CS 32 Professor Smallberg Lecture 2 Leo Gretzinger

Project 4 Report 3/14/19

1. I finished every method of every class to the best of my ability and I removed all bugs that I know about. The only thing is that when loading some of the extremely long genomes, the program just ends, there are no error messages or anything, I think the file is just too big (taking too long)…
2. I am pretty sure all of my methods meet the big O requirements in the spec – I used recursion wherever I could and only made specific for loops that kept the time complexity to the required limit.
3. Two methods:

Trie::find()

splits the function into two parts, whether an exact match in necessary or not

In the case of exact match necessary, I recursively compare the fragment with every node of the tree.

To do this I looped through the children of each node and if the current index of the

fragment ever equals a child’s label, recurse to the next node.

The pointer ends up at the node with the value for the fragment search.

In the case of exact match unnecessary, I recursively compare the fragment with every node of

The tree.

To do this I looped through the children of each node and if the current index of the

fragment ever equals the child’s label, recurse to the next node

If the character does not match, I mark that this fact and if the number of mismatching

characters ever becomes two, the fragment is invalid

If a match is found, the node is added to the vector

Return the resulting vector with the nodes of success matching

GenomeMatcher::findGenomesWithThisDNA()

Use the find method to create a vector of segment Values (contains a name and a position)

Looping through this vector, I isolate the genome found by the find method

In each iteration of the loop I then have access to the position and genome where the

match is found

Compare the fragment and the genome side by side, starting from the position found by

the segment value (so that mismatches can be counted from the beginning) and adding

to a “potential” string

If the fragment ever mismatch, increment the mismatch counter and if this is ever 2, go

to the next found value (iteration of the for loop)

If the potential string passes minimum length, add it to the vector of matches if

appropriate (longest in that genome)