RNAfriends

```
library(RNAfriends)
```

GitHub repository: https://github.com/rforbiodatascience25/group_09_package.git https://github.com/rforbiodatascience25/group_09_package.git

Authors:

- Selina Friesen - s252027
- Marta Sánchez - s254791
- Sergi Fornós - s253693
- Tommaso Ballocci - s257151
- Ivan Musalyk - s225210

Package description

This package is based on the central dogma of molecular biology, which states that a DNA filament (sequence of nucleotides) is transcribed into RNA, which is finally translated into aminoacid sequences (that form proteins).

With this package, you can recreate a DNA sequence, transcribe it, translate it and visualize the relative abundance of the aminoacids.

The functions that this package contains are:

- RNAfriends::makeDNA()
- RNAfriends::function2()
- RNAfriends::codonize()
- RNAfriends::RNA_aminoacid_translation()
- RNAfriends::plot_aminoacids_abundance()

makeDNA

Use this function to generate a random sequence of DNA, of the length that you prefer. generateseq requires an integer that will define the length of your DNA sequence.

Bear in mind that, to obtain meaningful results with this package, your DNA length should be >= 3.

Example:

```
RNAfriends::makeDNA(250)
#> [1] "TCAAGCAAATTTGATAATTCAGGTATCCCCGGGTTGGCCCGTAGTAGAAAAGATCTATGGTCATGGATACAACGTGACGTAAGGTTCGCAGGAGC
```

function2

To transcribe DNA into RNA, the nucleotide thymine (T) need to be changed into uracil (U), this function is responsible for this change.

```
DNAsequence <-"CTAAAAAAGGGGGGTTTTTTTTTTCCCCCCCCCCGTCAGCTACGTATGATGAGTAGATGCATATTTTT"
RNAfriends::function2(DNAsequence )
#> [1] "CUAAAAAAGGGGGGUUUUUUUUUCCCCCCCCCCGUCAGCUACGUAUGAUGAGUAGAUGCAUAUUUUU"
```

codonize

The function returns the codons of a DNA or RNA sequence.

```
sequence = "ATGACATCGAUGTTAATGACATCGAUGTTAATGACATCGAUGTTAATCGAUGTTAATGACATCGAUGTTAATGACATCGAUGTTAATGAGA

RNAfriends::codonize(sequence, start = 1)
#>  [1] "ATG" "ACA" "TCG" "AUG" "TTA" "ATG" "ACA" "TCG" "AUG" "TTA" "ATG" "ACA"
#> [13] "TCG" "AUG" "TTA" "ATC" "GAU" "GTT" "AAT" "GAC" "ATC" "GAU" "GTT" "AAT"
#> [25] "GAC" "ATC" "GAU" "GTT" "AAT" "GAG" "ACT" "TGA" "CAT" "CGA" "UGT" "AAT"
#> [37] "CGA" "UGT" "TAA" "TGA" "CAT" "TGA" "CAT" "CGA" "UGC" "GAU" "GTT" "AAT"
#> [49] "GAC" "ATC" "GAU" "GTT" "AAT" "GAG" "ACA" "TCG" "AUG" "TTA" "ATG" "ACA"
#> [61] "TCG" "AUG" "TTA" "ATG" "ACA" "TCG" "AUG" "TTA" "ATG" "ACA" "TCG" "ATC"
#> [73] "GAU" "GTT" "AAT" "GAC" "ATC" "GAU" "GTT" "AAT" "GAC" "ATC" "GAU" "GTT"
#> [85] "AAT" "GAC" "ATC" "GAU" "GTT"
```

RNA_aminoacid_translation

The function decodes a given RNA sequence to its aminoacid components.

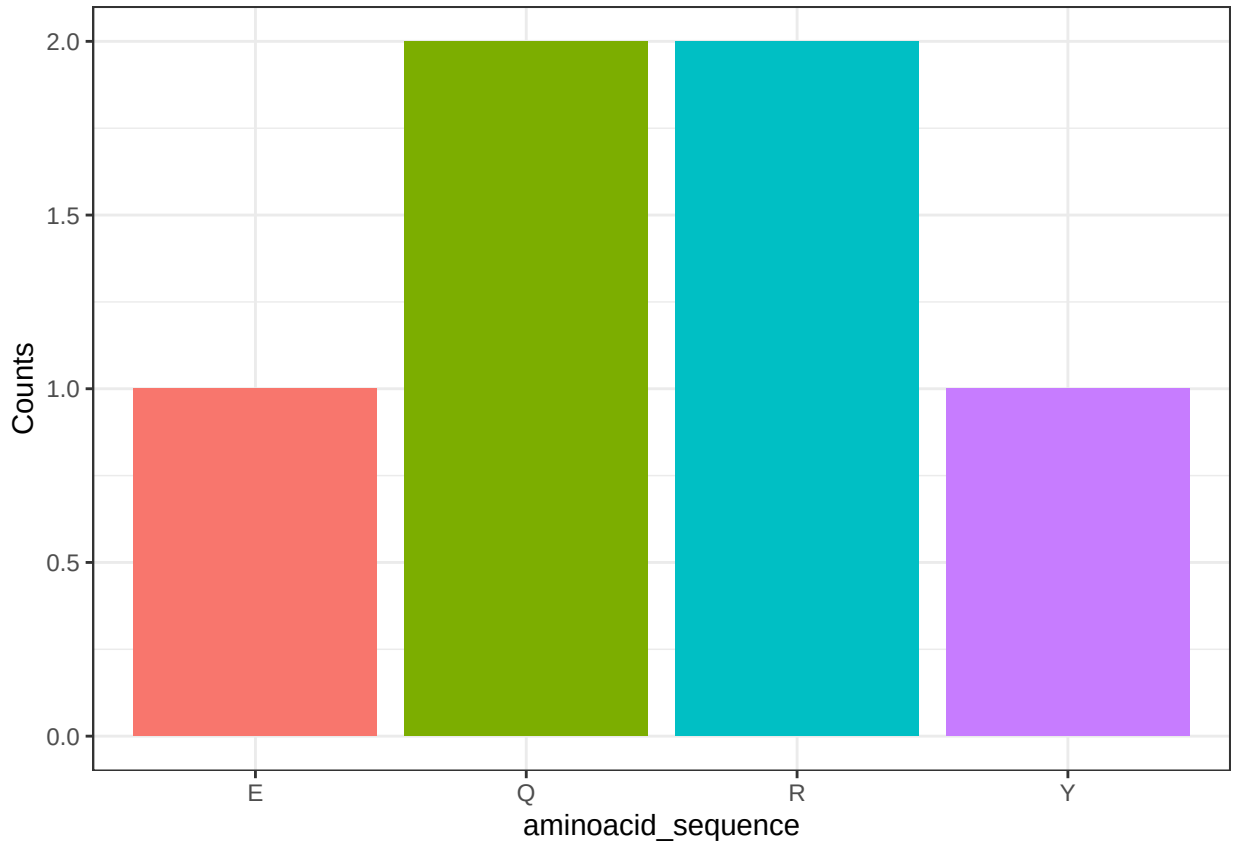
```
RNA_sequence = c("UUU", "AAA", "AUA")
RNA_sequence
#> [1] "UUU" "AAA" "AUA"
peptide_sequence = RNAfriends::RNA_aminoacid_translation(RNA_sequence)

peptide_sequence
#> [1] "FKI"
```

plot_aminoacids_abundance

The function plots the aminoacids abundance for a given peptide-sequence.

```
peptide_sequence = "QERRYQ"
RNAfriends::plot_aminoacids_abundance(peptide_sequence)
```



Combined example

Here we provide an example, where you can combine the functions.

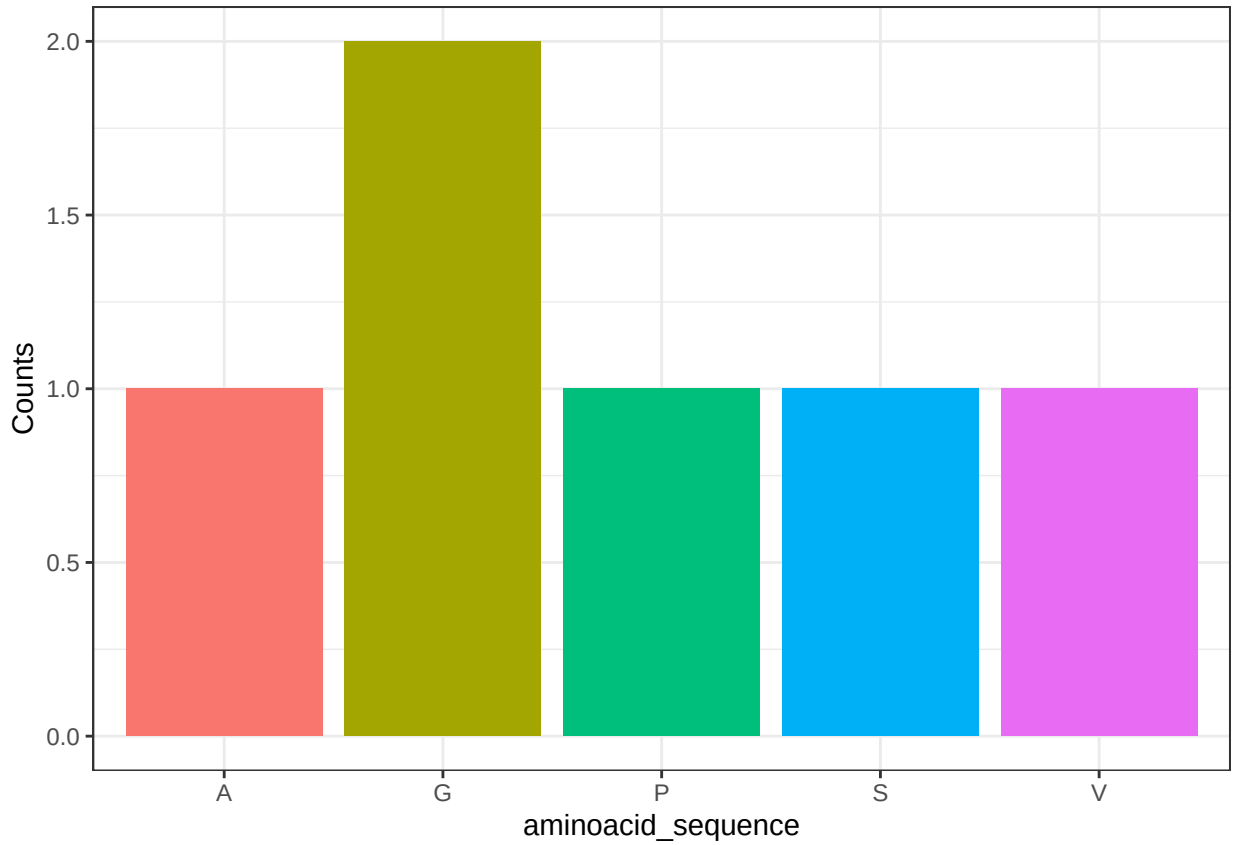
```
my_dna <- RNAfriends::makeDNA(20)
print(my_dna)
#> [1] "TCAGGTGCCGTAGGTCCGCG"

my_rna <- RNAfriends::function2(my_dna)
print(my_rna)
#> [1] "UCAGGUGCCGUAGGUCCGCG"

my_rna_codonized <- RNAfriends::codonize(my_rna, start = 1)
print(my_rna_codonized)
#> [1] "UCA" "GGU" "GCC" "GUA" "GGU" "CCG"

peptide_sequence = RNAfriends::RNA_aminoacid_translation(my_rna_codonized)
print(peptide_sequence)
#> [1] "SGAVGP"

RNAfriends::plot_aminoacids_abundance(peptide_sequence)
```



Use cases:

The package can be used to visualize a given DNA sequence and illustrate nicely what Aminoacid it can produces.

Additional functions could include some analysis of the produced peptide sequence.

Discussion Package dependencies

The packages are added via import statements in the description of the function.

- @importFrom ggplot2 ggplot aes geom_col theme_bw theme
- @importFrom stringr str_split boundary str_count

It is good to limit the number of dependencies, as less error could occur and the maintenance gets simpler. The installation speed is reduced with more packages.

However it cannot always be avoided, as one should reuse pre-existing working code (like the stringr and ggplot2 packages). Especially with ggplot2, the package has a complex functionality, so it doesn't make sense to recreate it from scratch.

Differences in dependency function calls:

Import from package function means that specified function is imported into the package's namespace, so it can be used directly without the package:: prefix. However it risks conflicts with redundant function names from other packages.

Loading a function via *package::function()* is an explicit call to another package. It doesn't hold the risk of unwanted behavior.