Enrichment Depletion Logo plots with String symbols using *Logolas*

Kushal K Dey, Dongyue Xie, Matthew Stephens

[1em] Dept. of Statistics, The University of Chicago
*Correspondending Email: kkdey@uchicago.edu

February 4, 2018

Abstract

Sequence logo plots have become a standard graphical tool for visualizing sequence motifs in DNA, RNA or protein sequences. However standard logo plots primarily highlight enrichment of symbols, and may fail to highlight interesting depletions. Current alternatives that try to highlight depletion often produce visually cluttered logos. We introduce a new sequence logo plot, the EDLogo plot, that highlights both enrichment and depletion, while minimizing visual clutter. We provide an easy-to-use and highly customizable R package Logolas to produce a range of logo plots, including EDLogo plots. This software also allows elements in the logo plot to be strings of characters, rather than a single character, extending the range of applications beyond the usual DNA, RNA or protein sequences. We illustrate our methods and software on applications to transcription factor binding site motifs, protein sequence alignments and cancer mutation signature profiles. Our new EDLogo plots, and flexible software implementation, can help data analysts visualize both enrichment and depletion of characters (DNA sequence bases, amino acids, etc) across a wide range of applications.

***Logolas* version:** 2.0.1 [1]

[1]This document used the vignette from *Bioconductor* package *CountClust, DESeq2* as *knitr* template

Contents

1 Introduction

Compared to the existing packages for plotting sequence logos (*seqLogo*, *seq2Logo*, *motifStack* etc), *Logolas* offers several new features that makes logo visualization a more generic tool with potential applications in a much wider scope of problems.

- **Enrichment Depletion Logo (EDLogo)** : General logo plotting softwares highlight only enrichment of certain symbols, but Logolas allows the user to highlight both enrichment and depletion of symbols at any position, leading to more parsimonious and visually appealing representation.

- **String symbols** : General logo building softwares have limited library of symbols usually restricted to English alphabets. Logolas allows the user to plot symbols for any alphanumeric string, comprising of English alphabets, numbers, punctuation marks, arrows etc. It also provides an easy interface for the user to create her own logo and add to the library of symbols that can be plotted.

- **Dirichlet Adaptive Shrinkage** : Logolas provides a statistical approach to adaptively scale the heights of the logos based on the number of aligned sequences.

- **Better customizations** : Logolas offers several new color palettes, fill and border styles, several options for determining heights of the logos etc. Also, they can be plotted in multiple panels and combined with ggplot2 graphics.

2 *Logolas* Installation

Logolas loads as dependencies the following CRAN-R package : *grid*, *gridExtra*, *SQUAREM*, *LaplacesDemon*, *Matrix*, *RColorBrewer*.

The Bioc version of *Logolas* can be installed as follows

```
source("http://bioconductor.org/biocLite.R")
biocLite("Logolas")
```

For installing the developmental version of *Logolas* from Github, the user is required to have the *devtools* package and then run the following command.

```
devtools::install_github('kkdey/Logolas')
```

Load *Logolas* into R

```
library(Logolas)
```

3 Data Type

3.1 Data Format

Logolas accepts two data formats as input

- a vector of aligned character sequences (may be DNA, RNA or amino acid sequences), each of same length (see Example 1 below)

- a positional frequency (weight) matrix, termed PFM (PWM), with the symbols to be plotted along the rows and the positions of aligned sequences, from which the matrix is generated, along the columns. (see Example 2)

3.2 String Data example

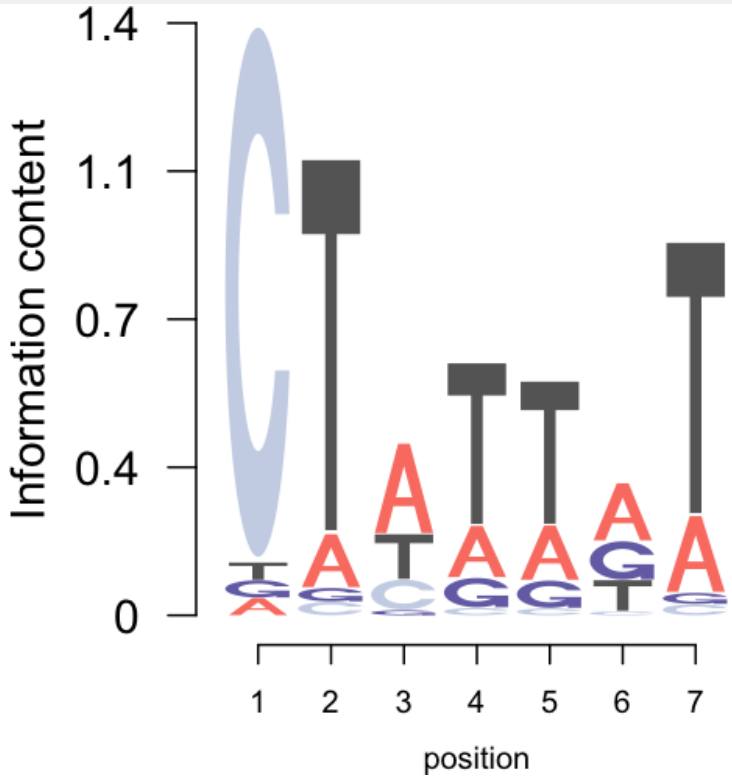
Consider aligned strings of characters

```
sequence <- c("CTATTGT", "CTCTTAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTGAAT",
              "CTTAGAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTTTAT", "CTATAGT",
              "CTATTTT", "CTTATAT", "CTATATT", "CTCATTT", "CTTATTT", "CAATAGT",
              "CATTTGA", "CTCTTAT", "CTATTAT", "CTTTTAT", "CTATAAT", "CTTAGGT",
```

```
                    "CTATTGT", "CTCATGT", "CTATAGT", "CTCGTTA", "CTAGAAT", "CAATGGT")
```

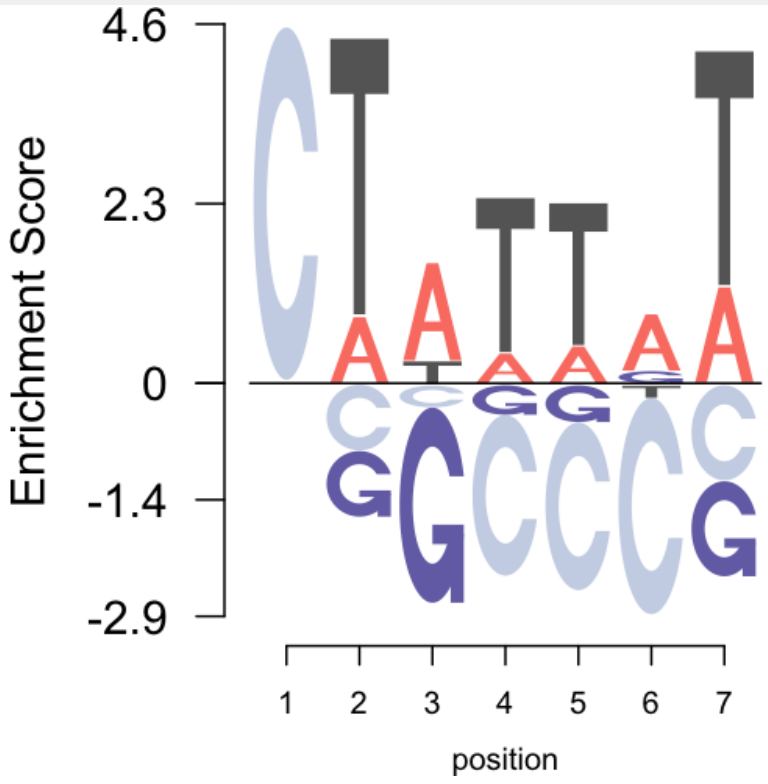
The logo plots (both standard and Enrichment Depletion Logo) can be plotted using the **logomaker()** function.

```
logomaker(sequence, type = "Logo")
```



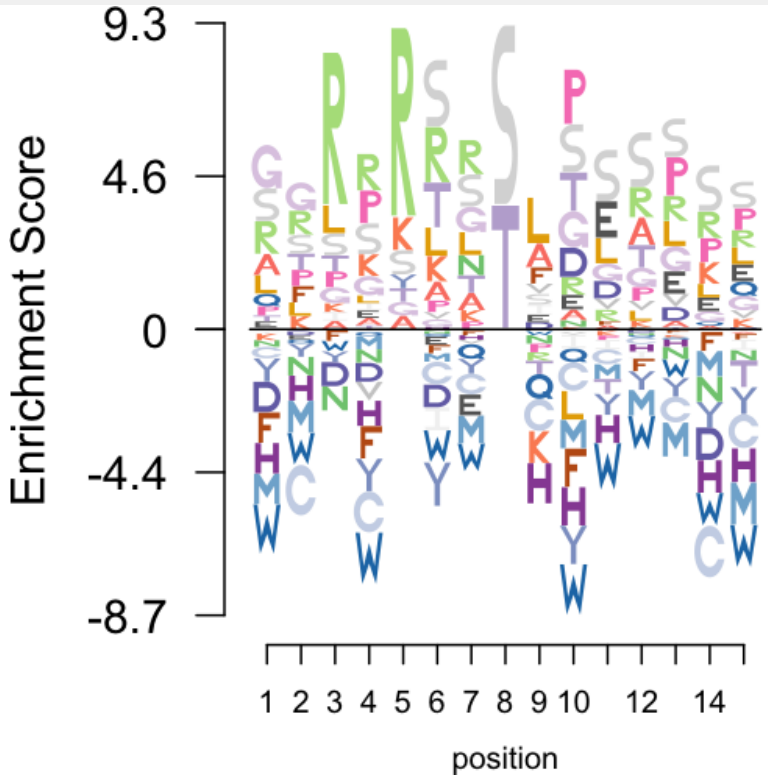
Enrichment Depletion Logo plots with String symbols using *Logolas*

```
logomaker(sequence, type = "EDLogo")
```



Instead of DNA.RNA sequqnce as above, one can also use amino acid character sequences.

```
library(ggseqlogo)
data(ggseqlogo_sample)
sequence <- seqs_aa$AKT1
logomaker(sequence, type = "EDLogo")
```



3.3 Positional Frequency (Weight) Matrix

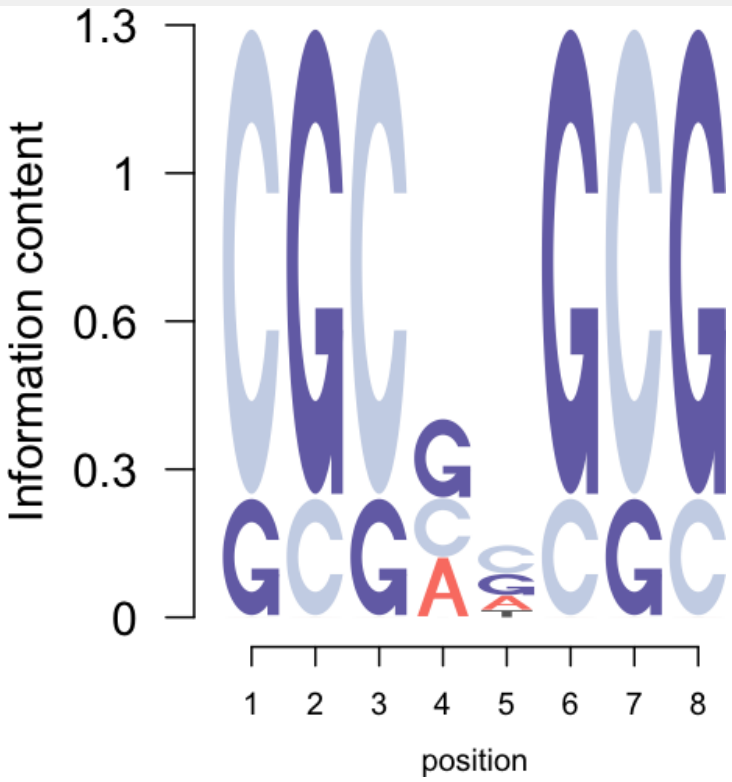
We now see an example of positional weight matrix (PWM) as input to **logomaker()**.

```
data(seqlogo_example)
```

We plot the logo plots for this PWM matrix.

The `return_heights= TRUE` outputs the information content at each position for the standard logo plot (type = "Logo") and the heights of the stacks along the positive and negative Y axis, along with the breakdown of the height due to different characters for the EDLogo plot (type = "EDLogo").

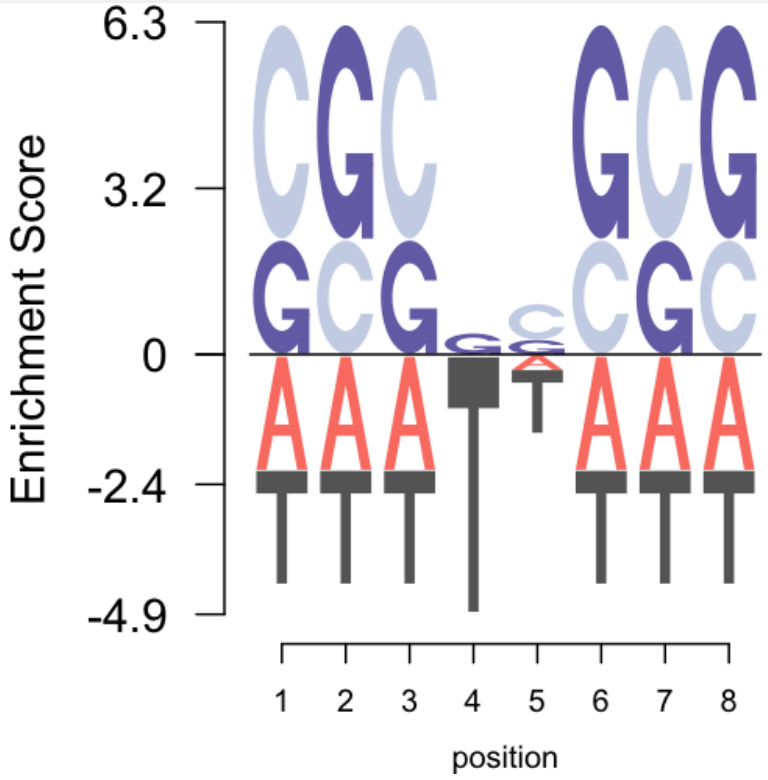
```
logomaker(seqlogo_example, type = "Logo", return_heights = TRUE)
```




```
## [1] 1.275 1.275 1.275 0.428 0.153 1.275 1.275 1.275
```

Enrichment Depletion Logo plots with String symbols using *Logolas*

```
logomaker(seqlogo_example, type = "EDLogo", return_heights = TRUE)
```




```
## $pos_ic
##     1     2     3     4     5     6     7     8
## 6.311 6.311 6.311 0.403 0.965 6.311 6.311 6.311
##
## $neg_ic
##    1    2    3    4    5    6    7    8
## 4.36 4.36 4.36 4.93 1.49 4.36 4.36 4.36
##
## $table_mat_pos_norm
##       1     2     3 4     5     6     7     8
## A 0.000 0.000 0.000 0 0.000 0.000 0.000 0.000
## C 0.654 0.346 0.654 0 0.709 0.346 0.654 0.346
## G 0.346 0.654 0.346 1 0.291 0.654 0.346 0.654
## T 0.000 0.000 0.000 0 0.000 0.000 0.000 0.000
##
## $table_mat_neg_norm
##     1   2   3 4     5   6   7   8
## A 0.5 0.5 0.5 0 0.188 0.5 0.5 0.5
## C 0.0 0.0 0.0 0 0.000 0.0 0.0 0.0
## G 0.0 0.0 0.0 0 0.000 0.0 0.0 0.0
## T 0.5 0.5 0.5 1 0.812 0.5 0.5 0.5
```

4 Configuring Logos

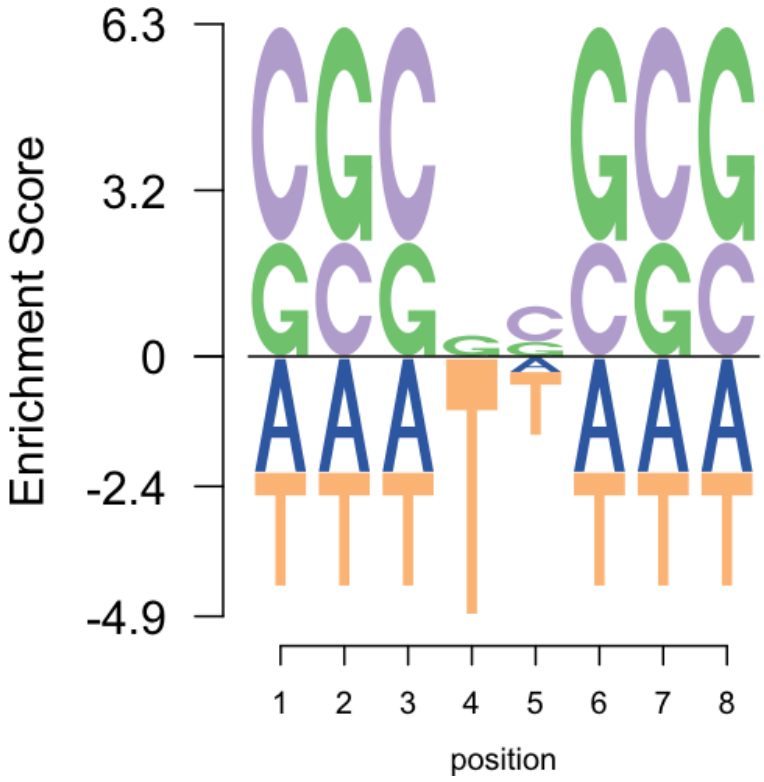
4.1 Coloring schemes

The **logomaker()** function provides three arguments to set the colors for the logos, a `color_type` specifying the scheme of coloring used, `colors` denoting the cohort of colors used and a `color_seed` argument determining how sampling is done from this cohort.

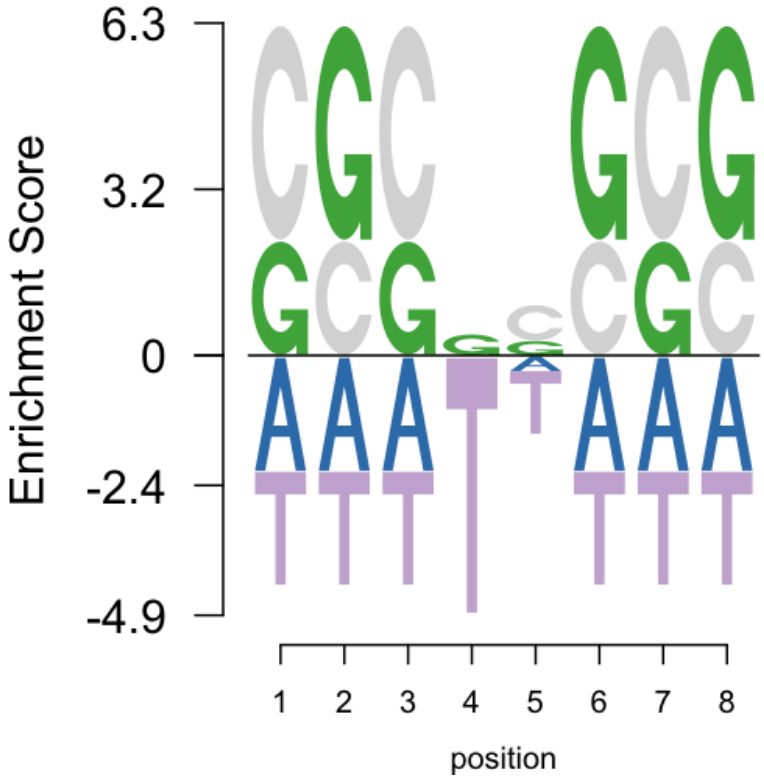
The `color_type` argument can be of three types, `per_row`, `per_column` and `per_symbol`. **colors** element is a cohort of colors (chosen suitably large) from which distinct colors are chosen based on distinct `color_type`. The number of colors chosen is of same length as number of rows in table for `per-row` (assigning a color to each string), of same length as number of columns in table for `per-column` (assuming a color for each column), or a distinct color for a distinct symbol in `per-symbol`. The length of **colors** should be as large as the number of colors to be chosen in each scenario. The default `color_type` is `per-row` and default **colors** comprises of a large cohort of nearly 70 distinct colors from which colors are sampled using the `color_seed` argument.

```
logomaker(seqlogo_example, color_type = "per_row",
          colors = c("#7FC97F", "#BEAED4", "#FDC086", "#386CB0"),
          type = "EDLogo")
```

Enrichment Depletion Logo plots with String symbols using *Logolas*



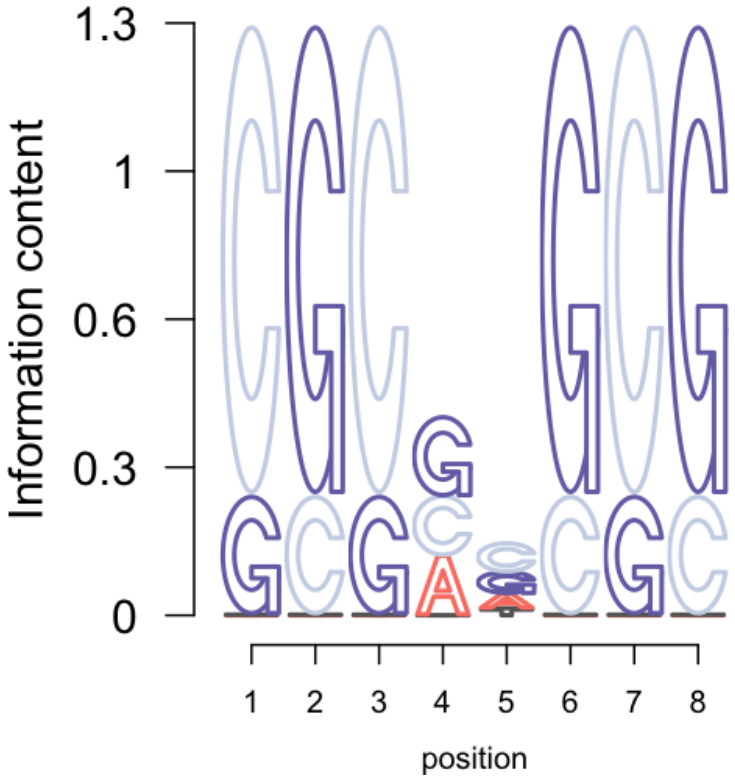

```
logomaker(seqlogo_example, type = "EDLogo", color_seed = 1500)
```



4.2 Styles of symbols

Besides the default style with filled symbols for each character, one can also use characters with border styles. For the standard logo plot, this is accomplished by the `tofill` control argument.

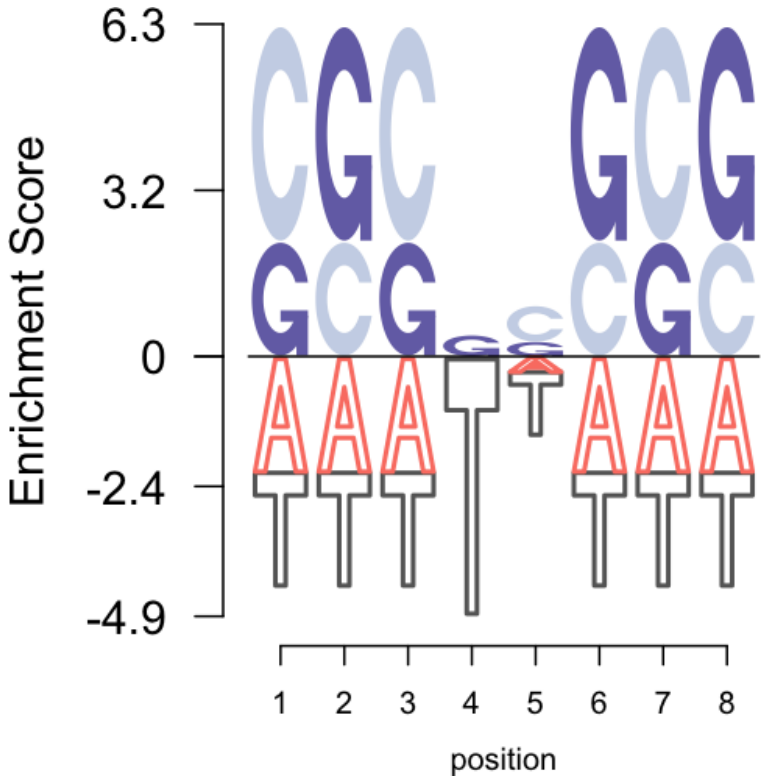
```
logomaker(seqlogo_example, type = "Logo",
          logo_control = list(control = list(tofill= FALSE)))
```



For an EDLogo plot, the arguments `tofill_pos` and `tofill_neg` represent the coloring scheme for the positive and the negative axes in an EDLogo plot.

```
logomaker(seqlogo_example, type = "EDLogo",
          logo_control = list(control = list(tofill_pos = TRUE,
                                             tofill_neg = FALSE)))
```

Enrichment Depletion Logo plots with String symbols using *Logolas*



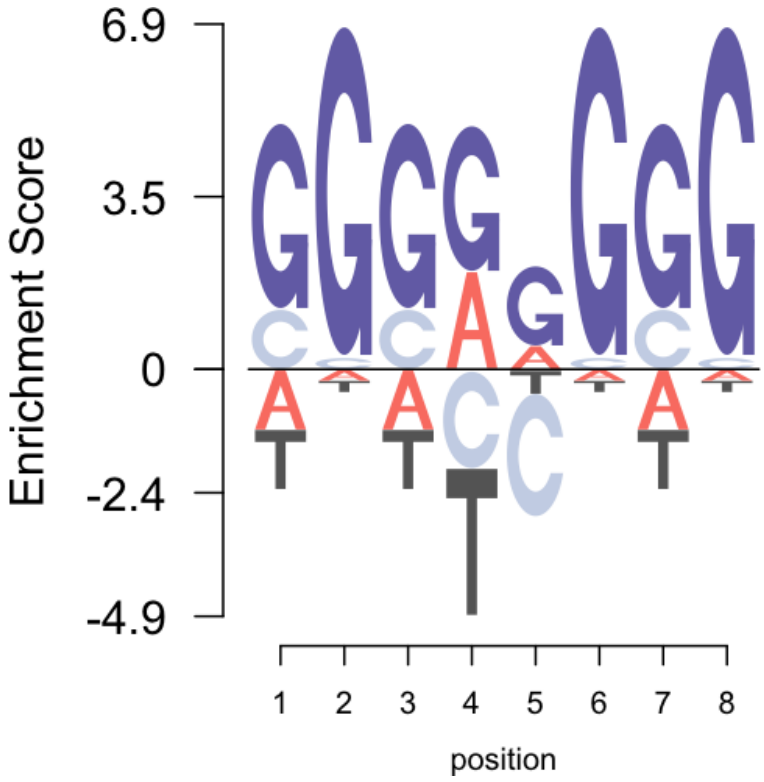
4.3 Background Info

Logolas allows the user to scale the data based on a specified background information. The background information can be incorporated in the argument bg. The default value is NULL, in which case equal probability is assigned to each symbol. The user can however specify a vector (equal to in length to the number of symbols) which specifies the background probability for each symbol and assumes this background probability to be the same across the columns (sites), or a matrix, whose each cell specifies the background probability of the symbols for each position.

First example with bg as a vector.

```
bg <- c(0.05, 0.90, 0.03, 0.05)
names(bg) <- c("A", "C", "G", "T")
logomaker(seqlogo_example, bg=bg, type = "EDLogo")
```

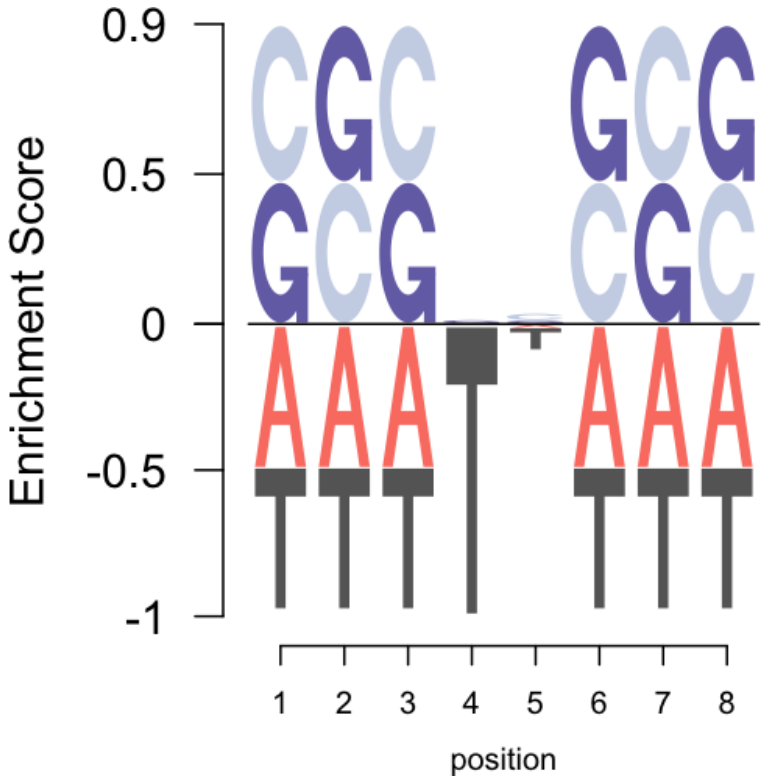
Enrichment Depletion Logo plots with String symbols using *Logolas*



Second example with bg as a matrix.

```
logomaker(seqlogo_example, bg=(seqlogo_example+1e-02), type = "EDLogo")
```

Enrichment Depletion Logo plots with String symbols using *Logolas*

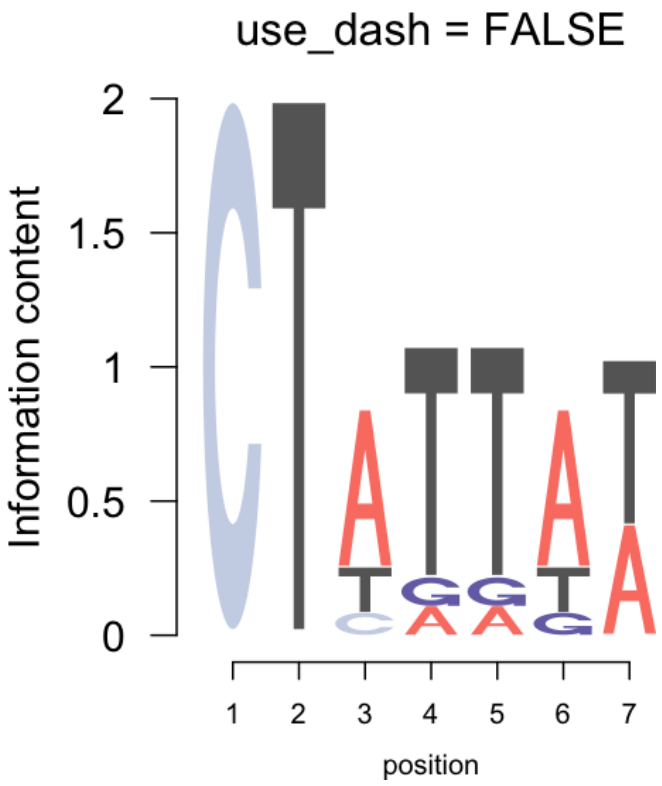


5 Adaptive scaling of logos

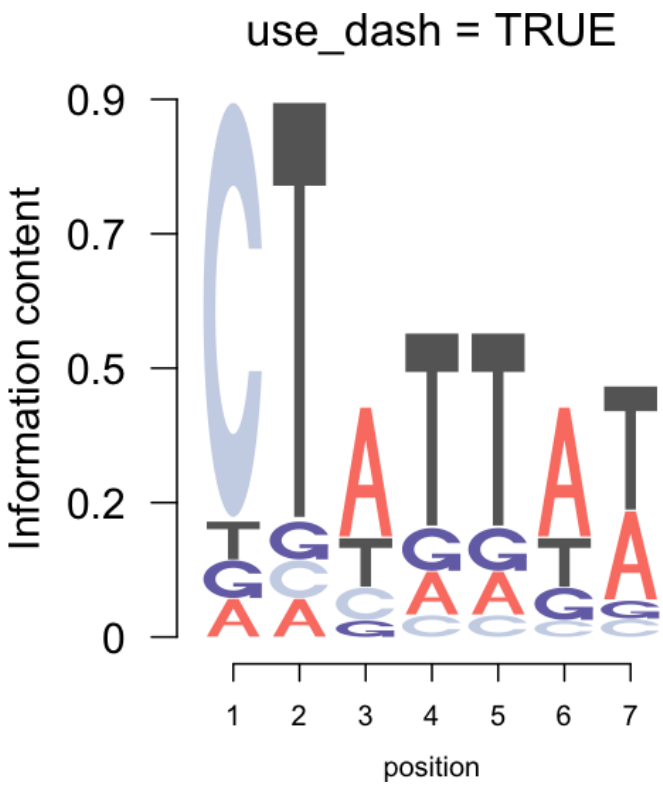
Logolas allows the user to perform adaptive scaling of the stack heights in a logo plot based on the number of aligned sequences, using the use_dash argument. This scaling is performed only when the data input into the **logomaker()** function is a vector of sequences or a position frequency (PFM) matrix. We show an example with and without the use_dash argument.

```
sequence <- c("CTATTGT", "CTCTTAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTGAAT",
              "CTTAGAT", "CTATTAA", "CTATTTA", "CTATTAT")
logomaker(sequence, use_dash = FALSE, type = "Logo",
          logo_control = list(pop_name = "use_dash = FALSE"))
```

Enrichment Depletion Logo plots with String symbols using *Logolas*




```
logomaker(sequence, type = "Logo", logo_control = list(pop_name = "use_dash = TRUE"))
```



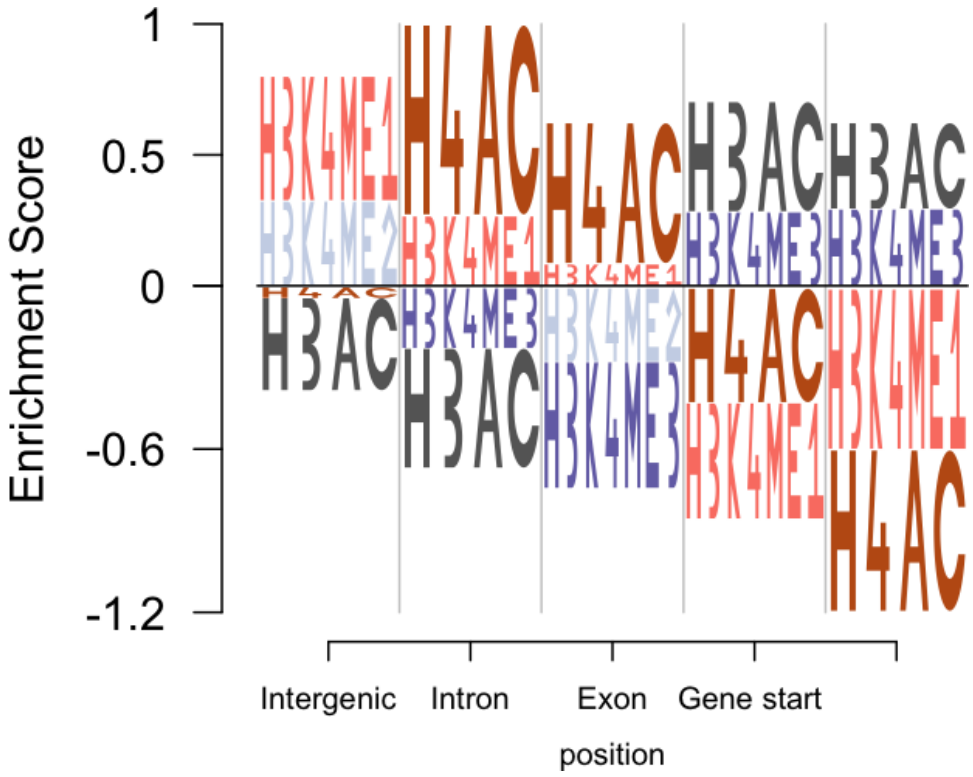
The adaptive scaling is performed by the Dirichlet Adaptive Shrinkage method, the details of which can be viewed at our **dashr** package repository https://github.com/kkdey/dashr.

6 String symbols

Logolas allows the user to plot symbols not just for characters as we saw in previous examples, but for any alphanumeric string. We present two examples - one for representing mutation signature and another for representing histone marks composition.

Histone marks string symbols example

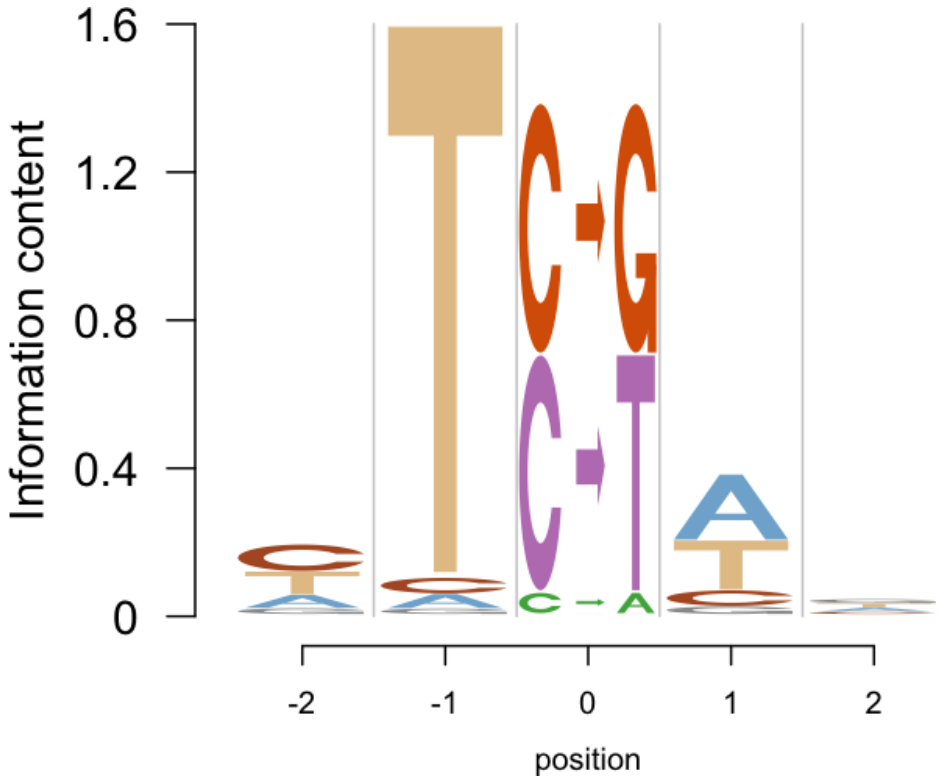
```
data("histone_marks")
logomaker(histone_marks$mat, bg=histone_marks$bgmat, type = "EDLogo")
```



Mutation signature string and character mix example.

```
data("mutation_sig")
logomaker(mutation_sig, type = "Logo", color_seed = 3000)
```

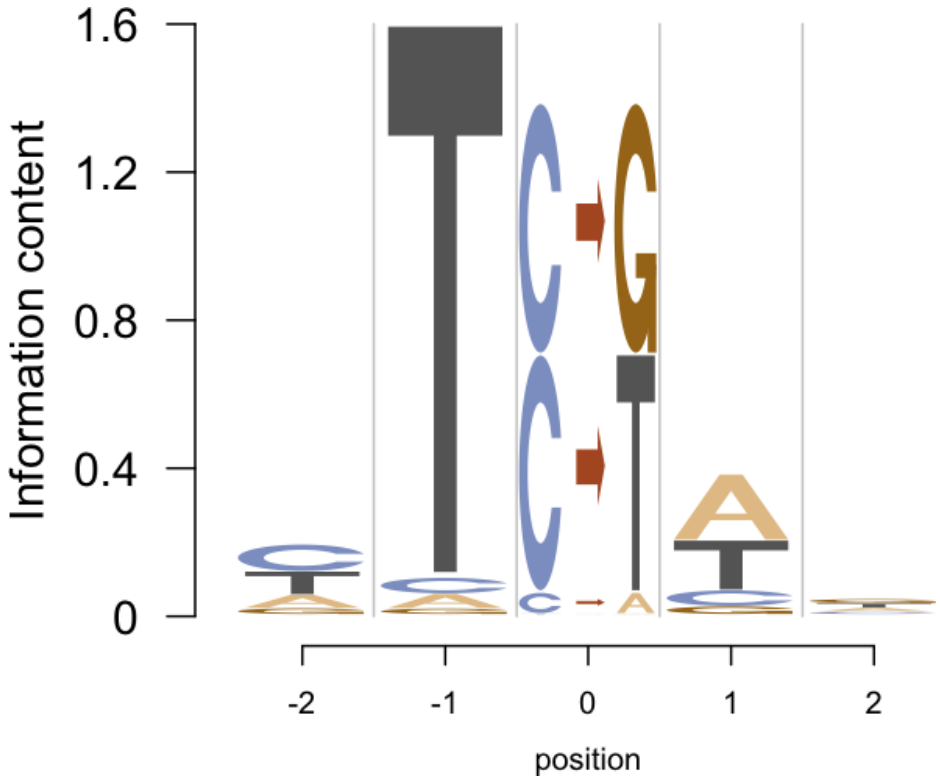
Enrichment Depletion Logo plots with String symbols using *Logolas*



The user may want to have distinct colors for distinct symbols. This is where we use the *persymbol* option for `color_type`.

```
logomaker(mutation_sig, type = "Logo", color_type = "per_symbol",  color_seed = 2300)
```

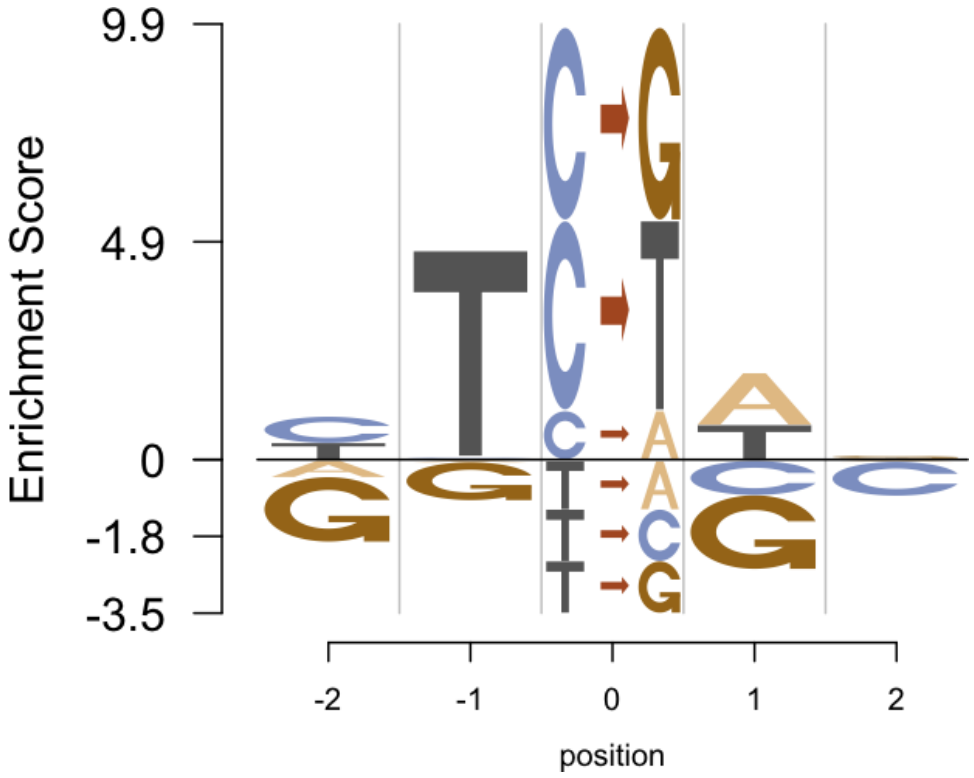
Enrichment Depletion Logo plots with String symbols using *Logolas*



The corresponding EDLogo

```
logomaker(mutation_sig, type = "EDLogo", color_type = "per_symbol",  color_seed = 2300)
```

Enrichment Depletion Logo plots with String symbols using *Logolas*



7 Extras

7.1 Consensus Sequence

Logolas provides a new nomenclature to geneerate consensus sequence from a positional frequency (weight) matrix or from a vector of aligned sequences. This is performed by the `GetConsensusSeq()` function.

```
sequence <- c("CTATTGT", "CTCTTAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTGAAT",
              "CTTAGAT", "CTATTAA", "CTATTTA", "CTATTAT")
GetConsensusSeq(sequence)

## [1] "C T (Ag) T T (Ac) (TA)"
```

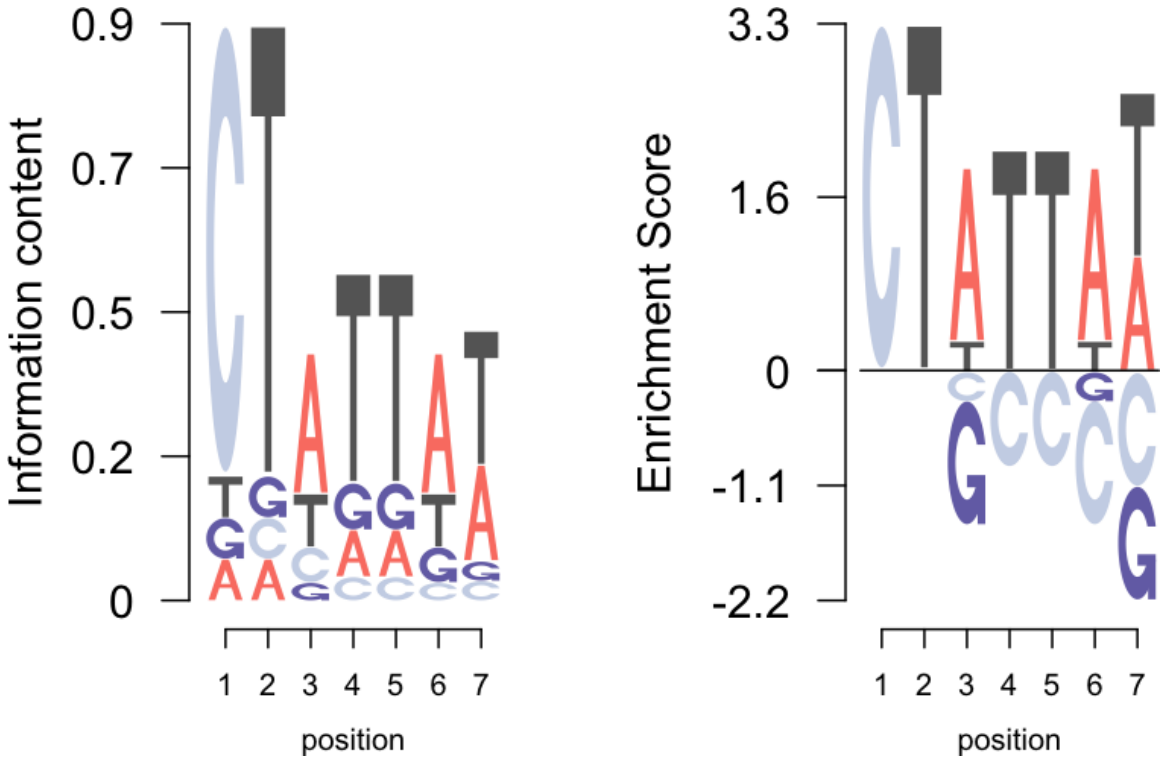
In the sequence, a position represented by (Ag) would mean enrichment in A and depletion in G at that position. One can input a PWM or PFM matrix with A, C, G and T as row names in the `GetConsensusSeq()` function as well.

7.2 Multiple panels plots

Logolas plots can be plotted in multiple panels, as depicted below.

```
sequence <- c("CTATTGT", "CTCTTAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTGAAT",
              "CTTAGAT", "CTATTAA", "CTATTTA", "CTATTAT")
Logolas::get_viewport_logo(1, 2, heights_1 = 20) ## first arg: num of rows in panel, secon
library(grid)
seekViewport(paste0("plotlogo", 1))
logomaker(sequence, type = "Logo", logo_control = list(newpage = FALSE))

seekViewport(paste0("plotlogo", 2))
logomaker(sequence, type = "EDLogo", logo_control = list(newpage = FALSE))
```

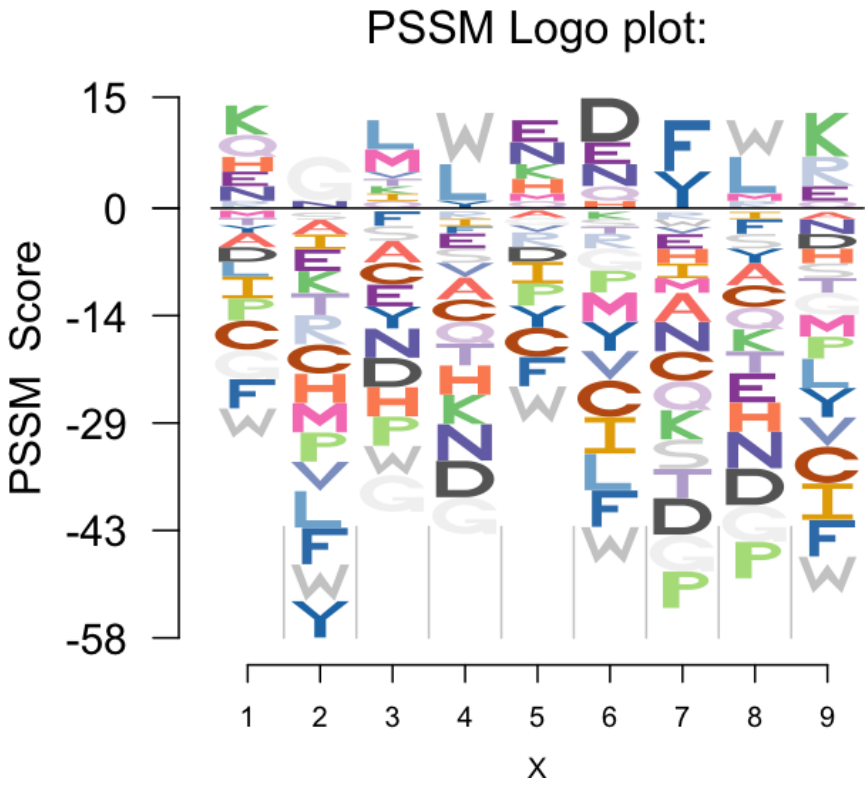


In the same way, ggplot2 graphics can also be combined with *Logolas* plots.

7.3 PSSM logos

While **logomaker** takes a PFM, PWM or a set of aligned sequences as input, sometimes, some position specific scores are only available to the user. In this case, one can use the `logo_pssm()` in *Logolas* to plot the scoring matrix.

```
data("pssm")
logo_pssm(pssm, control = list(round_off = 0))
```



The `round_off` comtrol argument specifies the number of points after decimal allowed in the axes of the plot.

8 Acknowledgements

The authors would like to acknowledge Oliver Bembom, the author of 'seqLogo' for acting as an inspiration and providing the foundation on which this package is created. We also thank Peter Carbonetto, Edward Wallace and John Blischak for helpful feedback and discussions.

9 Session Info

```
sessionInfo()
```

```
## R version 3.4.2 (2017-09-28)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ggseqlogo_0.1 Logolas_2.0.1 knitr_1.17
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.15         XVector_0.18.0       magrittr_1.5
##  [4] zlibbioc_1.24.0      IRanges_2.12.0       BiocGenerics_0.24.0
##  [7] munsell_0.4.3        gridBase_0.4-7       SQUAREM_2017.10-1
## [10] colorspace_1.3-2     rlang_0.1.6          stringr_1.2.0
## [13] highr_0.6            plyr_1.8.4           tools_3.4.2
## [16] parallel_3.4.2       gtable_0.2.0         htmltools_0.3.6
## [19] yaml_2.1.14          lazyeval_0.2.1       rprojroot_1.2
## [22] digest_0.6.12        tibble_1.3.4         RColorBrewer_1.1-2
## [25] ggplot2_2.2.1        S4Vectors_0.16.0     evaluate_0.10.1
## [28] LaplacesDemon_16.1.0 rmarkdown_1.8        stringi_1.1.6
## [31] compiler_3.4.2       Biostrings_2.46.0    scales_0.5.0
## [34] backports_1.1.0      stats4_3.4.2         BiocStyle_2.6.0
```

References

[1] Bembom O (2016). seqLogo: Sequence logos for DNA sequence alignments. R package version 1.40.0.

[2] Omar Wagih (2014). RWebLogo: plotting custom sequence logos. R package version 1.0.3. https://CRAN.R-project.org/package=RWebLogo

[3] Jianhong Ou and Lihua Julie Zhu (2015). motifStack: Plot stacked logos for single or multiple DNA, RNA and amino acid sequence. R package version 1.14.0.

[4] Shiraishi Y, Tremmel G, Miyano S, Stephens M (2015) A Simple Model-Based Approach to Inferring and Visualizing Cancer Mutation Signatures. PLoS Genet 11(12): e1005657. doi: 10.1371/journal.pgen.1005657

[5] Koch CM, Andrews RM, Flicek P, et al (2007). The landscape of histone modifications across 1% of the human genome in five human cell lines. Genome Research. 2007;17(6):691-707. doi:10.1101/gr.5704207.