Lab 12 Report

Ben Joye

For this lab, I chose to reuse my Levenshtein distance algorithm from lab 11 to analyze the similarity of the DNA sequences.

* 1st Step: Read the file containing all of the DNA information, and store it in an easily accessible way. I separated the animal names and the DNA sequences into two separate vectors, but so that the indexes match.
* 2nd Step: Put the matching animal names and DNA sequences into a map so that the sequence can be retrieved via the name.
* 3rd Step: Read in which animal from the list the user wants to compare everything to.
* 4th Step: Loop through the animals and calculate the Levenshtein distance for each animal compared the one the user chose.
  + This step takes a long time depending on how many DNA sequences there are and how long they are.
  + Details on the implementation of the Levenshtein Distance Algorithm can be found in the Lab 11 Report.
* 5th Step: Store all the distances in a vector. Also create a map connecting each distance to the animal name.
* 6th Step: Sort the vector so the smallest number is first.
* 7th Step: Loop through the sorted vector, and print, in order, the name corresponding to each distance and the distance itself.

How to Run:

* make and make run to compile and run.
* Enter the name of the animal you want to compare to (must be exact match from the list printed or else an error will occur)
* Wait for the similarities to be calculated.

What I would change and do better next time:

* Break up the distance calculations into smaller strands. I think doing it this way would be more accurate.
* Reading in the data took me a while to figure out, I could probably simplify it because it only works for this very specific format.

Sources:

https://bitesizebio.com/9445/the-beginners-guide-to-dna-sequence-alignment/