Bioinformatics Lab Assignment 4: K-Mers and the Origin of Replication

By Noah Segal-Gould

Background: For biologists who are interested in the replication of DNA, it is often useful to determine where that particular replication began in a sequence of nucleotides. This location is called the oriC or origin of replication. During replication, the DnaA box typically binds to sequences around 9 nucleotides in length, which also typically occur in a larger sequence multiple times. So, the frequency of nucleotide substrings of some length k (called k-mers) can be utilized to find the origin of replication in a given nucleotide sequence because a higher frequency implies that it may be the origin of replication. In a programming language like Python, it is possible to find these substrings as well as their frequencies in some larger string, and then print the most frequent substrings in a way which may be useful to biologists who are interested in finding the sequence which may be the origin of replication.

Methods: The two most important functions in my program are `find_reading_frames_k_mers` and `find_non_reading_frames_k_mers`. Both return dictionaries. In the first case, the dictionary contains 6 keys (from 1 through 6), each of which refers to a value which is a reading frame dictionary. Each of those dictionaries then contains a dictionary with 3 keys. These keys are 3, 6, and 9. These refer to the 3-mers, 6-mers, and 9-mers which were assigned to be found for each reading frame. For each key, the value in the dictionary is the list of the most common k-mers in that reading frame for that value of k (3, 6, and 9). This is accomplished by using `re.findall` to split apart the nucleotide sequence for each frame into groups of substrings of length k, and then using the `Counter` object from the `collections` module in Python to find the top 4 most common occurrences in each list. In the second case, the dictionary is similar but does not have 6 keys for each reading frame at its outermost level. Instead, the only key is the integer 1, which applies to the value of a dictionary with 3 keys which is similar to the one in the previous example. However, instead of finding all consecutive substrings of lengths 3, 6, and 9, I used a `find_k_mers` function I adapted from the English Wikipedia page for K-Mers. This function takes as inputs a string and some integer k, then outputs a list of K-Mers. The major difference between this and the regular expressions used in the reading frames case is that when finding possible substrings, this function includes partial substrings of previous matches as per the example in part 3a of the assignment description. Thus, in terms of the data structures used to represent their outputs, the object returned by `find_reading_frames_k_mers` is six times the size of `find_non_reading_frames_k_mers`. I wrote one more function to print tables. It always prints the top four most frequent K-Mers in order for each k (3, 6, and 9) either for all 6 reading frames or just for the sequence without considering reading frames at all. As its input, the `print_table` function takes first a nucleotide sequence string and second a boolean True or False for its `use_reading_frames` keyword argument, which is True by default. Finally, I run all my tests in a `main` function.

Results: When the program is run, the `main` function is called and the tests are shown. I acquired a nucleotide sequence for Vibrio Cholerae and one for Thermotoga Petrophila from the

course Moodle 2. For both of these sequences, two K-Mers tables are printed: the one which includes reading frames and the one which does not. For each group of 3, 6, and 9-mers, the top 4 most common substrings are shown. The output of the program is as follows:

```
K-Mers for Vibrio Cholerae, with reading frames:
-------------------------------------------------
1|3|TGA          GAT          CTT          CTC
 |6|GATCAA       TGGCCA       ATCAAT       CGTAAG
 |9|ATCAATGAT    CAACGTAAG    CTTCTAAGC    ATGATCAAG
 ================================================
2|3|ATC          GAT          CAT          TCT
 |6|ATCAAG       ACTTGT       CATGAT       TCAATG
 |9|TCAATGATC    AACGTAAGC    TTCTAAGCA    TGATCAAGG
 ================================================
3|3|TGA          TCA          ATG          CAA
 |6|TGATCA       CAATGA       TCAACG       TAAGCT
 |9|CAATGATCA    ACGTAAGCT    TCTAAGCAT    GATCAAGGT
 ================================================
4|3|ACT          CTA          GAA          GAG
 |6|CTAGTT       ACCGGT       GCAAAG       CTAGTA
 |9|GTAGCAAAG    CGAGAACTA    GTTCGACGA    TTCTTACTA
 ================================================
5|3|AGT          ACT          TAC          AAA
 |6|ACTAGT       AGTAGC       AGAACT       ACGACG
 |9|AGAACTAGT    TCGACGACG    CTTACTAGT    AAATGCCTT
 ================================================
6|3|TAG          CTA          GTA          AGA
 |6|TAGTTC       GTACTA       TGAACA       TAGTAG
 |9|GAGAACTAG    TTCGACGAC    TCTTACTAG    AAAATGCCT
 ================================================

K-Mers for Vibrio Cholerae, without reading frames:
---------------------------------------------------
1|3|TGA          ATC          GAT          TCA
 |6|TGATCA       ATGATC       GATCAA       ATCAAG
 |9|ATGATCAAG    CTCTTGATC    TCTTGATCA    CTTGATCAT
 ================================================

K-Mers for Thermotoga Petrophila, with reading frames:
------------------------------------------------------
1|3|AAC          TTT          AAA          TAT
 |6|AGGTTT       GGTGGT       CCTACC       GTGGTA
 |9|AACTCTATA    CCTCCTTTT    TGTCGAATT    TGTGTGATT
 ================================================
2|3|ATT          TAC          ACT          TTT
 |6|GTGGTA       AATTGA       ACTTAC       CTACCA
 |9|ACTCTATAC    CTCCTTTTT    GTCGAATTT    GTGTGATTT
 ================================================
3|3|AAA          ACC          ATT          TTT
 |6|TGGTAG       CTGAAA       CTTACC       ATTTCA
 |9|CTCTATACC    TCCTTTTTG    TCGAATTTG    TGTGATTTA
 ================================================
4|3|TTT          AAA          TTG          CAT
```

```
 |6|TCCAAA       CACCAT       GGATGG       CCACCA
 |9|TTTCTCCAC    AACTAGACT    CGTCATATT    GATTATTAT
 ================================================
5|3|TTT          TGG          TAA          AAA
 |6|ACCATC       GACTTT       ATGGTG       TAAAGT
 |9|TCTCCACCA    CTAGACTTT    TCATATTAA    TTATTATCG
 ================================================
6|3|TAA          ATG          TGA          AAA
 |6|ACCATT       TTAACT       GATGGT       TGAATG
 |9|TTCTCCACC    ACTAGACTT    GTCATATTA    ATTATTATC
 ================================================

K-Mers for Thermotoga Petrophila, without reading frames:
------------------------------------------------------------
1|3|AAA          TTT          ATT          ACC
 |6|TACCAC       ACCTAC       CCTACC       CTACCA
 |9|ACCTACCAC    TGGTAGGTT    GGTAGGTTT    AAACCTACC
 ================================================
```

For Vibrio Cholerae, in its non-reading frames table there exists a sequence along with its own reverse complement. The same is true of Thermotoga Petrophila. I have highlighted these sequences in this report. Because these sequences occur with their complements with such frequency, they are likely to be the origins of replication for the DNA nucleotide sequence.

Conclusion: I completed the tasks assigned in the group work assignment which was given out in class. I tried to make my outputs look nice, as well. I was unable to answer the final question regarding the most frequent 9-mer in the human genome.

Acknowledgements: I worked in a group on this assignment. Included in this group were Zoe Terhune and Aaron Krapf. I adapted some code from the English language Wikipedia article on K-Mers.