DiffLogo user guide

Hendrik Treutler

November 24, 2026

1 Introduction

The DiffLogo tool is a R package for the visualization of differences between multiple motifs for different alphabets. The user supplies a set of motifs each represented as position weight matrices (PWMs) [1]. The DiffLogo package supports the comparison of two motifs by a single difference logo and the comparison of multiple motifs by a table of difference logos. Difference logos are based on the idea behind the well-known sequence logo [2], i.e. motifs are visualized position-wise based on two functions. First, the *stackHeight* function computes the height of each stack. Second, the *baseDistribution* function breaks down the stack height on the individual characters. The user is able to parametrise the individual functions with arbitrary functions *stackHeight* and *baseDistribution*. Default implementations of both functions are provided. DiffLogo supports alignment of input PWMs for both pairwise and multiple comparison cases.

2 Download and import library

The user is able to download the R package DiffLogo from GitHub[1]. After installing the package, the user is able to import DiffLogo.

```
> library(DiffLogo)
```

[1] http://github.com/mgledi/DiffLogo

3 Import PWMs

PWMs can be represented as object of type pwm, data.frame, or matrix. The user is able to import motifs from any source in one of these formats.

```
> library(MotifDb)
> ## import motifs
> hitIndeces = grep ('CTCF', values (MotifDb)$geneSymbol, ignore.case=TRUE)
> list    = as.list(MotifDb[hitIndeces])
> sequenceCounts = as.numeric(values (MotifDb)$sequenceCount[hitIndeces])
> names(sequenceCounts) = names(list)
> pwm1 = reverseComplement(list$"Hsapiens-JASPAR_CORE-CTCF-MA0139.1"[, 2:18]) ## t
> pwm2 = list$"Hsapiens-jolma2013-CTCF"
> n1 = sequenceCounts["Hsapiens-JASPAR_CORE-CTCF-MA0139.1"]
> n2 = sequenceCounts["Hsapiens-jolma2013-CTCF"]
> ## DiffLogo can also handle motifs of different length
> pwm_long = reverseComplement(list$"Hsapiens-JASPAR_CORE-CTCF-MA0139.1") ## rever
> pwm_short = list$"Hsapiens-jolma2013-CTCF"
```

Here, we import two motifs from the transcription factor CTCF from package *MotifDb* [3]. Alternatively, there are example PWMs in folder *extdata/pwm* as tab–delimited files shipped with the package DiffLogo. (extracted from [4]).

```
> ## import five DNA motifs from matrix
> motif_folder = "extdata/pwm"
> motif_names_dna = c("H1-hESC", "MCF7", "HeLa-S3", "HepG2", "HUVEC")
> motifs_dna = list()
> for (name in motif_names_dna) {
+   fileName = paste(motif_folder,"/",name,".pwm",sep="")
+   file = system.file(fileName, package = "DiffLogo")
+   motifs_dna[[name]] = getPwmFromPwmFile(file)
+ }
> sampleSizes_dna = c("H1-hESC"=100, "MCF7"=100, "HeLa-S3"=100, "HepG2"=100, "HUVE
> ## import three DNA motifs from table
> motif_folder = "extdata/alignments"
> motif_names_dna2 = c("Mad", "Max", "Myc")
> motifs_dna2 = list()
> for (name in motif_names_dna2) {
```

```
+   fileName = paste(motif_folder,"/",name,".txt",sep="")
+   file = system.file(fileName, package = "DiffLogo")
+   motifs_dna2[[name]] = getPwmFromAlignmentFile(file)
+ }
> ## import three ASN motifs from fasta files
> motif_folder = "extdata/alignments"
> motif_names_asn = c("F-box_fungi.seq", "F-box_metazoa.seq", "F-box_viridiplantae
> motifs_asn = list()
> for (name in motif_names_asn) {
+   fileName = paste(motif_folder,"/",name,".fa",sep="")
+   file = system.file(fileName, package = "DiffLogo")
+   motifs_asn[[name]] = getPwmFromFastaFile(file, FULL_ALPHABET)
+ }
```

Here, we import a set of five motifs from the folder *extdata/pwm*.

4 Plot sequence logo

The user is able to examine motifs using the classical sequence logo from package *seqLogo* [5].

```
> ## plot classic sequence logo
> library(seqLogo)
> seqLogo::seqLogo(pwm = pwm1)
```

Using the package DiffLogo, the user is also able to plot sequence logos using custom functions for stack height and base distribution. In case of *stackHeight=informationContent* and *baseDistribution=probabilities*, the result is equivalent to the result of package *seqLogo*

```
> ## plot custom sequence logo
> par(mfrow=c(2,1), pin=c(3, 1), mar = c(2, 4, 1, 1))
> DiffLogo::seqLogo(pwm = pwm1)

[1] "pwm must be of class matrix or data.frame. Trying to convert"

> DiffLogo::seqLogo(pwm = pwm2, stackHeight = sumProbabilities)

[1] "pwm must be of class matrix or data.frame. Trying to convert"

> par(mfrow=c(1,1), pin=c(1, 1), mar=c(5.1, 4.1, 4.1, 2.1))
```

5 Plot difference logo

The user is easily able to plot a difference logo for a pair of motifs.

```
> ## plot DiffLogo
> diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"

> ## diffLogoFromPwm is a convenience function for
> diffLogoObj = createDiffLogoObject(pwm1 = pwm1, pwm2 = pwm2)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"

> diffLogo(diffLogoObj)
> ## mark symbol stacks with significant changes
> diffLogoObj = enrichDiffLogoObjectWithPvalues(diffLogoObj, n1, n2)
> diffLogo(diffLogoObj)
> ## plot DiffLogo for PWMs of different length
> diffLogoFromPwm(pwm1 = pwm_long, pwm2 = pwm_short, align_pwms=TRUE)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

6 Plot table of difference logos

The user is easily able to plot a table of difference logos for a set of motifs.
By default, before drawing diffLogo, motifs will multiply aligned.

```
> ## plot table of difference logos for CTFC motifs (DNA)
> diffLogoTable(PWMs = motifs_dna)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
> ## plot table of difference logos for E-Box motifs (DNA)
> diffLogoTable(PWMs = motifs_dna2)
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
> ## plot table of difference logos for F-Box motifs (ASN)
> diffLogoTable(PWMs = motifs_asn, alphabet = FULL_ALPHABET)
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
> ## CTFC motifs (DNA) with asterisks to indicate significant differences
> diffLogoTable(PWMs = motifs_dna, sampleSizes = sampleSizes_dna)
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
> resolution = 300
> width = size * widthToHeightRatio
> height = size
> ## export single DiffLogo as pdf document
> fileName = "Comparison_of_two_motifs.pdf"
> pdf(file = fileName, width = width, height = height)
> diffLogoFromPwm(pwm1 = pwm1, pwm2 = pwm2)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"

> dev.off()

pdf
  2

> ## export DiffLogo table as png image
> fileName = "Comparison_of_multiple_motifs.png"
> png(
+   filename = fileName, res = resolution,
+   width = width * resolution, height = height * resolution)
> diffLogoTable(PWMs = motifs_dna)

[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

```
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

8 PWMs alignment

User is able to align a set of PWMs. Hanging ends are filled with an uniform distribution and details are given in the list from function `multipleLocalPwmsAlignment`.

```
> ## align pwms and than plot seqLogo of them
> pwms = list(pwm_long, pwm_short)
> multiple_pwms_alignment = multipleLocalPwmsAlignment(pwms)
> aligned_pwms = extendPwmsFromAlignmentVector(pwms, multiple_pwms_alignment$align
> ## seqLogo of aligned pwms
> layout(mat = matrix(data = 1:4, nrow = 2, ncol = 2))
> DiffLogo::seqLogo(pwms[[1]], main = "Unaligned")

[1] "pwm must be of class matrix or data.frame. Trying to convert"

> DiffLogo::seqLogo(pwms[[2]])

[1] "pwm must be of class matrix or data.frame. Trying to convert"

> DiffLogo::seqLogo(aligned_pwms[[1]], main = "Aligned")

[1] "pwm must be of class matrix or data.frame. Trying to convert"

> DiffLogo::seqLogo(aligned_pwms[[2]])

[1] "pwm must be of class matrix or data.frame. Trying to convert"
```

Literature

[1] http://en.wikipedia.org/wiki/Position_weight_matrix
[2] Schneider TD, Stephens RM. 1990. Sequence Logos: A New Way to Display Consensus Sequences. Nucleic Acids Res. 18:6097-6100
[3] Shannon P (2014). MotifDb: An Annotated Collection of Protein-DNA Binding Sequence Motifs. R package version 1.10.0.
[4] Eggeling, R., Gohr, A., Keilwagen, J., Mohr, M., Posch, S., Smith, A.D., Grosse, I.: On the value of intra-motifdependencies of human insulator protein ctcf. PLoS ONE 9(1), 85629 (2014). doi:10.1371/journal.pone.0085629
[5] Bembom O. seqLogo: Sequence logos for DNA sequence alignments. R package version 1.34.0.