hmw2_writeup

Jason Needham

11/5/2019

Problem 1

Approach

In order write a script in R that could be called from the command line that take a fasta file and a sequence id (or text file containing a list of sequence ids), I began with the general script we wrote in class. That script, hoever, needed to be tweaked since the problem itself asked that the script took a *single* sequence if but in class, we wrote it to take a text file list of ids. I solved this by using the grepl() function to see if the second argument contatined ".txt". If the argument contains ".txt", then the script treats it like a file containing a lsit of sequence id's and parses it appropriately. If, however, the argument does not have ".txt", then it treats it like a single sequence id (as per the original homework question), and calls the corresponding fasta sequence.

The next thing that needed tweaking from the original script we wrote in class, was to write a way to handle sequence id's that *may* or *may not* have *.SomeNumber* attached to the end of them. This was handled by placing an "or" statement within the search pattern:

```
pattern <- "^(\\S+)\\.|^(\\S+)$"
```

This way the pattern can either have the period **or** the numbers go up to the end of the line. Either way, the sequence id is captured.

Finally, I had to determine a way to get the sequence id called into standard output to be viewed in the command line. As I was unable to find in the documentation to get WriteXStringSet() to print to stdout(), I used WriteXStringSet() to write the output to a file, then used a regular ol' write.table() command to print to stdout()

Code

Please see the accompanying **extractr.R** file for the code.

Results

To test the code, call:

```
Rscript extractr.R <fastaFile> <sequence ID -or- text file containing sequence IDs>
<nameForTheOutfile>
```

Problem 2

Approach

The first part of solving this problem was to set up the same general framework as the first homework problem by using the commandArge() function to get the scoring matrix and and fasta filenames, as well as to call in the Bioststrings package. To get the scoring matrix into R, I used the read.table() function

with header set to True, and the skip option set to 6 to skip the comment lines. This set my column names and row names to the corresponding amino acid letters, which made calling the appropriate values from matrix as easy as directly calling matrix[AA_1, AA_2]. If a different matrix is used with a different number of comment lines, this would have to be altered. For some reason, however, the final '*' could be used as a rowname but not a columnname (as it was converted to an 'X.'), which would break the scoring system at the end of the for-loop. For this reason, the asterisks were all converted to "X.".

Importing the fasta file used the same Biostrings readAAStringSet() function as was used in the previous script. Since there are only two fasta files which should be aligned, the scripts iterate through each amino acid in the first string and compares it with the corresponding numbered amino acid in the second string. It returns each amino acid as a character which is used to call the corresponding value in the matrix. The scores are added with every iteration and indels are set to add 0 to the score. If, however, the user wished to scored indels differently, that number can be altered.

Finally, a print command was used to call the score of the two fasta sequences within the command line using print() with a stdout() option.

Code

To see the code, open the accompaning blosumScore.R file.

Results

Running the BLOSUM62 matrix along with the dataset given on the course website gave a score of 104 for the 107 long strong of amino acids. This can be checked by running:

```
Rscript blosumScore.R BLOSUM62 ex_align.fasta
```

in the command line. This should return the string:

```
"The score for the detected sequences is: 104"
```