Blaise Bowman

CIS 4930: Bioinformatics

Analysis of Algorithm Performance with Varying String Lengths

To test the performance of the algorithm, I generated pairs of random strings with lengths of: 50, 100, 250, 500, 1000, 2,000 and 5,000 characters. I generated these pairs of strings with sequenceGenerator.cpp, which is ran separately from bowman.cpp, since generating random strings within bowman.cpp would yield inaccurate measurements, and with each test of bowman.cpp, the strings would change. The pairs of sequences that I used for evaluation for each string length are included in the zip file.

For the randomly generated strings, we are only concerned with part A – D. So, to test the randomly generated strings, I commented out the code associated with “assignment1.o5.” I separated the code to determine if there are multiple alignments from the code that actually displays all optimal alignments for a given pair of DNA sequences. I was able to determine execution time and memory consumption when computing all optimal alignments for pairs of sequences up to 50 characters in length. However, when I tried to calculate all optimal alignments for DNA sequences 100 characters in length, my computer would freeze.

To test the performance of the algorithm, I utilized Linux. I transferred bowman.cpp and sequenceGenerator.cpp to my remote Linux server with WinSCP and placed them within their own directory. The commands used are described in bowman-readme.txt.

1. String length of 50

real 0m0.024s or 00h:00m:0.024s

user 0m0.007s or 00h:00m:0.007s

sys 0m0.000s or 00h:00m:0.000s

Peak Memory: 276,824B = 0.0003 GB

1. String length of 100

real 0m0.025s or 00h:00m:0.025s

user 0m0.013s or 00h:00m:0.013s

sys 0m0.000s or 00h:00m:0.000s

Peak Memory: 832,872 B = 0.0008 GB

1. String length of 250

real 0m0.088s or 00hh:00mm:0.088s

user 0m0.062s or 00hh:00mm:0.062s

sys 0m0.004s or 00hh:00mm:0.004s

Peak Memory: 4,654,552 B = 0.0042 GB

1. String length of 500

real 0m0.294s or 00hh:00mm:0.294s

user 0m0.252s or 00hh:00mm:0.252s

sys 0m0.008s or 00hh:00mm:0.008s

Peak Memory: 18,234,872 B = 0.017 GB

1. String length of 1000

real 0m0.991s or 00hh:00mm:0.991 s

user 0m0.963s or 00hh:00mm:0.963s

sys 0m0.028s 00hh:00mm:0.028s

Peak Memory: 70,496,976 B = 0.0657 GB

1. String length of 1500

real 0m2.421s or 00hh:00mm:2.421s

user 0m2.179s or 00hh:00mm:2.179s

sys 0m0.184s or 00hh:00mm:0.184s

Peak Memory: 162, 539, 800B = 0.1514 GB

1. String length of 2,000

real 0m4.235s or 00hh:00mm:4.235s

user 0m3.855s or 00hh:00mm:3.855s

sys 0m0.321s or 00hh:00mm:0.321s

Peak Memory: 288,692,408 B = 0.2689 GB

1. String length of 5,000

real 0m25.247s or 00hh:00mm:25.427s

user 0m24.908s or 00hh:00mm:24.908s

sys 0m1.230s or 00hh:00mm:1.230s

Peak Memory: 1,801,607,009 B = 1.6779 GB

The longest string length I tested was 5,000.