Chris Hainzl

**Steps**

1. Inside the ‘main’ function, open the 'dna.txt' file for reading.
2. Read the file into a string variable named 'dna'.
3. After the file has been read into the variable, close the file.
4. Create another string variable named ‘new\_dna’, which will store the original ‘dna’ string, but with the newline characters removed (so it does not mess up our iterations).
5. Calculate the longest possible kmer length by dividing the length of the string stored in the ‘new\_dna’ variable by 2.
6. Using a for loop starting from 5, and continuing onto the longest possible kmer length, call on the 'find\_k\_mers' function. This will take the 'new\_dna' variable and the current kmer length as parameters.
7. Inside the 'find\_k\_mers' function, open a file named ‘kmers.txt’ for appending. This will help ensure that no previously read kmers are overwritten.
8. Inside a for loop, create a dictionary named ‘k\_mer\_dictionary’ using Python’s dict() function. The for loop should be set up like this: for index in range(length of DNA sequence – kmer length + 1)
9. Determine the current kmer that is being analyzed by slicing the DNA sequence like this: kmer = dna\_sequence[index : index + kmer length]
10. Using Python’s count() function, determine how many times the kmer appears in the DNA sequence, and then associate that count with the kmer in ‘k\_mer\_dictionary’.
11. If the count is greater than 1 (meaning the kmer appears more than once), print out the contents of ‘k\_mer\_dictionary’, then write the kmer to ‘kmers.txt’.
12. After finishing the iterations, close the ‘kmers.txt’ file before exiting the program.

**Pseudocode**

procedure find\_k\_mers(dna\_sequence : string, k : integer):

k\_mers\_file = openWrite(‘kmers.txt’)

for i := 0 to length of dna\_sequence – k + 1:

kmer\_dictionary = dict()

kmer = dna\_sequence[i : i + k]

kmer\_dictionary[kmer] = dna\_sequence.count(kmer)

if number associated with kmer in kmer\_dictionary > 1:

print(kmer\_dictionary)

k\_mers\_file.write(kmer + '\n')

k\_mers\_file.close()

procedure main():

dna\_file = openRead(‘dna.txt’)

dna = dna\_file.read()

dna\_file.close()

new\_dna = dna.replace(‘\n’, ‘’)

max\_kmer\_length = length of new\_dna // 2

for i := 5 to max\_kmer\_length:

find\_k\_mers(new\_dna, i)