Exam 4

BIO/DSP 439/539

The genome of an organism consists of a very long sequence of nucleic acids. From a computational perspective, this can be thought of as a long string of letters where the alphabet consists only of A, C, G, and T (e.g. ATGTCTGTCTGAA). This string can be divided into overlapping substrings of length k. For example, if k = 2 then the substrings for the example sequence above are AT, TG, GT, TC, CT, TG, GT, TC, CT, TG, GA, and AA.

Genome assemblies are constructed from sequenced genome fragments by identifying each substring (called a k-mer) and the substring that comes after it in the sequence. Finding the order of all of the substrings in the fragments produces a whole genome. For example, if k = 2 then the substring TG in the example is either followed by GT or GA.

Write a python script that, when run using the command line, gets information about each k-mer and its subsequent k-mers in a file of sequence fragments (provided). This would be the starting point to assemble a genome from these sequences; however, the process of genome assembly is beyond the scope of this class.

To achieve this goal:

- Start by identifying all different observed substrings of size 2 in the example.
- Extend this work to identify all different observed substrings of any size, specified as k.

In your final submission include the following:

1. Define a function to identify all substrings of size k, where k is specified as an argument, for a single sequence (also specified as an argument), and all unique possible subsequent substrings of each substring.
2. Define a function that uses the prior function to identify all possible substrings and their subsequent substrings for all sequences read from a file.
3. In my contrived example genome, there is at least one value of k where every substring has only one possible substring that follows it. Define a function to identify the smallest value using the prior functions.
4. Be sure that all your functions have appropriate docstrings.
5. Be sure that all code is commented fully. Note that with the proliferation of ChatGPT as well as stack overflow, if you do not provide explanatory comments I will assume that you don't understand the code and you will be graded accordingly.
6. Write a script to *thoroughly* test each of your functions that I can run with pytest.
7. Use the main function in your script to input the sequence data filename from the command line (you can find it as reads.fa in the BIO439539 folder in the shared folder) and print the value identified by function 3.
8. Create a github repository including an *informative* README (in markdown) to submit your work. I should be able to clone the repo, run pytest, and run the script on my computer without changing any of your code.

Submit the link to your github repo.