Package 'TFBSTools'

October 8, 2014

Version 1.2.0

Date 2014-03-13

Title Software package for transcription factor binding site (TFBS) analysis

Description TFBSTools is a package for the analysis and manipulation of transcription factor binding sites and transcription factor profile matrices.

Author Ge Tan <ge.tan09@imperial.ac.uk>

Maintainer Ge Tan <ge.tan09@imperial.ac.uk>

Imports Biostrings(>= 2.29.19), RSQLite(>= 0.11.4), seqLogo,GenomicRanges(>= 1.13.50), caTools(>= 1.14), XVector(>= 0.2.0),rtracklayer(>= 1.22.0), BSgenome(>= 1.30.0), IRanges(>= 1.19.38), methods, gtools(>= 3.0.0), CNEr(>= 0.99.8),BiocParallel(>= 0.5.6), DirichletMultinomial(>= 1.5.1)

Depends R (>= 3.0.1)

Suggests JASPAR2014(>= 0.99.3), RUnit, BiocGenerics

SystemRequirements meme

License GPL-2

URL http://jaspar.genereg.net/

Type Package

biocViews MotifAnnotation, GeneRegulation, MotifDiscovery

NeedsCompilation yes

LazyData yes

Collate AllGenerics.r AllClasses.r show-methods.r util-methods.r XMatrix-methods.r XMatrixList-methods.r SiteSet-methods.r coercion-methods.r DB-methods.r PairwiseAlignment-methods.r ICM-methods.r PWM-methods.r Motif-methods.r PFM-methods.r Wrappers-methods.r DirichletMixture.r

R topics documented:

`TFBSTools-package` *TFBS software package.*

Description

TFBS includes a set of tools for transcription factor binding site detection and analysis as well as database interface functions for JASPAR, etc.

Author(s)

Ge Tan

calConservation-methods *calConservation method*

Description

Calculate the conservation score for a pairwise alignment given a smooth window size.

Usage

```
calConservation(aln1, aln2, windowSize=51L, which="1")
```

Arguments

aln1	A DNAString object , a DNAStringSet or a character object, which contains the pairwise alignments. When the last two objects have a length of 2, the argument aln2 can be missing.
aln2	A DNAString, a character object or missing.
windowSize	The size of the sliding window (in nucleotides) for calculating local conservation in the alignment. This should be an odd value.
which	The conservation profile of Which sequence in the alignments is computed. It can be "1" or "2".

Value

A numeric vector with the same length of alignment is returned.

Author(s)

Ge Tan

See Also

searchAln

deleteMatrixHavingID *JASPAR database operations*

Description

The functions to initialize, store matrix or delete matrix in JASPAR database.

Usage

```
  ## S4 method for signature character
deleteMatrixHavingID(x, IDs)
  ## S4 method for signature SQLiteConnection
deleteMatrixHavingID(x, IDs)
  ## S4 method for signature JASPAR2014
deleteMatrixHavingID(x, IDs)
  ## S4 method for signature character,PFMatrixList
storeMatrix(x, pfmList)
  ## S4 method for signature SQLiteConnection,PFMatrixList
storeMatrix(x, pfmList)
  ## S4 method for signature JASPAR2014,PFMatrixList
storeMatrix(x, pfmList)
  ## S4 method for signature character,PFMatrix
storeMatrix(x, pfmList)
  ## S4 method for signature SQLiteConnection,PFMatrix
storeMatrix(x, pfmList)
  ## S4 method for signature JASPAR2014,PFMatrix
storeMatrix(x, pfmList)
  ## S4 method for signature SQLiteConnection
initializeJASPARDB(x)
  ## S4 method for signature character
initializeJASPARDB(x)
  ## S4 method for signature JASPAR2014
initializeJASPARDB(x)
```

Arguments

x	a character vector of length 1 for the path of JASPAR SQLite file, or a SQLiteConnection object.
IDs	JASPAR stable IDs.
pfmList	The PFMatrixList object, or pfm object.

Examples

```
  ## Not run:
    library(JASPAR2014)
    deleteMatrixHavingID(JASPAR2014, "MA0003")
    storeMatrix(JASPAR2014, pfm)
    initializeJASPARDB("jaspar.sqlite")

## End(Not run)
```

dmmEM-methods *dmmEM method*

Description

Training Dirichlet mixture models parameters for matrix.

Usage

```
dmmEM(x, K=6, alg=c("C", "R"))
```

Arguments

x	x can be a matrix, PFMatrixList or JASPAR2014 to be trained.
K	The maximal number of components to test in the mixture model when alg is "C". The fixed number of components to use when alg is "R". The default is 6.
alg	The algorithm to use. "C" uses the implementation from DirichletMultinomial package which has more advanced feature and performance. "R" uses our initial implemention in R.

Details

When using the implementation from DirichletMultinomial package, the final number of components can be 1:K. An internal selection will be made based on the maximum likelihood.

When using the implementation of R, the number of component is fixed to K.

Value

A list of trainned alpha0, pmix and likelihood during the training.

Methods

```
signature(x = "ANY")
signature(x = "matrix")
signature(x = "PFMatrixList")
```

Author(s)

Ge Tan

See Also

rPWMDmm

Examples

```
data(MA0003.2)
data(MA0004.1)
pfmList <- PFMatrixList(pfm1=MA0003.2, pfm2=MA0004.1, use.names=TRUE)
dmmParameters <- dmmEM(pfmList, K=6, alg="C")
```

---

getMatrixByID *Basic JASPAR database search functionis* getMatrixByID, getMatrixByName

---

Description

This method fetches matrix data under the given ID or name from the database and returns a XMatrix object.

Usage

```
  ## S4 method for signature character
getMatrixByID(x, ID)
  ## S4 method for signature SQLiteConnection
getMatrixByID(x, ID)
  ## S4 method for signature JASPAR2014
getMatrixByID(x, ID)
  ## S4 method for signature character
getMatrixByName(x, name)
  ## S4 method for signature SQLiteConnection
getMatrixByName(x, name)
  ## S4 method for signature JASPAR2014
getMatrixByName(x, name)
```

Arguments

x	a character vector of length 1 for the path of JASPAR SQLite file, a SQLiteConnection object, or a JASPAR2014 object.
ID	a character vector of length 1 for the JASPAR stable ID. See more details below.
name	a character vector of length 1 for the JASPAR profile name.

Details

For getMatrixByID, ID is a string which refers to the stable JASPAR ID (usually something like "MA0001") with or without version numbers. "MA0001" will give the latest version on MA0001, while "MA0001.2" will give the second version, if existing.

For getMatrixByName, according to the current JASPAR data model, name is not necessarily a unique identifier. Also, names change over time. In the case where there are several matrices with

the same name in the database, the function fetches the first one and prints a warning. You've been warned. Some matrices have multiple versions. The function will return the latest version. For specific versions, use getMatrixByID(ID.version)

Value

A XMatrix object is returned. The exact type of the object depending on the type argument. NA is returned if matrix with the given ID/name is not found.

Author(s)

Ge Tan

See Also

getMatrixSet

Examples

```
library(JASPAR2014)
db = file.path(system.file("extdata", package="JASPAR2014"), "JASPAR2014.sqlite")
pfm = getMatrixByID(db, ID="MA0003")
pfm2 = getMatrixByName(db, name="TFAP2A")
pfm3 = getMatrixByID(JASPAR2014, ID="MA0003")
```

---

getMatrixSet *Advanced JASPAR database search functions* get_MatrixSet

---

Description

This function fetches matrix data for all matrices in the database matching criteria defined by the named arguments and returns a PFMatrixList object

Usage

```
  ## S4 method for signature character
getMatrixSet(x, opts)
  ## S4 method for signature SQLiteConnection
getMatrixSet(x, opts)
  ## S4 method for signature JASPAR2014
getMatrixSet(x, opts)
```

Arguments

x	a character vector of length 1 for the path of JASPAR SQLite file, a SQLiteConnection object, or a JASPAR2014 object.
opts	a search options list. See more details below.

Details

The search options include three categories:

(1) Database basic criterias:

`all=c(TRUE, FALSE)`

`ID`, `name`

`collection=c("CORE", "CNE", "PHYLOFACTS", "SPLICE", "POLII", "FAM", "PBM", "PBM_HOMEO", "PBM_HLH"`

`all_versions=c(TRUE,FALSE)`

`species`, `matrixtype=c("PFM", "PWM", "ICM")`

(2) Tags based criterias:

`class`: "Zipper-Type"", "Helix-Turn-Helix", etc.

`type`: "SELEX", "ChIP-seq", "PBM", etc.

`tax_group`: "plants", "vertebrates", "insects", "urochordat", "nematodes", "fungi".

`family`, `medline`,

(3) Further criterias:

`min_ic` (minimum total information content of the matrix)

`length` (minimum sites length)

`sites` (minimum average sites number per base)

When `all` is `TRUE`, it will get all the matrices and has higher priority over other options. Then ID has the second highest priority, and will ignore all the followiing options. The rest options are combined in search with AND, while multiple elements under one options have the logical operator OR.

Author(s)

Ge Tan

See Also

`getMatrixByID`, `getMatrixByName`

Examples

```
library(JASPAR2014)
db = file.path(system.file("extdata", package="JASPAR2014"), "JASPAR2014.sqlite")
opts = list()
opts[["species"]] = 9606
opts[["name"]] = "RUNX1"
#opts[["class"]] = "Ig-fold"
opts[["type"]] = "SELEX"
opts[["all_versions"]] = TRUE
siteList = getMatrixSet(db, opts)
siteList2 = getMatrixSet(JASPAR2014, opts)
```

MA0004.1 *Some example PFM matrices.*

Description

Some example PFM matrices from JASPAR 2014.

Usage

```
data(MA0004.1)
       data(MA0003.2)
```

Format

The format is: PFMatrix object.

Details

Some examples PFM matrices from JASPAR 2014.

Source

http://jaspar.genereg.net/

Examples

```
data(MA0004.1)
data(MA0003.2)
## maybe str(MA0004.1) ; plot(MA0004.1) ...
```

MotifSet *Class* "MotifSet"

Description

This MotifSet object is a container for storing the generated motifs from Motif identification softwares, such as MEME.

Usage

```
## Constructor
MotifSet(motifList=GRangesList(), motifEvalues=numeric(),
         subjectSeqs=DNAStringSet())
```

Arguments

motifList	A GRangesList. Each GRanges store the starts, ends, strand, seqnames and scores information of one motif sites sequences.
motifEvalues	A numeric vector of the E values generated from MEME for each motif.
subjectSeqs	A DNAStringSet object. It stores the original sequences which are scanned by the software.

Methods

[signature(x = "MotifSet"): Getter

consensusMatrix signature(x = "MotifSet")(x, as.prob = FALSE, shift = 0L, width = NULL, ...): Calculate the consensus matrix. Other arguments, please check the consensusMatrix in Biostrings package.

length signature(x = "MotifSet"): Returns the number of motifs.

sitesSeq signature(x = "MotifSet")(x, n=10L, type="none"): Gets the sites sequences.

n is the number of bases to include from flanking region.

type controls "all", "left", "right" or "none" flanking sequences are included.

Author(s)

Ge Tan

See Also

runMEME

Examples

```
## Not run:
  motifSet = runMEME(file.path(system.file("extdata", package="TFBSTools"), "crp0.s"),
                 binary="/usr/local/Cellar/meme/4.9.0-p4/bin/meme",
                 arguments="-nmotifs 3")
  sitesSeq(motifSet, type="all")
  sitesSeq(motifSet, type="none")
  consensusMatrix(motifSet)

## End(Not run)
```

---

permuteMatrix-methods *permuteMatrix method*

Description

This method simply shuffles the columns in matrices. This can either be done by just shuffling columns within each selected matrix, or by shuffling columns almong all selected matrices.

Usage

```
permuteMatrix(x, type="intra")
```

Arguments

x	A matrix which meets the PFM standard, PFMatrix object, or PFMatrixList obejct.
type	The type of shuffling. It can be "intra" or "inter", which shuffle within each matrix, or between all the matrix.

Value

A object with shuffled matrix.

Author(s)

Ge Tan

Examples

```
library(JASPAR2014)
pfmSubject = getMatrixByID(JASPAR2014, ID="MA0043")
pfmQuery = getMatrixByID(JASPAR2014, ID="MA0048")
opts = list()
opts[["class"]] = "Ig-fold"
pfmList = getMatrixSet(JASPAR2014, opts)
foo = permuteMatrix(pfmQuery)
foo1 = permuteMatrix(pfmList, type="intra")
foo2 = permuteMatrix(pfmList, type="inter")
```

PFMSimilarity-methods *PFMSimilarity method*

Description

Given a PFMatrix or a normal matrix, align it with another set of PFMatrix to assess the similarity.

Usage

```
PFMSimilarity(pfmSubject, pfmQuery, openPenalty=3, extPenalty=0.01)
```

Arguments

pfmSubject	A matrix, PFMatrix or PFMatrixList object, which is compared with query matrix.
pfmQuery	A matrix, PFMatrix or IUPAC character object.
openPenalty	The gap open penalty used in the modified Needleman-Wunsch algorithm. By default, it is 3.
extPenalty	The gap extension penalty used in the modified Needleman-Wunsch algorithm. By default, it is 0.01.

Value

For each pfmSubject, an absolute score and a relative percentage score is returned. The maximum absolute score is 2*the width of the smaller matrix in the comparison pair.

Author(s)

Ge Tan

References

Sandelin, A., H glund, A., Lenhard, B., & Wasserman, W. W. (2003). Integrated analysis of yeast regulatory sequences for biologically linked clusters of genes. Functional & Integrative Genomics, 3(3), 125-134. doi:10.1007/s10142-003-0086-6

Examples

```
library(JASPAR2014)
## one to one comparison
pfmSubject = getMatrixByID(JASPAR2014, ID="MA0043")
pfmQuery = getMatrixByID(JASPAR2014, ID="MA0048")
score = PFMSimilarity(pfmSubject, pfmQuery)
## one to several comparsion
opts = list()
opts[["class"]] = "Ig-fold"
pfmList = getMatrixSet(JASPAR2014, opts)
scores = PFMSimilarity(pfmList, pfmQuery)
```

PWMSimilarity-methods *PWMSimilarity method*

Description

This function measures the similarity of two PWM matrix in three measurements: "normalised Euclidean distance", "Pearson correlation" and "Kullback Leibler divergence".

Usage

```
PWMSimilarity(pwmSubject, pwmQuery, method=c("Euclidean", "Pearson", "KL"))
```

Arguments

pwmSubject	A matrix or PWMatrix or PWMatrixList object.
pwmQuery	A matrix or PWMatrix object.
method	The method can be "Euclidean", "Pearson", "KL".

Details

When pwmSubject and pwmQuery have different number of columns, the smaller PWM will be shifted from the start position of larger PWM and compare all the possible alignments. Only the smallest distance, divergence or largest correlation will be reported.

Methods

```
signature(pwmSubject = "matrix", pwmQuery = "matrix")
signature(pwmSubject = "matrix", pwmQuery = "PWMatrix")
signature(pwmSubject = "PWMatrix", pwmQuery = "matrix")
signature(pwmSubject = "PWMatrix", pwmQuery = "PWMatrix")
signature(pwmSubject = "PWMatrixList", pwmQuery = "matrix")
signature(pwmSubject = "PWMatrixList", pwmQuery = "PWMatrix")
signature(pwmSubject = "PWMatrixList", pwmQuery = "PWMatrixList")
```

References

Linhart, C., Halperin, Y., & Shamir, R. (2008). Transcription factor and microRNA motif discovery: The Amadeus platform and a compendium of metazoan target sets. Genome Research, 18(7), 1180-1189. doi:10.1101/gr.076117.108

See Also

PFMSimilarity

Examples

```
data(MA0003.2)
data(MA0004.1)
pwm1 = toPWM(MA0003.2, type="prob")
pwm2 = toPWM(MA0004.1, type="prob")
PWMSimilarity(pwm1, pwm2, method="Euclidean")
```

rPWMDmm-methods *rPWMDmm method*

Description

This function sample matrice from trainned Dirichlet mixture model based on selected matrices.

Usage

```
rPWMDmm(x, alpha0, pmix, N=1, W=6)
```

Arguments

x	x can be a matrix, PFMatrixList. The count matrix on which the sampling is based.
alpha0	The trained Dirichlet mixture parameters.
pmix	The trained mixing proportions.
N	The number of matrice to sample.
W	The desired width of matrice from the sampling.

Details

This feature enables the users to generate random Position Frequency Matrices (PFMs) from selected profiles.

We assume that each column in the profile is independent and described by a mixture of Dirichlet multinomials in which the letters are drawn from a multinomial and the multinomial parameters are drawn from a mixture of Dirichlets. Within this model each column has its own set of multinomial parameters but the higher level parameters – those of the mixture prior is assumed to be common to all Jaspar matrices. We can therefore use a maximum likelihood approach to learn these from the observed column counts of all Jaspar matrices. The maximum likelihood approach automatically ensures that matrices receive a weight relative to the number of counts it contains.

Drawing samples from the prior distribution will generate PWMs with the same statistical properties as the Jaspar matrices as a whole. PWMs with statistical properties like those of the selected profiles can be obtained by drawing from a posterior distribution which is proportional to the prior times a multinomial likelihood term with counts taken from one of the columns of the selected profiles.

Each 4-dimensional column is sampled by the following three-step procedure: 1. draw the mixture component according to the distribution of mixing proportions, 2. draw an input column randomly from the concatenated selected profiles and 3. draw the probability vector over nucleotides from a 4-dimensional Dirichlet distribution. The parameter vector alpha of the Dirichlet is equal to the sum of the count (of the drawn input) and the parameters of the Dirichlet prior (of the drawn component).

Draws from a Dirichlet can be obtained in the following way from Gamma distributed samples: (X1,X2,X3,X4) = (Y1/V,Y2/V,Y3/V,Y4/V) ~ Dir(a1,a2,a3,a4) where V = sum(Yi) ~ Gamma(shape = sum(ai), scale = 1).

Value

A list of matrices from the sampling.

Methods

```
signature(x = "PFMatrix")
signature(x = "matrix")
signature(x = "PFMatrixList")
```

Note

This code is based on the Matlab code original written by Ole Winther, binf.ku.dk, June 2006.

Author(s)

Ge Tan

References

L. Devroye, "Non-Uniform Random Variate Generation", Springer-Verlag, 1986

Kimura, T., Tokuda, T., Nakada, Y., Nokajima, T., Matsumoto, T., & Doucet, A. (2011). Expectation-maximization algorithms for inference in Dirichlet processes mixture. Pattern Analysis and Applications, 16(1), 55-67. doi:10.1007/s10044-011-0256-4

See Also

dmmEM

Examples

```
data(MA0003.2)
data(MA0004.1)
pfmList <- PFMatrixList(pfm1=MA0003.2, pfm2=MA0004.1, use.names=TRUE)
dmmParameters <- dmmEM(pfmList, 6)
rPWMDmm(MA0003.2, dmmParameters$alpha0, dmmParameters$pmix, N=1, W=6)
```

---

runMEME *Wrapper function for MEME*

Description

This function builds position frequency matrices using an external program MEME written by Bailey and Elkan.

Usage

```
  ## S4 method for signature character
runMEME(x, binary="meme", seqtype="DNA",
                                arguments=list(), tmpdir=tempdir())
  ## S4 method for signature DNAStringSet
runMEME(x, binary="meme", seqtype="DNA",
                                  arguments=list(), tmpdir=tempdir())
```

Arguments

x	A chracter vector of the path of fasta file or a XStringSet.
binary	The path of MEME binary. By default, we assume the meme is in the PATH.
seqtype	The sequence type. "AA" and "DNA" are allowed.
arguments	A list: the addtional arguments for meme. This list takes the parameter of MEME as names of the elements, and the values of the parameters as the elements. For examples, arguments=list("-nmotifs"=3).
tmpdir	A character vector to change the defult tmp directory.

Value

A MotifSet object is returned.

Author(s)

Ge Tan

References

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., et al. (2009). MEME SUITE: tools for motif discovery and searching. Nucleic acids research, 37(Web Server issue), W202-8. doi:10.1093/nar/gkp335

See Also

MotifSet

Examples

```
  ## Not run:
    motifSet = runMEME("/Users/gtan/src/meme_4.9.1/tests/crp0.s",
                   binary="/usr/local/Cellar/meme/4.9.0-p4/bin/meme",
                   arguments="-nmotifs 3")

## End(Not run)
```

searchAln *searchAln method*

Description

Scans a pairwise alignment of nucleotide sequences with the pattern represented by the PWMatrix. It reports only those hits that are present in equivalent positions of both sequences and exceed a specified threshold score in both, AND are found in regions of the alignment above the specified conservation cutoff value.

Usage

```
searchAln(pwm, aln1, aln2, seqname1="Unknown1", seqname2="Unknown2",
          min.score="80%", windowSize=51L,
          cutoff=0.7, strand="*", type="any", conservation=NULL,
          mc.cores=1)
```

Arguments

pwm	A PWMatrix object or a PWMatrixList object.
aln1	A DNAString, character, DNAStringSet or Axt object can be used to represent the pairwise alignment. When the last two objects are used and have a length of 2, the argument aln2 can be missing.
aln2	A DNAString, character. It can be missing when aln1 is DNAStringSet or Axt object.
seqname1,seqname2	A chracter object for the name of sequence. "Unknown1" and "Unknown2" are used by default. These two arguments are ignored when aln1 is Axt. seqname is taken from Axt.
min.score	The minimum score for the hit. Can be given an character string in th format of "80%" or as a single absolute value.
windowSize	The size of the sliding window (in nucleotides) for calculating local conservation in the alignment. This should be an odd value.
cutoff	The conservation cutoff (between 0 and 1) for including the region in the results of the pattern search.
strand	When searching the alignment, we can search the positive strand or negative strand. While strand is "*", it will search both strands and return the results based on the positvie strand coordinate.
type	This argument can be "any" or "all". When it is "any", one motif will be kept if the maximal conservation value of the motif is larger than the cutoff. When it is "all", one motif will be kept if the minimal conservation value of the motif is larger than the cutoff.
conservation	A vector of conservation profile. If not supplied, the conservation profile will be computed internally on the fly.
mc.cores	The number of cpu threads to use when searching Axt. 1 is assigned by default.

Value

A SitePairSet object is returned when pwm is a PWMatrix, while a SitePairSetList is returned when pwm is a PWMatrixList.

Author(s)

Ge Tan

See Also

searchSeq

Examples

```
data(MA0003.2)
pwm <- toPWM(MA0003.2)
## two character objects
aln1 <- "ACTTCACCAGCTCCCTGGCGGTAAGTTGATC---AAAGG---AAACGCAAAGTTTTCAAG"
aln2 <- "GTTTCACTACTTCCTTTCGGGTAAGTAAATATATAAATATATAAAAATATAATTTTCATC"
sitePairSet <- searchAln(pwm, aln1, aln2, min.score="50%",
                         cutoff=0.5, strand="*", type="any")

## scan axt object
library(CNEr)
axtFilesHg19DanRer7 <- file.path(system.file("extdata", package="CNEr"),
                                 "hg19.danRer7.net.axt")
axtHg19DanRer7 <- readAxt(axtFilesHg19DanRer7)
sitePairSet <-  searchAln(pwm, axtHg19DanRer7, min.score="80%",
                          windowSize=51L, cutoff=0.7, strand="*",
                          type="any", conservation=NULL, mc.cores=2)
```

---

searchPairBSgenome-methods

searchPairBSgenome method

Description

Given a chain file for liftover from one genome to another, it searches two BSgenome with a PWMatrix, and only reports the hits that are presents in two genomes with equivalent positions.

Usage

```
searchPairBSgenome(pwm, BSgenome1, BSgenome2, chr1, chr2,
                   min.score="80%", strand="*", chain)
```

Arguments

pwm	A PWMatrix object or a PWMatrixList object.
BSgenome1, BSgenome2	A BSgenome class.
chr1, chr2	A character object, specifying the chromosomes you want to search.
min.score	The minimum score for the hit. Can be given an character string in th format of "80%" or as a single absolute value.
strand	When searching the alignment, we can search the positive "+" strand or negative "-" strand. While strand is "*", it will search both strands and return the results based on the positvie strand coordinate.
chain	A Chain object. It can be generated by import.chain from package rtracklayer. Please provide the chain from BSgenome1 to BSgenome2.

Value

A SitePairSet object is returned when pwm is a PWMatrix, while a SitePairSetList is returned when pwm is a PWMatrixList.

Author(s)

Ge Tan

See Also

searchAln

Examples

```
## Not run:
  library(rtracklayer)
  library(JASPAR2014)
  library(BSgenome.Hsapiens.UCSC.hg19)
  library(BSgenome.Mmusculus.UCSC.mm10)
  pfm = getMatrixByID(JASPAR2014, ID="MA0004.1")
  pwm=toPWM(pfm)
  chain = import.chain("Downloads/hg19ToMm10.over.chain")
  sitepairset = searchPairBSgenome(pwm, BSgenome.Hsapiens.UCSC.hg19,
                                   BSgenome.Mmusculus.UCSC.mm10,
                                   chr1="chr1", chr2="chr1",
                                   min.score="90

## End(Not run)
```

searchSeq *searchSeq method*

Description

It scans a nucleotide sequence with the pattern represented by a PWMatrix and identifies putative transcription factor binding sites.

Usage

```
searchSeq(x, subject, seqname="Unknown", strand="*", min.score="80%")
```

Arguments

x	x can be a PWMatrix object or a PWMatrixList object.
subject	A DNAString, XStringViews or MaskedDNAString object which will be scanned.
seqname	This is sequence name of the target sequence.
strand	When searching the sequence, we can search the positive strand or negative strand. While strand is "*", it will search both strands and return the results based on the positvie strand coordinate.
min.score	The minimum score for the hit. Can be given an character string in th format of "80

Value

A Site object is returned when x is a PWMatrix object, while a SiteList object is returned when x is a PWMatrixList.

Author(s)

Ge Tan

See Also

searchAln, matchPWM

Examples

```
data(MA0003.2)
pwm = toPWM(MA0003.2)
pwmList = PWMatrixList(pwm1=pwm, pwm2=pwm)
siteset = searchSeq(pwm, "GAATTCTCTCTTGTTGTAGTCTCTTGACAAAATG",
                    min.score="60%")
sitesetList = searchSeq(pwmList, "GAATTCTCTCTTGTTGTAGTCTCTTGACAAAATG",
                        min.score="60%", strand="*")
```

seqLogo *Plot a sequence logo for a ICMatrix object*

Description

This function takes a ICMatrix object and plot the sequence logo.

Usage

```
seqLogo(x, ic.scale = TRUE, xaxis = TRUE, yaxis = TRUE,
         xfontsize = 15, yfontsize = 15)
```

Arguments

x	x is a valid ICMatrix class.
ic.scale	A logical value. If TRUE, the total height of one column is proportional to the information content at that position. Otherwise, all the columns will have the same height.
xaxis	A logical value. If TRUE, the x-axis will be plotted.
yaxis	A logical value. If TRUE, the y-axis will be plotted.
xfontsize	A numeric value. The font size for x-axis.
yfontsize	A numeric value. The font size for y-axis.

Details

A sequence logo is a graphical representation of the matrix model, based on the information content of each position. The information content ranges from 0 (no base preference) to 2 (only 1 base used). If ic.scale is TRUE, the height of the logo at certain site is proportinal to the information content value. And each stacked base (A, C, G, T)'s height is also proportional to the information content of each base at that position, and sorted based on the character size.

Methods

```
signature(x = "ICMatrix")
```

Author(s)

Ge Tan

References

T D Schneider, R. M. S. (1990). Sequence logos: a new way to display consensus sequences. Nucleic acids research, 18(20), 6097.

See Also

toICM, ICMatrix,

Examples

```
data(MA0003.2)
icm = toICM(MA0003.2)
seqLogo(icm, ic.scale = TRUE)
```

SitePairSet *Class* "SitePairSet"

Description

The SitePairSet object is a container for storing two SiteSet objects. Usually it is used to hold the results returned by searchAln.

Usage

```
## Constructor
SitePairSet(siteset1, siteset2)
```

Arguments

siteset1, siteset2	Each SiteSet object is from one sequence in the pairwise alignment.

Methods

siteset1 `signature(x = "SitePairSet")`: Gets the first SiteSet object.

siteset2 `signature(x = "SitePairSet")`: Gets the second SiteSet object.

Author(s)

Ge Tan

See Also

SitePairSet, searchAln

SitePairSetList-class *Class* "SitePairSetList"

Description

The SitePairSetList class is a container for storing a collection of SitePairSet objects. Basically it is a SimpleList and is designed for manipulating the set of SitePairSet objects as a whole.

Usage

```
## Constructors:
  SitePairSetList(..., use.names=TRUE)
```

Arguments

...	The SitePairSet objects are supplied in .... A list of SitePairSet objects is also acceptable.
use.names	A logical value. When TRUE, the names of the SitePairSet will be kept.

Author(s)

Ge Tan

See Also

SitePairSet,

SiteSet *Class* "SiteSet"

Description

The SiteSet object is a container for storing a set of putative transcription factor binding sites on a nucleotide sequence (start, end, strand, score, pattern as a PWMatrix, etc.)

Usage

```
## Constructors:
SiteSet(views, score, strand="*", seqname="Unknown", sitesource="TFBS",
        primary="TF binding site", pattern)
```

Arguments

views	Object of class "XStringViews": It holds the start, end and the nucleotide sequence information of the transcription factor binding sites.
score	Object of class "numeric": A vector of PWM score for each putative binding site.
strand	Object of class "character": The binding site is from the positive ("+"), negative ("-") or unknown("*") strand.
seqname	Object of class "character": the seqname of the sequence which contains these binding sites.
sitesource	Object of class "character": Currently it is set to "TFBS"
primary	Object of class "character": Currently it is set to "TF binding site"
pattern	Object of class "PWMatrix": The PWMatrix object which is used to search the binding sites.

Methods

[`signature(x = "SiteSet")`: Getter function.

length `signature(x = "SiteSet")`: The number of binding sites in this SiteSet.

pattern `signature(x = "SiteSet")`: Returns the PWMatrix used.

relScore `signature(x = "SiteSet")`: Gets relative score (between 0.0 to 1.0) with respect of the score range of the associated pattern (PWMatrix).

score `signature(x = "SiteSet")`: Returns the score of each site.

seqname `signature(x = "SiteSet")`: Returns the sequence name of the sequence which contains these sites.

strand `signature(x = "SiteSet")`: Returns the strand information.

views `signature(x = "SiteSet")`: Returns the views object.

pvalues `signature(x = "SiteSet")(x)`: Calculates the empirical p-values for the scores. The background probability for sampling is based on the PWM matrix in the `SiteSet` object.

Author(s)

Ge Tan

See Also

searchSeq, searchAln, PWMatrix, SiteSetList, SitePairSet

Examples

```
data(MA0003.2)
pwm = toPWM(MA0003.2)
siteset = searchSeq(pwm, "GAATTCTCTCTTGTTGTAGTCTCTTGACAAAATG", min.score="60%")
writeGFF3(siteset, scoreType="absolute")
relScore(siteset)
pvalues(siteset)
```

SiteSetList *Class* "SiteSetList"

Description

The SiteSetList class is a container for storing a collection of SiteSet objects. Basically it is a SimpleList and is designed for manipulating the set of SiteSet objects as a whole.

Usage

```
## Constructors:
  SiteSetList(..., use.names=TRUE)
```

Arguments

...	The SiteSet objects are supplied in .... A list of SiteSet objects is also acceptable.
use.names	A logical value. When TRUE, the names of the SiteSet will be kept.

Methods

pvalues signature(x = "SiteSetList")(x): Calculates the empirical p-values for the scores.

Author(s)

Ge Tan

See Also

SiteSet, searchSeq, searchAln

Examples

```
data(MA0003.2)
data(MA0004.1)
pwmList = PWMatrixList(MA0003.2=toPWM(MA0003.2), MA0004.1=toPWM(MA0004.1))
sitesetList = searchSeq(pwmList, "GAATTCTCTCTTGTTGTAGTCTCTTGACAAAATG",
                        min.score="50%")
writeGFF3(sitesetList, scoreType="absolute")
pvalues(sitesetList)
```

toGRangesList-methods *toGRangesList function*

Description

Get the genomic coordinates from SitePairSetList.

Value

A list of two GRanges objects are returned, one for the target sequences and another for query sequences.

In the GRanges, strand is taken from the Axt object. In the meta-data columns, PWM matrix ID, the strand of matrix and match score are also returned.

Methods

signature(x = "SitePairSetList", axt = "Axt") Convert the relative coordinates to absolute coordinates.

Author(s)

Ge Tan

Examples

```
data(MA0003.2)
pwm <- toPWM(MA0003.2)
library(CNEr)
axtFilesHg19DanRer7 <- file.path(system.file("extdata", package="CNEr"),
                                 "hg19.danRer7.net.axt")
axtHg19DanRer7 <- readAxt(axtFilesHg19DanRer7)
sitePairSet <-  searchAln(pwm, axtHg19DanRer7, min.score="80%",
                          windowSize=51L, cutoff=0.7, strand="*",
                          type="any", conservation=NULL, mc.cores=2)
toGRangesList(sitePairSet, axtHg19DanRer7)
```

toICM *toICM method*

Description

Converts a raw frequency matrix (PFMatrix) to a information content matrix (ICMatrix). It takes the bases background frequencies, pseudocounts and schneider as parameters.

Usage

```
toICM(x, pseudocounts=0.8, schneider=FALSE, bg=c(A=0.25, C=0.25, G=0.25, T=0.25))
```

Arguments

x	For toPWM, a PFMatrix, rectangular DNAStringSet object ("rectangular" means that all elements have the same number of characters) with no IUPAC ambiguity letters, a rectangular character vector or a matrix with rownames containing at least A, C, G and T.
pseudocounts	A default value 0.8 is used.
schneider	This logical parameter controls whether a Schneider correction will be done. See more details below.
bg	bg is a vector of background frequencies of four bases with names containing A, C, G, T. When toPWM is applied to a PFMatrix, if bg is not specified, it will use the bg information contained in PFMatrix.

Details

The information content matrix has a column sum between 0 (no base preference) and 2 (only 1 base used). Usually this information is used to plot sequence log.

The information content at each position is computed

$$D = \log_2(nrow(pfm)) + colSums(postProbs * \log_2(postProbs)$$

$$icm = postProbs * D$$

If a Schneider correction will be done if requested. Please see the reference below for more comprehensive explanation.

Value

A ICMatrix object which contains the background frequency, pseudocounts and Schneider correction used.

Author(s)

Ge Tan

References

Schneider, T. D., Stormo, G. D., Gold, L., & Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. Journal of molecular biology, 188(3), 415-431.

See Also

toPWM, XMatrix, seqLogo

Examples

```
## Constructor a PFMatrix
pfm = PFMatrix(ID="MA0004.1", name="Arnt", matrixClass="Zipper-Type", strand="+",
      bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
      tags=list(family="Helix-Loop-Helix", species="10090", tax_group="vertebrates",
      medline="7592839", type="SELEX", ACC="P53762", pazar_tf_id="TF0000003",
      TFBSshape_ID="11", TFencyclopedia_ID="580"),
      profileMatrix=matrix(c(4L,  19L, 0L,  0L,  0L,  0L,
                             16L, 0L,  20L, 0L,  0L,  0L,
                             0L,  1L,  0L,  20L, 0L,  20L,
                             0L,  0L,  0L,  0L,  20L, 0L),
                           byrow=TRUE, nrow=4,
                           dimnames=list(c("A", "C", "G", "T")))
      )
 ## Convert it into a PWMatrix
icm = toICM(pfm, pseudocounts=0.8, schneider=TRUE)
```

`toPWM` *toPWM method*

Description

Converts a raw frequency matrix (PFMatrix) to a position weight matrix (PWMatrix). It takes the type, bases background frequencies, pseudocounts as parameters.

Usage

```
toPWM(x, type="log2probratio", pseudocounts=0.8,
      bg=c(A=0.25, C=0.25, G=0.25, T=0.25))
```

Arguments

`x`	For `toPWM`, a PFMatrix, rectangular DNAStringSet object ("rectangular" means that all elements have the same number of characters) with no IUPAC ambiguity letters, a rectangular character vector or a matrix with rownames containing at least A, C, G and T.
`type`	The type of PWM generated, should be one of "log2probratio" or "prob".
`pseudocounts`	pseudocounts is a numeric non-negative vector, which means you can specify different pseudocounts for each site. The values will be recycled if shorter than the length of sites. 0.8 is recommended. See the reference below for more details. In the TFBS perl module, the squared root of the column sum of the matrix, i.e., the number of motifs used to construct the PFM, is used.
`bg`	bg is a vector of background frequencies of four bases with names containing A, C, G, T. When toPWM is applied to a `PFMatrix`, if bg is not specified, it will use the bg information contained in `PFMatrix`.

Details

The raw position frequency matrix (PFM) is usually converted into a position weight matrix (PWM), also known as position specific scoring matrix (PSSM). The PWM provides the probability of each base at certain position and used for scanning the genomic sequences. The implementation here is slightly different from `PWM` in `Biostrings` package by choosing the pseudocounts. Pseudocounts is necessary for correcting the small number of counts or eliminating the zero values before log transformation.

$$postProbs = \frac{PFM + bg * pseudocounts}{ncol(PFM) + sum(bg) * pseudocounts}$$

$$priorProbs = bg/sum(bg)$$

$$PWM_{log2probratio} = \log_2 \frac{postProbs}{priorProbs}$$

$$PWM_{prob} = postProbs$$

Value

A `PWMatrix` object which contains the background frequency and pseudocounts used.

Author(s)

Ge Tan

References

Wasserman, W. W., & Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. Nature Publishing Group, 5(4), 276-287. doi:10.1038/nrg1315

Nishida, K., Frith, M. C., & Nakai, K. (2009). Pseudocounts for transcription factor binding sites. Nucleic acids research, 37(3), 939-944. doi:10.1093/nar/gkn1019

See Also

`toICM`, `XMatrix`, PWM

Examples

```
## Constructe a PFMatrix
pfm = PFMatrix(ID="MA0004.1", name="Arnt", matrixClass="Zipper-Type",
               strand="+", bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
               tags=list(family="Helix-Loop-Helix", species="10090",
                         tax_group="vertebrates",
                         medline="7592839", type="SELEX", ACC="P53762",
                         pazar_tf_id="TF0000003",
                         TFBSshape_ID="11", TFencyclopedia_ID="580"),
               profileMatrix=matrix(c(4L,  19L, 0L,  0L,  0L,  0L,
                                      16L, 0L,  20L, 0L,  0L,  0L,
                                      0L,  1L,  0L,  20L, 0L,  20L,
                                      0L,  0L,  0L,  0L,  20L, 0L),
```

```
                                    byrow=TRUE, nrow=4,
                                    dimnames=list(c("A", "C", "G", "T")))
                )
## Convert it into a PWMatrix
pwm = toPWM(pfm, type="log2probratio", pseudocounts=0.8)
```

---

writeGFF3-methods writeGFF3, writeGFF2 *functions*

Description

write the SiteSet, SitePairSet, SiteSetList, SitePairSetList into the GFF3 or GFF2 format.

Usage

```
writeGFF3(x, scoreType=c("absolute", "relative"))
writeGFF2(x, scoreType=c("absolute", "relative"))
```

Arguments

x	A SiteSet, SitePairSet, SiteSetList, or SitePairSetList object.
scoreType	The score column can have absolute value or relative value.

Author(s)

Ge Tan

XMatrix "XMatrix" *objects*

Description

XMatrix is a virtual class. No objects can be created from it directly. Three classes are derived from this class: PFMatrix, PWMatrix and ICMatrix.

PFMatrix is a class whose instances are objects representing raw position frequency matrices (PFMs).

PWMatrix is a class whose instances are objects representing position weight matrices (PWMs). Compared with PFMatrix, it has extra slot pseudocounts.

ICMatrix is a class whose instances are objects representing information content matrices (ICMs). Compared with PWMatrix, it has extra slot schneider.

Usage

```
## Constructors:
PFMatrix(ID="Unknown", name="Unknown", matrixClass="Unknown",
         strand="*", bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
         tags=list(), profileMatrix=matrix())
PWMatrix(ID="Unknown", name="Unknown", matrixClass="Unknown",
         strand="*", bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
         tags=list(), profileMatrix=matrix(), pseudocounts=numeric())
ICMatrix(ID="Unknown", name="Unknown", matrixClass="Unknown",
         strand="*", bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
         tags=list(), profileMatrix=matrix(), pseudocounts=numeric(),
         schneider=logical())
## Accessor-like methods:
## S4 method for signature XMatrix
ID(x)
## S4 method for signature XMatrix
bg(x)

## ... and more (see Methods)
```

Arguments

ID	Object of class "character": a unique identifier for each matrix.
name	Object of class "character": The name of the transcription factor. In JASPAR, as far as possible, the name is based on the standardized Entrez gene symbols. In the case the model describes a transcription factor hetero-dimer, two names are concatenated, such as RXR-VDR. In a few cases, different splice forms of the same gene have different binding specificity: in this case the splice form information is added to the name, based on the relevant literature.
matrixClass	Object of class "character": Structural class of the transcription factor, based on the TFCaT system
strand	Object of class "character": which strand is the binding sites sequences from.
bg	Object of class "numeric": Background frequencies of the four bases. By default, it is equally 0.25.
tags	Object of class "list": Some tags information about this model. Tags include: (1) "family": Structural sub-class of the transcription factor, based on the TFCaT system. (2) "species": The species source for the sequences, in NCBI tax IDs. (3) "tax_group": Group of species, currently consisting of 4 larger groups: vertebrate, insect, plant, chordate. (4) "medline": a ID to the relevant publication reporting the sites used in the mode building. (5) "type": Methodology used for matrix construction. (6) "ACC": A representative protein accession number in Genbank for the transcription factor. Human takes precedence if several exists. (6) "pazar_tf_id": a ID to PAZAR database.

(7) "TFBSshape_ID": a ID to TFBSshape database.

(8) "TFencyclopedia_ID": a ID to the Transcription Factor Encyclopedia.

(9) "comment": For some matrices, a curator comment is added.

`profileMatrix` Object of class `"matrix"`: This is the matrix information.

`pseudocounts` Object of class `"numeric"`: This is the pseudocounts used when computing ICM or PWM from PFM. By default, a threshold of 0.8 is used based on the previous research (doi:10.1093/nar/gkn1019).

`schneider` Obejct of class `"logical"`: this logical value indicates whether the schneider correction is used during the conversion from PFM to ICM.

`x` Object of class `XMatrix`.

Methods

bg `signature(x = "XMatrix")`: Gets the background base frequencies.

bg<- `signature(x = "XMatrix")`: Sets the background base frequencies.

ID `signature(x = "XMatrix")`: Gets the ID information.

ID<- `signature(x = "XMatrix")`: Sets the ID information.

length `signature(x = "XMatrix")`: Gets the pattern length in nucleotides (i.e. number of columns in the matrix).

reverseComplement `signature(x = "PWMatrix")`: Generates the reverse complement matrix object. Note than the strand is XMatrix will also be changed to the opossite strand.

as.matrix `signature(x = "XMatrix")`: Returns the matrix in the XMatrix class.

totalIC `signature(x = "ICMatrix")`: Returns the information content vector.

Matrix `signature(x = "XMatrix")`: Gets the matrix stored in `XMatrix` object.

Matrix<- `signature(x = "XMatrix")`: Sets the matrix stored in `XMatrix` object.

matrixClass `signature(x = "XMatrix")`: Gets the matrix type of a `XMatrix` object.

matrixClass<- `signature(x = "XMatrix")`: Sets the matrix type of a `XMatrix` object.

name `signature(x = "XMatrix")`: Gets the name information.

name<- `signature(x = "XMatrix")`: Sets the name information.

reverseComplement `signature(x = "PWMatrix")`: Reverts the column order in matrix of `PWMatrix` object and assigns each row to its complemntaory base.

strand `signature(x = "XMatrix")`: Gets the strand information of a `XMatrix` object.

tags `signature(x = "XMatrix")`: Gets a `list` object of tags information.

Author(s)

Ge Tan

See Also

`toPWM`, `toICM`

Examples

```
## ------------------------------------------------------------
## Constructor
## Note that there is no XMatrix() constructor,
## but an XMatrix family of constructors: PFMatrix(), PWMatrix(), ICMatrix()
pfm = PFMatrix(ID="MA0004.1", name="Arnt", matrixClass="Zipper-Type", strand="+",
      bg=c(A=0.25, C=0.25, G=0.25, T=0.25),
      tags=list(family="Helix-Loop-Helix", species="10090", tax_group="vertebrates",
      medline="7592839", type="SELEX", ACC="P53762", pazar_tf_id="TF0000003",
      TFBSshape_ID="11", TFencyclopedia_ID="580"),
      profileMatrix=matrix(c(4L,  19L, 0L,  0L,  0L,  0L,
                             16L, 0L,  20L, 0L,  0L,  0L,
                             0L,  1L,  0L,  20L, 0L,  20L,
                             0L,  0L,  0L,  0L,  20L, 0L),
                           byrow=TRUE, nrow=4,
                           dimnames=list(c("A", "C", "G", "T")))
      )

## Coersion
as.matrix(pfm)
as(pfm, "matrix")

## Methods
pwm = toPWM(pfm)
reverseComplement(pwm)
length(pfm)
```

XMatrixList *Class* "XMatrixList"

Description

The XMatrixList virtual class is a container for storing a collection of XMatrix objects. No object can be constructed directly from this virtual and it has three subclasses: PFMatrixList, PWMatrixList and ICMatrixList. Basically it is a SimpleList and is designed for manipulating the set of XMatrix objects as a whole.

Usage

```
## Constructors:
PFMatrixList(..., use.names=TRUE)
PWMatrixList(..., use.names=TRUE)
ICMatrixList(..., use.names=TRUE)

## Accessor-like methods:
## S4 method for signature XMatrixList
ID(x)
## S4 method for signature XMatrixList
name(x)
```

Arguments

...	The XMatrix objects are supplied in ....
use.names	A logical value. When TRUE, the names of the XMatrix will be kept.
x	A XMatrixList object.

Author(s)

Ge Tan

See Also

XMatrix,

Examples

```
data(MA0003.2)
data(MA0004.1)
pfmList <- PFMatrixList(pfm1=MA0003.2, pfm2=MA0004.1, use.names=TRUE)
```

Index