- tfhash, 13
- TFTargs, 14
- tftColl, 14
- tftCollMap, 15
- topTraitsOfTargets, 16
- URL_s3_tf, 16