Package 'motifRG'

November 21, 2019

Title A package for discriminative motif discovery, designed for high throughput sequencing dataset

Version 1.30.0

Date 2012-03-23

Author Zizhen Yao

Description Tools for discriminative motif discovery using regression methods

Imports Biostrings,IRanges,seqLogo,parallel,methods,grid,graphics,XVector

Maintainer Zizhen Yao <yzizhen@fhcrc.org>

License Artistic-2.0

LazyLoad yes

biocViews Transcription,MotifDiscovery

Depends R (>= 2.15), Biostrings (>= 2.26), IRanges, seqLogo, parallel, methods, grid, graphics, BSgenome, XVector, BSgenome.Hsapiens.UCSC.hg19

git_url https://git.bioconductor.org/packages/motifRG

git_branch RELEASE_3_10

git_last_commit 7ec8603

git_last_commit_date 2019-10-29

Date/Publication 2019-11-20

R topics documented:

`ctcf.motifs` *CTCF motifs predicted by motifRG*

Description

The output produced by findMotif.

Details

A list with following elements: motifs:a list motif descriptions of class Motif-class. category:input binary specification of foreground/background. mask.motifs:if `mask=T`, then mask.motifs contain the description of motif is based on motif matches after the input sequences being masked by previous motifs. In this case, "motifs" contained the unmasked motif descriptions.

References

Unpublished

`findMotif` *De-novo discovery of distriminative motifs*

Description

The function searches motifs that discriminate the given foregound and background sequences.

Usage

```
findMotif(all.seq,  category, weights = rep(1, length(all.seq)),
start.width=6,min.cutoff=5, min.ratio=1.3,
min.frac=0.01, both.strand=TRUE, flank=2, max.motif=5,
mask=TRUE,other.data=NULL, start.nmer=NULL,
enriched.only=F,n.bootstrap = 5, bootstrap.pvalue=0.1,is.parallel =
TRUE,mc.cores = 4,min.info=10,max.width=15,discretize=TRUE)
```

Arguments

`all.seq`	DNAStringSet; foreground and background sequences.
`category`	numeric vector; specify which sequences are foreground (with value 1), and background (value 0).
`weights`	numeric vector: the weights for all sequences. Default: 1
`start.width`	logical; the width for enumerating seed patterns
`min.cutoff`	numeric; the score cutoff required for seed selection. All scores are negative, the lower the better.
`min.ratio`	numeric; the minimum fold change of motif occurences in foreground vs background.
`min.frac`	numeric; the minimum fraction of fg/bg sequences containing the candidate motifs

both.strand	logical; if true, search both strands
flank	integer; the length for step-wise pattern extension at both ends on candidate motifs
max.motif	integer; the maximum number of output motifs
mask	logical; if true, mask previous motifs when searching for the next motif
other.data	if not NULL, a matrix with additional terms for the regression model for bias adjustment
start.nmer	if not NULL, a matrix with counts for user specified seed pattern in each sequence
enriched.only	logical; if true, only predict enriched motif
n.bootstrap	integer; the number of bootstrapping tests to estimate score variance
bootstrap.pvalue	numeric: the bootstrap t.test pvalues to determine the significance of improvement
is.parallel	logical;if true, runs in parallel mode, and requires "parallel" library
mc.cores	integer; the number of CPUs for paralel run
min.info	minimal information content for the motif to prevent it from being too degenerate
max.width	maximum width of the motif for extension
discretize	logical default TRUE

Value

return a list with following elements:

motifs	a list motif descriptions of class Motif-class
.	
category	input binary specification of foreground/background
mask.motifs	if mask=T, then mask.motifs contain the description of motif is based on motif matches after the input sequences being masked by previous motifs. In this case, "motifs" contained the unmasked motif descriptions.

Examples

```
MD.peak.seq <- readDNAStringSet(system.file("extdata","MD.peak.fa", package="motifRG"))
MD.control.seq <- readDNAStringSet(system.file("extdata","MD.control.fa", package="motifRG"))
category <- c(rep(1, length(MD.peak.seq)), rep(0, length(MD.control.seq)))
MD.motifs <- findMotif(append(MD.peak.seq, MD.control.seq),category, max.motif=3,enriched=TRUE)

### Get summary of motifs
summaryMotif(MD.motifs$motifs, MD.motifs$category)

### plot the dinucleotide representation of the first motif
plotMotif(MD.motifs$motifs[[1]]@match$pattern)

### Create table of motifs in Latex
motifLatexTable(MD.motifs, main="MD motifs")

### Create table of motifs in Html
motifHtmlTable(MD.motifs)
```

findMotifFasta *Wrapper function for findMotif using fasta input*

Description

Perform motif search on two input fasta files. This is a wrapper function for findMotif

Usage

```
findMotifFasta(fg.file, bg.file, ...)
```

Arguments

fg.file	character; foreground fasta filename
bg.file	character; background fasta filename
...	Other parameters passed to findMotif

findMotifFgBg *Wrapper function for findMotif using two sequence datasets*

Description

Perform motif search on two sequence datasets. This is a wrapper function for findMotif.

Usage

```
findMotifFgBg(fg.seq, bg.seq, ...)
```

Arguments

fg.seq	DNAStringSet; foreground sequences
bg.seq	DNAStringSet; background sequences
...	Other parameters passed to findMotif

getSequence *Fetch genomic sequences from GRanges Object*

Description

The function fetch genomic sequences given the coordinates and strand informaton

Usage

```
getSequence(gr, genome)
```

Arguments

gr	GRanges object; coordinates of sequences.
genome	BSgenome object

Motif-class *Motif objects*

Description

A Motif object contains general motif characteristics and and details of motif match

Details

A motif object has the following slots: score:absolute z-value based on the logistic regression model for the motif. sign:the sign of the motif:plus for enriched motif in the foreground sequences, and negative for depleted motif count:a numeric vector holding the number of matches in each sequence match:a data.frame with the following columns: match.strand:the strand on which the match is found; pattern:the motif match pattern; seq.id: on which sequence the match is found; pos:the position relative to sequence start of the match. pattern:the motif pattern consensus:the motif consensus pattern determined by the majority votes at each position using the following rule: the most dominate single nucleotide if its frequency is greater than 0.6, or the two most dominate nucleotide if combined frequency is greater than 0.8, or the three most dominate nucleotide if combined frequency is greater than 0.95

See Also

`findMotif` `summaryMotif` `plotMotif` `motifLatexTable`

motifLatexTable *create of table of motifs*

Description

create a latex table to be embedded in a latex document

Usage

```
 motifLatexTable(motifs,  main="", prefix="motif", dir=".", height=1,
 width=3,enriched.only=F, plot.pwm= TRUE,
summary.cols=c(1,7,8,9),use.mask=TRUE)
 motifHtmlTable(motifs, dir="html", prefix="motif", enriched.only=F,
plot.pwm= TRUE, summary.cols=c(1,7,8,9),use.mask=TRUE)
```

Arguments

`motifs`	result of findMotif
`main`	The title of table
`prefix`	The prefix for the filenames of motif logos
`dir`	The directory for storing motif logo files
`height, width`	size of the sequence logo
`enriched.only`	If true, list only enriched motifs
`plot.pwm`	If true, plot PWM logo instead of di-nucleotide logo

summary.cols	The selected columns of summary table created by summaryMotif included in the table
use.mask	If true, use masked motif match summary statistics

Value

motifLatexTable outputs a latex table to the stdout console. motifHtmlTable outputs a html file named as <preix>.html in "dir" directory.

See Also

findMotif

Examples

```
data(ctcf.motifs)
### Create table of motifs in Latex
motifLatexTable(ctcf.motifs, main="CTCF motifs", dir="motif")

### Create table of motifs in Html
motifHtmlTable(ctcf.motifs, dir="Html")
```

---

plotMotif *plot motif sequence matches*

Description

plot aligned sequences, revealing the independent position specificity and dependency among adjacent positions.

Usage

```
plotMotif(match, logodds=F, entropy=F, bg.ld=NULL, alphabet=c("A", "C", "G", "T"),has.box=T,...)
```

Arguments

match	motif match to be plotted. character or DNAStringSet object.
logodds	logical; if true, plot the enrichment/depletion of a adjacent pair relative to the independent model.
entropy	logical; if true, areas outside the core region of the motif are dimmed
bg.ld	Experimental features: background dinucleotide logodds against independent model. if logodds=T, then the background logodds will be substracted
alphabet	the alphabets used in the sequence. Do not change its value
has.box	logical; if true, plot the boundaries of the motif
...	other arguments passed to the lower level plot function

Details

X-axis refers to the positions of the motifs.

Y-axis correspond to the alphabets.

Letter sizes define the frequencies of the nucleotides at a given position.

Edges between the letters specifiy the dinucleotide relationship. The depth of the color correspond to the dinucleotide frequency. If `logodds=T`, thinner edges will be plotted between dependent pairs. The edge is colored red if the pair is depleted (relative to the expected frequency if the pair is independent), and green if the pair is enriched. The gradient of color red/green correspond to the level of dependency.

Examples

```
data(ctcf.motifs)
### plot the dinucleotide representation of the first motif
plotMotif(ctcf.motifs$motifs[[1]]@match$pattern)
plotMotif(ctcf.motifs$motifs[[1]]@match$pattern, logodds=TRUE)
plotMotif(ctcf.motifs$motifs[[1]]@match$pattern, logodds=TRUE, entropy=TRUE)
```

`refinePWMMotif` *create a PWM (Position Weight Model) model given a initial set of motif matches and input sequences*

Description

Create a PWM model given a initial set of motif matches and input sequences

Usage

```
  refinePWMMotif(motifs=NULL, seqs,  pwm.ld= NULL, max.iter=50,
tol=10^-4, mod="oops", null=rep(0.25, 4),pseudo=1, weights=rep(1,
length(seqs)), motif.weights=NULL)
```

Arguments

`motifs`	The initial set of motif matches. character vector or DNAStringSet object
`seqs`	Input sequences. character vector or DNAStringSet object
`pwm.ld`	The initial PWM matrixes in logodds transformation. Either "motifs" or "pwm.ld" is not NULL
`max.iter`	Maximum number of iterations for refinement
`tol`	Convergence criteria. The percentage of total PWM scores improvement required for convergence.
`mod`	Motif ocurrence model. If `mod=="oops"`, assume one motif match per sequence. If `mod=="zoops"`, assume zero or one motif match per sequence.
`null`	A numeric vector specifying the background model
`pseudo`	Pseudo counts for PWM construction
`weights`	a numeric vector specifying the weights for all sequences. Default: 1 for all sequences
`motif.weights`	a numeric vector specifying the weights for initial sets of motifs. Default: NULL

Value

Return a list with two elements:

model	a list with two elements. "prob": PWM model, sum of columns add to 1. "logodd": PWM model in logodds form, log2 of original matrix substract the background model
.	
match	a data.frame specifying the motif matches in each sequence. Columns are: "match": the sequence of the match, "score": PWM score, "strand", the strand of the match in the input sequence, "pos": start position of the motif match. If multiple matches are allowed, then "seq.id" specifies the index of the input sequence for the motif match.
score	Total PWM score of the motif matches

See Also

findMotif refinePWMMotifExtend

Examples

```
ctcf.seq <- readDNAStringSet(system.file("extdata","ctcf.fa", package="motifRG"))
data(ctcf.motifs)
### refine PWM model based on motif matches
pwm.match <- refinePWMMotif(ctcf.motifs$motifs[[1]]@match$pattern, ctcf.seq)
### plot traditional motif logo
library("seqLogo")
seqLogo(pwm.match$model$prob)
### plot dinucleotide motif logo
plotMotif(pwm.match$match$pattern)
### automatically extend PWM model
pwm.match.extend <-refinePWMMotifExtend(ctcf.motifs$motifs[[1]]@match$pattern, ctcf.seq)
### plot the new motif matches
plotMotif(pwm.match.extend$match$pattern)
```

---

refinePWMMotifExtend *create an extended PWM (Position Weight Model) model given a initial set of motif matches and input sequences*

Description

Create an extended PWM model given a initial set of motif matches and input sequences

Usage

```
refinePWMMotifExtend(motifs=NULL, seqs, pwm.ld=NULL, flank=3, extend.tol=10^-3, trim.rel.entropy
```

Arguments

motifs	The initial set of motif matches. character vector or DNAStringSet object
seqs	Input sequences. character vector or DNAStringSet object
pwm.ld	The initial PWM matrixes in logodds transformation. Either "motifs" or "pwm.ld" is not NULL
flank	The number of bases for extension on both sides of the motif. The extension will be iterated if the there is sufficient signal in the flanking region.
extend.tol	Convergence criteria for extension.
trim.rel.entropy	cutoff to be used to trim the uninformative flanking of a PWM model based on relative entropy against a null distribution.
null	NULL background distribution
max.width	The maximum width of PWM
...	other arguments passed to function refinePWMMotif

Details

Flanking regions with length equal to flank is still included in output for reference

Value

Same type of object returned by refinePWMMotif

See Also

findMotif refinePWMMotif

---

summaryMotif *summarize a list of motifs*

Description

Create a summary table of a list of motifs found by findMotif

Usage

```
summaryMotif(motifs, category)
```

Arguments

motifs	a list of motifs of class Motif-class
category	a vector of 0 or 1, specifying which sequences are foreground and background. Input for findMotif

Value

A data.frame with following columns:

`scores`	scores for each Motif. All values are negative. The absolute scales of the scores reflect the discriminative power of the motif for separating the foreground and background. Statistically, they correspond to the Z-values of the predictor(counts of the motifs in this case) in the logistic regression model
`signs`	sign of the motifs. TRUE for enriched motifs, FALSE for depleted motifs
`fg.hits, bg.hits`	Total number of hits in the foreground, and background sequences. If the motif is scanned on both strands of the input sequences, the counts on both strands are added.
`fg.seq, bg.seq`	The number of sequences that contain at least one motif match in the foreground, and the background
.	
`ratio`	The enrichment/depleted ratio of motifs
`fg.frac,bg.frac`	The fraction of foreground/background sequences that contain at least one motif match

See Also

`findMotif` `motifLatexTable` `motifHtmlTable`

Examples

```
data(ctcf.motifs)
###plot the summary statics of motif matches after masking previous motif ocurrences###
summaryMotif(ctcf.motifs$mask.motifs, ctcf.motifs$category)

###plot the summary statics of motif matches in the original sequences###
summaryMotif(ctcf.motifs$motifs, ctcf.motifs$category)
```

YY1.control *Background for YY1 ChIP-Seq peaks in HepG2*

Description

Background regions randomly sampled in the flanking region of YY1 ChIP-Seq peaks in HepG2.

References

Unpublished

YY1.peak *YY1 ChIP-Seq peaks in HepG2*

Description

a subset of 5000 YY1 ChIP-Seq peaks in HepG2 from ENCODE

References

Unpublished

Index