Jacob Wahbeh

CS32 Lec 1

March 14 2019

Project 4

1. I believe that I sorted out most of the bugs in the project. From testing however, I noticed that for the findRelatedGenomes and findGenomesWithThisDNA functions that if two genomes have the same sequence in the same position, the function sometimes does not pick out both sequences.
2. I met all the Big-O requirement except for findGenomesWithThisDNA and findRelatedGenomes. For this function I inserted all the matches found including duplicates in the same genome into a set. Then I pulled out the biggest of each genome into the vector. So, my function has time complexity of O(H\*F\*log(S)). Where S is the size of my set. For the second function I created a map for every genome in the library to a corresponding double. I had to iterate through every genome in the library twice. So, the time complexity was O(Q \* X + N) where N is the number of genomes in my library.

3.

Find Function

Declare a node pointer called current and set it equal to the root of the tree

Iterate through all of current’s children

if the child’s associated letter is the same as the first character in key

call the helper function with the child’s next node, key without the first character, and same Boolean for exactMatchOnly

return the vector that the helper function returned

Helper Function – parameters: Node pointer, string, and Boolean

If key’s size is 0 return the value of the Node pointer

Iterate through the children’s of the node pointer

If the associated letter equals the first letter in key and exactMatchOnly is true

call the helper function with the child’s next node, key without the first character, and true for the Boolean

If the associated letter equals the first letter in key and exactMatchOnly is false

call the helper function with the child’s next node, key without the first character, and false for the Boolean

If the associated letter does not equals the first letter in key and exactMatchOnly is false

call the helper function with the child’s next node, key without the first character, and true for the Boolean

return the combined vector consisting of all the vectors that the functions return

FindGenomesWithThisDNA()

If fragment size is less than minimum length or minimum length is less than minimum search length

Return false

Clear the matches vector

Create a set of DNA matches

Crete a vector of genome positions called prefix and set it equal to trie’s find function where the key is the first minimum search length characters of the fragment

If the vector is empty

Return false

iterate through all the genome positions in the prefix vector

create a string called suffix

create an integer called length and set it equal to the minimum search length

if prefix has enough characters after it to be a match

set suffix equal to the remaining characters after prefix

iterate through the remaining fragment

if the corresponding suffix character does not equal fragment’s

and exact match only is true

break;

if the suffix string and fragments have more than one deviation and

exact match only is false

break;

else

increment length

if length is greater than or equal to minimum length

add the DNA match to a set

iterate through the set

find the largest length from each genome

insert into vector matches

if matches is empty

return false

return true