Package 'Logolas'

April 16, 2019

Type Package

Encoding UTF-8

Title EDLogo Plots Featuring String Logos and Adaptive Scaling of Position-Weight Matrices

Version 1.6.0

Date 2018-02-06

Maintainer Kushal Dey <kkdey@uchicago.edu>

Description Produces logo plots highlighting both enrichment and depletion of characters, allows for plotting of string symbols, and performs scaling of position-weights adaptively, along with several fun stylizations.

License GPL (>= 2)

URL https://github.com/kkdey/Logolas

BugReports http://github.com/kkdey/Logolas/issues

LazyData TRUE

NeedsCompilation no

Depends R (>= 3.4)

Imports grid, SQUAREM, LaplacesDemon, stats, graphics, utils, ggplot2, gridBase, Biostrings

Suggests knitr, rmarkdown, BiocStyle, Biobase, devtools, xtable, gridExtra, RColorBrewer, seqLogo, ggseqlogo

biocViews SequenceMatching, Alignment, Software, Visualization, Bayesian

VignetteBuilder knitr

RoxygenNote 6.0.1

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

git_branch RELEASE_3_8

git_last_commit 67c730d

git_last_commit_date 2018-10-30

Date/Publication 2019-04-15

Author Kushal Dey [aut, cre],
Dongyue Xie [aut],
Peter Carbonetto [ctb],
Matthew Stephens [aut]

R topics documented:

`aRxiv` *Counts of the aRxiv field categories of 4 UChicago Statistics professors. A dataset of compositional counts of 10 aRxiv field categories for 4 professors' publications in aRxiv.*

Description

Counts of the aRxiv field categories of 4 UChicago Statistics professors. A dataset of compositional counts of 10 aRxiv field categories for 4 professors' publications in aRxiv.

Usage

```
aRxiv
```

Format

A matrix with 10 rows and 4 columns

`EBF1_disc1` *EBF1_disc1 transcription factor position weight matrix A dataset of composition probabilities of A, C, G and T in 10 positions of the motif.*

Description

EBF1_disc1 transcription factor position weight matrix A dataset of composition probabilities of A, C, G and T in 10 positions of the motif.

Usage

```
EBF1_disc1
```

Format

A matrix with 4 rows and 10 columns

`GetConsensusSeq` *Function for obtaining consensus sequence of DNA sequence symbols from a PWM matrix*

Description

uses a special nomenclature (we call it the Logolas nomenclature) to determine the consensus sequence of symbols based on the enrichment and depletion of the symbols at each position. This approach is an alternative to the getIUPAC() method used by the atSNP package.

Usage

```
GetConsensusSeq(data)
```

Arguments

`data`	The input data may be a vector of A, C, G and T sequences - representing aligned DNA or RNA sequences , or a matrix/ data frame with symbols of A, C, G and T along the rows of the matrix/data frame and the positions or sites of the aligned sequences along the columns.

Value

Returns the consensus sequence for the DNA/RNA sequence motif along the positions using the Logolas nomenclature (highlighting both enrichment and depletion).

Examples

```
pwm=matrix(c(0.8,0.1,0.1,0,
0.9,0.1,0,0,0.9,0.05,0.05,0,0.5,
0.4,0,0.1,0.6,0.4,0,0,0.4,0.4,0.1,
0.1,0.5,0,0.2,0.3,0.35,0.35,0.06,
0.24,0.4,0.3,0.2,0.1,0.4,0.2,0.2,
0.2,0.28,0.24,0.24,0.24,0.5,0.16,0.17,
0.17,0.6,0.13,0.13,0.14,0.7,0.15,0.15,0),
nrow = 4,byrow = FALSE)
rownames(pwm)=c('A','C','G','T')
colnames(pwm)=1:ncol(pwm)
GetConsensusSeq(pwm)

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

get_logo_heights *Get heights of logos in nlogomaker() under different scoring schemes*

Description

Generates total heights of the stack of logos in the positive and negative scales of the nlogomaker() logo plot along with the proportion of the height distributed between the logos to be plotted in the positive and the negative scales respectively under different scoring schemes.

Usage

```
get_logo_heights(table, ic = FALSE, score = c("diff", "log", "log-odds",
  "probKL", "ratio", "unscaled_log", "wKL"), bg = NULL, epsilon = 0.01,
  opt = 1, symm = TRUE, alpha = 1, hist = FALSE, quant = 0.5)
```

Arguments

table	The input table (data frame or matrix) of compositional counts or relative frequencies across different logos or symbols (specified along the rows) for different sites or positions or groups (specified along the columns).
ic	Boolean, denoting whether information content based scaling is used on top of the scoring scheme used or not. Default is FALSE
score	Can take either of the options - diff, log, log-odds, probKL, ratio, unscaled_log, wKL. Each option corresponds to a different scoring scheme. The most
bg	The background probability, which defaults to 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.
epsilon	An additive constant added to the PWM before scaling to eliminate log (0) type errors.
opt	Option parameter - taking values 1 and 2 - depending on whether median adjustment is done based on background corrected proportions or without background correction.
symm	A bool input, which if TRUE, the function uses symmetric KL divergence whereas if FALSE, the function uses non-symmetric KL divergence.
alpha	The Renyi entropy tuning parameter which is used in case of scaling of the bar heights by information criterion. The default tuning parameter value is 1, which corresponds to Shannon entropy.
hist	Whether to use the hist method or the information criterion method to determine the heights of the logos.
quant	The quantile to be adjusted for in computing enrichment and depletion scores. Defaults to 0.5, which corresponds to the median.

Value

Returns the heights of enrichment and depletion for diff approach to EDLogo.

Examples

```
m = matrix(rep(0,48),4,12)
m[1,] = c(0,0,2.5,7,0,0,0,0,0,0,1,0)
m[2,] = c(4,6,3,1,0,0,0,0,0,5,0,5)
m[3,] = c(0,0,0,0,0,1,8,0,0,1,1,2)
m[4,] = c(4,2,2.5,0,8,7,0,8,8,2,6,1)
rownames(m) = c("A", "C", "G", "T")
colnames(m) = 1:12
m=m/8
get_logo_heights(m, score = "log")
get_logo_heights(m, score = "log", ic = TRUE)
get_logo_heights(m, score = "wKL")
get_logo_heights(m, score = "probKL", ic = TRUE)
```

---

get_viewport_logo *Function for creating a multi-panel logo viewport*

Description

This is a void function to be run prior to a multi-panel Logolas plot for a fixed number of rows and columns of the panel.

Usage

```
get_viewport_logo(layout.rows, layout.cols, widths_1 = 6, heights_1 = 20)
```

Arguments

layout.rows	The number of rows in the panel
layout.cols	The number of columns in the panel
widths_1	The widths of each viewport in the panel
heights_1	The heights of each viewport in the panel

Value

Creates a panel of viewports for each row and each column, to be subsequently used for creating logo plots.

Examples

```
get_viewport_logo(layout.rows = 2, layout.cols = 2)
```

himalayan_fauna *Proportional abundances of different bird species in the Himalayan mountains A dataset of proportional composition of 140 bird species in 3 regions of the Himalayas.*

Description

Proportional abundances of different bird species in the Himalayan mountains A dataset of proportional composition of 140 bird species in 3 regions of the Himalayas.

Usage

```
himalayan_fauna
```

Format

A matrix with 140 rows and 3 columns

histone_marks *Compositional data of histone marks in different regions of genome along with background information A list containing two matrices storing composition of 5 histone marks (H3K4ME1, H3K4ME2, H3K4ME3, H3AC, H4AC) in 5 regions of the genome*

Description

Compositional data of histone marks in different regions of genome along with background information A list containing two matrices storing composition of 5 histone marks (H3K4ME1, H3K4ME2, H3K4ME3, H3AC, H4AC) in 5 regions of the genome

Usage

```
histone_marks
```

Format

A list with 2 matrices - each with 5 rows (histone marks) and 5 columns (regions of genome). One is the actual matrix (mat), the other is the background matrix (bgmat)

`logomaker` *Create logo plots from aligned sequences or positional frequency (weight) matrix*

Description

Takes as input a vector of character sequences (aligned to have the) same length or a positional frequency or weight matrix and plots the standard logo or the Enrichment Depletion (ED) Logo plots.

Usage

```
logomaker(data, type = c("Logo", "EDLogo"), use_dash = TRUE, bg = NULL,
  color_type = NULL, colors = NULL, color_seed = 2030,
  return_heights = FALSE, logo_control = list(), dash_control = list())
```

Arguments

`data`	The input data may be a vector of character sequences - representing aligned sequences of DNA, RNA or amino acids, or a matrix/ data frame with symbols of characters or strings of characters along the rows of the matrix/data frame and the positions or sites of the aligned sequences along the columns.
`type`	can either be "Logo" or "EDLogo" depending on if user wants to plot the standard Logo or the Enrichment Depletion Logo.
`use_dash`	Boolean. If TRUE, performs adaptive scaling of the consensus matrix from sequences or the positional frequency matrix based on the number of underlying seqeunces. Step automatically skipped if the user provides a positional weight matrix.
`bg`	The background probability, which defaults to 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.
`color_type`	A list specifying the coloring scheme. Defaults to NULL, for which, based on `color_seed`, a specific coloring scheme is chosen. The list contains two elements - `type` and `col`.The `type` can be of three types - "per-row", "per-column" and "per-symbol". The `col` element is a vector of colors, 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". For "per-symbol", the length of the `color_profile$col` should be same as library size of the logos, but if the vector of colors provided is more or less, we can downsample or upsample the colors as required. The colors are matched with the symbols in the `total_chars`.
`colors`	Add description here.
`color_seed`	A seed for choosing among multiple available coloring schemes in `color_profile`. The default choice is 2030. But the user can use any seed of her choice.

`return_heights` Boolean. If TRUE, the function returns the stack heights for the logo plot.For standard Logo (type = "Logo"), it returns the information content. For tyep = "EDLogo", it returns the total stack height along positive and negative axis, as well as the breakdown of the heights along different symbols along the two axis. Defaults to FALSE.

`logo_control` Control parameters for the logo plot. Check the input arguments from the `plogomaker` and `nlogomaker` functions.

`dash_control` Control parameters for the adaptive scaling `dash` function. Check the input arguments to the `dash` function in this package.

Value

Returns a standard or EDLogo plot of the sequence of the positional frequency matrix based on the `type` is equal to Logo or EDLogo.

Examples

```
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")

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

library(ggseqlogo)
data(ggseqlogo_sample)

sequence <- seqs_aa$AKT1
logomaker (sequence, type = "Logo")
logomaker (sequence, type = "EDLogo")

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

---

logo_pssm *Function to plot PSSM logo plot visualization.*

Description

stacks logos created by the `makemylogo` function on top of each other to build the PSSM logo plot.

Usage

```
logo_pssm(table, color_type = NULL, colors = NULL, color_seed = 2030,
  total_chars = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L",
  "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "zero",
  "one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "dot",
  "comma", "dash", "colon", "semicolon", "leftarrow", "rightarrow"),
  frame_width = NULL, yscale_change = TRUE, pop_name = NULL,
  addlogos = NULL, addlogos_text = NULL, newpage = TRUE, ylimit = NULL,
  xaxis = TRUE, yaxis = TRUE, xaxis_fontsize = 10, xlab_fontsize = 15,
  y_fontsize = 15, main_fontsize = 16, start = 0.001, xlab = "X",
  ylab = "PSSM  Score", col_line_split = "grey80", control = list())
```

Arguments

table	The input table (data frame or matrix) of PSSM scores (comprising of both positive and negative scores) across different logos or symbols (specified along the rows) and across different sites or positions or groups (specified along the columns).
color_type	A list specifying the coloring scheme. Defaults to NULL, for which, based on color_seed, a specific coloring scheme is chosen. The list contains two elements - type and col.The type can be of three types - "per-row", "per-column" and "per-symbol". The col element is a vector of colors, 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". For "per-symbol", the length of the color_profile$col should be same as library size of the logos, but if the vector of colors provided is more or less, we can downsample or upsample the colors as required. The colors are matched with the symbols in the total_chars.
colors	Add description here.
color_seed	Add description here.
total_chars	The total number of character symbols in the user library. The default is the default library provided by Logolas, but the user can add symbols that he creates to this list.
frame_width	The width of the frames for individual site/postion/column in the logo plot. As default, all the columns have same width, equal to 1.
yscale_change	If TRUE, adjusts the Y axis scale based on the size of the bars, else keeps it to the maximum value possible, which is ceiling(max(ic) under ic_computer defined IC criteria.
pop_name	User can mention a name of the population for which the logo plot is created. Defaults to NULL when no population name is mentioned.
addlogos	Vector of additional logos/symbols defined by user
addlogos_text	Vector of the names given to the additional logos/symbols defined by user.
newpage	if TRUE, plots the logo plot in a new page. Defaults to TRUE.
ylimit	The limit of the Y axis.
xaxis	Binary specifying if there should be a X axis in the logo plot or not. Defaults to TRUE.
yaxis	Binary specifying if there should be a Y axis in the logo plot or not. Defaults to TRUE.

xaxis_fontsize	The size of the X-axis axis ticks.
xlab_fontsize	The size of the X-axis label.
y_fontsize	The size of the Y-axis font.
main_fontsize	The size of the title.
start	The starting point in Y axis for the first logo. Default is 0.0001 which is very close to 0.
xlab	X axis label
ylab	Y axis label
col_line_split	The color of the line split between the consecutive groups or blocks
control	control parameters fixing whether the symbols should be filled with color or border colored (tofill_pos,tofill_neg), the viewport configuration details for the plot (viewport.margin.bottom, viewport.margin.left, viewport.margin.top, viewport.margin.right) etc.

Value

Plots the logo plot for the PSSM scoring data, with column names representing the sites/blocks and the row names denoting the symbols for which logos are plotted

Examples

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

---

makemylogo *Logo maker for a given English alphanumeric with common punctuations*

Description

Plots logo for a given english symbol or name that contains English alphabets, numbers or punctuations like dots, dashes, etc. This is the skeleton used by the logomaker function of the package to create distinct logos for distinct alphanumeric symbols.

Usage

```
makemylogo(name, tofill = TRUE, colfill = "orange", lwd = 10,
  plot = FALSE, total_chars = c("A", "B", "C", "D", "E", "F", "G", "H", "I",
  "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X",
  "Y", "Z", "zero", "one", "two", "three", "four", "five", "six", "seven",
  "eight", "nine", "dot", "comma", "dash", "colon", "semicolon", "leftarrow",
  "rightarrow"), addlogos = NULL, addlogos_text = NULL)
```

Arguments

name	A English name, or alphanumeric, containing English alphabets, numbers, dots, dashes, arroww, colons, semicolons, comma among punctuations.
tofill	A binary indicating whether to use fill the logo symbols with color in colfill or to use it for the bordering.
colfill	The color used for the symbol
lwd	The line width for the symbol.
plot	binary, if FALSE, returns only the co-ordinates of the symbol in the [0,1] X [0,1] grid, along with block id labels and their corresponding colors. If TRUE, plots the symbol with specified color in a new grid window.
total_chars	The total number of character symbols in the user library. The default is the default library provided by Logolas, but the user can add symbols that he creates to this list.
addlogos	Vector of additional logos/symbols defined by user
addlogos_text	Vector of the names given to the additional logos/symbols defined by user.

Value

Along with symbol plot, if plot is TRUE, returns a list with the following items.

x	X co-ordinates of the logo in the [0,1] X [0,1] grid window
y	Y co-ordinates of the logo in the [0,1] X [0,1] grid window
id	id vector representing blocks in the logo co-ordinates
fill	a vector equal to the number of distinct ids or blocks in the logo, whose elements correspond to colors of these blocks

Examples

```
makemylogo("KUSHAL")
cols = RColorBrewer::brewer.pal.info[RColorBrewer::brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(RColorBrewer::brewer.pal, cols$maxcolors,
rownames(cols)))
makemylogo("Evening", plot=TRUE, colfill=col_vector)
```

---

mutation_sig *Compositional mutational signature data, with mismatch and flanking base frequencies reported A dataset of compositional weights for mismatches in the "0"the position (center of signature) and that of the bases A, C, G, T for two left flanking bases (-1, -2) and two right flanking bases (1, 2).*

Description

Compositional mutational signature data, with mismatch and flanking base frequencies reported A dataset of compositional weights for mismatches in the "0"the position (center of signature) and that of the bases A, C, G, T for two left flanking bases (-1, -2) and two right flanking bases (1, 2).

Usage

```
mutation_sig
```

Format

A matrix with 10 rows (4 bases and 6 types of mismatches) and 5 columns (center - 0, flanking bases 1,2 on the left and flanking bases 1, 2 on the right of the center)

---

nlogomaker *Main workhorse function that builds negative logo plots*

Description

stacks logos created by the makemylogo function on top of each other to build the logo plot.

Usage

```
nlogomaker(table, ic = FALSE, score = c("diff", "log", "log-odds", "probKL",
  "ratio", "unscaled_log", "wKL"), color_profile, total_chars = c("A", "B",
  "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q",
  "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "zero", "one", "two", "three",
  "four", "five", "six", "seven", "eight", "nine", "dot", "comma", "dash",
  "colon", "semicolon", "leftarrow", "rightarrow"), bg = NULL,
  frame_width = NULL, yscale_change = TRUE, pop_name = NULL,
  addlogos = NULL, addlogos_text = NULL, newpage = TRUE, yrange = NULL,
  xaxis = TRUE, yaxis = TRUE, xaxis_fontsize = 10, xlab_fontsize = 15,
  y_fontsize = 15, main_fontsize = 16, start = 0.001, xlab = "X",
  ylab = "Enrichment Score", col_line_split = "grey80", control = list())
```

Arguments

table	The input table (data frame or matrix) of counts across different logos or symbols (specified along the rows) and across different sites or positions or groups (specified along the columns).
ic	Boolean, denoting whether information content based scaling is used on top of the scoring scheme used or not. Default is FALSE
score	Can take either of the options - diff, log, log-odds, probKL, ratio, unscaled_log, wKL. Each option corresponds to a different scoring scheme. The most recommended option is log.
color_profile	A list containing two elements - "type" and "col". The type can be of three types - "per-row", "per-column" and "per-symbol". The "col" element is a vector of colors, 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". For "per-symbol", the length of the color_profile$col should be same as library size of the logos, but if the vector of colors provided is more or less, we can downsample or upsample the colors as required. The colors are matched with the symbols in the total_chars

total_chars	The total number of character symbols in the user library. The default is the default library provided by Logolas, but the user can add symbols that he creates to this list.
bg	The background probability, which defaults to 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.
frame_width	The width of the frames for individual site/postion/column in the logo plot. As default, all the columns have same width, equal to 1.
yscale_change	If TRUE, adjusts the Y axis scale based on the size of the bars, else keeps it to the maximum value possible, which is `ceiling(max(ic)` under `ic_computer` defined IC criteria.
pop_name	User can mention a name of the population for which the logo plot is created. Defaults to NULL when no population name is mentioned.
addlogos	Vector of additional logos/symbols defined by user
addlogos_text	Vector of the names given to the additional logos/symbols defined by user.
newpage	if TRUE, plots the logo plot in a new page. Defaults to TRUE.
yrange	The limit of the Y axis.
xaxis	Binary specifying if there should be a X axis in the logo plot or not. Defaults to TRUE.
yaxis	Binary specifying if there should be a Y axis in the logo plot or not. Defaults to TRUE.
xaxis_fontsize	The size of the X-axis axis ticks.
xlab_fontsize	The size of the X-axis label.
y_fontsize	The size of the Y-axis font.
main_fontsize	The size of the title.
start	The starting point in Y axis for the first logo. Default is 0.0001 which is very close to 0.
xlab	X axis label
ylab	Y axis label
col_line_split	The color of the line split between the consecutive groups or blocks
control	control parameters fixing whether the height of the logos is detrmined by IC or histogram proportions (`hist`), the scales for the plot (`scale0`, `scale1`), the additive factor epsilon added to log transform to avoid log(0) errors (`epsilon`), the quantile adjustment (`quant`), whether to use symmetric KL for scaling (`symm`), whether the symbols should be filled with color or border colored (`tofill_pos, tofill_neg`), the Renyi alpha parameter for the entropy calculation (`alpha`), the gap between ylabel and y-axis and xlabel and x-axis texts (`gap_ylab`, `gap_xlab`), the viewport configuration details for the plot (`viewport.margin.bottom`,`viewport.margin.left`, `viewport.margin.top`, `viewport.margin.right`), whether the height of the logos would be fixed apriori or determined by the PWM matrix as in seqLogo (`use_seqLogo_heights`) etc.

Value

Plots the logo plot for the table data, with column names representing the sites/blocks and the row names denoting the symbols for which logos are plotted

Examples

```
mFile <- system.file("Exfiles/pwm1", package="seqLogo")
m <- read.table(mFile)
p <- seqLogo::makePWM(m)
pwm_mat <- slot(p,name = "pwm")
pwm_mat[,4] <- c(0.3, 0.3, 0.35, 0.05)
mat1 <- cbind(pwm_mat[,c(3,4)], rep(NA,4), pwm_mat[,c(5,6)]);
colnames(mat1) <- c("-2", "-1", "0", "1", "2")
mat2 <- cbind(rep(NA,6), rep(NA,6),
            c(0.8, 0.10, 0.03, 0.03, 0.0, 0),
            rep(NA,6), rep(NA,6))
rownames(mat2) <- c("C>T", "C>A", "C>G",
                    "T>A", "T>C", "T>G")

table <- rbind(mat1, mat2)

cols = RColorBrewer::brewer.pal.info[RColorBrewer::brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(RColorBrewer::brewer.pal, cols$maxcolors,
                  rownames(cols)))

total_chars = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
                "K", "L", "M", "N", "O",
               "P", "Q", "R", "S", "T", "U", "V",
               "W", "X", "Y", "Z", "zero", "one", "two",
               "three", "four", "five", "six", "seven",
               "eight", "nine", "dot", "comma",
               "dash", "colon", "semicolon", "leftarrow", "rightarrow")

set.seed(20)
col_vector[c(1,3,7, 20, 43)] <- c("red", "blue",
                                  "orange", "green", "gray")
color_profile <- list("type" = "per_symbol",
                      "col" = col_vector)

nlogomaker(table,
           color_profile = color_profile,
           yrange = 1.2)
```

`pssm` *Position specific scoring matrix data A dataset of position specific sores of various amino acids in 9 positions binding domain.*

Description

Position specific scoring matrix data A dataset of position specific sores of various amino acids in 9 positions binding domain.

Usage

```
pssm
```

Format

A matrix with 20 rows (aminoa acids) and 9 columns

`seqlogo_example` *An example of the position weight matrix taken from seqLogo package A dataset of composition probabilities of A, C, G and T in 8 positions of the motif.*

Description

An example of the position weight matrix taken from seqLogo package A dataset of composition probabilities of A, C, G and T in 8 positions of the motif.

Usage

```
seqlogo_example
```

Format

A matrix with 4 rows and 8 columns

`UMI` *An example of the position weight matrix of the bases in the first 30 positions from ends of sequenced reads (together with the barcode) A dataset of composition probabilities of A, C, G and T in 30 positions from the 5' end of reads.*

Description

An example of the position weight matrix of the bases in the first 30 positions from ends of sequenced reads (together with the barcode) A dataset of composition probabilities of A, C, G and T in 30 positions from the 5' end of reads.

Usage

```
UMI
```

Format

A matrix with 4 rows and 30 columns

Index