Scott Cagnard

Programming Assignment 2

March 7, 2016

CSC 201

Programming Assignment #3----Report

The main objective of this programming assignment was to get more comfortable with string operations and manipulation in Python, as well as using conditional statements and loops to solve a simplified version of a real world application.

Throughout the program I use the primary variable DNA, and other similar variables such as DNA1, DNA2, and such, to represent all the DNA strings that the user will input throughout the program. The other primary variable I use is count (count1, count2, .etc.), which is very helpful when we use if statements and for loops within the program. Using the variable count helps us total the amount of matches between two strings and also determine the validity of a string when we compare it to the length of the string. In this instance, these variables are acting as local variables as they are a serving a specific purpose each time a new DNA string is entered. In order to find the mirrored string, we must simply use the .replace function and replace “A” with “T”, “T” with “G”, “C” with “G”, and “G” with “C”. To produce a reverse string, we must simply take the total string and put it in reverse, using [::-1]. To determine the validity of a DNA string, we have to create an if statement within a for loop. The for loop allows us to detect each character within a given string, and we create an if statement with the condition that the each character in the DNA string must equal “A”, “T”, “C”, or “G”. We also have a variable count set equal to zero, so each time one of the four specific characters is detected, the count goes up one. If the DNA string is valid, then the length of the string should equal the variable count once the for loop is complete. We also find the mirrored strings of two given strings, and if any of them are equal, then we know that those two given strings must be part of the same fragment. When trying to calculate the longest length of the matching subsequence between the two strings, I was unable to return one single value, but returned a list instead. I had to verbally address that the highest number from the list was the longest length, because the if statement caused the variable count to reset to zero each time a match was not found, so the highest number provide the longest match between the two DNA strings. As well, I was unable to produce a matching reference sequence for the longest matching subsequence, but was able to produce the reference sequence of the most number of matches between the entered string. I was able to do this by using an if statement within a for loop. In this particular case, the for loop detects characters in two strings. I compared the entered string by the use and each of the three reference strings separately, and using variations of the variable count, was able to get count1, count2, and count3 each equaling the total number of matches of the entered DNA string and ref1, ref2, and ref3, all respectively.