Package ‘MotIV’

March 26, 2015

Type Package
Title Motif Identification and Validation
Version 1.23.0
Date 2014-04-09
Author Eloi Mercier, Raphael Gottardo
Maintainer Eloi Mercier <emercier@bcgsc.ca>, Raphael Gottardo
<rgottard@fhcrc.org>
Depends R (>= 2.10), BiocGenerics (>= 0.1.0)
Imports graphics, grid, methods, IRanges (>= 1.13.5), Biostrings (>= 1.24.0), lattice, rGADEM, stats, utils
Suggests rtracklayer
Description This package makes use of STAMP for comparing a set of motifs to a given database (e.g. JASPAR). It can also be used to visualize motifs, motif distributions, modules and filter motifs.
License GPL-2
biocViews Microarray, ChIPchip, ChIPSeq, GenomicSequence, MotifAnnotation
NeedsCompilation yes

R topics documented:

alignments-class ........................................ 2
as.data.frame .............................................. 3
combineMotifs ............................................ 4
exportAsRangedData ...................................... 5
exportAsTransfacFile ..................................... 6
filter ....................................................... 7
filter-class ............................................... 9
filters-methods ......................................... 10
FOXA1_rGADEM ........................................... 10
alignments-class

Class "alignments"

Description

This object contains the alignments found by a MotIV analysis.

Objects from the Class

Objects can be created by calls of the form new("alignments", TF, evalue, sequence, match, strand).

Slots

- **TF** Object of class "TF"
- **evalue** The e-value of the alignment.
- **sequence** The input sequence aligned.
- **match** The TF sequence which has been matched.
- **strand** The strand of the alignment.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>
as.data.frame

See Also

motiv, matches, transcriptionFactor

---

Coerce to a Data Frame

Description

This function coerces a MotIV object into a data frame.

Usage

## S4 method for signature 'motiv'
as.data.frame(x)

Arguments

x

An object of class motiv.

Details

'as.data.frame' returns a data frame.
This object regroups all the TF identified by MotIV with the corresponding evalue and alignments.

Value

A data.frame object.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

See Also

data.frame, viewAlignments

Examples

####Database and Scores####
path <- system.file(package="MotIV")
jasper <- readPWMFile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

####Input####
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs, trimPWMedge, threshold=1)
combineMotifs

### Description

This function combines motifs according to a set of filters.

### Usage

```r
# S4 method for signature 'motiv,filters'
combineMotifs(x, y, name=NULL, exact=TRUE, verbose=TRUE)
```

### Arguments

- **x**: An object of class `motiv`.
- **y**: A filter or a set of filter.
- **name**: Name(s) to be given for similar motifs.
- **verbose**: If `FALSE`, no output will be print.
- **exact**: If `TRUE`, search only for perfect name match.

### Details

This function is used to consider some motifs as a unique motif or similar motifs.

Many filters could be pass in argument separated by coma. They will be considered independently (comma is considered as OR).

If a name or a vector of name is provided, it will be used to assign new name for similar motif to the corresponding filter. Else, a generic name is used.

### Value

A `motiv` object.

### Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

### See Also

`setFilter`, `filter`, `split`
exportAsRangedData

Examples

```
###Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

###Input###
data(FOXAl_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

###Analysis###
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

###Filters###
f.foxal<-setFilter(name="", tfname="FOXAl", top=3, evauleMax=10^-5)
f.ap1 <- setFilter (tfname="AP1", top=3)
f.foxal.ap1 <- f.foxal | f.ap1
foxa1.filter <- filter(foxa1.analysis.jaspar, f.foxal.ap1, exact=FALSE, verbose=TRUE)
foxa1.filter.combine <- combineMotifs(foxa1.filter, c(f.foxal, f.ap1), exact=FALSE, name=c("FOXAl", "AP1"), verbose=TRUE)
```

`exportAsRangedData` Export MotIV Results

Description

Export your

Usage

`exportAsRangedData(x, y, correction=TRUE)`

Arguments

- `x` An object of class `motiv`.
- `y` The rGADEM type object associated with the `motiv` object.
- `correction` If `TRUE`, corrects the position according to the alignment.

Details

Use this function to export the results into a RangedData object.

Value

An object of type RangedData.
Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

Examples

```r
### Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMFile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

### Input ###
data(FOXAl_rGADoM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs, trim PWM edge, threshold=1)

### Analysis ###
foxal.analysis.jaspar <- motifMatch(inputPWM=motifs, align="SWU", cc="PCC", database=jaspar, DBscores=jaspar.scores, summary=foxal.analysis.jaspar)

### Filters ###
f.foxal <- setFilter(name="", tfname="FOXAl", top=3, eva lueMax=10^-5)
f.apl <- setFilter (tfname="API", top=3)
f.foxal.apl <- f.foxal | f.apl
goal.filter <- filter(foxal.analysis.jaspar, f.foxal.apl, exact=FALSE, verbose=TRUE)
goal.split <- split(foxal.analysis.jaspar, c(f.foxal, f.apl), drop=FALSE, exact=FALSE, verbose=TRUE)
goal.filter.combIe <- combineMotifs(foxal.filter, c(f.foxal, f.apl), exact=FALSE, name=c("FOXAl", "API"), verbose=TRUE)

### Plots ###
# plot(foxal.filter.combIe, ncol=2, top=5, rev=FALSE, main="FOXAl", bysim=TRUE)
# plot(foxal.filter.combIe, gadem, ncol=2, type="distribution", correction=TRUE, group=FALSE, bysim=TRUE, strand=FALSE)
# plot(foxal.filter.combIe, gadem, type="distance", correction=TRUE, group=TRUE, bysim=TRUE, main="FOXAl", strand=FALSE)

### Ranged Data ###
foxal.rd <- exportAsRangedData(foxal.filter.combIe["FOXAl"], gadem)
apl.rd <- exportAsRangedData(foxal.filter.combIe["API"], gadem)
```

exportAsTransfacFile  

Write Transfac Files

Description

Export an object of class motif as a Transfac file type.

Usage

```r
## S4 method for signature 'motiv'
exportAsTransfacFile(x, file)
## S4 method for signature 'list'
exportAsTransfacFile(x, file)
```
Arguments

x An object of class motiv to be export.
file A character string naming a file.

Details

This function is made to provide standard output file used by STAMP. It take an object of class motiv and write two files named *.matched.txt and *.match_pairs.txt containing alignments and identified PWMs.

For more information about the Transfac file format, please refer to http://www.benoslab.pitt.edu/stamp/help.html.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

Examples

```r
# Database and Scores
path <- system.file(package="MotIV")
jaspar <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jaspar.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

# Input
data(FOX1_rGADEM)
recs <- getPWM(gadem)
recs.trimed <- lapply(recs,trimPWMedge, threshold=1)

# Analysis
fox1.analysis.jaspar <- motifMatch(inputPWM=recs,align="SWU",cc="PCC",database=jaspar, DBscores=jaspar.scores)

# Filters
f.fox1<-setFilter(name="", tfname="FOX1", top=3, evaluMax=10**-5)
f.ap1 <- setFilter (tfname="AP1", top=3)
f.fox1.ap1 <- f.fox1 | f.ap1
fox1.filter <- filter(fox1.analysis.jaspar, f.fox1.ap1, exact=FALSE, verbose=TRUE)
fox1.split <- split(fox1.analysis.jaspar, c(f.fox1, f.ap1), drop=FALSE, exact=FALSE, verbose=TRUE)
fox1.filter.combine <- combineMotifs(fox1.filter, c(f.fox1, f.ap1), exact=FALSE, name=c("FOX1", "AP1"), verbose=TRUE)

# Export
exportAsTransfacFile(fox1.filter.combine, file="fox1_analysis")
```

Description

This function selects motifs according to a set of filters.
Usage

```r
## S4 method for signature 'motiv, filters'
filter(x, f, exact=FALSE, verbose=TRUE)
```

Arguments

- `x` An object of class `motiv`.
- `f` A filter or a set of filters for `motiv` object.
- `verbose` If `FALSE`, no output will be print.
- `exact` If `TRUE`, search only for perfect name match.

Details

This function is used to select motifs that correspond to the filters. Many filters could be pass in argument separated by coma. They will be considered independently.

Value

A `motiv` object.

Author(s)

Eloi Mercier <<emer@chibi.ubc.ca>>

See Also

`setFilter`, `split`, `combine`

Examples

```r
### Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPMMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

### Input###
data(FOX1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

### Analysis###
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar, DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

### Filters###
f.foxa1<-setFilter(name="", tfname="FOX1", top=3, evalMax=10^-5)
f.ap1 <- setfilter (tfname="AP1", top=3)
f.foxa1.ap1 <- f.foxa1 | f.ap1
foxa1.filter <- filter(foxa1.analysis.jaspar, f.foxa1.ap1, exact=FALSE, verbose=TRUE)
foxa1.split <- split(foxa1.analysis.jaspar, c(f.foxa1, f.ap1), drop=FALSE, exact=FALSE, verbose=TRUE)
```
Description

This object information to be apply as filter.

Details

This class filter is used to selected motif objects according the filter’s arguments.

Objects from the Class

Objects can be created by calls of the form new("filter", name, tfname, top, eva valoreMax, lengthMax, valid).

Slots

- name: A name or a list of names.
- tfname: A transcription factor name or a list of TF names.
- eva valoreMax: An e-value between 0 and 1.
- top: Defined the depth of the filter.
- lengthMax: The maximum motif length.
- valid: The alignment that should be considered as valid.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

See Also

setFilter, filter, split, combine

Examples

showClass("filter")
Description

Methods for filters object

Usage

```r
## S4 method for signature 'filter'
summary(object)
## S4 method for signature 'filters'
summary(object)
## S4 method for signature 'filter'
names(x)
## S4 method for signature 'filters'
names(x)
```

Arguments

- `object`: An object of class `filter`.
- `x`: An object of class `filter`.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

See Also

`setFilter`, `filter`, `split`, `combine`

Examples

```r
showClass("filter")
```

---

**FOXA1_rGADEM**  
*Dataset for FOXA1*

Description

This dataset contains results obtained by rGADEM for the FOXA1 transcription factor.

Usage

`gadem`
References

http://genomebiology.com/2008/9/9/R137

Examples

#####Database and Scores#####
path <- system.file(package="MotIV")
data(jaspar2010)
data(jaspar2010_scores)

#####Input#####
data(FOXA1_rGadem)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

#####Analysis#####
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

---

generateDBScores  Database Scores Functions

Description

This functions are used to generate scores of a PWM database.

Usage

generateDBScores(inputDB,cc="PCC",align="SWU",nRand=1000,go=1,ge=0.5)
readDBScores(file)
writeDBScores(x, file)

Arguments

inputDB  A list of PWM corresponding to the database.
c  The metric name to be used :
align  The Alignment method to be used.
go  Gap open penalty.
ge  Gap extension penalty.
nRand  The number of random PWM to be generated. The more higer it is, the more accurate score will be.
file  A character string naming a file.
x  A numeric matrix corresponding to a score.
generateDBScores

Details

The score reflects the bias of the database. It is used to compute more precisely e-value alignments.

generateDBScores: Based on database properties (such as length, zero rate, invariant columns), nRand matrix are generated. A score is calculated for each matrix length with the specified alignment method and metric.

The score is associated to a database and a alignment method and metric so you don’t have to generate it each time you use the same database. Use the writeDBScores and readDBScores instead. readDBScores: Read a score file. writeDBScores: Write a score file.

Value

A numeric matrix. Columns correspond respectively to the first matrix length, second matrix length, variance, mean, matrix number, distance min and max.

Warning

Because of each matrix is compared to each other, computing time is exponential. You should be aware of this fact before providing a high nRand. 5000 is a good time/accuracy rate choice.

Author(s)

Shaun Mahony, modified by Eloi Mercier <emercier@chibi.ubc.ca>

References


See Also

'readDBScores', 'writeDBScores'

Examples

```r
# Database and Scores
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path, "/extdata/jaspar2010.txt", sep=""))
jasper.scores <- generateDBScores(inputDB=jaspar, cc="PCC", align="SWJ", nRand=1000)
#writeDBScores(jasper.scores,paste(path,"/extdata/jaspar_PCC_SWJ.scores",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWJ.scores",sep=""))
```
getGademPWM

Description

This function selects the PWMs contained in an object of type gadem.

Usage

getGademPWM(y)

Arguments

y A gadem object.

Value

A list of PWM.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

Examples

```r
### Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC SWJ.scores",sep=""))

### Input###
data(FOXA1_rGADEM)
# motifs <- getGademPWM(gadem) # depreciated
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)
```

getPWM

Get PWMs from a motif object

Description

Get PWMs from a motif object.

Usage

```r
## S4 method for signature 'motiv'
getPWM(x)
```
Arguments

x  An object of class `motiv`.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

Description

Jaspar database and Jaspar score.

Usage

```r
jaspar
jaspar.scores
```

Details

Jaspar is a well-known transcription factor database. Version 2010 contains 130 non-redundant matrix of TF binding sites.

The Jaspar scores have been computed with Pearson Correlation Coefficient and Smith-Waterman Ungapped alignments.

Source

http://jaspar.genereg.net/

References

Albin Sandelin, Wynand Alkema, Per Arne Engstrom, Wyeth W. Wasserman and Boris Lenhard, 
JASPAR: an open-access database for eukaryotic transcription factor binding profiles, 

See Also

generateDBscores, motifMatch
Examples

```r
### Database and Scores###
path <- system.file(package="MotIV")
data(jaspar2010)
data(jaspar2010_scores)

### Input ###
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

### Analysis ###
foxa1.analysis.jaspar <- motifMatch(inputP WM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores
tsummary(foxa1.analysis.jaspar )
```

makePWM

Constructing a pwm object

Description

This function constructs an object of class pwm from a matrix. It checks that the matrix has correct dimensions and that columns add up to 1.0.

Usage

```r
makePWM(pwm, alphabet="DNA")
```

Arguments

- `pwm`: Matrix representing the position weight matrix
- `alphabet`: Character the alphabet making up the sequence. Currently, only "DNA" is supported.

Value

An object of class pwm.

Author(s)

Oliver Bembom, <bembom@berkeley.edu>

Examples

```r
mfile <- system.file("Exfiles/pwm1", package="seqLogo")
m <- read.table(mfile)
pwm <- makePWM(m)
```
matches-class  

**Class** "matches"

**Description**

This object contains the name of the input motif and all the matches found.

**Objects from the Class**

Objects can be created by calls of the form `new("matches", name, aligns, similarity, valid)`.

**Slots**

- `name`  Motif name.
- `aligns`  Alignments found by `motifMatch`.
- `similarity`  The optional name given to the motif.
- `valid`  The alignment that should be considered as valid.

**Author(s)**

Eloi Mercier <<emercier@chibi.ubc.ca>>

**See Also**

`motiv`, `alignments`, `tf`

**Examples**

```r
showClass("matches")
```

---

**motifDistances**  

**Clustering PWMs Computation**

**Description**

Set of functions to perform clustering of PWMs.

**Usage**

```r
motifDistances(inputPWM, DBscores=jaspar.scores, cc="PCC", align="SWU", top=5, go=1, ge=0.5)
motifHclust(x,...)
motifCutree(tree,k=NULL, h=NULL)
```
motifMatch

Arguments

- `inputPWM`, `DBscores`, `cc`, `align`, `top`, `go`, `ge`
  Option for the PWMs distances computation. Refer to motifMatch.

- `x, ...`
  Arguments to pass to the hclust function. See hclust.

- `tree, k, h`
  Arguments to pass to the cutree function. See cutree.

Details

This function are made to perform motifs clustering.

The ‘motifDistances’ function computes the distances between each pair of motifs using the specified alignment.

The ‘motifHclust’ and ‘motifCutree’ functions are simple redefinition of ‘hclust’ and ‘cutree’.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

Examples

```r
# Database and Scores
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

# Input
data(FOX1_gadem)
motifs1 <- getPWM(gadem)
motifs1.trimmed <- lapply(motifs1, trimPWMedge, threshold=1)

# Analysis
foxa1.analisis.jaspar <- motifMatch(inputPWM=motifs1, align="SWU", cc="PCC", database=jaspar, DBscores=jaspar.scores)

# Clustering

d <- motifDistances(getPWM(foxa1.analisis.jaspar))
hc <- motifHclust(d)
plot(hc)
f <- motifCutree(hc, k=2)
foxa1.combine <- combineMotifs(foxa1.analisis.jaspar, f, exact=FALSE, name=c("Group1", "Group2"), verbose=TRUE)
```

motifMatch  Motifs Matches Analysis

Description

Search for motifs matches corresponding to PWM.
motifMatch

Usage

motifMatch(inputPWM, database=jaspar, DBscores=jaspar.scores, cc="PCC", align="SWU", top=5, go=1, ge=1)

Arguments

inputPWM A list of PWM.
database A list of PWM corresponding to the database.
DBscores A matrix object containing the scores associated to the database.
cc The metric name to be used
align The Alignment method to be used.
top The number of identified transcription factors per motif.
go Gap open penalty.
ge Gap extension penalty.

details

For a set of PWMs given by inputPWM, this function realizes alignments with each motif of the database and returns the top best motifs. If no database is provided, the function will use jaspar by loading data(jaspar2010). If no DBscores is given, jaspar.scores from data(jaspar2010_scores) will be used.

The e-value is computed according the metric name cc and is corrected by the DBscores.

Value

A motif object.

Author(s)

Eloi Mercier <emerrier@chibi.ubc.ca>

References


See Also

generateDBScores

Examples

### Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMFile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

### Input###
data(FOXA1_rGDEM)
motiv-class

motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

####Analysis####
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

motiv-class  

Class "motiv"

Description

This object contains all informations about the motiv analysis.

Objects from the Class

Objects can be created by calls of the form new("motiv",input, bestMatch,argv).

Slots

input  List of input PWM.

bestMatch  Object of class "matches".

argv  List of arguments used.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

See Also

matches,alignments,transcriptionFactor

Examples

showClass("motiv")
Description

Methods for motif objects.

Usage

```r
## S4 method for signature 'motiv'
summary(object)
## S4 method for signature 'motiv'
names(x)
## S4 method for signature 'motiv'
length(x)
## S4 method for signature 'motiv'
similarity(x)
## S4 method for signature 'motiv'
x[i,j=ANY, bysim=TRUE, ..., exact=TRUE, ignore.case=FALSE, drop=FALSE]
```

Arguments

- `object` An object of class `motiv`.
- `x` An object of class `motiv`.
- `i` A string representing a motif name.
- `j` NOT USED.
- `bysim` If TRUE, select by similarity name.
- `...` Further potential arguments passed to methods.
- `ignore.case` if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.
- `exact` If TRUE, search only for perfect name match.
- `drop` If TRUE, no match motifs will be dropped.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

See Also

- matches,
- alignments,
- tf

Examples

```r
showClass("motiv")
```
occurences

Motifs Occurences and Co-occurences

Description

Get the number of motifs occurrences and co-occurrences from a rGDEM object.

Usage

occurences(gadem)
cococcurrences(x)

Arguments

gadem An object of type rGDEM.
x A contingency table.

Value

occurences returns the contingency table of the number of motifs per sequences. This object can be put in cococcurrences to return the number of sequences where two motifs appear together.

Author(s)

Eloi Mercier <<emerier@chibi.ubc.ca>>

Examples

data("FOXA1_rGDEM")
oc <- occurences (gadem)
coc <- cococcurrences (oc)
coc

plot-methods

Plot Motiv

Description

This functions are used to vizualise and validate motif analysis.
Usage

```r
## S4 method for signature 'motiv,ANY'
plot(x, y=NULL, main=NULL, sub=NULL, ncol=0, nrow=0, top=3, bysim=TRUE, rev=FALSE, trim=0.05, cex)

## S4 method for signature 'motiv,gadem'
plot(x, y, sort=FALSE, group=FALSE, main=NULL, sub=NULL, ncol=0, nrow=0, xlim=NULL, correction=TRUE,
```

Arguments

- `x` An object of class `motiv`.
- `y` The GADEM type object associated with the `motiv` object.
- `ncol, nrow` A numeric value giving the number of columns and rows to plot.
- `top` A numeric value giving the number of best matches per motif to display.
- `rev` A logical value. If `TRUE`, print reverse motif for negative strand.
- `main` An overall title for the plot: see `title`.
- `sub` A sub title for the plot: see `title`.
- `type` What type of plot should be drawn. Possible values are: distribution to display the binding sites distribution within the peaks or distance to show the pairwise distance between motifs.
- `strand` If `TRUE`, distribution will be plot for both forward and reverse strand.
- `group` If `TRUE`, similar motifs will be grouped.
- `sort` If `TRUE`, motifs will be plot according their computed variance.
- `bysim` If `TRUE`, the 'similar' field (defined with the `combine` function) will be print instead of the original name.
- `xlim` A numeric vectors of length 2, giving the x coordinates ranges.
- `correction` If `TRUE`, corrects the position according to the alignment.
- `trim` A numeric value. Define the minimun information content value for which the logo letters are shown.
- `col, border, lwd, lty` Define respectively the color, the border, the line wide and the line type of both curve and histogram. See 'par'.
- `nclass` A numerical value giving the number of class for the histogram.
- `bw` The smoothing bandwidth to be used to calculate the density. See `density`.
- `cex, vcol` A numerical value giving the amount by which plotting text should be magnified relative to the default.

Details

A single `motiv` object (usually provied by `motifMatch`) will plot the list of identified transcription factors for each motif. With `rev=TRUE`, the transcription factor logo will be print to correspond to the real alignment instead of original TF PWM.
Giving a motiv object and a gadem object with type="distribution" will show the motif repartition within gadem peaks. If strand=TRUE, a distinct distribution is made for forward and reverse strand.

A var.test is automatically made to help to distinguish centered distribution. The distribution with lowest variance is assign as "reference" distribution to compute the var.test statistic. With sort=TRUE, distribution are plot according decreasing statistic.

type="distance" indicates to compute and plot the distance between each pair of motif. It aslo provided Venn diagramm that returns the proportion of common sequences per pair of motif.

The group argument indicates to consider similar motif as a single motif.

With correction=TRUE the motif position is corrected accoring to the alignment. It means that the gap/"N" contained in the alignments are removed to give a corrected start and end position.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

Examples

```r
####Database and Scores####
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWJ.scores",sep=""))

####Input######
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

####Analysis######
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

####Filters######
f.foxa1<-setFilter(name="", tfname="FOXA1", top=3, evalMax=10^-5)
f.api1 <- setFilter (tfname="AP1", top=3)
f.foxa1.ap1 <- f.foxa1 | f.api1
foxa1.filter <- filter(foxa1.analysis.jaspar, f.foxa1.ap1, exact=FALSE, verbose=TRUE)
foxa1.split <- split(f.foxa1.analysis.jaspar, c(f.foxa1, f.ap1)) , drop=FALSE, exact=FALSE, verbose=TRUE)
foxa1.filter.combined <- combineMotifs(foxa1.filter, c(f.foxa1, f.ap1), exact=FALSE, name=c("FOXA1", "AP1"), verbose=TRUE)

####Plots######
plot(foxa1.filter.combined, ncol=2,top=5, rev=FALSE, main="FOXA", bysim=TRUE)
plot(foxa1.filter.combined,gadem,ncol=2, type="distribution", correction=TRUE, group=FALSE, bysim=TRUE, strand=FALSE)
plot(foxa1.filter.combined ,gadem,type="distance", correction=TRUE, group=TRUE, bysim=TRUE, main="FOXA", strand=FALSE)
```
readGademPWMFile  
**Read Gadem File**

**Description**

This function is used to read a gadem file containing PWM.

**Usage**

```r
readGademPWMFile(file)
```

**Arguments**

- `file`  
  File name.

**Details**

This function is made to read typical output file from Gadem (v1.2). Standard name is 'observedPWMs.txt'.

**Author(s)**

Eloi Mercier <emercier@chibi.ubc.ca>

**Examples**

```r
### Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWJ.scores",sep=""))

### Input###
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimmed <- lapply(motifs,trimPWMedge, threshold=1)
```

readPWMfile  
**Read Transfac File**

**Description**

This function is used to read standard Transfac type file.

**Usage**

```r
readPWMfile(file)
```
seqLogo2

Arguments

file Transfac file's name.

Details

This function is designed to read standard Transfac type file. For more information about the format, please refer to [http://mcast.sdsc.edu/doc/transfac-format.html](http://mcast.sdsc.edu/doc/transfac-format.html)

Value

A list of matrix.

Author(s)

Eloi Mercier &lt;emerclier@chibi.ubc.ca&gt;

Examples

```r
database and scores
path <- system.file(package="MotIF")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))
```

seqLogo2  
*Plot a sequence logo for a given position weight matrix*

Description

This function takes the 4xW position weight matrix of a DNA sequence motif and plots the corresponding sequence logo.

Usage

```r
seqLogo2(pwm, ic.scale=TRUE, xaxis=TRUE, yaxis=TRUE, xfontsize=15, yfontsize=15, vmargins=c(0,0),hmargins=c(0,0))
```

Arguments

- **pwm** numeric The 4xW position weight matrix.
- **ic.scale** logical If TRUE, the height of each column is proportional to its information content. Otherwise, all columns have the same height.
- **xaxis** logical If TRUE, an X-axis will be plotted.
- **yaxis** logical If TRUE, a Y-axis will be plotted.
- **xfontsize** numeric Font size to be used for the X-axis.
- **yfontsize** numeric Font size to be used for the Y-axis.
- **vmargins** numeric Vertical margins.
- **hmargins** numeric Horizontal margins.
setFilter

Description

This function is use to set a motif filter.

Usage

setFilter(name = ",

tfname = ",
evalueMax = 1,
top = 10,
lengthMax = 100,
valid = NULL)

Arguments

name A name or a list of names.
tfname A transcription factor name or a list of TF names.
evalueMax An evalue between 0 and 1.
top Defines the depth of the filter.
lengthMax The maximum motif length.
valid The alignment that should be considered as valid.
Value
A filter object.

Author(s)
Eloi Mercier <emercier@chibi.ubc.ca>

See Also
filter, split, combine

Examples

```r
### Database and Scores####
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

### Input####
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

### Analysis####
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jasper, DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar )

### Filters####
f.FOXA1<-setfilter(name="", tfname="FOXa1", top=3, evallueMax=10^-5)
f.AP1 <- setfilter (tfname="AP1", top=3)
f.foxal.ap1 <- f.FOXA1 | f.AP1
foxal.filter <- filter(foxa1.analysis.jaspar, f.foxal.ap1, exact=FALSE, verbose=TRUE)
foxal.split <- split(foxa1.analysis.jaspar, c(f.FOXA1, f.AP1), drop=FALSE, exact=FALSE, verbose=TRUE)
foxal.filter.combine <- combineMotifs(foxal.filter, c(f.FOXA1, f.AP1), exact=FALSE, name=c("FOXA1", "AP1"),
```
split-methods

Arguments

x  An object of class motiv (usually provided by motifMatch).

f  A filter or a set of filters for motif object.

drop  If TRUE, no match motifs will be dropped.

verbose  If FALSE, no output will be printed.

exact  If TRUE, search only for perfect name match.

...  Further potential arguments passed to methods.

Details

This function is used to split motifs that correspond to the filters.

Many filters could be passed in argument separated by comma. They will be considered independently (comma is considered as OR).

Value

A list of motiv objects.

Author(s)

Eloi Mercier <emerrier@chibi.ubc.ca>

See Also

setfilter, filter, combine

Examples

Database and Scores

```R
path <- system.file(package="MotIF")
jasper <- readPWMFile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))
```

Input

```R
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimmed <- lapply(motifs,trimPWMedge, threshold=1)
```

Analysis

```R
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar, DBscores=jaspar.scores)
summary(foxa1.analysis.jaspar)
```

Filters

```R
f.foxa1 <- setfilter(name="", tfname="FOXA1", top=3, evalMax=10^-5)
f.ap1 <- setfilter(tfname="AP1", top=3)
f.fox1.ap1 <- f.fox1 | f.ap1
foxa1.filter <- filter(foxa1.analysis.jaspar, f.fox1.ap1, exact=FALSE, verbose=TRUE)
foxa1.split <- split(f.fox1.analysis.jaspar, c(f.fox1, f.ap1), drop=FALSE, exact=FALSE, verbose=TRUE)
foxa1.filter.combined <- combineMotifs(foxa1.filter, c(f.fox1, f.ap1), exact=FALSE, name=c("FOXA1", "AP1"), verbose=TRUE)
```
transcriptionFactor-class

Transcription Factor Class

Description

This object contains the Transcription Factor name and PWM.

Objects from the Class

Objects can be created by calls of the form new("transcriptionFactor", name, pwm).

Slots

name  TF name.
pwm   TF PWM.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

See Also

motiv, matches, alignments

Examples

showClass("transcriptionFactor")

trimPWMedge

Trim PWM edge

Description

This function is used to cut edges with low information content.

Usage

trimPWMedge(x, threshold=1)

Arguments

x  A matrix representing a PWM.
threshold  A transcription factor name or a list of TF names.
Value

A PWM.

Author(s)

Eloi Mercier <<emercier@chibi.ubc.ca>>

See Also

makePWM

Examples

```r
#####Database and Scores#####
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SNU.scores",sep=""))

#####Input#####
data(FOXA1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)
```

---

**viewAlignments**

*Print Motifs Alignments*

**Description**

This function return a list of the alignments of a motiv object for each motif.

**Usage**

`viewAlignments(x)`

**Arguments**

x 
An object of class motiv (usualy provied by motifMatch).

**Details**

This function shows the alignements for each motif.

**Author(s)**

Eloi Mercier <<emercier@chibi.ubc.ca>>

**See Also**

as.data.frame
**Examples**

```r
#######Database and Scores#######
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

#######Input#######
data(FOX1a_rGADEM)
motifs <- getPWM(gadem)
motifs.trimated <- lapply(motifs,trimPWMedge, threshold=1)

#######Analysis#######
foxal1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores)
summary(foxal1.analysis.jaspar )
viewAlignments(foxal1.analysis.jaspar )
```

**Description**

This function return a list of the identified motifs contained in a `motiv` object.

**Usage**

```r
## S4 method for signature 'motiv'
viewMotifs(x,n=100)
```

**Arguments**

- `x` An object of class `motiv` (usally provied by `motifMatch`).
- `n` The number of motifs shown.

**Details**

This function shows the number of identified motif.

**Value**

A list of motifs names.

**Author(s)**

Eloi Mercier <<emercier@chibi.ubc.ca>>
Examples

```r
###Database and Scores###
path <- system.file(package="MotIV")
jasper <- readPWMfile(paste(path,"/extdata/jaspar2010.txt",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.scores",sep=""))

###Input###
data(FOX1_rGADEM)
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

###Analysis###
fox1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jasper,DBscores=jaspar.scores)
viewMotifs(fox1.analysis.jaspar, 5)
```
Index

Topic **classes**
- alignments-class, 2
- filter-class, 9
- filters-methods, 10
- matches-class, 16
- motif-class, 19
- transcriptionFactor-class, 29

Topic **methods**
- as.data.frame, 3
- combineMotifs, 4
- filter, 7
- motif-methods, 20
- plot-methods, 21
- split-methods, 27

Topic **misc**
- exportAsRangedData, 5
- exportAsTransfacFile, 6
- FOXA1_rGADEM, 10
- generateDBScores, 11
- getGademPWM, 13
- getPWM, 13
- jaspar2010, 14
- makePWM, 15
- motifDistances, 16
- motifMatch, 17
- occurrences, 21
- readGademPWMfile, 24
- readPWMfile, 24
- seqLogo2, 25
- setFilter, 26
- trimPWMedge, 29
- viewAlignments, 30
- viewMotifs-methods, 31
- [,motiv,ANY,ANY-method (motiv-methods), 20
- [,motiv-method (motiv-methods), 20
- & filters, filters-method (filters-methods), 10
- alignments-class, 2
- as.data.frame, 3
- as.data.frame, motiv-method (as.data.frame), 3
- as.data.frame-methods (as.data.frame), 3
- combineMotifs, 4
- combineMotifs, motiv, filters-method (combineMotifs), 4
- combineMotifs, motiv, list-method (combineMotifs), 4
- combineMotifs-method (combineMotifs), 4
- cooccurrences (occurrences), 21
- exportAsRangedData, 5
- exportAsTransfacFile, 6
- exportAsTransfacFile, list-method (exportAsTransfacFile), 6
- exportAsTransfacFile, motiv-method (exportAsTransfacFile), 6
- exportAsTransfacFile-methods (exportAsTransfacFile), 6
- filter, 7
- filter, motiv, filters-method (filter), 7
- filter, motiv, list-method (filter), 7
- filter-class, 9
- filter-methods (filter), 7
- filters-methods, 10
- FOXA (FOXa1_rGADEM), 10
- FOXA1_rGADEM, 10
- gadem (FOXa1_rGADEM), 10
- generateDBScores, 11
- getGademPWM, 13
- getPWM, 13
- getPWM, motiv-method (getPWM), 13
- getPWM-methods (getPWM), 13
- jaspar (jaspar2010), 14
- jaspar2010, 14

33
length, motiv-method (motiv-methods), 20
makePWM, 15
matches-class, 16
motifCutree (motifDistances), 16
motifDistances, 16
motifHclust (motifDistances), 16
motifMatch, 17
motiv (motiv-methods), 20
motiv-class, 19
motiv-methods, 20
names, filter-method (filters-methods), 10
names, filters-method (filters-methods), 10
names, motiv-method (motiv-methods), 20
names<-, motiv-method (motiv-methods), 20
occurrences, 21
plot (plot-methods), 21
plot, motiv, ANY-method (plot-methods), 21
plot, motiv, gadem-method (plot-methods), 21
plot, motiv-method (plot-methods), 21
plot-methods, 21
readDBScores (generateDBScores), 11
readGademPWMFile, 24
readPWMfile, 24
seqLogo2, 25
setFilter, 26
show, filter-method (filters-methods), 10
show, motiv-method (motiv-methods), 20
similarity (motiv-methods), 20
similarity, list-method (motiv-methods), 20
similarity, motiv-method (motiv-methods), 20
split, motiv, filters-method (split-methods), 27
split, motiv, list-method (split-methods), 27
split-methods, 27
summary, filter-method (filters-methods), 10
summary, filters-method (filters-methods), 10
summary, list-method (motiv-methods), 20
transcriptionFactor-class, 29
trimPWMedge, 29
viewAlignments, 30
viewMotifs, viewMotifs-method
(viewMotifs-methods), 31
viewMotifs-methods, 31
writeDBScores (generateDBScores), 11