Matthew Daniel

CS 415

March 23, 2017

Sub Project 2a Write-Up

**Genetic Algorithm Description:**

I wrote a genetic algorithm in Python that creates a population of size 50 comprised of strings of length 50 whose characters are selected randomly from the amino dictionary in scoringMatrix.py from lecture. Each creature is assigned a fitness score based upon how many English words of 2 or more characters it contains. To get the number of English words, I search through the string character-by-character, checking to see if the character forms an English word of length 2 to the end of the string using a package called “Enchant.” This process takes O(n2) time, so the algorithm runtime is around 70 seconds. I used a mutation rate of 20% to attempt to escape local maxima. This mutation rate seemed to be successful during assignment 1, so it makes sense to use it here. Crossover occurs in the same way as the in-class algorithm, swapping two characters between two strings with a probability of 0.5.

**Data & Analysis Description**

Each iteration, the most fit creature is written to a file. These creatures are strings of length 50, and there will be 1000 of them, one from each iteration. One file contains the creatures only, while another contains the iteration number as well for comparison later. A third file contains the first and final most-fit creatures that the algorithm produces. A separate program, analyze.py, reads in the creatures from the unlabeled file and reorders them. I will apply a global alignment algorithm to the unlabeled creatures and compare them with creature from the original population. I will attempt to use the Blossom-50 scoring matrix to assess the alignments. Since my fitness function has nothing to do with biological processes, this may be unsuccessful, and I may have to engineer another scoring matrix. I will assume that the lower the alignment score a creature gets, the later it was produced (the higher its relative age). I will sort the creatures by alignment score descending, then compare my array to the file that contains the original iteration information. I will measure the effectiveness of the alignment by the average distance a creature is calculated from where it ought to be by comparison with the unaltered data.

**Potential Issues**

I made sure that my GA codes is easily modifiable to alter the number of iterations, number of creatures outputted to the files, etc. so that I can make changes should my analysis prove unsuccessful. One change that I might make entails modifying my mutation function to favor amino acids with higher comparative scores when mutating a character. I suspect that this may increase the effectiveness of Blossom-50 if I encounter problems with the alignment scores yielding useful results.