Package ‘MotIV’

January 21, 2017

Type   Package
Title  Motif Identification and Validation
Version 1.30.0
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, S4Vectors, IRanges (>= 1.13.5),
       Biostrings (>= 1.24.0), lattice, rGADM, utils
Suggests rtracklayer
SystemRequirements GNU Scientific Library >= 1.6
       (http://www.gnu.org/software/gsl/)
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 ........................................................... 8
  filters-methods ......................................................... 9
  FOXA1_rGADEM .......................................................... 10
  generateDBScores ....................................................... 10
  getGademPWM ........................................................... 12
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 as been matched.
- **strand** The strand of the alignment.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

See Also

motiv, matches, transcriptionFactor
Description

This function coerces a MotIV object into a data frame.

Usage

```r
## 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

```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(FDXA1_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) foxa1.analysis.jaspar.df = as.data.frame(foxa1.analysis.jaspar)
head(foxa1.analysis.jaspar.df)
```
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 (coma 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 &lt;emercier@chibi.ubc.ca&gt;

**See Also**

`setFilter`, `filter`, `split`

**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)
```
exportAsRangedData

Description

Export your

Usage

descartasRangedData(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 <<emerier@chibi.ubc.ca>>

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.trimmed <- lapply(motifs, trimPWMedge, threshold=1)
### Analysis

```r
foxa1.analysis.jaspar <- motifMatch(inputPWM=motifs,align="SWU",cc="PCC",database=jasper,DBscores=jaspar.scores,top=5)
summary(foxa1.analysis.jaspar )
```

### Filters

```r
f.foxa1<-setFilter(name="", tfname="FOXA1", top=3, evalueMax=10^-5)
f.ap1 <- setFilter(tfname="API1", 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)
foxa1.filter.combine <- combineMotifs(foxa1.filter, c(f.foxa1, f.ap1), exact=FALSE, name=c("FOXA1", "API1"), drop=FALSE, exact=FALSE, verbose=TRUE)
```

### Plots

```r
#plot(foxa1.filter.combine, ncol=2,top=5, rev=FALSE, main="FOXA1", bysim=TRUE)
#plot(foxa1.filter.combine,gadem,ncol=2, type="distribution", correction=TRUE, group=FALSE, bysim=TRUE, strand=FALSE, sort=TRUE, main="FOXA1")
#plot(foxa1.filter.combine,gadem, type="distance", correction=TRUE, group=TRUE, bysim=TRUE, main="FOXA1", strand=FALSE, method=3, xlim=c(-100,100), darg=list(bw=8))
```

### RangedData

```r
foxa1.rd <- exportAsRangedData(foxa1.filter.combine["FOXA1"], gadem)
ap1.rd <- exportAsRangedData(foxa1.filter.combine["API1"], gadem)
```

---

**exportAsTransfacFile**  
*Write Transfac Files*

#### Description

Export an object of class `motiv` 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 refere to [http://www.benoslab.pitt.edu/stamp/help.html](http://www.benoslab.pitt.edu/stamp/help.html).

#### Author(s)

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

#####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(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,top=5)
summary(foxa1.analysis.jaspar )

#####Filters#####
f.foxa1<-setFilter(name="", tfname="FOXA1", top=3, evalueMax=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)
foxa1.filter.combine <- combineMotifs(foxa1.filter, c(f.foxa1, f.ap1), exact=FALSE, name=c("FOXA1", "AP1"), verbose=TRUE)

#####Export#####
exportAsTransfacFile(foxa1.filter.combine, file="foxa1_analysis")

filter
---

Filter Motifs

Description

This function selects motifs according to a set of filters.

Usage

## 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 filter 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 filter could be pass in argument separated by coma. They will be considered independently.

Value

A motiv object.
Author(s)

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

See Also

setFilter, split, combine

Examples

#####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(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,top=5)
summary(foxa1.analysis.jaspar )

#####Filters#####
f.foxa1<-setFilter(name="", tfname="FOXA1", top=3, evalueMax=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)
foxa1.filter.combine <- combineMotifs(foxa1.filter, c(f.foxa1, f.ap1), exact=FALSE, name=c("FOXA1", "AP1"),

---

**filter-class**

*Class “filter”*

**Description**

This object information to be apply as filter.

**Details**

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

**Objects from the Class**

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

**Slots**

- **name** A name or a list of names.
- **tfname** A transcription factor name or a list of TF names.
- **evalueMax** 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.
**filters-methods**

### 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

```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(inputPWM=motifs,align="SWU",cc="PCC",database=jaspar,DBscores=jaspar.scores,top=5)
summary(foxa1.analysis.jaspar )
```

generateDBScores  

Database Scores Functions

Description

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

Usage

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

Arguments

inputDB  A list of PWM corresponding to the database.
cc       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 higher it is, the more accurate score will be.
file     A character string naming a file.
x        A numeric matrix corresponding to a score.

Details

The score reflects the bias of the database. It is used to compute more precisely e-value alignments.
generateDBScores : Based on database properties (suchs 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 compare to each other, computing time is exponential. You should be aware of this fact before provided 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=""))
#jaspar.scores <- generateDBScores(inputDB=jaspar,cc="PCC",align="SWU",nRand=1000)
#writeDBScores(jaspar.scores,paste(path,"/extdata/jaspar_PCC_SWU.scores",sep=""))
jasper.scores <- readDBScores(paste(path,"/extdata/jaspar2010_PCC_SWU.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
#####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(FOXA1_rGADEM)
# motifs <- getGademPWM(gadem) #depreciated
motifs <- getPWM(gadem)
motifs.trimed <- lapply(motifs,trimPWMedge, threshold=1)

getPWM

Description
Get PWMs from a motiv object.

Usage
## S4 method for signature 'motiv'
getPWM(x)

Arguments
x  An object of class motiv.
Description
Jaspar database and Jaspar score.

Usage
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

See Also
generateDBscores, motifMatch

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, top=5)
summary(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.
motifDistances

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

See Also
motiv, alignments, tf

Examples
showClass("matches")

motifDistances  Clustering PWMs Computation

Description
Set of functions to perform clustering of PWMs.

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

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>>
motifMatch

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(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,top=5)

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

motifMatch

**Motifs Matches Analysis**

**Description**

Search for motifs matches corresponding to PWM.

**Usage**

```r
motifMatch(inputPWM, database=jaspar, DBscores=jaspar.scores, cc="PCC", align="SWU", top=5, go=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`. 
**motiv-class**

Value

A motiv object.

Author(s)

Eloi Mercier <emercier@chibi.ubc.ca>

References


See Also

generateDBScores

Examples

```
#####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(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,top=5)
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>
motiv-methods

See Also
matches, alignments, transcriptionFactor

Examples
showClass("motiv")

motiv-methods  Motiv methods

Description
Methods for motiv objects.

Usage

## 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
showClass("motiv")
Motifs Occurrences and Co-occurrences

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

Usage
occurrences(gadem)
cooccurrences(x)

Arguments

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

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

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

Examples
data("FOXA1_rGADEM")
oc <- occurrences(gadem)
coc <- cooccurrences(oc)
coc

Plot Motiv

Description
This functions are used to visualise and validate motif analysis.

Usage
## 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, ... col=c("blue", "red"), border=c("black", "black"), lwd=2, lty=1, nclass=20, bw="nrd0", cex=1, vcol=c("red", "green"))
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 printed 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 minimum information content value for which the logo letters are shown.

col, border, lwd, lty
- Define respectively the color, the border, the line width 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 provided by `motifMatch`) will plot the list of identified transcription factors for each motif. With `rev=TRUE`, the transcription factor logo will be printed 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 partition 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 assigned as "reference" distribution to compute the `var.test` statistic. With `sort=TRUE`, distribution are plotted according decreasing statistic.

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

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

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

**Description**

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

**Usage**

```r
readGademPWMFile(file)
```

**Arguments**

- `file`  
  File's name.

**Details**

This function is made to read typically output file from Gadem (v1.2). Standard name is `observed-PWMs.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_SWU.scores",sep=""))

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

---

**readPWMfile**  
*Read Transfac File*

**Description**

This function is use to read standard Transfac type file.

**Usage**

```r
readPWMfile(file)
```

**Arguments**

- `file`  
  Transfac file's name.

**Details**

This function is designed to read standard Transfac type file. For more information about the format, please refere 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 <<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=""))
```
**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), size=1, trim=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.
- `size` numeric Graphic size.
- `trim` numeric Print nucleotide only if the information content is superior to this trim threshold.

**Details**

Within each column, the height of a given letter is proportional to its frequency at that position. If `ic.scale` is TRUE, the height of each column in the plot indicates the information content at that position of the motif. Otherwise, the height of all columns are identical.

This is an internal function for the package MotIV. User should prefer the `seqLogo` function from the package seqLogo to visualize individual motif.

**Value**

None.

**Author(s)**

Oliver Bembom, <bembom@berkeley.edu>, modified by Eloi Mercier <<emercier@chibi.ubc.ca>>

**Examples**

```r
#mFile <- system.file("Exfiles/pwm1", package="seqLogo")
#m <- read.table(mFile)
#pwm <- makePWM(m)
#seqLogo2(pwm)
```
setFilter

Set Motif Filter

Description

This function is used to set a motif filter.

Usage

```r
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 <emerier@chibi.ubc.ca>

See Also

- filter
- split
- combine

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(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, top = 5)
summary(foxa1.analysis.jaspar)

# Filters
f.foxa1 <- setFilter(name = "", tfname = "FOXA1", top = 3, evalueMax = 10^-5)
f.ap1 <- setFilter(tfname = "AP1", top = 3)
f.foxa1.ap1 <- f.foxa1 | f.ap1
```
split-methods

Description

This function splits a `motiv` object according to filters.

Usage

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

Arguments

- `x`: An object of class `motiv` (usually provided by `motifMatch`).
- `f`: A filter or a set of filters for `motiv` 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` object.

Author(s)

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

See Also

`setFilter`, `filter`, `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=jaspar,DBscores=jaspar.scores,top=5)
summary(foxa1.analysis.jaspar )

#####Filters#####
f.foxa1<-setFilter(name=", tfname="FOXA1", top=3, evalueMax=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)
foxa1.filter.combine <- combineMotifs(foxa1.filter, c(f.foxa1, f.ap1), exact=FALSE, name=c("FOXA1", "AP1"),
```

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 <emerrier@chibi.ubc.ca>

See Also

motiv, matches, alignments

Examples

```
showClass("transcriptionFactor")
```
trimPWMedge

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 <eemercier@chibi.ubc.ca>

See Also
makePWM

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

viewAlignments

Print Motifs Alignments

Description
This function returns a list of the alignments of a motif object for each motif.

Usage
viewAlignments(x)
Arguments

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

Details

This function shows the alignments for each motif.

Author(s)

Eloi Mercier <emercler@chibi.ubc.ca>

See Also

as.data.frame

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

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

---

viewMotifs-methods

Print Identified Motifs

Description

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

Usage

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

Arguments

x                   An object of class motiv (usually provided 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

####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(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)
viewMotifs(foxa1.analysis.jaspar, 5)
Index

*Topic classes
  alignments-class, 2
  filter-class, 8
  filters-methods, 9
  matches-class, 14
  motif-class, 17
  transcriptionFactor-class, 26

*Topic methods
  as.data.frame, 3
  combineMotifs, 4
  filter, 7
  motif-methods, 18
  plot-methods, 19
  split-methods, 25

*Topic misc
  exportAsRangedData, 5
  exportAsTransfacFile, 6
  FOXA1_rGADEM, 10
  generateDBScores, 10
  getGademPWM, 12
  getPWM, 12
  jasper2010, 13
  makePWM, 14
  motifDistances, 15
  motifMatch, 16
  occurences, 19
  readGademPWMFile, 21
  readPWMFile, 22
  seqLogo2, 23
  setFilter, 24
  trimPWMedge, 27
  viewAlignments, 27
  viewMotifs-methods, 28
  [,motiv,ANY,ANY-method (motiv-methods), 18
  [,motiv-method (motiv-methods), 18
  &,filters,filters-method (filters-methods), 9
  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
  cooccurences (occurences), 19
  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, 8
  filter-methods (filter), 7
  filters-methods, 9
  FOXA (FOXA1_rGADEM), 10
  FOXA1_rGADEM, 10
  gadem (FOXA1_rGADEM), 10
  generateDBScores, 10
  getGademPWM, 12
  getPWM, 12
  getPWM,motiv-method (getPWM), 12
  getPWM-methods (getPWM), 12
  jasper (jaspar2010), 13
  jasper2010, 13
  length,motiv-method (motiv-methods), 18
  makePWM, 14
  matches-class, 14
  motifCutree (motifDistances), 15
  motifDistances, 15
  motifHclust (motifDistances), 15
  motifMatch, 16
  motif (motiv-methods), 18
motiv-class, 17
motiv-methods, 18

names, filter-method (filters-methods), 9
names, filters-method (filters-methods), 9
names, motiv-method (motiv-methods), 18
names<-, motiv-method (motiv-methods), 18
occurences, 19

plot (plot-methods), 19
plot, motiv, ANY-method (plot-methods), 19
plot, motiv, gadem-method (plot-methods), 19
plot, motiv-method (plot-methods), 19
plot-methods, 19

readDBScores (generateDBScores), 10
readGademPWMFile, 21
readPWMFile, 22

seqLogo2, 23
setFilter, 24
show, filter-method (filters-methods), 9
show, motiv-method (motiv-methods), 18
similarity (motiv-methods), 18
similarity, list-method (motiv-methods), 18
similarity, motiv-method (motiv-methods), 18
split, motiv, filters-method (split-methods), 25
split, motiv, list-method (split-methods), 25
split-methods, 25
summary, filter-method (filters-methods), 9
summary, filters-method (filters-methods), 9
summary, list-method (motiv-methods), 18
summary, motiv-method (motiv-methods), 18

transcriptionFactor-class, 26
trimPWMedge, 27

viewAlignments, 27
viewMotifs (viewMotifs-methods), 28
viewMotifs, motiv-method (viewMotifs-methods), 28
viewMotifs-methods, 28

writeDBScores (generateDBScores), 10