Package ‘DiffLogo’

October 12, 2016

Type Package

Title DiffLogo: A comparative visualisation of sequence motifs

Version 1.2.1

Depends R (>= 1.8.0), stats, cba,

Suggests knitr, testthat, seqLogo, MotifDb

VignetteBuilder knitr

Description DiffLogo is an easy-to-use tool to visualize motif differences.

License GPL (>= 2)

URL `https://github.com/mgledi/DiffLogo/`

BugReports `https://github.com/mgledi/DiffLogo/issues`

biocViews Software, SequenceMatching, MultipleComparison, MotifAnnotation, Visualization

Collate 'alphabet.R' 'baseDistrs.R' 'diffSeqLogo.R' 'preconditions.R' 'seqLogo.R' 'stackHeights.R' 'utilities.R'

NeedsCompilation no

Author Martin Nettling [aut],
Hendrik Treutler [aut, cre],
Jan Grau [aut, ctb],
Jens Keilwagen [aut, ctb],
Stefan Posch [aut],
Ivo Grosse [aut]

Maintainer Hendrik Treutler <`htreutle@ipb-halle.de`>

R topics documented:

`Alphabet` *built alphabet*

Description

builts an object of class Alphabet from the given set of symbols and colors

Usage

```
Alphabet(chars, cols)
```

Arguments

`chars`	set of symbols
`cols`	set of colors; one for each symbol

Value

the Alphabet object

Author(s)

Martin Nettling

Examples

```
DNA = Alphabet(c("A","C","G","T"),c("green4","blue","orange","red"))
```

`ASN` *ASN alphabet*

Description

the amino acid alphabet (20 symbols), i.e. A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y

Usage

```
ASN
```

Format

```
List of 3
 $ chars: chr [1:20] "A" "C" "D" "E" ...
 $ cols : chr [1:20] "Light green" "Green" "Dark green" "Dark green" ...
 $ size : int 20
 - attr(*, "class")= chr "Alphabet"
```

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/alignments"
motif_name = "calamodulin_1"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = getPwmFromAlignment(readLines(file), ASN, 1)
seqLogo(pwm = motif, alphabet=ASN)
```

`createDiffLogoObject` *DiffLogo object*

Description

Creates a DiffLogo object

Usage

```
createDiffLogoObject(pwm1, pwm2, stackHeight = shannonDivergence,
  baseDistribution = normalizedDifferenceOfProbabilities, alphabet = DNA)
```

Arguments

`pwm1`	representation of the first position weight matrix (PWM) of type pwm, data.frame, or matrix
`pwm2`	representation of the second position weight matrix (PWM) of type pwm, data.frame, or matrix
`stackHeight`	function for the height of a stack at position i
`baseDistribution`	function for the heights of the individual bases
`alphabet`	of type Alphabet

Value

DiffLogo object

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoObj = createDiffLogoObject(pwm1 = pwm1, pwm2 = pwm2)
diffLogo(diffLogoObj)
```

`differenceOfICs` *normalized information content differences*

Description

information content differences normalized by the sum of absolute information content differences for the given pair of probability vectors

Usage

```
differenceOfICs(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

a vector with one result for each symbol

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, baseDistribution = differenceOfICs)
```

`diffLogo` *Draw DiffLogo*

Description

Draws the difference of two sequence logos.

Usage

```
diffLogo(diffLogoObj, ymin = 0, ymax = 0, sparse = FALSE)
```

Arguments

`diffLogoObj`	a DiffLogoObject created by the function createDiffLogoObject
`ymin`	minimum value on the y-axis
`ymax`	maximum value on the y-axis
`sparse`	if TRUE margins are reduced and tickmarks are removed from the logo

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoObj = createDiffLogoObject(pwm1 = pwm1, pwm2 = pwm2)
diffLogo(diffLogoObj)
```

`diffLogoFromPwm` *Draw DiffLogo from PWM*

Description

Draws the difference of two sequence logos.

Usage

```
diffLogoFromPwm(pwm1, pwm2, ymin = 0, ymax = 0,
  stackHeight = shannonDivergence,
  baseDistribution = normalizedDifferenceOfProbabilities, sparse = FALSE,
  alphabet = DNA)
```

Arguments

`pwm1`	representation of the first position weight matrix (PWM) of type pwm, data.frame, or matrix
`pwm2`	representation of the second position weight matrix (PWM) of type pwm, data.frame, or matrix
`ymin`	minimum value on the y-axis
`ymax`	maximum value on the y-axis
`stackHeight`	function for the height of a stack at position i
`baseDistribution`	function for the heights of the individual bases
`sparse`	if TRUE margins are reduced and tickmarks are removed from the logo
`alphabet`	of type Alphabet

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2)
```

`diffLogoTable` *Draw DiffLogo-table*

Description

Draws a table of DiffLogos.

Usage

```
diffLogoTable(PWMs, stackHeight = shannonDivergence,
  baseDistribution = normalizedDifferenceOfProbabilities,
  uniformYaxis = TRUE, sparse = TRUE, showSequenceLogosTop = TRUE,
  enableClustering = TRUE, treeHeight = 0.5, margin = 0.02, ratio = 1,
  alphabet = DNA, ...)
```

Arguments

`PWMs`	a list/vector of position weight matrices (PWMs) each of type pwm, data.frame, or matrix
`stackHeight`	function for the height of a stack at position i
`baseDistribution`	function for the heights of the individual bases
`uniformYaxis`	if TRUE each DiffLogo is plotted with the same scaling of the y-axis
`sparse`	if TRUE margins are reduced and tickmarks are removed from the logo
`showSequenceLogosTop`	if TRUE the classical sequence logos are drawn above each column of the table

`enableClustering`	if TRUE the motifs are reordered, so that similar motifs have a small vertical and horizontal distance in the table
`treeHeight`	the height of the plotted cluster tree above the columns of the table; set equal to zero to omit the cluster tree
`margin`	the space reseverved for labels
`ratio`	the ratio of the plot; this is needed to determine the margin sizes correctly
`alphabet`	of type Alphabet
`...`	set of parameters passed to the function 'axis' for plotting

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

diffLogoTable(motifs)
```

`DNA` *DNA alphabet*

Description

the DNA alphabet, i.e. A, C, G, T

Usage

```
DNA
```

Format

```
List of 3
 $ chars: chr [1:4] "A" "C" "G" "T"
 $ cols : chr [1:4] "green4" "blue" "orange" "red"
 $ size : int 4
 - attr(*, "class")= chr "Alphabet"
```

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_name = "HepG2"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = as.matrix(read.delim(file,header=FALSE))
seqLogo(pwm = motif, alphabet=DNA)
```

`getPwmFromAlignment` *Create PWM from alignment*

Description

Creates a matrix-representation of a PWM from a set of sequences

Usage

```
getPwmFromAlignment(alignment, alphabet, pseudoCount)
```

Arguments

`alignment`	a vector or list of sequences each with equal length
`alphabet`	of type Alphabet
`pseudoCount`	the number of pseudo-observations for each character in the alphabet

Value

PWM as matrix

Author(s)

Hendrik Treutler

Examples

```
motif_folder= "extdata/alignments"
motif_name = "calamodulin_1"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = getPwmFromAlignment(readLines(file), ASN, 1)
seqLogo(pwm = motif, alphabet=ASN)
```

`informationContent` *information content*

Description

the information content for the given probability vector

Usage

```
informationContent(p)
```

Arguments

`p`	probability vector representing the symbol distribution

Value

an object consisting of height a ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_name = "HepG2"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = as.matrix(read.delim(file,header=FALSE))
seqLogo(pwm = motif, stackHeight = informationContent)
```

`lossOfAbsICDifferences` *the change of information content*

Description

the change of information content for the given probability vectors

Usage

```
lossOfAbsICDifferences(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

an object consisting of height and ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, stackHeight = lossOfAbsICDifferences)
```

`normalizedDifferenceOfProbabilities`

normalized probability differences

Description

probability differences normalized by the sum of absolute probability differences for the given pair of probability vectors

Usage

```
normalizedDifferenceOfProbabilities(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

a vector with one result for each symbol

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, baseDistribution = normalizedDifferenceOfProbabilities)
```

`probabilities` *probabilities*

Description

the given probabilities

Usage

```
probabilities(p)
```

Arguments

`p`	probability vector representing the symbol distribution

Value

the given vector

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_name = "HepG2"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = as.matrix(read.delim(file,header=FALSE))
seqLogo(pwm = motif, baseDistribution = probabilities)
```

`RNA` *RNA alphabet*

Description

the RNA alphabet, i.e. A, C, G, U

Usage

```
RNA
```

Format

```
List of 3
 $ chars: chr [1:4] "A" "C" "G" "U"
 $ cols : chr [1:4] "green4" "blue" "orange" "red"
 $ size : int 4
 - attr(*, "class")= chr "Alphabet"
```

Author(s)

Martin Nettling

`seqLogo` *Draw sequence logo*

Description

Draws the classic sequence logo.

Usage

```
seqLogo(pwm, sparse = FALSE, drawLines = 0.5,
  stackHeight = informationContent, baseDistribution = probabilities,
  alphabet = DNA)
```

Arguments

`pwm`	representation of a position weight matrix (PWM) of type pwm, data.frame, or matrix
`sparse`	if TRUE margins are reduced and tickmarks are removed from the logo
`drawLines`	distance between background lines
`stackHeight`	function for the height of a stack at position i
`baseDistribution`	function for the heights of the individual bases
`alphabet`	of type Alphabet

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_name = "HepG2"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = as.matrix(read.delim(file,header=FALSE))
seqLogo(pwm = motif)
```

`shannonDivergence` *shannon divergence*

Description

the shannon divergence for the given pair of probability vectors

Usage

```
shannonDivergence(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

an object consisting of height and ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, stackHeight = shannonDivergence)
```

`sumOfAbsICDifferences` *sum of absolute information content differences*

Description

the sum of absolute information content differences for the given pair of probability vectors

Usage

```
sumOfAbsICDifferences(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

an object consisting of height and ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}
```

```
pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, stackHeight = sumOfAbsICDifferences)
```

`sumOfAbsProbabilityDifferences`

sum of absolute probability differences

Description

the sum of absolute probability differences for the given pair of probability vectors

Usage

```
sumOfAbsProbabilityDifferences(p1, p2)
```

Arguments

`p1`	probability vector representing the first symbol distribution
`p2`	probability vector representing the second symbol distribution

Value

an object consisting of height and ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_names = c("HepG2","MCF7","HUVEC","ProgFib")
motifs = list()
for (name in motif_names) {
  fileName = paste(motif_folder,"/",name,".txt",sep="")
  file = system.file(fileName, package = "DiffLogo")
  motifs[[name]] = as.matrix(read.delim(file,header=FALSE))
}

pwm1 = motifs[[motif_names[[1]]]]
pwm2 = motifs[[motif_names[[2]]]]

diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2, stackHeight = sumOfAbsProbabilityDifferences)
```

`sumProbabilities` *sum of probabilities, i.e. 1.0*

Description

the sum of probabilities for the given probability vector, i.e. 1.0

Usage

```
sumProbabilities(p)
```

Arguments

`p`	probability vector representing the symbol distribution

Value

an object consisting of height and ylab

Author(s)

Martin Nettling

Examples

```
motif_folder= "extdata/pwm"
motif_name = "HepG2"
fileName = paste(motif_folder,"/",motif_name,".txt",sep="")
file = system.file(fileName, package = "DiffLogo")
motif = as.matrix(read.delim(file,header=FALSE))
seqLogo(pwm = motif, stackHeight = sumProbabilities)
```

Index