Package ‘kmeRs’

July 17, 2024

Title K-Mers Similarity Score Matrix and HeatMap

Version 2.1.0

Description Similarity Score Matrix and HeatMap for nucleic and amino acid k-mers. Similarity score is evaluated by Point Accepted Mutation (PAM) and BLOcks SUbstitution Matrix (BLOSUM). The 30, 40, 70, 120, 250 and 62, 45, 50, 62, 80, 100 matrix versions are available for PAM and BLOSUM, respectively. Alignment is evaluated by local and global alignment.

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

URL https://github.com/urniaz/kmeRs

BugReports https://github.com/urniaz/kmeRs/issues

biocViews Software

Imports utils, stats, Biostrings, BiocGenerics, pwalign

Suggests RColorBrewer, knitr, rmarkdown, unittest, testthat (>= 3.0.0)

RoxygenNote 7.3.1

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Rafal Urniaz [aut, cre] (<https://orcid.org/0000-0003-0192-2165>),
Jason Lin [ctb]

Maintainer Rafal Urniaz <rafal.urniaz@cantab.net>

Repository CRAN

Date/Publication 2024-07-17 11:00:02 UTC

Contents

`kmeRs_generate_kmers` *kmeRs generate kmers*

Description

kmeRs generate kmers

Usage

```
kmeRs_generate_kmers(k, bases)
```

Arguments

`k`	times
`bases`	follow the kmeRs_similarity_matrix()

`kmeRs_heatmap` *K-mer similarity score heatmap*

Description

The `kmeRs_heatmap` function generates a heatmap from similarity score matrix

Usage

```
kmeRs_heatmap(
  x,
  cexRow = NULL,
  cexCol = NULL,
  col = NULL,
  Colv = NA,
  Rowv = NA
)
```

Arguments

`x`	matrix calculated by `kmeRs_similarity_matrix` function
`cexRow`	= NULL
`cexCol`	= NULL
`col`	color palette, when NULL the default palette is applied
`Colv`	when different from NA, the column dendrogram is shown
`Rowv`	when different from NA, the row dendrogram is shown

Value

heatmap from results

Examples

```
# Use RColorBrewer to generate a figure similar to publication
library(RColorBrewer)
h.palette <- rev(brewer.pal(9, "YlGnBu"))
q0 <- c("GATTACA", "ACAGATT", "GAATTAC", "GAAATCT", "CTATAGA", "GTACATA", "AACGATT")
example <- kmeRs_similarity_matrix(q0, submat = "BLOSUM62")
kmeRs_heatmap(kmeRs_score(example), col = h.palette)
```

---

kmeRs_score *Sort a k-mer Similarity Matrix*

Description

The `kmeRs_score` function sums the partial scores and sort the data.frame to indicate the most 'different' k-mers

Usage

```
kmeRs_score(x, decreasing = FALSE)
```

Arguments

`x`	the similarity matrix calculated by `kmeRs_similarity_matrix` function
`decreasing`	when TRUE, results are sorted decreasing

Value

sorted similarity matrix with global.score column added; is returned as a data.frame

Examples

```
# Calculate the example BLOSUM62 matrix and score the result

example <- kmeRs_similarity_matrix(kmers_given = c("A", "T", "C", "G"), submat = "BLOSUM62")
kmeRs_score(example)
```

`kmeRs_show_alignment` *Calculate and Show Alignment Between Two Compared K-mers*

Description

The `kmeRs_show_alignment` function aligns and shows calculated alignment between two DNA or RNA sequences

Usage

```
kmeRs_show_alignment(
  kmer_A,
  kmer_B,
  seq.type = "AA",
 submat = ifelse(test = (match.arg(toupper(seq.type), c("DNA", "AA")) == "AA"), yes =
    "BLOSUM62", no = NA),
  na.match = ifelse(is.na(submat), yes = 2, no = NA),
  na.mismatch = ifelse(is.na(submat), yes = -3, no = NA),
  align.type = "global",
  verbose = TRUE,
  ...
)
```

Arguments

`kmer_A` given k-mer A

`kmer_B` given k-mer B

`seq.type` type of sequence in question, either 'DNA' or 'AA' (default)

`submat` substitution matrix version, defaults to 'BLOSUM62'; other choices include 'BLOSUM45', 'BLOSUM50', 'BLOSUM62', 'BLOSUM80', 'BLOSUM100', 'PAM30', 'PAM40', 'PAM70', 'PAM120' and 'PAM250'; this parameter is ignored if `na.match` and `na.mismatch` are specified

`na.match` for DNA sequences, what should the score for exact match be?

`na.mismatch` for DNA sequences, what should the score for mismatches be?

`align.type` "global" or "local"

`verbose` = TRUE

`...` other parameters, e.g. gap opening/extension penalties (`gapOpening`, `gapExtension`) for generating a DNA base substitution matrix

Value

alignment is returned as a data frame

Examples

```
# Example DNA alignment with gap opening and extension penalties of 1 and 0
# with default base match/mismatch values

kmeRs_show_alignment(kmer_A = "AAATTTCCCGGG", kmer_B = "TCACCC",
    seq.type = "DNA", gapOpening = 1, gapExtension = 0)
```

---

kmeRs_similarity_matrix
Pairwise Similarity Matrix

Description

The kmeRs_similarity_matrix function generates a pairwise similarity score matrix for for k length given k-mers vs. all possible k-mers combination. The pairwise similarity score is calculated using PAM or BLOSUM substitution matrix; 30, 40, 70, 120, 250 and 62, 45, 50, 62, 80, 100 matrix versions are available for PAM or BLOSUM, respectively. The results are evaluated by global similarity score; higher similarity score indicates more similar sequences for BLOSUM and opposite for PAM matrix.

Usage

```
kmeRs_similarity_matrix(
  q = NULL,
  x = NULL,
  align.type = "global",
  k = 3,
  seq.type = "AA",
 submat = ifelse(test = (match.arg(toupper(seq.type), c("DNA", "AA")) == "AA"), yes =
    "BLOSUM62", no = NA),
  compare.all = FALSE,
  save_to_file = NULL,
  ...
)
```

Arguments

q — query vector with given k-mers

x — kmers to search the query vector against. If unspecified, q will be compared to either other k-mers within q (compare.all = FALSE), or all possible combinations specified by the parameter k

align.type — type of alignment, either global or local. global uses Needleman-Wunsch global alignment to calculate scores, while local represents Smith-Waterman local alignment instead `k` length of k-mers to calculate the similarity matrix for, defaults to 3; e.g. for DNA, N = 4^3 = 64 combinations if `k = 3`;

`seq.type` type of sequence in question, either 'DNA' or 'AA' (default); this will also modify `q` accordingly, if `q` is unspecified.

`submat` substitution matrix, default to 'BLOSUM62'; other choices are 'BLOSUM45', 'BLOSUM50', 'BLOSUM62', 'BLOSUM80', 'BLOSUM100', 'PAM30', 'PAM40', 'PAM70', 'PAM120' or 'PAM250'

`compare.all` if `TRUE`, the query vector will be compared to all possible combinations of k-mers (defaults to `FALSE`)

`save_to_file` if specified, the results will be saved to the path in comma-separated format (.CSV)

`...` other parameters, e.g. gap opening/extension penalties (`gapOpening`, `gapExtension`), or DNA match/mismatch scores (`na.match`, `na.mismatch`)

Value

similarity matrix is returned as a data.frame

Examples

```
# Simple BLOSUM62 similarity matrix for all amino acid nucleotides
kmeRs_similarity_matrix(submat = "BLOSUM62")
```

kmeRs_statistics *Calculate row and column statistics for a k-mer similarity matrix*

Description

The `kmeRs_statistics` function calculates basic statistics and returns the similarity matrix with calculated results or summarized table with statistics only when margin.only is set to TRUE

Usage

```
kmeRs_statistics(x, margin.only = FALSE, digits = 2)
```

Arguments

`x` Similarity matrix computed by `kmeRs_similarity_matrix`

`margin.only` Should only margin statistics be displayed? Defaults to `FALSE`

`digits` rounding digits, defaults to 2

Value

data.frame with results

Examples

```
# Simple BLOSUM62 similarity matrix for DNA nucleotides
# Sample heptamers
q0 <- c("GATTACA", "ACAGATT", "GAATTAC", "GAAATCT", "CTATAGA", "GTACATA", "AACGATT")
# Compute similarity matrix
example <- kmeRs_similarity_matrix(q0, submat = "BLOSUM62")
# Result as a full matrix
kmeRs_statistics(example)

# Result a summary statistics table
kmeRs_statistics(example, margin.only = TRUE)
```

---

kmeRs_transcript_to_complementary

Translate Given K-mers To Complementary Sequences

Description

The kmeRs_transcript_to_complementary function transcripts DNA given k-mers to complementary sequences

Usage

```
kmeRs_transcript_to_complementary(kmers_given)
```

Arguments

kmers_given	vector contains given k-mers

Value

vector contains complementary sequences

Examples

```
# Returns complementary sequence to GATTACA

kmeRs_transcript_to_complementary('GATTACA')
```

kmeRs_twoSeqSim *kmeRs_twoSeqSim*

Description

Supporting func to kmeRs_show_alignment

Usage

```
kmeRs_twoSeqSim(
  kmer_A,
  kmer_B,
  seq.type = "AA",
  submat = ifelse(test = (match.arg(toupper(seq.type), c("DNA", "AA")) == "AA"), yes =
   "BLOSUM62", no = NA),
  na.match = ifelse(is.na(submat), yes = 2, no = NA),
  na.mismatch = ifelse(is.na(submat), yes = -3, no = NA),
  align.type = "global",
  verbose = TRUE,
  ...
)
```

Arguments

`kmer_A`	given k-mer A
`kmer_B`	given k-mer B
`seq.type`	type of sequence in question, either 'DNA' or 'AA' (default)
`submat`	substitution matrix version, defaults to 'BLOSUM62'; other choices include 'BLOSUM45', 'BLOSUM50', 'BLOSUM62', 'BLOSUM80', 'BLOSUM100', 'PAM30', 'PAM40', 'PAM70', 'PAM120' and 'PAM250'; this parameter is ignored if `na.match` and `na.mismatch` are specified
`na.match`	for DNA sequences, what should the score for exact match be?
`na.mismatch`	for DNA sequences, what should the score for mismatches be?
`align.type`	"global" or "local"
`verbose`	= TRUE
`...`	other parameters, e.g. gap opening/extension penalties (`gapOpening`, `gapExtension`) for generating a DNA base substitution matrix

Index