

# Package ‘rtfbsdb’

April 13, 2015

**Version** 0.1.0

**Date** 2015-04-12

**Title** Parse TF motifs from public databases, read into R, and scan using 'rtfbs'.

**Author** Charles G. Danko <dankoc@gmail.com>

**Maintainer** Charles G. Danko <dankoc@gmail.com> Zhong Wang <zw355@cornell.edu>

**Depends** R (>= 2.6)

**Imports** rphast, rtfbs, bigWig, parallel, grid, cluster, methods, latticeExtra, lattice

**LinkingTo**

**Suggests** RCurl, stringr

**Description**

Convenience functions to read and scan DNA sequences using Position Weight Matrices (PWMs)

**License** GPL version 3 or newer

**biocViews** Sequencing, Analysis

**LazyLoad** yes

## R topics documented:

CisBP.db-class . . . . .	2
CisBP.download . . . . .	3
CisBP.extdata . . . . .	4
CisBP.find . . . . .	4
CisBP.group . . . . .	5
CisBP.zipload . . . . .	7
tfbs . . . . .	8
tfbs-class . . . . .	9
tfbs.clusterMotifs . . . . .	10
tfbs.compareTFsite . . . . .	11
tfbs.db-class . . . . .	12
tfbs.dirs . . . . .	13
tfbs.drawLogo . . . . .	14
tfbs.drawLogosForClusters . . . . .	15

tfbs.getDistanceMatrix . . . . .	16
tfbs.getExpression . . . . .	16
tfbs.scanTFsite . . . . .	18
tfbs.selectByGeneExp . . . . .	20
tfbs.selectByRandom . . . . .	21

<b>Index</b>	<b>23</b>
--------------	-----------

---

CisBP.db-class	Class "CisBP.db"
----------------	------------------

---

## Description

The motif library from CisBP web site.

Link: <http://cisbp.ccb.utoronto.ca/>

## Objects from the Class

Objects can be created by calls of the form `CisBP.extdata()`, `CisBP.zipload()`, `CisBP.download()`.

## Slots

**species:** String indicating the species name defined in the CisBP dataset.

**zip.file:** String indicating the filename of temporary data file.

**zip.url:** String indicating the download source.

**file.tfinfo:** String indicating the TF filename, default is TF\_Information.txt.

## Extends

Class "`tfbs.db`", directly.

## Methods

**CisBP.find** Build a `tfbs` object by querying the meta file of CisBP dataset and subsetting the results.

**CisBP.group** Get the statistical summary by grouping the field in the CisBP dataset.

## References

Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., ... & Hughes, T. R. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6), 1431-1443.

## Examples

```
showClass("CisBP.db")
```

---

CisBP.download	Download CisBP dataset.
----------------	-------------------------

---

## Description

Download TF data file from CisBP dataset and store it to temporary folder

## Usage

```
CisBP.download(species = "Homo_sapiens",  
               url = "http://cisbp.ccbbr.utoronto.ca/bulk_archive.php")
```

## Arguments

species	String, indicating the species name in the CisBP dataset
url	String, the URL of bulk downloads from CisBP dataset, default is <a href="http://cisbp.ccbbr.utoronto.ca/bulk_archive.php">http://cisbp.ccbbr.utoronto.ca/bulk_archive.php</a>

## Value

A CisBP object (class name: CisBP.db) is returned with four items:

species	String indicating the species name
zip.file	String indicating the filename of temporary data file.
zip.url	String indicating the download source
file.tfinfo	String indicating the TF filename, default is TF_Information.txt.

## References

Weirauch, M. T., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., ... & Hughes, T. R. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6), 1431-1443.

## See Also

See Also as [CisBP.zipload](#), [CisBP.extdata](#).

## Examples

```
#download human dataset  
db1 <- CisBP.download("Homo_sapiens");  
  
#download mouse dataset  
db2 <- CisBP.download("Mus_musculus");
```

---

CisBP.extdata	<i>Load internal CisBP dataset.</i>
---------------	-------------------------------------

---

**Description**

Build a CisBP object from the internal zip file stored in this package

**Usage**

```
CisBP.extdata(species)
```

**Arguments**

species	String, only valid for human and mouse species, i.e. Homo_sapiens or Mus_musculus
---------	-----------------------------------------------------------------------------------

**Value**

A CisBP object (class name: CisBP.db) is returned with four items:

species	String indicating the species name defined in the CisBP dataset.
zip.file	String indicating the filename of temporary data file.
zip.url	String indicating the download source
file.tfinfo	String indicating the TF filename, default is TF_Information.txt.

**See Also**

See Also as [CisBP.zipload](#), [CisBP.download](#).

**Examples**

```
#reading data from inner file
db.human <- CisBP.extdata("Homo_sapiens")
```

---

CisBP.find	<i>Query the CisBP dataset.</i>
------------	---------------------------------

---

**Description**

Build a tfbs object by querying the meta file of CisBP dataset and subsetting the results.

**Usage**

```
CisBP.find(cisbp.db,
  tf_name = NULL,
  tf_status = NULL,
  family_name = NULL,
  motif_type = NULL,
  msource_id = NULL,
  motif_info_type = 1)
```

**Arguments**

<code>cisbp.db</code>	A CisBP object, including the file of TF_Information.txt.
<code>tf_name</code>	String, indicting the TF_name field will be used to select motifs.
<code>tf_status</code>	String, indicting the TF_Status field will be used to select motifs.
<code>family_name</code>	String, indicting the Family_Name field will be used to select motifs.
<code>motif_type</code>	String, indicting the Motif_Type field will be used to select motifs.
<code>msource_id</code>	String, indicting the MSource_Identifier field will be used to select motifs.
<code>motif_info_type</code>	Number,indicating which TF meta file will be used. Available values are 1 for TF_Information.txt, 2 for TF_Information_all_motifs.txt and 3 for F_Information_all_motifs_plus.txt.

**Value**

A tfbs object is returned with PWM matrices, see Also as [tfbs](#)

**See Also**

See Also as [tfbs](#)

**Examples**

```
# Load the internal CisBP dataset
db_human <- CisBP.extdata("Homo_sapiens");

# Load all motifs and return a tfbs object.
tfs0 <- CisBP.find(db_human);

# Query the motifs by the conditins and return a tfbs object
tfs1 <- CisBP.find(db_human, family_name="Homeodomain", tf_status="D",
  motif_type="ChIP-seq", msource_id= "MS01_1.01", motif_info_type=1 );

# Query the motifs by the conditins and return a tfbs object
tfs2 <- CisBP.find(db_human, family_name="Homeodomain", tf_status="D", motif_info_type=1);

# Query the motifs by the conditins and return a tfbs object
tfs3 <- CisBP.find(db_human, motif_type="ChIP-seq", motif_info_type=1 );

# Query the motifs by the conditins and return a tfbs object
tfs4 <- CisBP.find(db_human, motif_info_type=2);
```

---

CisBP.group

*Summarize the motif number.*


---

**Description**

Get the statistical summary by grouping the field in the CisBP dataset.

**Usage**

```
CisBP.group(cisbp.db,
            group_by=c("tf_name", "tf_species", "tf_status", "family_name",
                       "motif_type", "msource_id"),
            motif_info_type=1)
```

**Arguments**

<code>cisbp.db</code>	A CisBP object including the TF_Information.txt.
<code>group_by</code>	String, indicating which field will be used to group values. Available values are <code>tf_name</code> , <code>tf_species</code> , <code>tf_status</code> , <code>family_name</code> , <code>motif_type</code> and <code>msource_id</code> .
<code>motif_info_type</code>	Number, indicating which TF meta file will be used. Available values are 1 for TF_Information.txt, 2 for TF_Information_all_motifs.txt and 3 for F_Information_all_motifs_plus.txt.

**Details**

Three TF information files in CisBP dataset.

- 1: TF\_Information.txt : (direct motifs) or (no direct but inferred motifs with 90%)
- 2: TF\_Information\_all\_motifs.txt: (direct motifs) and (inferred motifs above the threshold)
- 3: F\_Information\_all\_motifs\_plus.txt: All motifs

**Value**

A data frame returned includes two columns

<code>group_by</code>	Values of grouping field
<code>number</code>	Counts of group value

**See Also**

See Also as [CisBP.find](#)

**Examples**

```
# Load the internal CisBP dataset
db_human <- CisBP.extdata("Homo_sapiens");

# Group the motif count by the column of family_name in TF_Information.txt
gr1 <- CisBP.group(db_human, group_by="family_name", motif_info_type=1 );

# Group the motif count by the column of tf_status in TF_Information.txt
gr2 <- CisBP.group(db_human, group_by="tf_status", motif_info_type=1 );

# Group the motif count by the column of tf_status in TF_Information_all_motifs.txt
gr3 <- CisBP.group(db_human, group_by="tf_status", motif_info_type=2);

# Group the motif count by the column of tf_status in F_Information_all_motifs_plus.txt
gr4 <- CisBP.group(db_human, group_by="tf_status", motif_info_type=3);
```

---

CisBP.zipload	<i>Load the zipped CisBP file.</i>
---------------	------------------------------------

---

## Description

Build a CisBP object from the zipped CisBP file.

## Usage

```
CisBP.zipload(zip.file, species = "Homo_sapiens")
```

## Arguments

zip.file	String, indicating the zipped file data
species	String, indicating the species name in the CisBP database

## Value

A CisBP object (class name: CisBP.db) is returned with four items:

species	String indicating the species name
zip.file	String indicating the filename of temporary data file.
zip.url	String indicating the download source
file.tfinfo	String indicating the TF filename, default is TF_Information.txt.

## See Also

See Also as [CisBP.extdata](#), [CisBP.download](#).

## Examples

```
# Download the dataset
db2 <- CisBP.download("Mus_musculus");

# Loading the zip file, the db2 and db3 have same TF data.
# Here is an example to show how to use CisBP.zipload.
# We don't need to download it by CisBP.download and then load it by CisBP.zipload
db3 <- CisBP.zipload(db2@zip.file, species="Mus_musculus");
```

---

tfbs

---

Create a tfbs object from the supplied PWM files.

---

## Description

Create a tfbs object from the supplied PWM files.

## Usage

```
tfbs(filenamees,
      names,
      species="Homo_sapiens",
      extra_info = NULL, ...)
```

## Arguments

filenamees	Vector of PWM files
names	Vector of unique gene symbols.
species	String indicating species name
extra_info	Data frame including meta information for all motifs., Default: NULL
...	Parameters,such as pseudocount, force_even, and the parameters used in read.table function.

## Value

A tfbs object (class name: tfbs) including all PWM matrices.The all attributes are as follows:

TFID	Vector of non-unique ID for TF.
species	String indicating the species name
ntfs	Number of motifs in matrix.
pwm	A list including PWM matics.
filename	Vector of PWM filename.
mgisymbols	Unique gene symbols for TF.
extra_info	Data frame, including extra information for PWMs, it maybe different with motif dataset, default:NULL.
distancematrix	Distance matrix between motifs returned by <code>tfbs.getDistanceMatrix</code> , default:NULL.
expressionlevel	Data frame indicatig the result of expression level returned by <code>tfbs.getExpression</code> , default:NULL.

The tfbs object can be created by the function of `tfbs`, `tfbs.dir`, `CisBP.find`.



## Examples

```
# M3590_1.01 PAX5 ENSG00000196092
# M3590_1.01 PAX5 ENSG00000196092
fs1 <- system.file("extdata", "M3590_1.01.pwm", package="rtfbsdb")
fs2 <- system.file("extdata", "M3591_1.01.pwm", package="rtfbsdb")

cat(fs1, "\n");

tfs <- tfbs( c( fs1, fs2 ), names=c("M3590_1.01", "M3591_1.01"),
             header=TRUE, sep="\t" , row.names=1 );
str(tfs);
```

---

tfbs-class	<i>Class "tfbs"</i>
------------	---------------------

---

## Description

Tfbj object is a collection of motif PWM data. Some functions are provided based on the PWM and GENCODE data, such as clustering, search and compare.

## Objects from the Class

Objects can be created by calls of the function of `CisBP.find`, `tfbs.dirs` and `tfbs`.

## Slots

**TFID** Vector of non-unique ID for TF.

**species** String indicating the species name

**ntfs** Number of motifs in matrix.

**pwm** A list including PWM matics.

**filename** Vector of PWM filename.

**mgisymbols** Unique gene symbols for TF.

**extra\_info** Data frame, including extra information for PWMs, it maybe different with motif dataset, default:NULL.

**distancematrix** Distance matrix between motifs returned by `tfbs.getDistanceMatrix`, default:NULL.

**expressionlevel** Data frame indicatig the result of expression level returned by `tfbs.getExpression`, default:NULL.

## Methods

**tfbs.getDistanceMatrix** Calcuate a distance matrix with Pearson's R values

**tfbs.getExpression** Estimate gene expression of target TF.

**tfbs.clusterMotifs** Cluster the specified motifs and drawing the heatmap.

**tfbs.scanTFsite** Find TF sites from genome data within the BED ranges.

**tfbs.compareTFsite** Comparative TFBS search with the BED ranges

**tfbs.selectByGeneExp** Select the motifs with minimum p-value from each group of clustering.

**tfbs.selectByRandom** Select the motifs randomly from each group of clustering.

**tfbs.drawLogosForClusters** Draw the motif logos by one group per page.

**tfbs.drawLogo** Draw the logo for a single TF motif.

## Examples

```
showClass("tfbs")
```

---

```
tfbs.clusterMotifs
```

*Clustering the specified motifs and drawing the heatmap.*

---

## Description

Clustering the specified motifs and drawing the heatmap.

## Usage

```
tfbs.clusterMotifs(tfbs,
  subset = NA,
  pdf.heatmap = NA,
  method = NA,
  group.k = NA)
```

## Arguments

<code>tfbs</code>	A tfbs object returned by <code>CisBP.find</code> , <code>tfbs.dirs</code> or other functions.
<code>subset</code>	Vector, the indexes of partial motifs if not all motifs are clustered.
<code>pdf.heatmap</code>	String, a PDF filename for heatmap.
<code>method</code>	String, available values are "agnes" and "cors".
<code>group.k</code>	Integer, if the method of agnes is used to do clustering, the parameter of k is optional to use as preset group number.

## Details

This result of clustering will be used in the

## Value

A matrix with 2 columns is returned, 1st column is the index of motifs and 2nd column is the group number of clustering.

## See Also

See Also as [tfbs.selectByGeneExp](#) and [tfbs.selectByRandom](#)

## Examples

```
# Load the internal CisBP data set
db <- CisBP.extdata("Homo_sapiens");

# Create a tfbs object by querying the meta file of CisBP dataset.
tfs <- CisBP.find(db, motif_type="ChIP-seq", motif_info_type=1 );

# Calculate the distance matrix
tfs <- tfbs.getDistanceMatrix( tfs, ncores=1 );

# Cluster the motifs using the "cors" method
cluster1 <- tfbs.clusterMotifs(tfs, pdf.heatmap = "test-heatmap1.pdf", method="cors" );
show(cluster1);

# draw motif logos on one group per page.
tfbs.drawLogosForClusters(tfs, cluster1, "test-cluster1.pdf")

# Cluster the motifs using the "agnes" function
cluster2 <- tfbs.clusterMotifs(tfs, pdf.heatmap = "test-heatmap2.pdf", method="agnes" );
show(cluster2);
```

---

tfbs.compareTFsite *Comparative TFBS search with the BED ranges*

---

## Usage

```
tfbs.compareTFsite(tfbs,
  file.twoBit,
  positive.bed,
  negative.bed,
  file.prefix,
  usemotifs=NA,
  fdr=0.1,
  threshold=NA,
  background.order=2,
  background.length=100000,
  ncores=3)
```

## Arguments

tfbs	A tfbs object, see also <a href="#">tfbs</a>
file.twoBit	String, the file name of genome data( hg19.2bit, mm10.2bit)
positive.bed	Data frame, bed-formatted down-regulatory ranges
negative.bed	Data frame, bed-formatted up-regulatory ranges
file.prefix	String, the prefix for outputted file
usemotifs	Number, the index of which motifs are used to compare. All motifs are used if NA.
fdr	Numeric value, False Discovery Rate (FDR) of possible binding sites, only binding sites with FDR less than this value can be selected.

threshold	Numeric value, only sites with scores above this threshold are returned (default = 6)
background.order	Number, order of Markov model to build background
background.length	Number, length of the sequence to simulate background
ncores	Number, computing nodes in parallel environment.

### Value

A data frame with the following columns:

tf.name	Motif name.
Npos	TF site count found in positive ranges.
Nneg	TF site count found in negative ranges.
assoc.pvalue	p-value calculated by fisher test.
starch	Binary filename of detected TF sites.

### Examples

```
file.dREG.H.change.bed <- "/home/zw355/src/rtfbs_db/rtfbsdb/test/dREG.H.change.bed"
file.dREG.all.bed <- "/home/zw355/src/rtfbs_db/rtfbsdb/test/dREG.all.bed"
file.twoBit <- "/local/storage/data/hg19/hg19.2bit"

db <- CisBP.extdata("Homo_sapiens");
tfs <- CisBP.find(db, family_name="AP-2");

dREG_H_change_bed <- read.table(file.dREG.H.change.bed, header=FALSE);
dREG_all_bed <- read.table(file.dREG.all.bed, header=FALSE);

t <- tfbs.compareTFsite( tfs,
  file.twoBit,
  dREG_H_change_bed,
  dREG_all_bed,
  file.prefix="test.db",
  ncores = 1);
```

---

tfbs.db-class	<i>Class</i> "tfbs.db"
---------------	------------------------

---

### Description

Abstract class for motif dataset. The CisBP class is a son class of tfbs.db.

### Objects from the Class

Now code or function can be used to create this class.

**Slots**

`species`: Species name.

**Methods**

No methods defined with class "tfbs.db" in the signature.

**Examples**

```
showClass("tfbs.db")
```

---

tfbs.dirs	Create a tfbs object from the folders.
-----------	----------------------------------------

---

**Description**

Create a tfbs object from all the PWM files found in the supplied folders.

**Usage**

```
tfbs.dirs(...,
  species = "Homo_sapiens",
  args.read.motif = NULL,
  pattern = glob2rx("*.pwm"),
  recursive = FALSE)
```

**Arguments**

<code>...</code>	Multiple strings, one or more folders can be used in this function.
<code>species</code>	String, including the species name.
<code>args.read.motif</code>	List, including <i>pseudocount</i> , <i>force_even</i> or other parameters used in <code>read.table</code> function.
<code>pattern</code>	String, a character vector specifying regular expression and wildcards.
<code>recursive</code>	Logical, indicating the loading recursively descends into subfolders or not, default: FALSE.

**Details**

Two parameters in the list of `args.read.motif` can be used:  
*pseudocount*: log value for zero value in PWM matrix, default is -7.  
*force\_even*: whether the PWM matrix with odd size needs to be even.

**Value**

A tfbs object collecting all the PWM files in the specified folders. For the details of tfbs object, please see [tfbs.db](#)

**See Also**

The structure of tfbs object is described in [tfbs.db](#)

**Examples**

```
fs.dir <- system.file("extdata","", package="rtfbsdb")
tfs <- tfbs.dirs( fs.dir,
  args.read.motif = list(pseudocount=-7, header=TRUE, sep="\t" , row.names=1) );
str(tfs);
```

---

tfbs.drawLogo	<i>Draw single motif logo</i>
---------------	-------------------------------

---

**Description**

Draw the logo for a single TF motif.

**Usage**

```
tfbs.drawLogo(tfbs, index)
```

**Arguments**

tfbs	A tfbs object
index	Number, indicating the motif index.

**Value**

No return values.

**See Also**

See Also as [tfbs](#)

**Examples**

```
db <- CisBP.extdata("Homo_sapiens");
tfs <- CisBP.find(db, family_name="AP-2");
pdf("test-logos.pdf");
for(i in 1:10) tfbs.drawLogo(tfs, i );
dev.off();

#unlink("test-logos.pdf");
```

---

`tfbs.drawLogosForClusters`*Draw the motif logos by clustering.*

---

## Description

Draw the motif logos by one group per page.

## Usage

```
tfbs.drawLogosForClusters(tfbs, cluster.mat, pdf.logos)
```

## Arguments

<code>tfbs</code>	A tfbs object.
<code>cluster.mat</code>	A matrix with 2 columns returned by <code>tfbs.clusterMotifs</code> , 1st column is the index of motifs and 2nd column is the group number of clustering.
<code>pdf.logos</code>	String indicating a PDF filename.

## Value

No return value.

## See Also

See Also as `tfbs.clusterMotifs`

## Examples

```
# Load the internal CisBP data set
db <- CisBP.extdata("Homo_sapiens");

# Create a tfbs object by querying the meta file of CisBP dataset.
tfs <- CisBP.find(db, motif_type="ChIP-seq", motif_info_type=1 );

# Calculate the distance matrix
tfs <- tfbs.getDistanceMatrix( tfs, ncores=1);

# Cluster the motifs using the "cors" method
cluster1 <- tfbs.clusterMotifs(tfs, pdf.heatmap = "test-heatmap1.pdf", method="cors" );
show(cluster1);

# draw motif logos on one group per page.
tfbs.drawLogosForClusters(tfs, cluster1, "test-cluster1.pdf")
```

---

```
tfbs.getDistanceMatrix
```

*Calculate a distance matrix with Pearson's R values.*

---

### Description

Compare any two motifs and return a matrix with Pearson's R values.

### Usage

```
tfbs.getDistanceMatrix(tfbs, ncores = 3, BG = log(c(0.25, 0.25, 0.25, 0.25)))
```

### Arguments

tfbs	A tfbs object.
ncores	Number, the number of cores to use simultaneously.
BG	The log value of probabilities for nucleotide A, C, G and T as Background computing.

### Value

A tfbs object with new distance matrix (@distancematrix).

### Examples

```
db <- CisBP.extdata("Homo_sapiens");
tfs <- CisBP.find(db, family_name="AP-2");
tfs0 <- tfbs.getDistanceMatrix(tfs, ncores=1);
```

---

```
tfbs.getExpression
```

*Estimate gene expression of target TF.*

---

### Description

Gets expression level of target TF.  
USE extra\_info\$DBID to find gene information encoded by GENCODE V21

### Usage

```
tfbs.getExpression(tfbs,
  file.bigwig.plus, file.bigwig.minus,
  file.twoBit=NA,
  gencode.ext.rdata=NA)
```



**Arguments**

`tfbs` A `tfbs` object outputted by `CisBP.extdata()`, `CisBP.download()`, `CisBP.zipload()`.

`file.bigwig.plus` String, indicating bigwig file for strand plus(+).

`file.bigwig.minus` String, indicating bigwig file for strand minus(-).

`file.twoBit` String, indicating the binary data of sequence.

`gencode.ext.rdata` Gencode RDATA file encoded by this package.

**Details**

For each motif, the occurrence ranges can be queried by the gene ID in the GENCODE v22 database(`file:gencode.v22.annotation`). After the searching, one range obtained from the merge of the multiple ranges will be used to detect the reads count in the specified bigwig files(including plus and minus). The probability of each motif can be calculated by the reads count and lambda.

The lambda is determined by the following formulation:

```
r.lambda = 0.04 * sum(reads_in_all_chromosomes)/10751533/1000.
```

The dataset of GENCODE v21 has been compiled into RDATA file and delivered in this package.

The `gencode_transcript_ext` object can be accessed after the following command is executed successfully.

```
load( system.file("extdata", "gencode_human21_transcript_ext.rdata", package = "tfbs") )
```

**Value**

A `tfbs` object with new expression data frame including the following columns:

<code>Motif_ID</code>	Motif_ID from CisBP dataset or other data source.
<code>DBID</code>	DBID from CisBP dataset or other data source.
<code>chr</code>	String, chromosome name.
<code>start</code>	Integer, start position in which gene ID can be detected.
<code>end</code>	Integer, end position in which gene ID can be detected.
<code>strand</code>	String, + or -, indicating the strand direction.
<code>bed_length</code>	Integer, the length of range which gene ID can be detected.
<code>reads</code>	The reads number queried by BigWig function from the bigwig files( plus and minus)
<code>lambda</code>	The lambda parameter in poisson distribution.
<code>prob</code>	The probability calculated based on poisson distribution.

**See Also**

See Also as `tfbs`, `~~~`

## Examples

```
# Load the internal CisBP data set
db.human <- CisBP.extdata("Homo_sapiens");

# Create a tfbs object by querying the meta file of CisBP dataset.
tfs <- CisBP.find(db.human, motif_type="ChIP-seq", motif_info_type=1 );

if(0)
{
file.bigwig.plus <- "/work/03350/tg826494/hg19/GSM1480327_K562_PROseq_plus.bw";
file.bigwig.minus <- "/work/03350/tg826494/hg19/GSM1480327_K562_PROseq_minus.bw";

tfs <- tfbs.getExpression(tfs, file.bigwig.plus, file.bigwig.minus );
}
```

---

tfbs.scanTFsite	<i>Find TF sites from genome data within the BED ranges</i>
-----------------	-------------------------------------------------------------

---

## Description

Find TF sites from genome data within the BED ranges. Please notice that this package does not provided genome data such as hg19.2bit, mm10.2bit.

## Usage

```
tfbs.scanTFsite(tfbs,
  file.twoBit,
  dnase.peaks.bed,
  file.prefix="scan.db",
  usemotifs=NA,
  ncores=3,
  return.type=c("matches", "posteriors", "maxposterior", "writedb"),
  threshold=6, ...)
```

## Arguments

tfbs	A tfbs object returned by <code>CisBP.find</code> , <code>tfbs</code> , <code>tfbs.dirs</code> .
file.twoBit	String, the file name of genome data( hg19.2bit or mm10.2bit)
dnase.peaks.bed	Data frame, bed-formatted peak information
file.prefix	String, the prefix for outputted file, only used when the <code>return.type</code> is <i>writedb</i>
usemotifs	Vector indtciting indexes of motif to be used in scanning.
ncores	Number, comupting nodes in parallel environment.
return.type	String, four available values explained in th details
threshold	Numeric value, only sites with scores above this threshold are returned (default = 6)
...	Any parameters used in the function of <code>score.ms</code> .

## Details

(1) Four options are available for the function of `tfbs.scanTFsite` as follows.

- `matches` returns all matching motifs.
- `writedb` writes a bed file with matches. Assumes that `sort-bed` and `starch` tools are available in `$PATH`
- `posteriors` returns the posteriors at each position.
- `maxposterior` returns the `max(posterior)` in each dnase-1 peak.

(2) In order to make the binary file with the parameter of `writedb`, make sure that `starchcat` and `sort-bed` command (in BEDOPS) can be accessed from R environment. If not, please put the folder in `$PATH`.

## Value

The option of *matches* returns a list including the result of every motif, which result is BED style data frame with the following columns.

<code>chrom</code>	chromosome
<code>chromStart</code>	start position
<code>chromEnd</code>	chromosomeend position
<code>name</code>	
<code>score</code>	
<code>strand</code>	strand

The option of *writedb* will return a binary BED filename in which store all bed ranges.

The option of *posteriors* will return a list for each motif returned by `score.ms` function. Scores represent the motif <92>match score<92>, or the product of the probability of observing each base under the motif or background models. Scores are returned under the motif model for all positions in the sequence, on both forward and reverse strands, and under the background model.

The option of *maxposterior* will return a probability matrix which the row indicates the range of dnase peak and the column indicates the motif.

## Examples

```
library(rtfbsdb);

file.dREG.H.change.bed <- "/home/zw355/src/rtfbs_db/rtfbsdb/test/dREG.H.change.bed"
file.twoBit <- "/local/storage/data/hg19/hg19.2bit"

db <- CisBP.extdata("Homo_sapiens");
tfs <- CisBP.find(db, family_name="AP-2");

dREG_H_change_bed <- read.table(file.dREG.H.change.bed, header=FALSE);
```

```

t0 <- tfbs.scanTFsite( tfs,
  file.twoBit,
  dREG_H_change_bed,
  file.prefix="test.db",
  ncores = 1);

str(t0[[1]])

t1 <- tfbs.scanTFsite( tfs,
  file.twoBit,
  dREG_H_change_bed,
  file.prefix="test.db",
  return.type="writedb",
  ncores = 1);

t1

t2 <- tfbs.scanTFsite( tfs,
  file.twoBit,
  dREG_H_change_bed,
  file.prefix="test.db",
  return.type="posteriors",
  ncores = 1);

str(t2[[1]]);

t3 <- tfbs.scanTFsite( tfs,
  file.twoBit,
  dREG_H_change_bed,
  file.prefix="test.db",
  return.type="maxposterior",
  ncores = 1);

str(t3);

```

---

```
tfbs.selectByGeneExp
```

*Motif selection by gene expression level.*

---

## Description

Select the motifs with minimum p-value from each group of clustering.

## Usage

```
tfbs.selectByGeneExp(tfbs, cluster.mat)
```

## Arguments

tfbs	A tfbs object with the data frame of gene expression level.
cluster.mat	A matrix with 2 columns returned by <code>tfbs.clusterMotifs</code> , 1st column is the index of motifs and 2nd column is the group number of clustering.

**Details**

The function of `tfbs.getExpression` should be successfully called and the results of gene expression should be returned before this function is called. The indexes of selected motifs will be used in the function of `tfbs.compareTFsite` or `tfbs.scanTFsite`.

**Value**

A vector of motif indexes is returned.

**See Also**

See Also as `tfbs.selectByRandom`, `tfbs.getExpression`

**Examples**

```
db <- CisBP.extdata("Homo_sapiens");

tfs <- CisBP.find(db, family_name="AP-2");

if(0)
{
  tfs <- tfbs.getExpression(tfs, file.bigwig.plus, file.bigwig.minus, file.hg19);

  cluster1 <- tfbs.clusterMotifs(tfs, pdf.heatmap="test-AP2-heatmap.pdf" );

  usemotif <- tfbs.selectByGeneExp(tf, cluster1);
}
```

---

```
tfbs.selectByRandom
```

*Random motif selection*

---

**Description**

Select the motifs randomly from each group of clustering.

**Usage**

```
tfbs.selectByRandom(tfbs, cluster.mat)
```

**Arguments**

<code>tfbs</code>	A <code>tfbs</code> object.
<code>cluster.mat</code>	A matrix with 2 columns returned by <code>tfbs.clusterMotifs</code> , 1st column is the index of motifs and 2nd column is the group number of clustering.

**Details**

The indexes of selected motifs can be used in the function of `tfbs.compareTFsite` or `tfbs.scanTFsite`.

**Value**

A vector of motif indexes is returned.

**See Also**

See Also as [tfbs.selectByGeneExp](#), [tfbs.getExpression](#)

**Examples**

```
db <- CisBP.extdata("Homo_sapiens");  
tfs <- CisBP.find(db, family_name="AP-2");  
tfs <- tfbs.getDistanceMatrix(tfs, ncores=1);  
cluster1 <- tfbs.clusterMotifs(tfs, pdf.heatmap="test-AP2-heatmap.pdf" );  
usemotif <- tfbs.selectByRandom(tfs, cluster1);  
show(usemotif);
```

# Index

## \*Topic **CisBP object**

- CisBP.download, 3
- CisBP.extdata, 4
- CisBP.find, 4
- CisBP.group, 5
- CisBP.zipload, 7

## \*Topic **Clustering**

- tfbs.clusterMotifs, 10
- tfbs.drawLogosForClusters, 15
- tfbs.selectByGeneExp, 20
- tfbs.selectByRandom, 21

## \*Topic **Comparing**

- tfbs.compareTFsite, 11

## \*Topic **Distance matrix**

- tfbs.getDistanceMatrix, 16

## \*Topic **Gene expression**

- tfbs.getExpression, 16

## \*Topic **Logo**

- tfbs.drawLogo, 14
- tfbs.drawLogosForClusters, 15

## \*Topic **Scanning**

- tfbs.scanTFsite, 18

## \*Topic **Selection**

- tfbs.selectByGeneExp, 20
- tfbs.selectByRandom, 21

## \*Topic **classes**

- CisBP.db-class, 2
- tfbs-class, 9
- tfbs.db-class, 12

## \*Topic **tfbs object**

- CisBP.find, 4
- tfbs, 8
- tfbs.clusterMotifs, 10
- tfbs.compareTFsite, 11
- tfbs.dirs, 13
- tfbs.drawLogo, 14
- tfbs.drawLogosForClusters, 15
- tfbs.getDistanceMatrix, 16
- tfbs.getExpression, 16
- tfbs.scanTFsite, 18
- tfbs.selectByGeneExp, 20
- tfbs.selectByRandom, 21

CisBP.db-class, 2

CisBP.download, 3, 4, 7

CisBP.extdata, 3, 4, 7

CisBP.find, 4, 6, 8

CisBP.find, CisBP.db-method  
(CisBP.db-class), 2

CisBP.group, 5

CisBP.group, CisBP.db-method  
(CisBP.db-class), 2

CisBP.zipload, 3, 4, 7

<http://cisbp.cabr.utoronto.ca/>,  
2

[http://cisbp.cabr.utoronto.ca/bulk\\_archive.php](http://cisbp.cabr.utoronto.ca/bulk_archive.php),  
3

tfbs, 5, 8, 8, 11, 14, 17

tfbs-class, 9

tfbs.clusterMotifs, 10, 15, 20, 21

tfbs.clusterMotifs, tfbs-method  
(tfbs-class), 9

tfbs.compareTFsite, 11, 21

tfbs.compareTFsite, tfbs-method  
(tfbs-class), 9

tfbs.db, 2, 13, 14

tfbs.db-class, 12

tfbs.dir, 8

tfbs.dirs, 13

tfbs.drawLogo, 14

tfbs.drawLogo, tfbs-method  
(tfbs-class), 9

tfbs.drawLogosForClusters, 15

tfbs.drawLogosForClusters, tfbs-method  
(tfbs-class), 9

tfbs.getDistanceMatrix, 8, 9, 16

tfbs.getDistanceMatrix, tfbs-method  
(tfbs-class), 9

tfbs.getExpression, 8, 9, 16, 21, 22

tfbs.getExpression, tfbs-method  
(tfbs-class), 9

tfbs.scanTFsite, 18, 21

tfbs.scanTFsite, tfbs-method  
(tfbs-class), 9

tfbs.selectByGeneExp, 10, 20, 22

`tfbs.selectByGeneExp`, `tfbs`-method  
    (*tfbs-class*), [9](#)  
`tfbs.selectByRandom`, [10](#), [21](#), [21](#)  
`tfbs.selectByRandom`, `tfbs`-method  
    (*tfbs-class*), [9](#)