CAP5510 - Bioinformatics Fall 2022

Team Members:-
Amarjeet Kumar (3881 7064)
Nikhil Yerra (9545 3265)
Ujwala Guttikonda(5791 4323)

Aim:- To acquire knowledge on sequence alignment by implementing the following three simulations:-
1. Simulator for sequence generator
2. Simulator for sequence partitioning
3. Sequence assembler

Implementation Details

i) Simulator for sequence generator:-

This program will generate a set of new sequences based on the parent sequence using probability-based mutation.

The following are the input parameters:-
1. string: F1 = input file name which will contain DNA sequence in FASTA format
2. integer: k = number of sequences
3. real number: p = mutation probability in [0:1] interval
4. string: F2 = output file name which will contain mutated DNA sequences in FASTA format

We have used the Bio and SeqIo from the Python library to implement the functionality.

To run the program:-
hw-1.py inputfile.fasta 10 0.005 output1.fasta

ii) Simulator for sequence partitioning:

This program will generate fragments from a given input of mutated sequences. The output of the previous implementation will be the input file here.

The following are the input parameters:-
1. string: F1 = input file name which will contain a set of DNA sequences in FASTA format
2. integer: x = minimum fragment length
3. integer: y = maximum fragment length (x ≤y)
4. integer: z = maximum ACCEPTABLE fragment length (x ≤z ≤y)
5. string: F2 = output file name which will contain a set of DNA sequences in FASTA format

For every DNA sequence in the given input file, it will generate the fragments of DNA sequence in FASTA format. The length of the fragment will be chosen randomly between x and y, the maximum allowed length being z.

To run the program:-
hw-2.py output1.fasta 100 200 220 output2.fasta

iii) Simulator for sequence assembler:

The program will generate and assemble DNA fragments given in the input file which will be the output of the previous program and generate a single long DNA sequence as per the given criteria.

The following are the input parameters:-
1. string: F1 = input file name which will contain a set of DNA sequences in FASTA format
2. integer: s = score for match (positive integer)
3. integer: r = penalty for replace (negative integer)
4. integer: g = penalty for delete/insert (negative integer)
5. string: F2 = output file name which will contain a set of DNA sequences in FASTA format

For every DNA sequence in the given input file, it will use a greedy strategy to determine the best possible alignment score using a dovetail alignment for each pair of sequences.

To run the program:-
hw-3.py output2.fasta 1 -1 -3 output3.fasta