Guided Logolas Tutorial

Kushal Dey, Dongyue Xie, Matthew Stephens

2018-04-19

We present an elaborate guided tutorial of how to use the **Logolas** R package. A pdf version of this vignette can be found here.

Features of Logolas

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.

Installation

Logolas loads as dependencies the following CRAN-R package : `grid`, `gridExtra`, `SQUAREM`, `LaplacesDemon`, `Matrix`, `RColorBrewer`. To run this vignette, the user also would be required to install the `ggseqlogo` package.

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 the Logolas package.

```
library(Logolas)
```

Accepted Data Types

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)

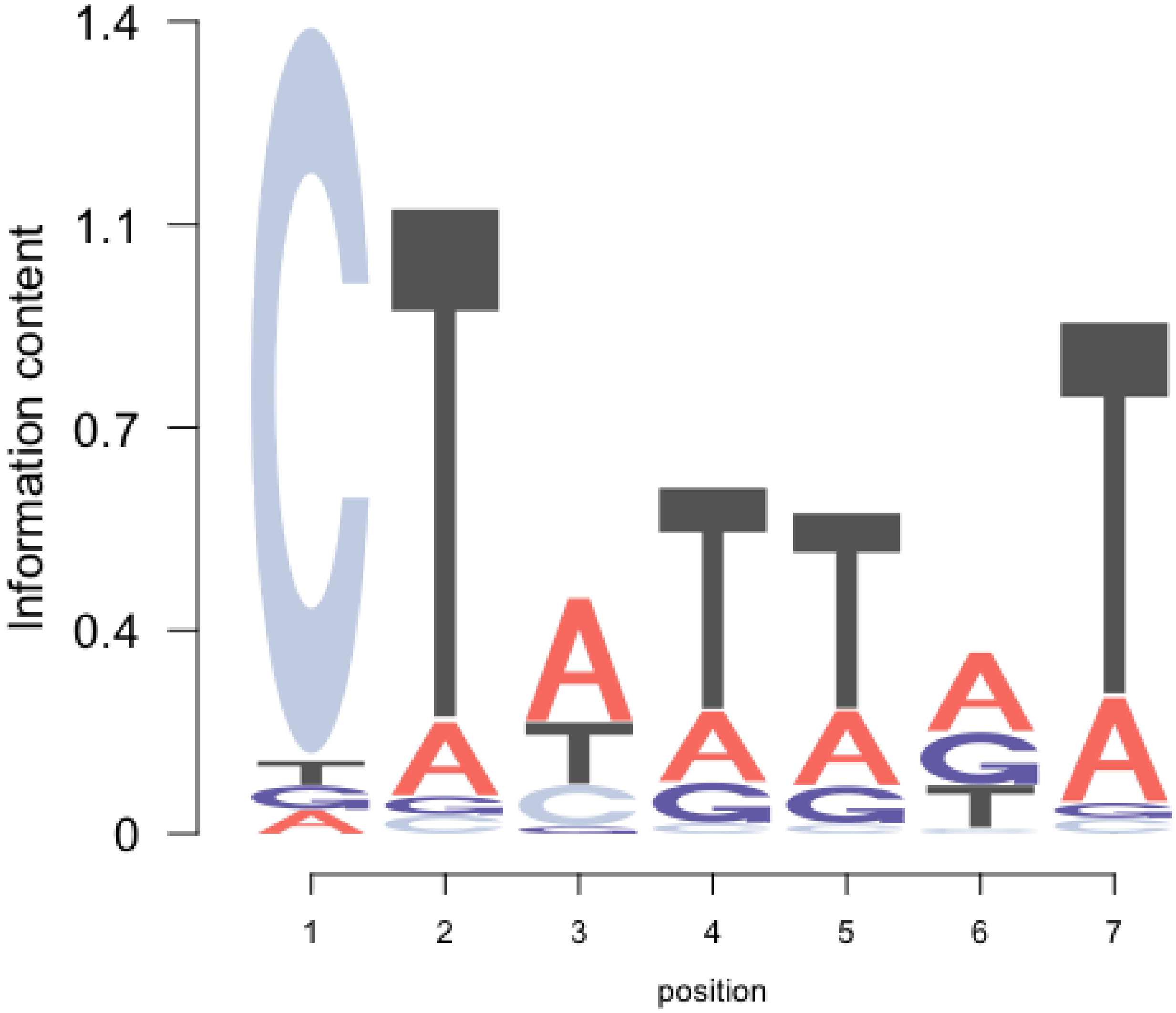
String Data example

Consider aligned strings of characters

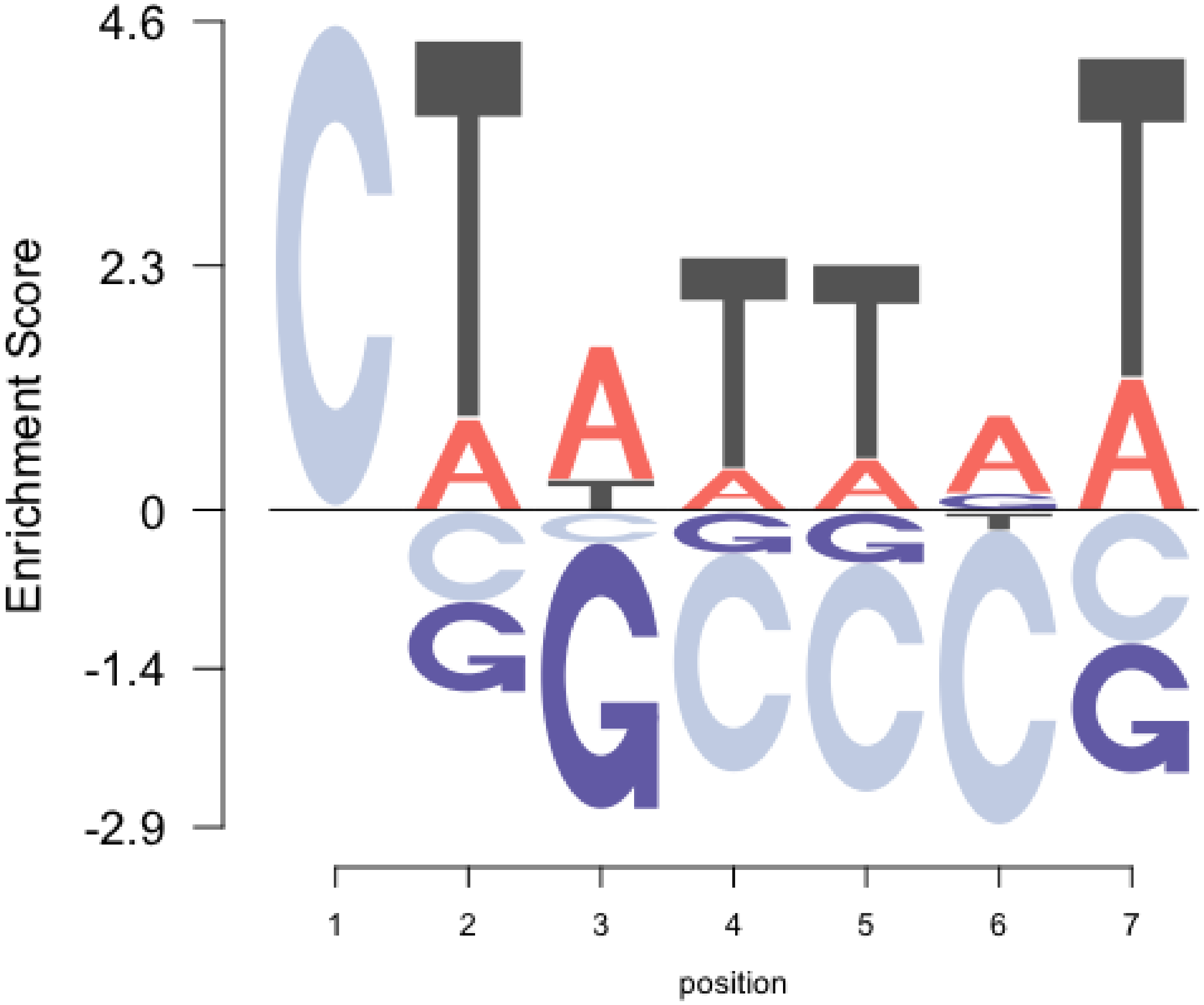
```
sequence <- c("CTATTGT", "CTCTTAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTGAAT",
              "CTTAGAT", "CTATTAA", "CTATTTA", "CTATTAT", "CTTTTAT", "CTATAGT",
              "CTATTTT", "CTTATAT", "CTATATT", "CTCATTT", "CTTATTT", "CAATAGT",
              "CATTTGA", "CTCTTAT", "CTATTAT", "CTTTTAT",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

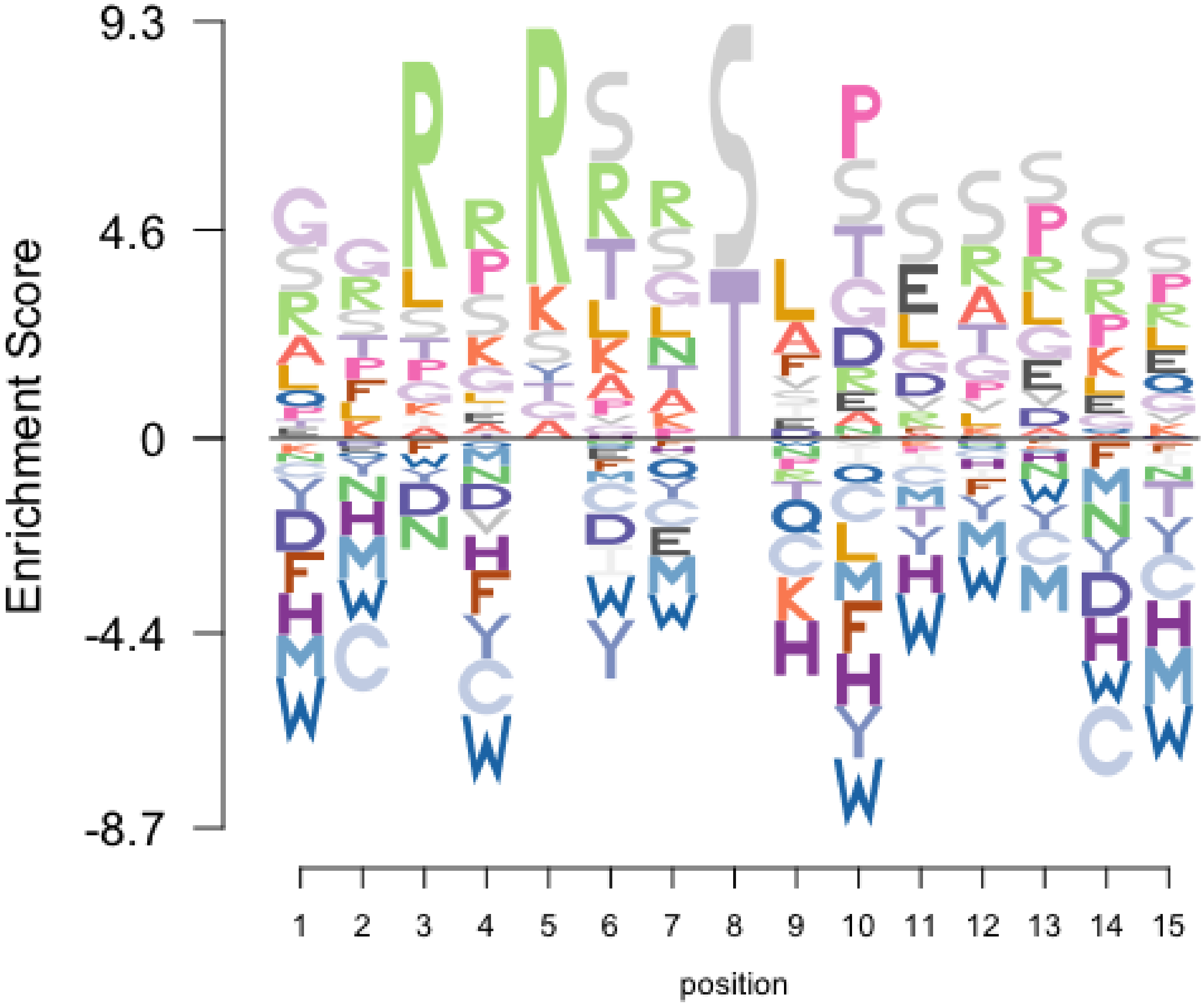


```
logomaker(sequence, type = "EDLogo")
```



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

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



Positional Frequency (Weight) Matrix

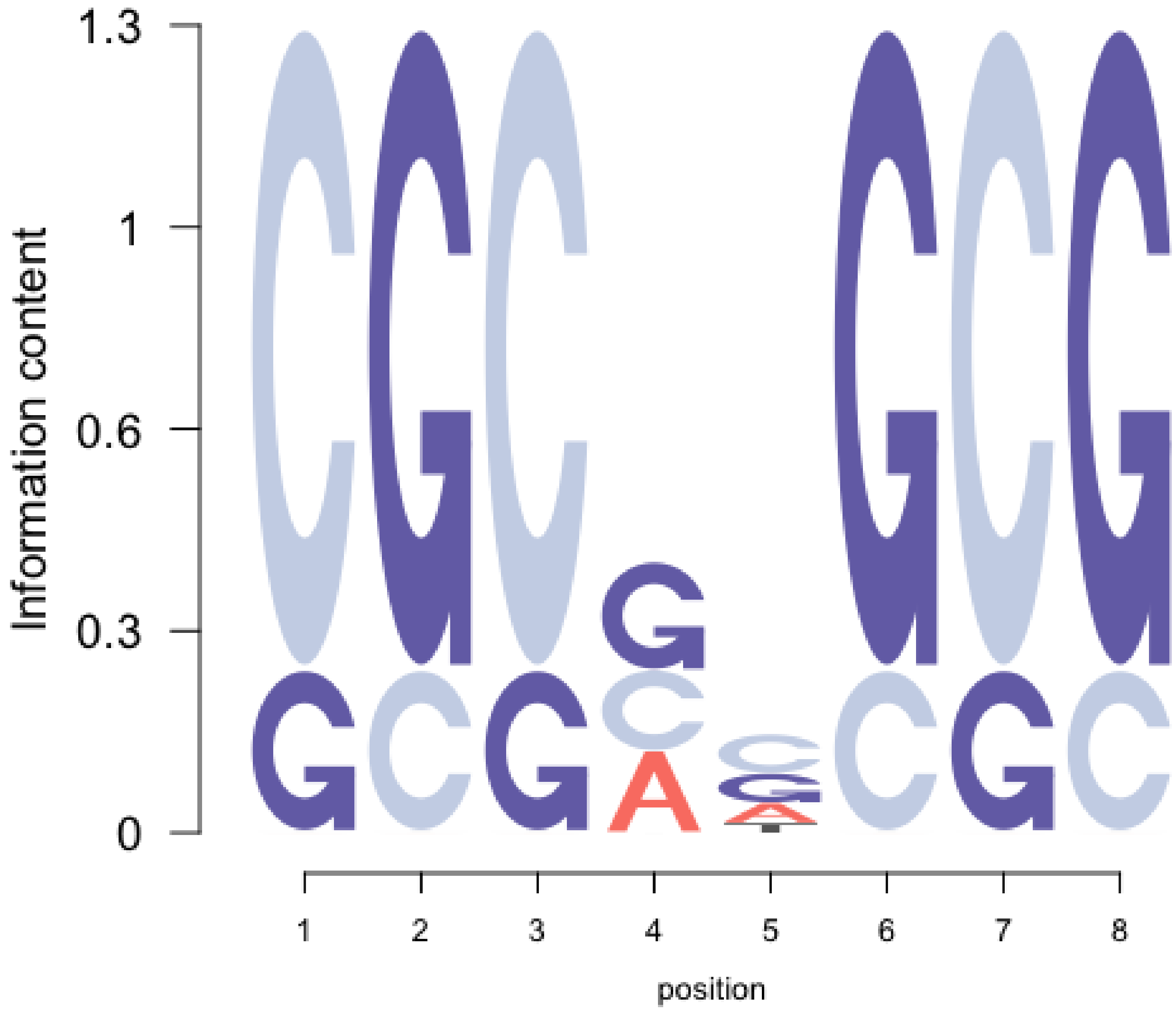
We now see an example of positional weight matrix (PWM) as input to **logomaker()**.

```
data("seqlogo_example")
seqlogo_example
```

```
##     1   2   3   4   5   6   7   8
## A 0.0 0.0 0.0 0.3 0.2 0.0 0.0 0.0
## C 0.8 0.2 0.8 0.3 0.4 0.2 0.8 0.2
## G 0.2 0.8 0.2 0.4 0.3 0.8 0.2 0.8
## T 0.0 0.0 0.0 0.0 0.1 0.0 0.0 0.0
```

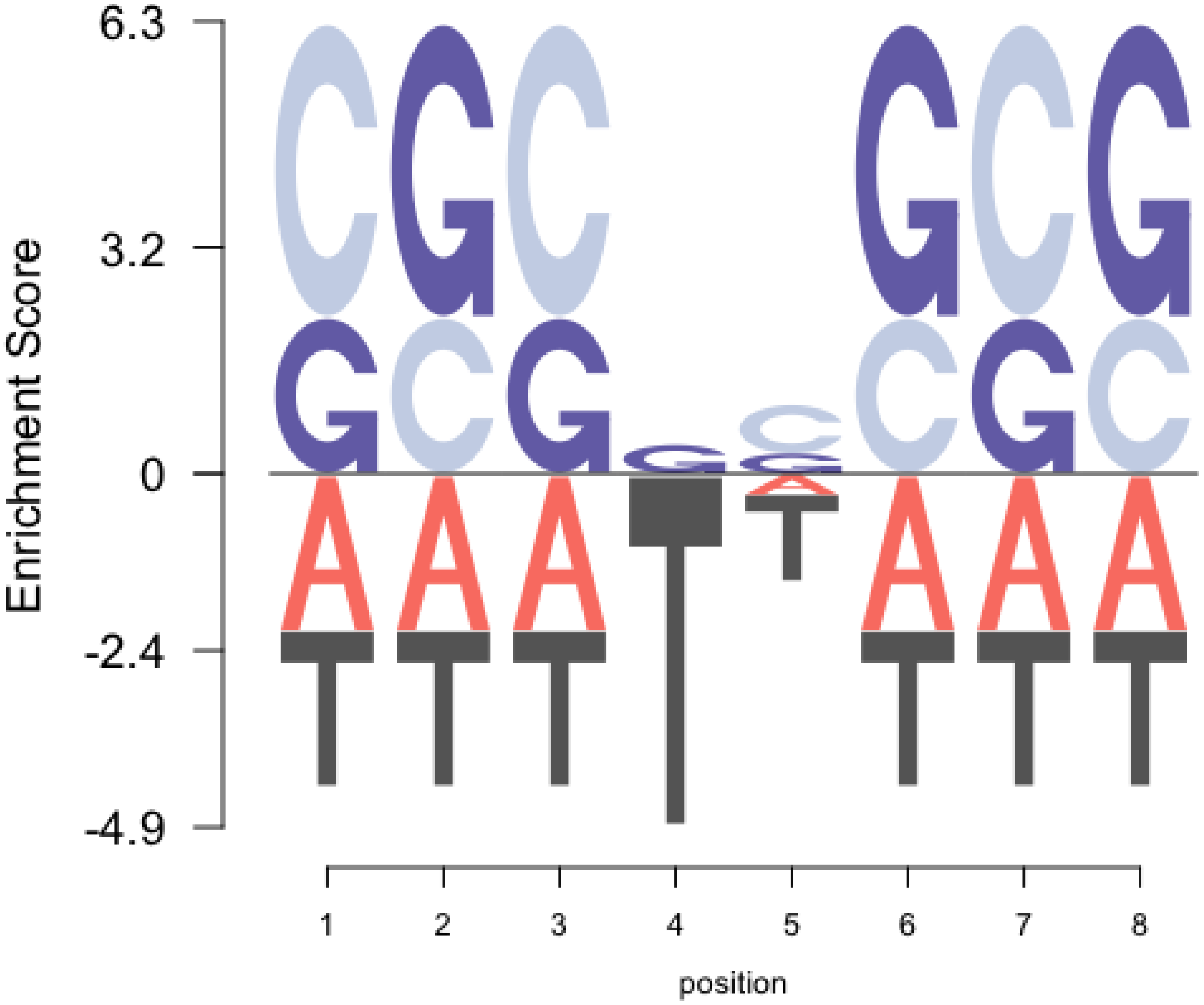
We plot the logo plots for this PWM matrix.

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



```
## [1] 1.2752 1.2752 1.2752 0.4277 0.1534 1.2752 1.2752 1.2752
```

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



```
## $pos_ic
##      1      2      3      4      5      6      7      8
## 6.3105 6.3105 6.3105 0.4031 0.9645 6.3105 6.3105 6.3105
##
## $neg_ic
##     1     2     3     4     5     6     7     8
## 4.364 4.364 4.364 4.925 1.493 4.364 4.364 4.364
##
## $table_mat_pos_norm
##        1      2      3 4     5      6      7      8
## A 0.0000 0.0000 0.0000 0 0.000 0.0000 0.0000 0.0000
## C 0.6542 0.3458 0.6542 0 0.709 0.3458 0.6542 0.3458
## G 0.3458 0.6542 0.3458 1 0.291 0.6542 0.3458 0.6542
## T 0.0000 0.0000 0.0000 0 0.000 0.0000 0.0000 0.0000
##
## $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
```

The 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”).

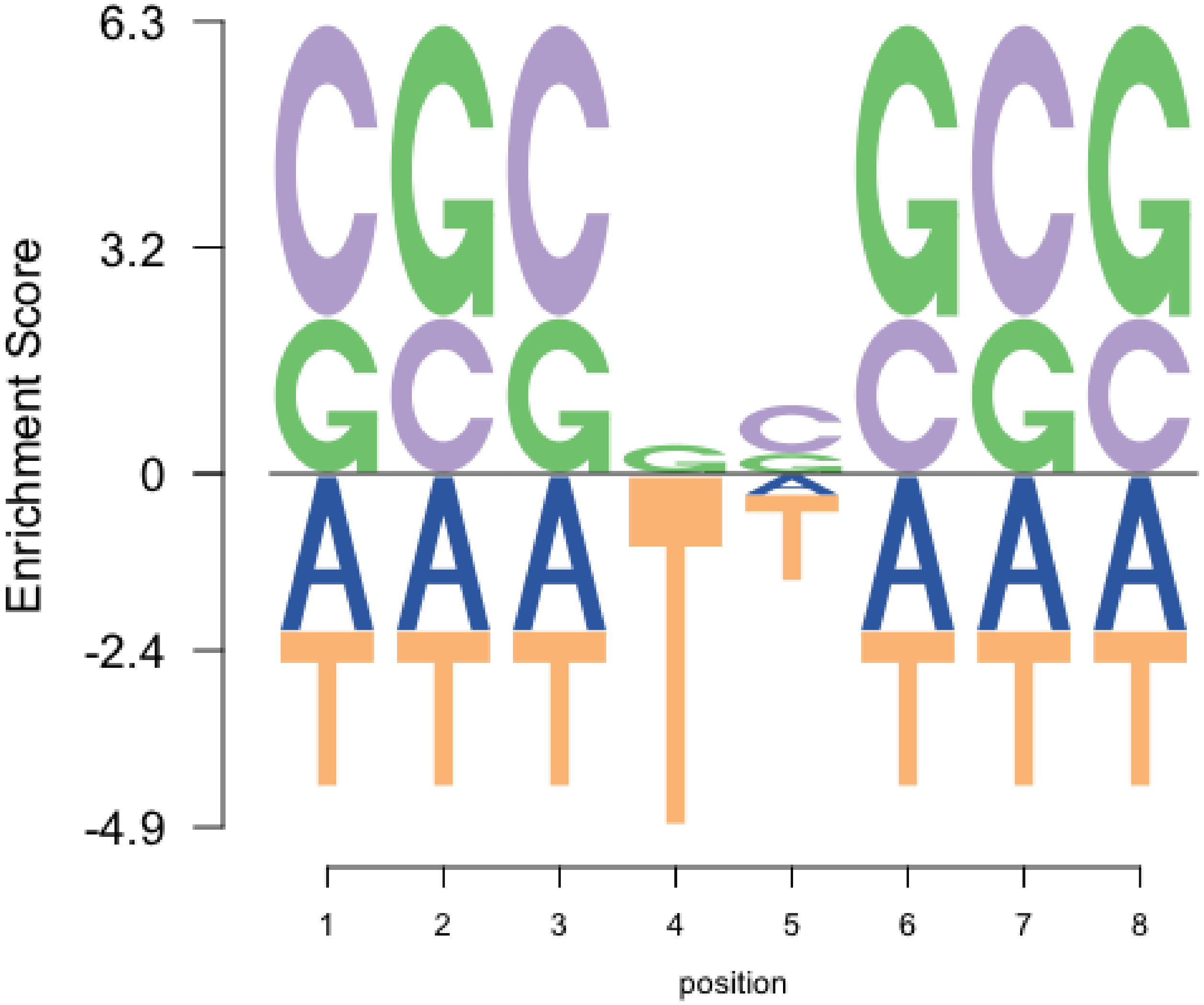
Configuring Logos

Coloring schemes

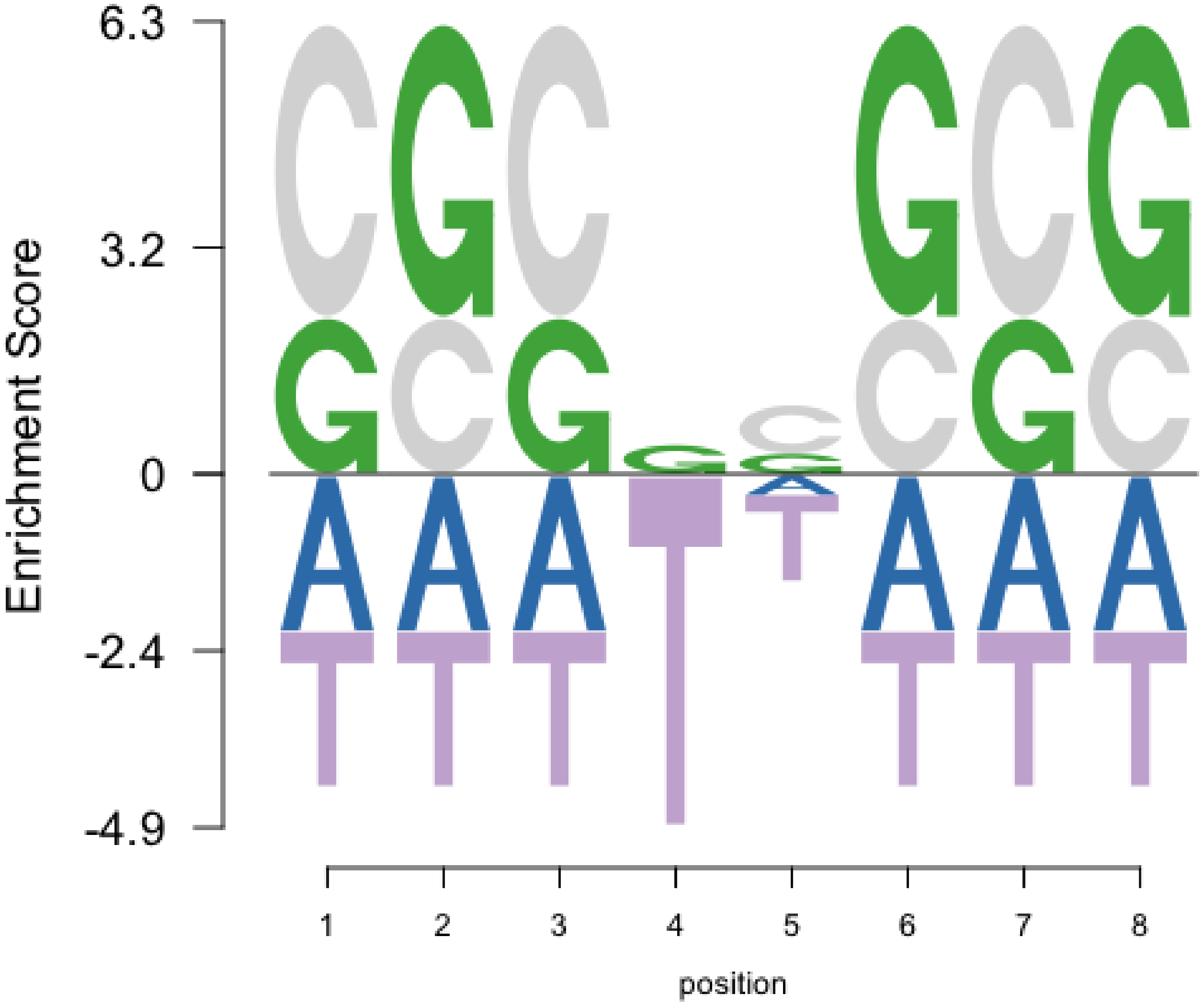
The **logomaker()** function provides three arguments to set the colors for the logos, a **color_type** specifying the scheme of coloring used, **colors** denoting the cohort of colors used and a **color_seed** argument determining how sampling is done from this cohort.

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.

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



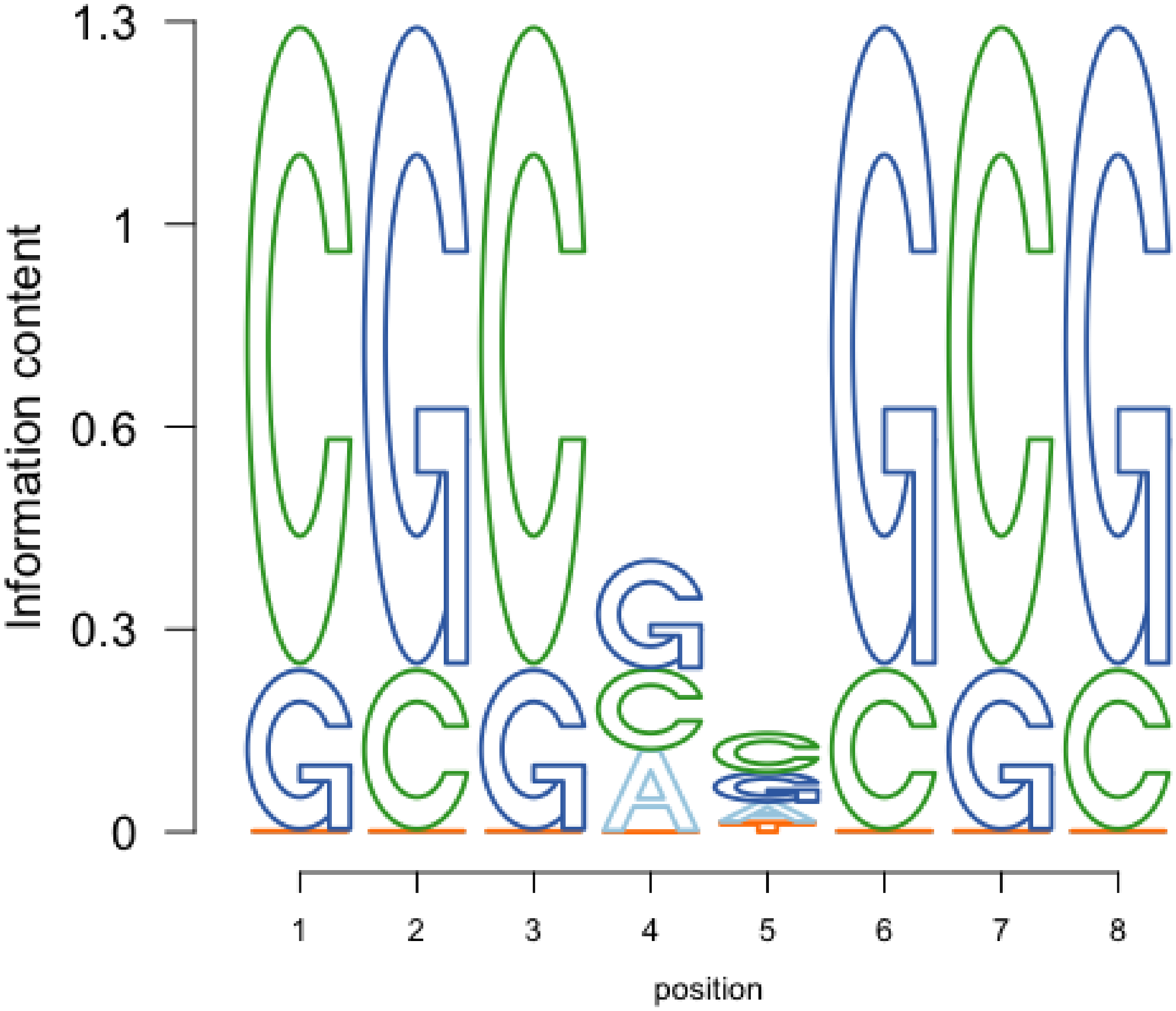
```r
logomaker(seqlogo_example, type = "EDLogo", color_seed = 1500)
```



Styles of symbols

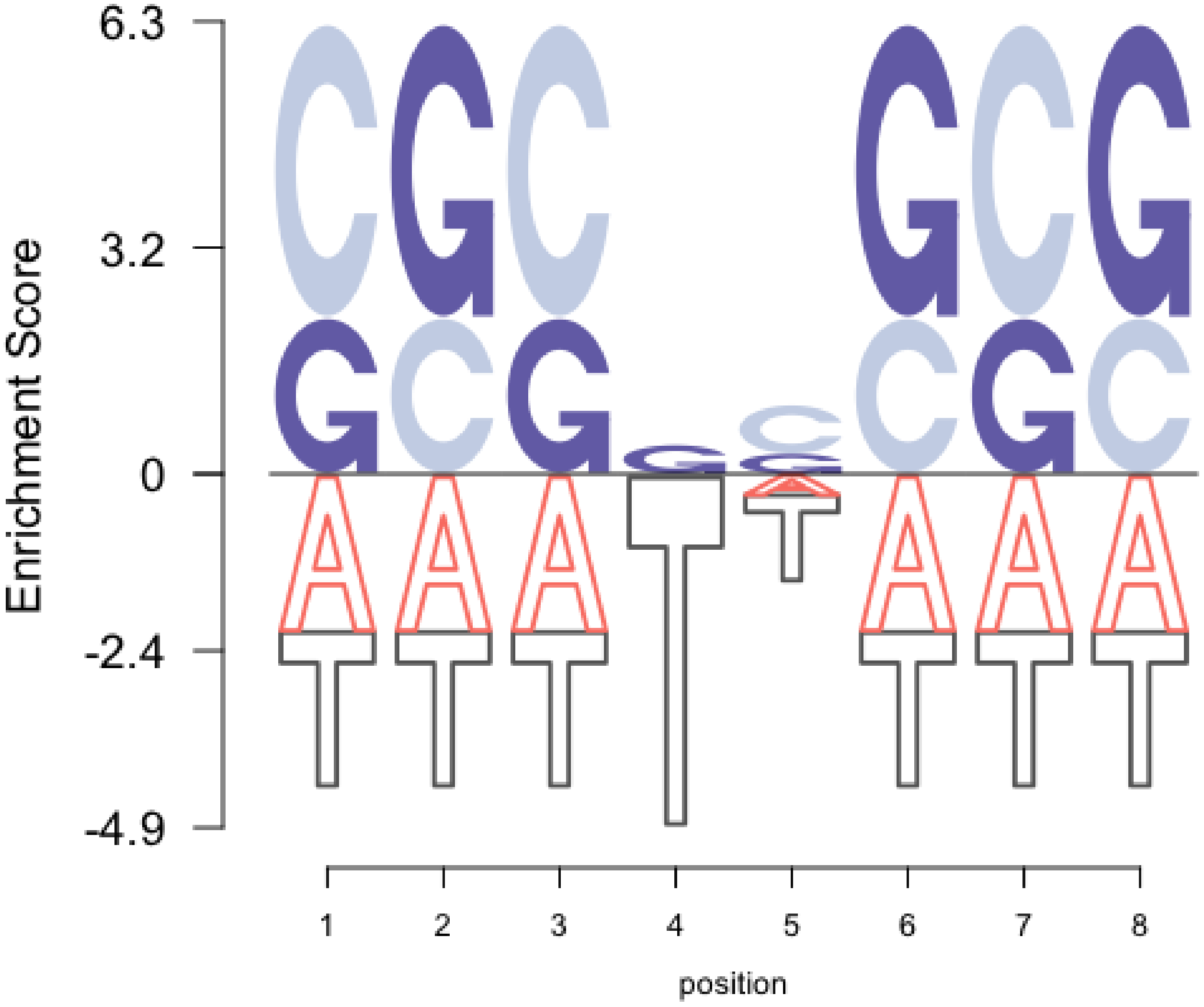
Besides the default style with filled symbols for each character, one can also use characters with border styles. For the standard logo plot, this is accomplished by the `tofill` control argument.

```r
logomaker(seqlogo_example, type = "Logo",
          logo_control = list(control = list(tofill= FALSE)), color_seed = 4000)
```



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)))
```

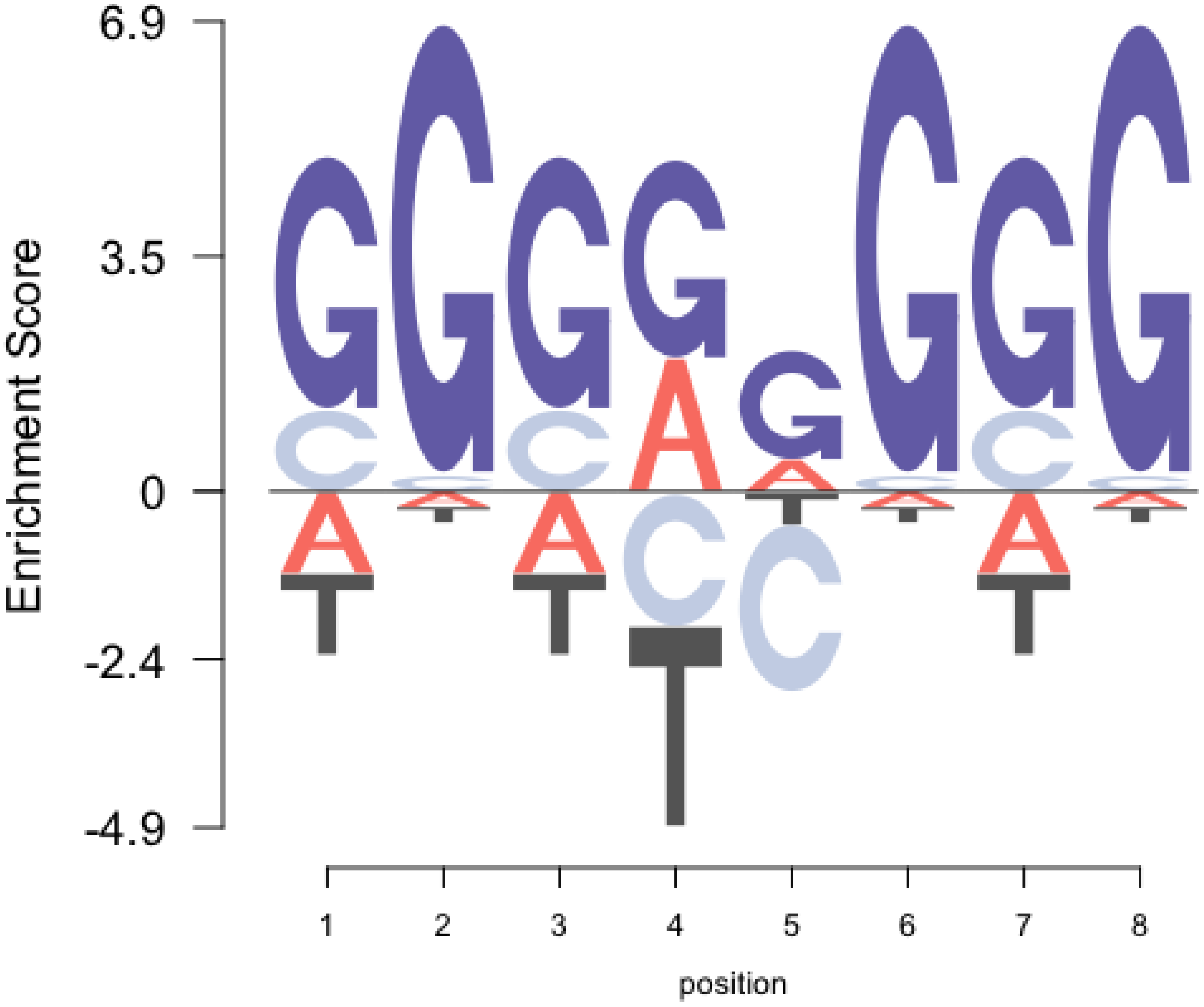


Background Info

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.

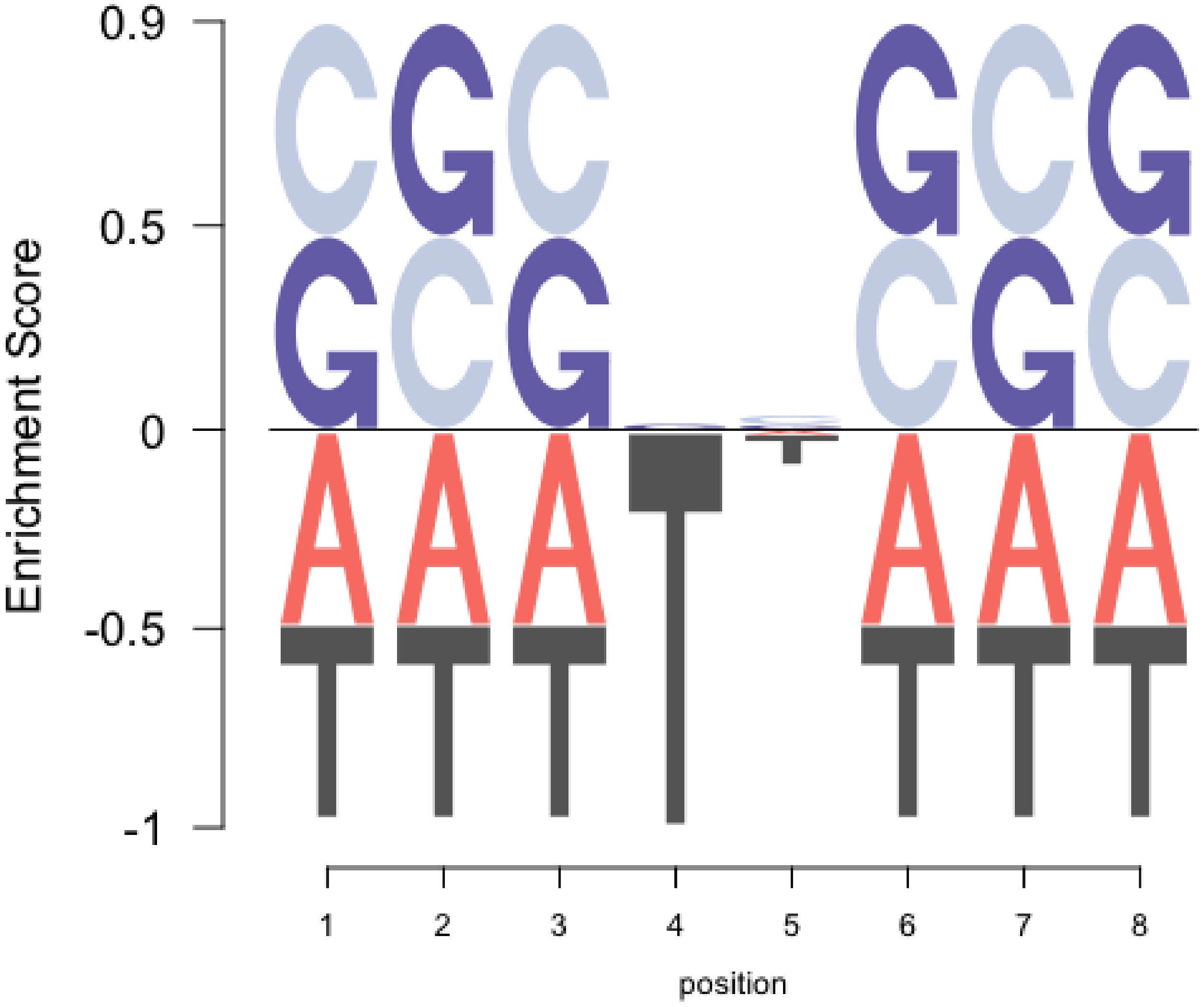
First example with `bg` as a vector.

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



Second example with bg as a matrix.

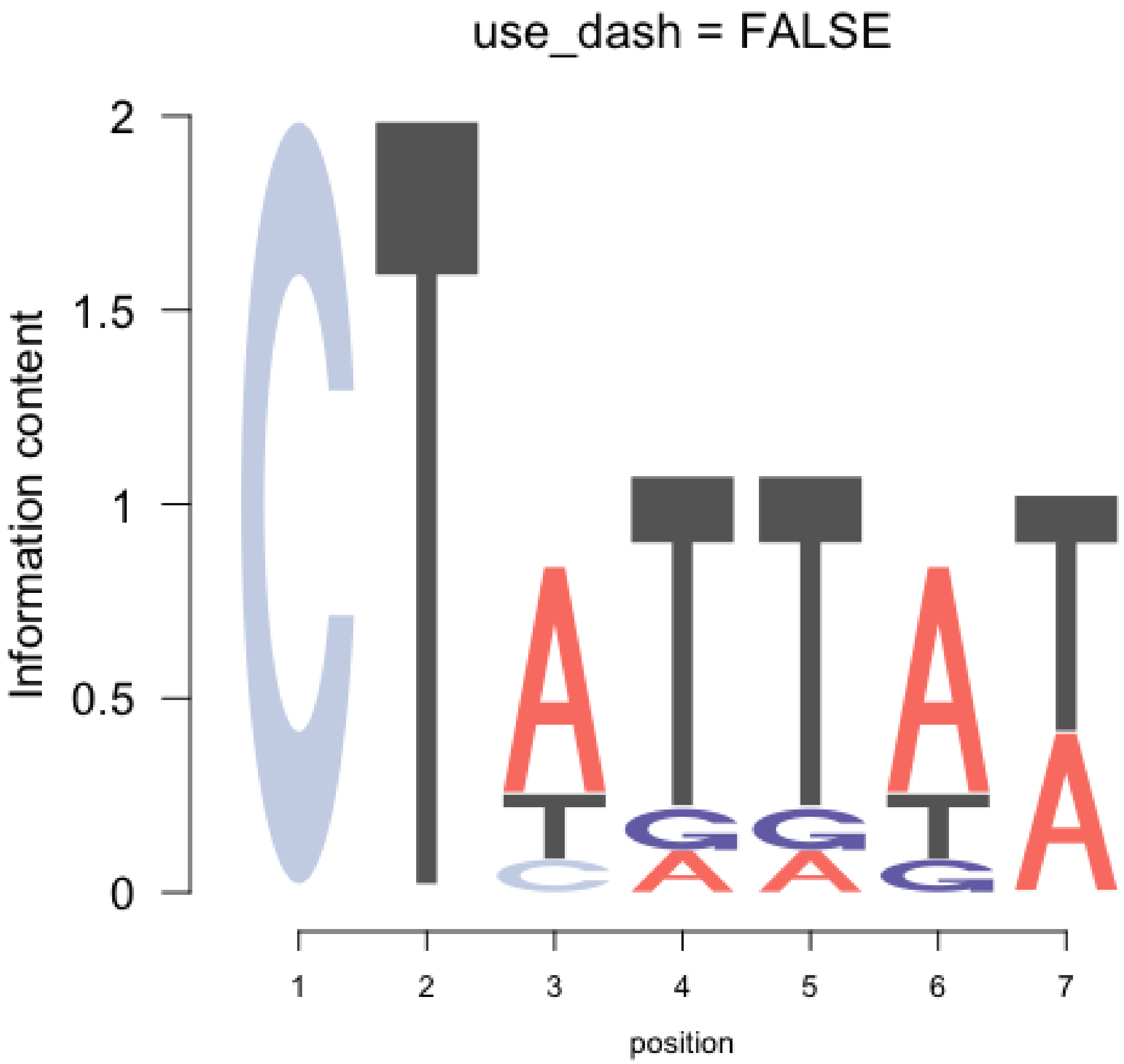
```
logomaker(seqlogo_example, bg=(seqlogo_example+1e-02), type = "EDLogo")
```



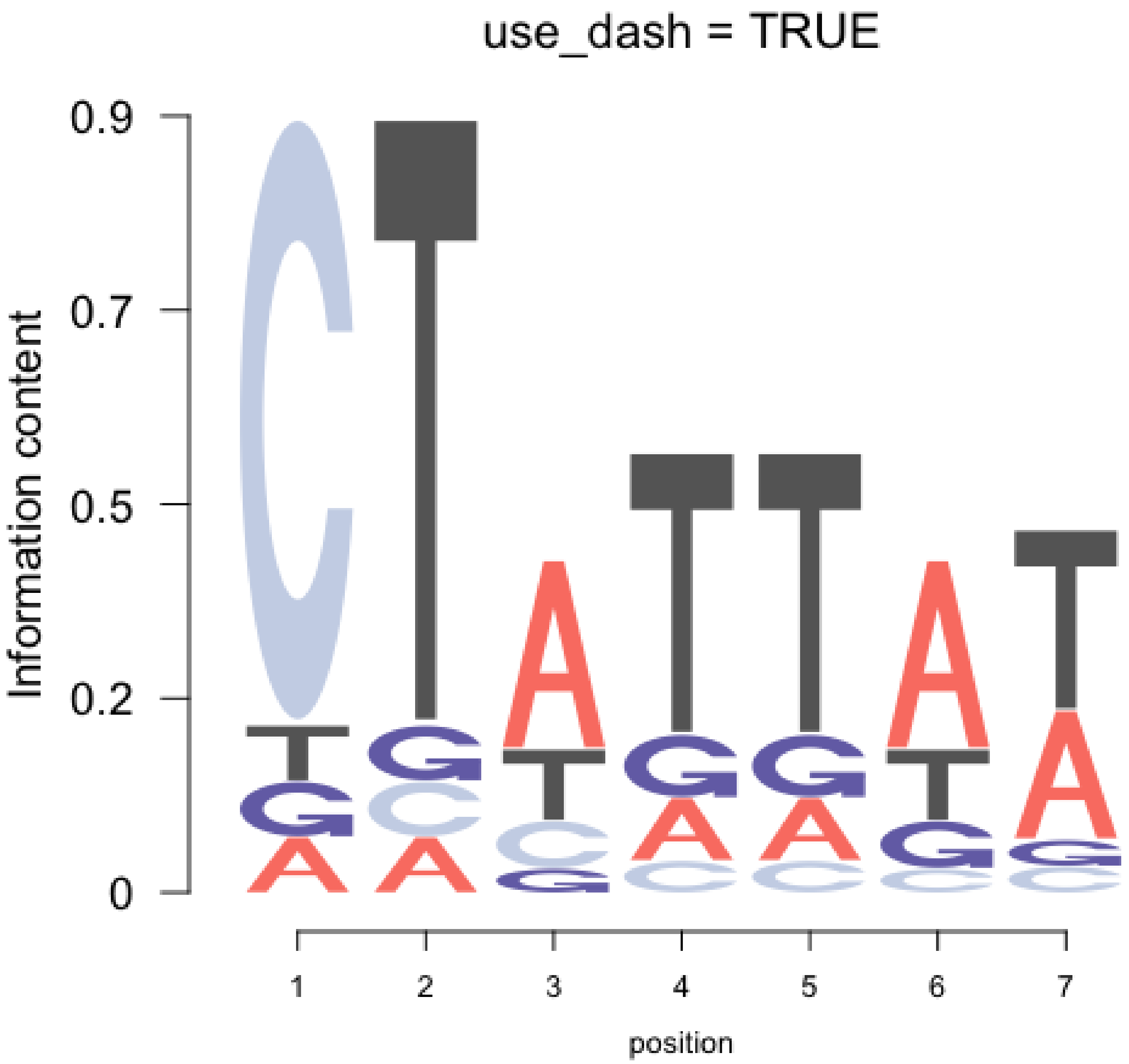
Adaptive scaling of logos (dash)

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"))
```



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



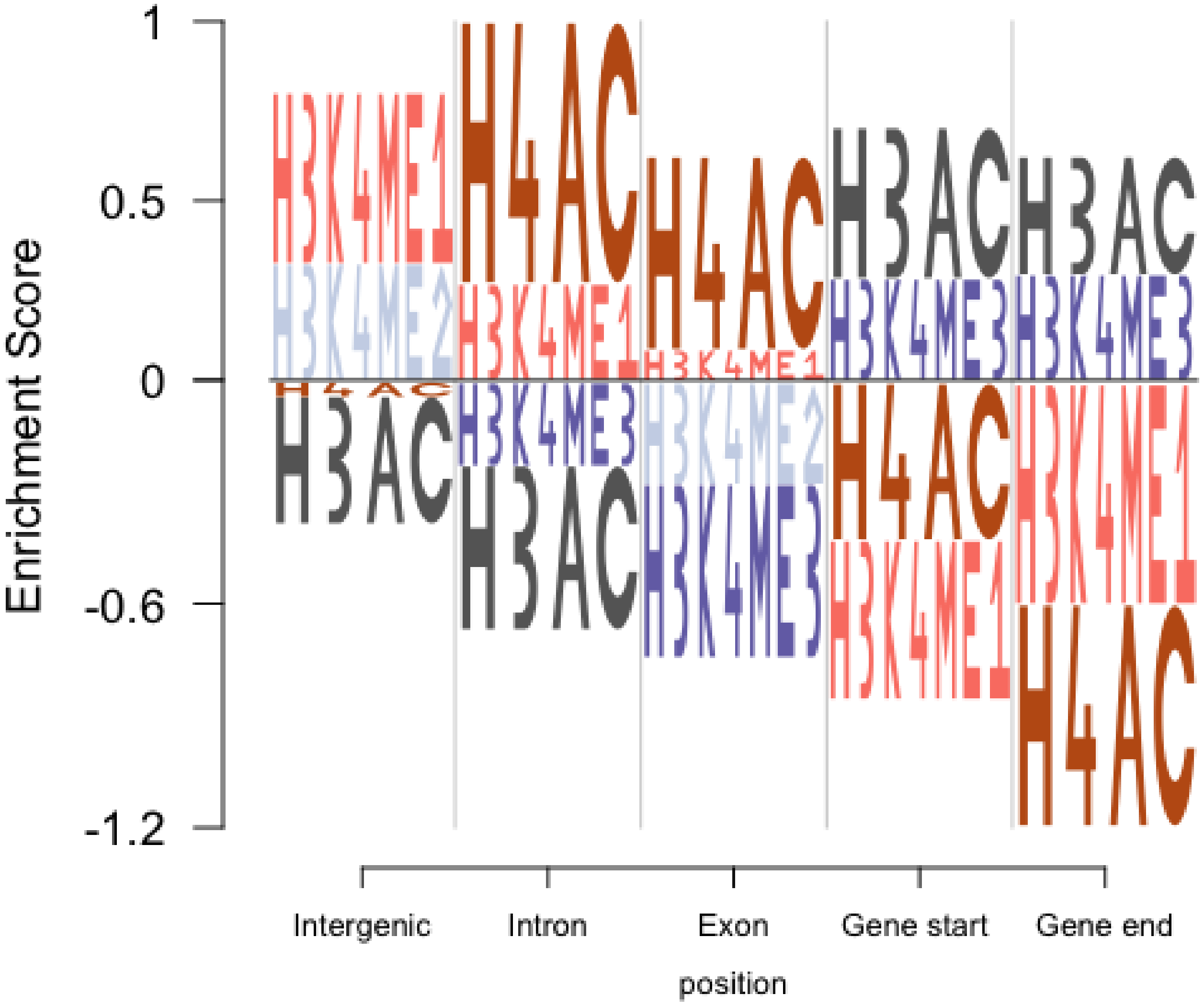
The adaptive scaling is performed by the Dirichlet Adaptive Shrinkage method, the details of which can be viewed at our dashr package.

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.

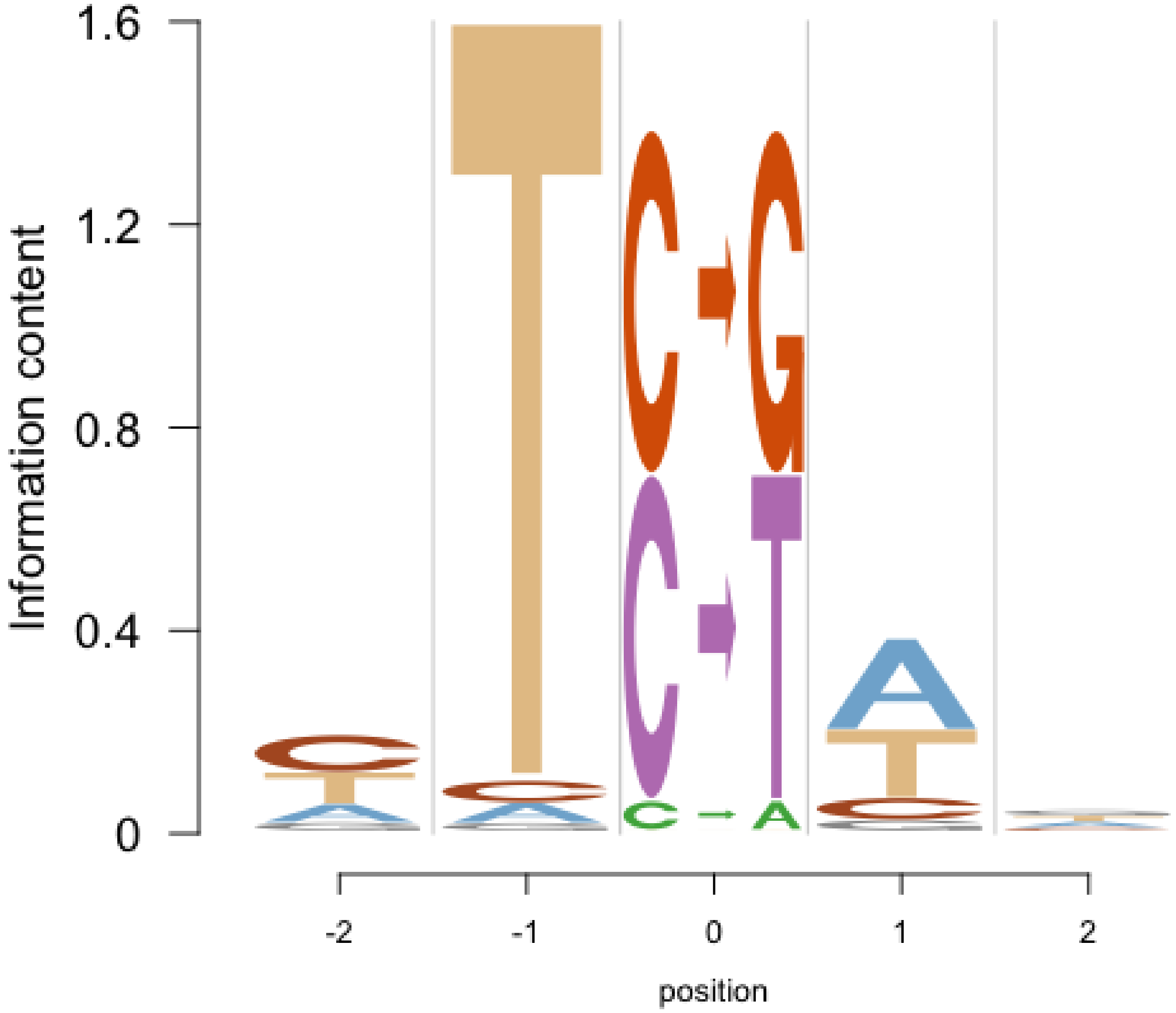
Histone marks string symbols example

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



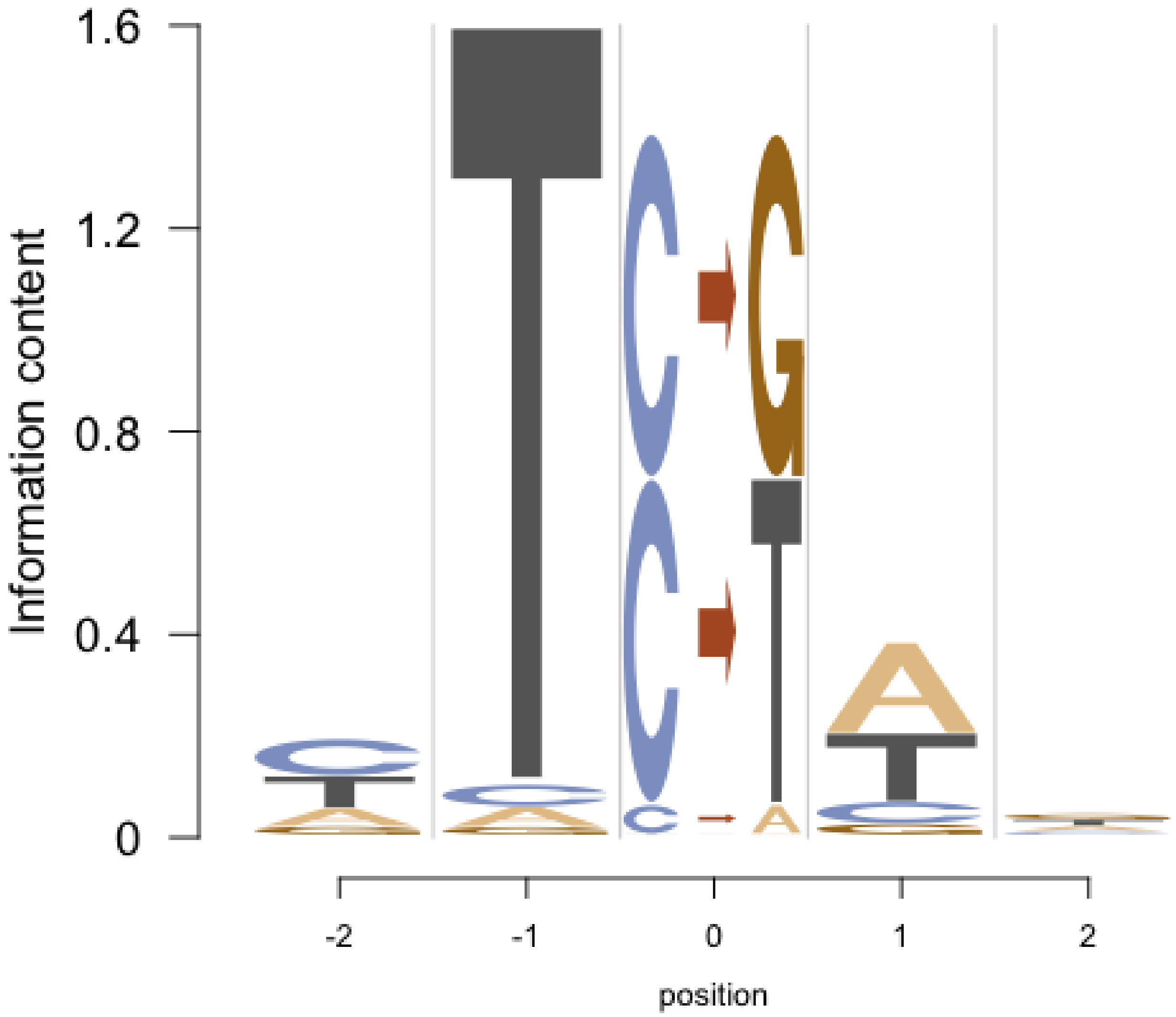
Mutation signature string and character mix example.

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



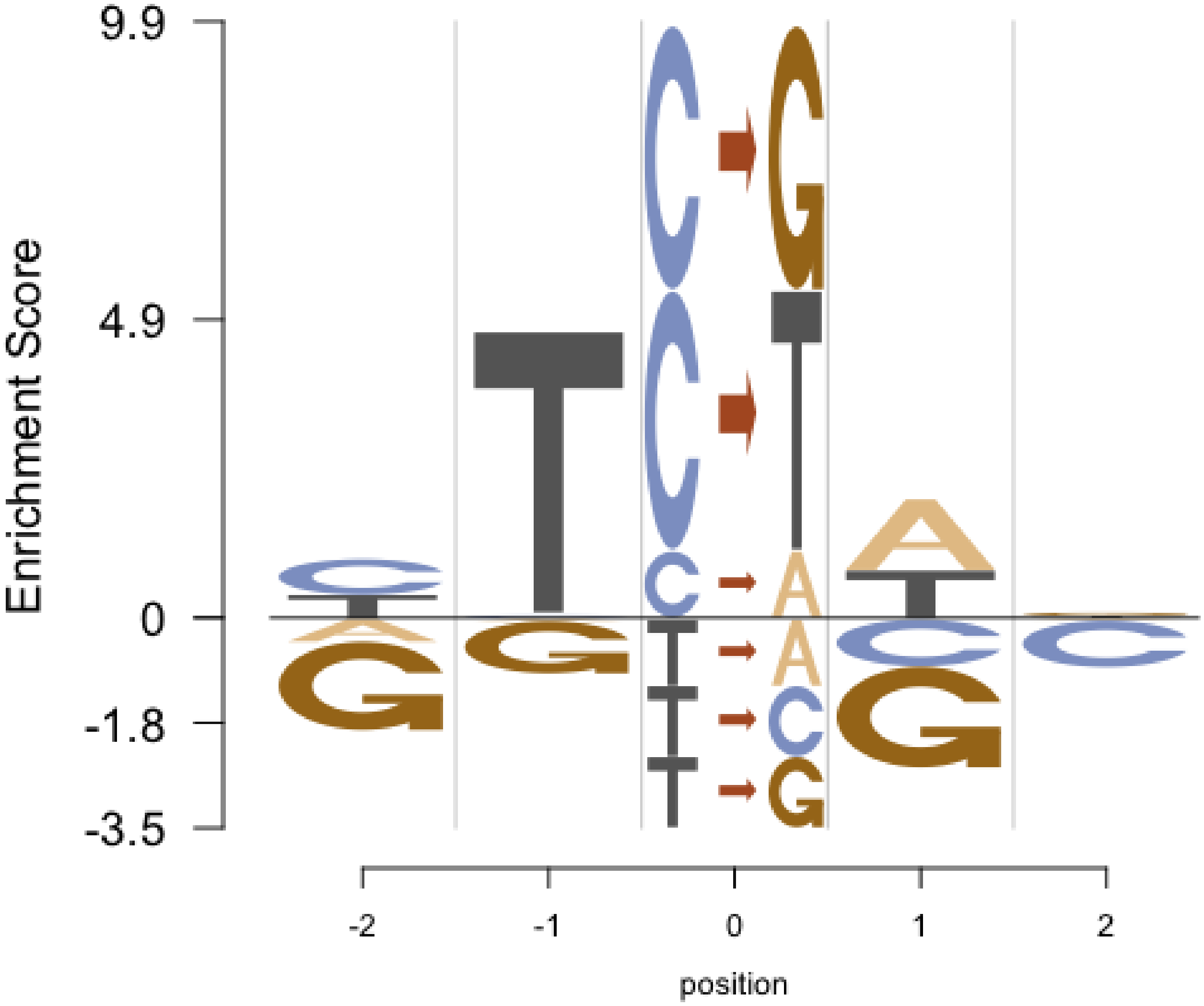
The user may want to have distinct colors for distinct symbols. This is where we use the `per_symbol` option for **color_type**.

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



The corresponding EDLogo

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



Extras

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)"
```

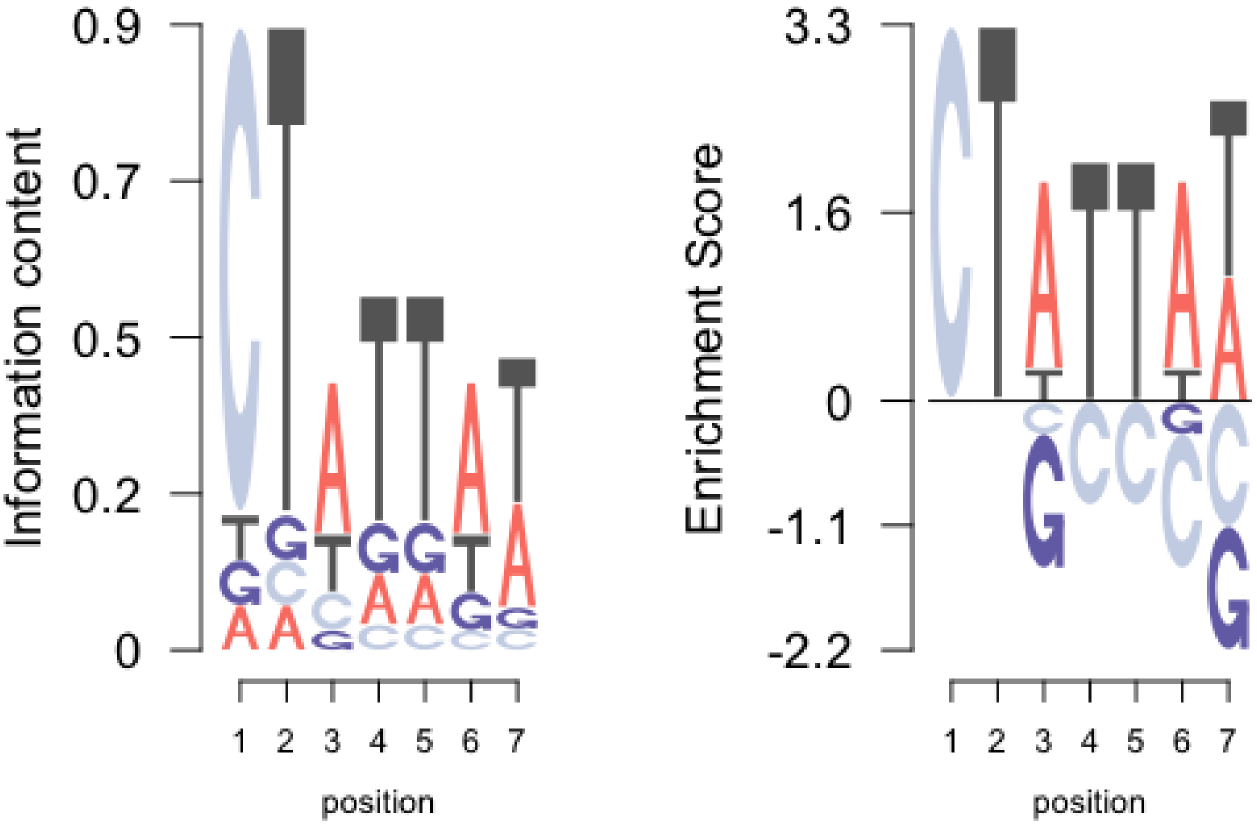
In the sequence, a position represented by (Ag) would mean enrichment in A and depletion in G at that position. One can input a PWM or PFM matrix with A, C, G and T as row names in the **GetConsensusSeq()** function as well.

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)
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))
```

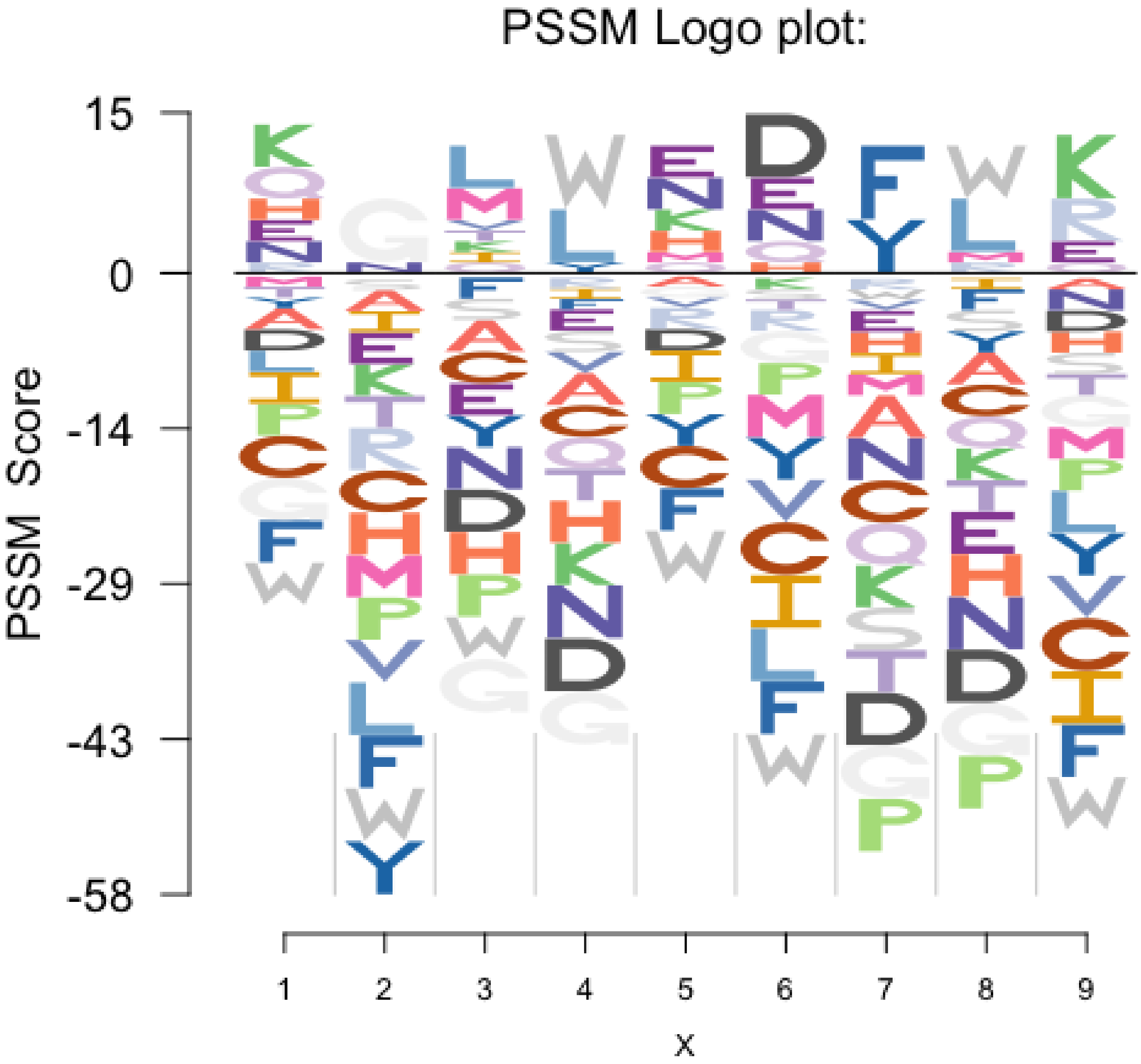


In the same way, ggplot2 graphics can also be combined with **Logolas** plots.

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.

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.

Session Info

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.4
```

```
##
## 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_1.4.1 rmarkdown_1.9
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.15         XVector_0.18.0       knitr_1.20
##  [4] magrittr_1.5         zlibbioc_1.24.0      IRanges_2.12.0
##  [7] BiocGenerics_0.24.0  munsell_0.4.3        gridBase_0.4-7
## [10] SQUAREM_2017.10-1    colorspace_1.3-2     rlang_0.2.0.9000
## [13] stringr_1.3.0        plyr_1.8.4           tools_3.4.3
## [16] parallel_3.4.3       gtable_0.2.0         htmltools_0.3.6
## [19] yaml_2.1.18          lazyeval_0.2.1       rprojroot_1.3-2
## [22] digest_0.6.15        tibble_1.4.2         RColorBrewer_1.1-2
## [25] ggplot2_2.2.1        S4Vectors_0.16.0     evaluate_0.10.1
## [28] LaplacesDemon_16.1.0 stringi_1.1.6        pillar_1.2.1
## [31] compiler_3.4.3       Biostrings_2.46.0    scales_0.5.0
## [34] backports_1.1.2      stats4_3.4.3
```

Thank you for using Logolas !

If you have any questions, you can either open an issue in our Github page or write to Kushal K Dey (kkdey@uchicago.edu). Also please feel free to contribute to the package. You can contribute by submitting a pull request or by communicating with the said person.