Overview

SequenceTools is a package for processing and analysing biological sequence alignments, with a focus is on protein.

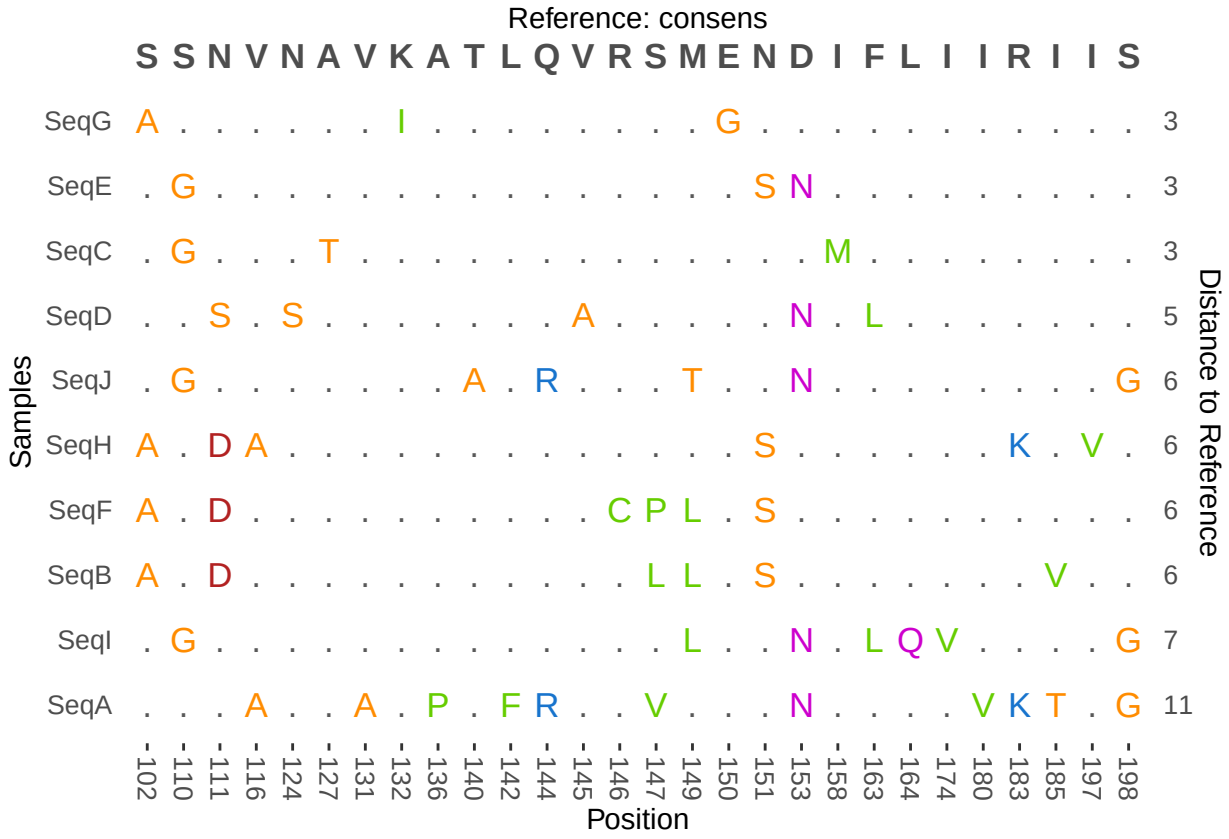
Installation

```
# Currently the only way to install it is over github
install.packages("devtools")
devtools::install_github("ltschmitt/SequenceTools")
```

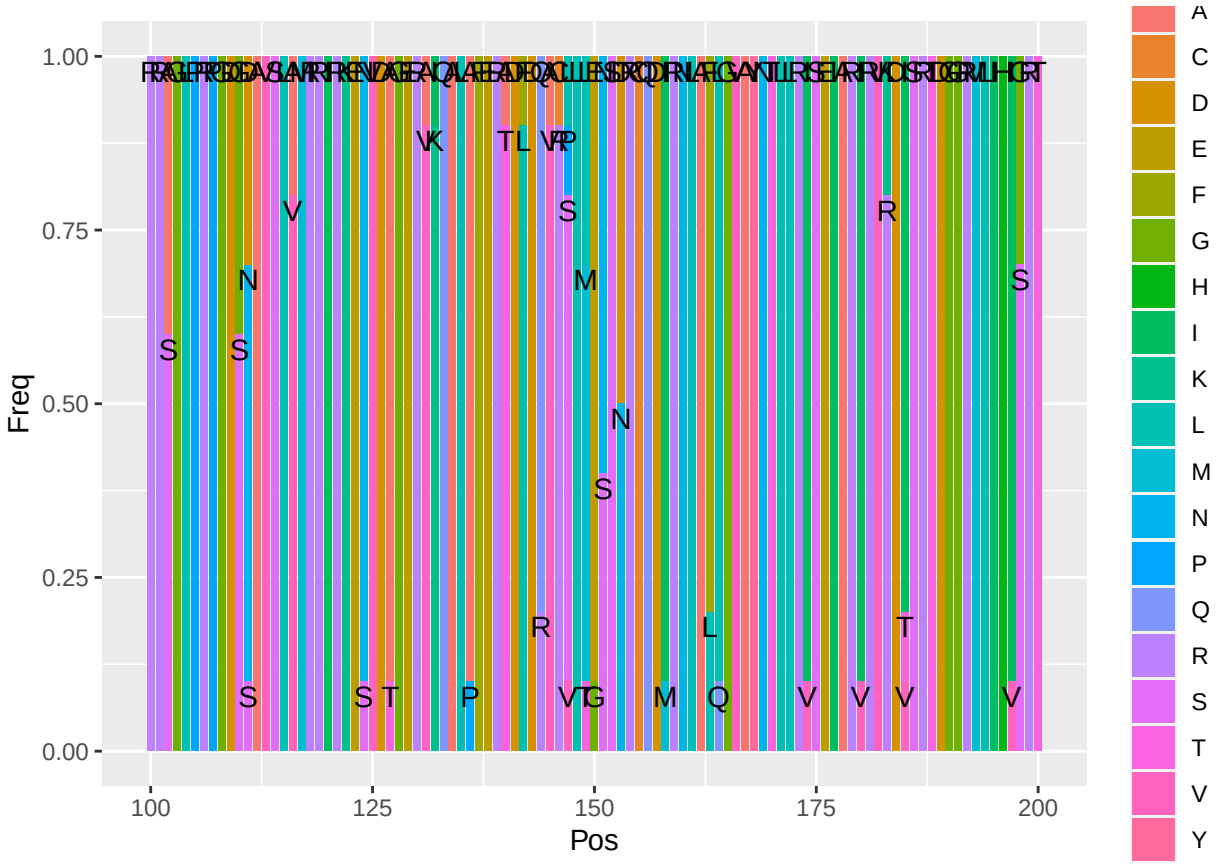
Usage

You can read in fasta files of DNA or amino acid single letter using *read_alignments*. The output is a named character vector, which makes further processing uncomplicated. An example how to read in a fasta file, generate a consensus sequence and to plot the alignment of all the reads.

```
library(SequenceTools)
library(tidyverse)

# read in single line fasta
seqs = read_alignments(input = 'data_raw/cre-variants.fa', naming = 'headers')
#> reading 1 sequence file(s)...

# make consensus sequence, sequences need to be same length and the consensus is made with the sequence
consens = generate_consensus(setNames(seqs, rep('consens',length(seqs))))
#> Warning in generate_consensus(setNames(seqs, rep("consens", length(seqs)))): Frequency tie for sampl
#> Warning in generate_consensus(setNames(seqs, rep("consens", length(seqs)))): Frequency tie for sampl
#> Warning in generate_consensus(setNames(seqs, rep("consens", length(seqs)))): Frequency tie for sampl
#> Warning in generate_consensus(setNames(seqs, rep("consens", length(seqs)))): Frequency tie for sampl
#> Warning in generate_consensus(setNames(seqs, rep("consens", length(seqs)))): Frequency tie for sampl

# plot alignment with consens as reference, makes ggplot element
plot_alignment(Alignment = seqs, Reference = consens, seqrange = 100:200) + theme(legend.position = 'no
```




```
# convert the sequences to a "long" format which makes working with tidyverse tools very easy
seqs = read_alignments(input = 'data_raw/cre-variants.fa', naming = 'filenames')
#> reading 1 sequence file(s)...
lseqs = alignments2long(seqs, get_frequencies = T)
head(lseqs)
#> # A tibble: 6 x 5
#>   Sample         Pos AA        n  Freq
#>   <fct>        <int> <chr> <int> <dbl>
#> 1 cre-variants     1 M        10   1
#> 2 cre-variants     2 A         1   0.1
#> 3 cre-variants     2 S         6   0.6
#> 4 cre-variants     2 T         3   0.3
#> 5 cre-variants     3 D         5   0.5
#> 6 cre-variants     3 N         5   0.5

lseqs %>% filter(Pos %in% 100:200) %>% ggplot(aes(x = Pos, y = Freq, fill = AA, label = AA)) + geom_col
```



Alternatives

This package was developed to suite my preferences, your might want something different. Here are some other R packages I found to be useful with similar functions:

DECIPHER sequence tool package sequence alignment tool