Code ▾

Assignment 5 - Analysis

- Nadejda Boev (20056079)
- Due Date - 2022/02/16
- Github user - 16nbb1
- Github link - https://github.com/16nbb1/Biol432_A5_Rentrez (https://github.com/16nbb1/Biol432_A5_Rentrez)

Hide

```
url <- "https://upload.wikimedia.org/wikipedia/commons/f/f3/Borrelia_burgdorferi_%28CDC-PHIL_-6631%29_lores.jpg"
knitr::include_graphics(url)
```

Learn more about Borrelia burgdorferi from Wikipedia (https://en.wikipedia.org/wiki/Borrelia_burgdorferi)

Loading in packages and Sequences csv

Hide

```
library(dplyr)
(df  = read.csv("./Sequences.csv"))
```

Name <chr>
>HQ433692.1 Borrelia burgdorferi strain QLZP1 16S ribosomal RNA gene, partial sequence
>HQ433694.1 Borrelia burgdorferi strain CS4 16S ribosomal RNA gene, partial sequence
>HQ433691.1 Borrelia burgdorferi strain GL18 16S ribosomal RNA gene, partial sequence

3 rows | 1-1 of 2 columns

Counting A/T/C/G

Count the number of each base pair (A, T, C and G), in each of the three sequences

- I've looped through all my rows and done a strsplit and searched annd tabulated each individual letter from all the sequences in each row i
- This gets appended to the larger dataframe
- I've then printed out the sequence and printed out the full row using the row index

Hide

```
for (i in 1:nrow(df)) {

  df[i,'A'] = (data.frame(rbind(table(strsplit(df$Sequence[i], ""))))$A)
  df[i,'T'] = (data.frame(rbind(table(strsplit(df$Sequence[i], ""))))$T)
  df[i,'C'] = (data.frame(rbind(table(strsplit(df$Sequence[i], ""))))$C)
  df[i,'G'] = (data.frame(rbind(table(strsplit(df$Sequence[i], ""))))$G)

  # PRINTING  OUT
  print(df$Sequence[i])
  print(df[i,])

}
```

```
[1] "AGCATGCAAGTCAAACGAGATGTAGCAATACATCTAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAA
TACCGAATAAGGTCAATTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGTTAGTTGGTAGGGTAAATGCCTACCAAGGC
GATGATAAGTAACCGGCCTGAGAGGGTGAACGGTCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGAC
GGAGCGACACTGCGTGAATGAAGAAGGTCGAAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACGAAGTGATGACGTTAATTTATGAATAAGCCC
CGGCTAATTACGTGCCAGCAGCCGCGGTAATACG"
```

◂	A	T	C	G
	<int>	<int>	<int>	<int>
	154	114	82	131

1 row | 4-7 of 6 columns

```
[1] "AGCATGCAAGTCAAACGGGATGTAGCAATACATTCAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAA
TACCGAATAAGGTCAGTTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGCTAGTTGGTAGGGTAAATGCCTACCAAGGC
AATGATAAGTAACCGGCCTGAGAGGGTGAACGGTCACACTGGAACTGAGATACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGAC
GGAGCGACACTGCGTGAATGAAGAAGGTCGAAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACAAAGTGATGACGTTAATTTATGAATAAGCCC
CGGCTAATTACGTGCCAGCAGCAGCGGTAATACG"
```

◂	A	T	C	G
	<int>	<int>	<int>	<int>
	155	114	81	131

1 row | 4-7 of 6 columns

```
[1] "AGCATGCAAGTCAAACGAGATGTAGTAATACATCTAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAA
TACCGAATAAGGTCAATTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGTTAGTTGGTAGGGTAAATGCCTACCAAGGC
GATGATAAGTAACCGGCCTGAGAGGGTGAACGGTCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGAC
GGAGCGACACTGCGTGAATGAAGAAGGTCGAAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACGAAGTGATGACGTTAATTTATGAATAAGCCC
CGGCTAATTACGTGCCAGCAGCCGCGGTAATACG"
```

◂	A	T	C	G
	<int>	<int>	<int>	<int>
	154	115	81	131

1 row | 4-7 of 6 columns

Calculate GC Content (% of nucleotides that are G or C) and create a final table showing GC content

- Since I have the A/T/C/G totals from above, I can use dplyr's mutate to calculate the number of G/Cs across the whole sequence
    - I've presented it as a percentage
- I then renamed and shuffled the column names to match the exapple

Hide

```
(gc = df %>%
  mutate(GC = sprintf("%0.1f%%", GC_Content = 100*(C+G)/ (A+T+C+G)),
         ID = gsub(">", "", unlist(lapply(strsplit(df$Name, " "), '[[', 1)))) %>%
  select(ID, GC) %>%
  rename('Sequence ID'= ID, 'GC Content'= GC))
```

Sequence ID	**GC Content**
<chr>	<chr>
HQ433692.1	44.3%
HQ433694.1	44.1%
HQ433691.1	44.1%

3 rows