Algorithms in Bioinformatics

Problem set 1

Hesam Montazeri and Alireza Fotuhi Siahpirani
Department of Bioinformatics, IBB, University of Tehran

Due date: Ordibehesht 27, 1402

Programming assignments

P1: Graph Traversal Algorithms

a) Write a function that takes the name of a .csv file as input, reads the list of undirected edges from the file, and creates an adjacency matrix from the interactions. Assume that the input is a simple graph. See this file for an example: https://docs.google.com/spreadsheets/d/1Q7ULYvXlxNX80SXIZYBvIayAwfh_YNjT_t5a5xNlXyE/edit?usp=sharing
b) Write a function that, given the adjacency matrix of a graph, performs DFS and returns the resulting tree as an adjacency matrix.
c) Write a function that, given the adjacency matrix of a graph, performs BFS and returns the resulting tree as an adjacency matrix.

P2: Write a function that, given two DNA sequences, performs global alignment. The inputs to the functions are: sequence 1, sequence 2, match reward (positive value), mismatch penalty (negative value), gap penalty (negative value). You should return the global alignment with the highest score, and the corresponding score.

P3: A gene expression data is simulated for ten healthy participants (rows 1 to 10) and ten cancer patients (rows 11 to 20) and on 12500 genes. The goal is to find significantly different genes between the two groups. In the simulated gene expression matrix, only the first 1250 genes are different between two groups. Develop three various statistics (as we defined in the lecture) and compute empirical p-values. Compare your approaches with the t-test and Wilcoxon test in terms of type I and type II errors and other criteria of your choice. Perform a comprehensive analysis.

P4: Burrows-Wheeler Transform

Implement the following functions efficiently for the BWT. Use partial suffix arrays and checkpoints in your implementation.

a) Constructing the BWT
b) Inverting the BWT
c) Exact pattern matching

You may use the existing implementations for suffix arrays if any. Write a complete report. Testing your functions on the E Coli genome for this problem.

P5: Hidden Markov Models
Implement Viterbi algorithm, forward algorithm, Viterbi learning in the context of the problem Q3 of the written problems.

Written problems

Q1: Suffix tree and Suffix array

a) Build the suffix tree for the string "banana".
b) Build the suffix array for the same string.

Q2: Burrows Wheeler Transform

a) Construct the BWT for the string "BIOINFORMATICS".
b) Given the BWT "UERV$DYLFEOOS", decompress it to obtain the original string.
c) Find all occurrences of the pattern "YOU".

Q3. Define an HMM model for predicting the weather, given the following parameters.

- Hidden states: sunny, cloudy, rainy.
- Observable states: hot, warm, cool.
- Transition probabilities:
  - sunny -> sunny = 0.7, sunny -> cloudy = 0.2, sunny -> rainy = 0.1;
  - cloudy -> sunny = 0.3, cloudy -> cloudy = 0.5, cloudy -> rainy = 0.2;
  - rainy -> sunny = 0.2, rainy -> cloudy = 0.4, rainy -> rainy = 0.4.
- Emission probabilities:
  - sunny -> hot = 0.4, sunny -> warm = 0.4, sunny -> cool = 0.2;
  - cloudy -> hot = 0.2, cloudy -> warm = 0.6, cloudy -> cool = 0.2;
  - rainy -> hot = 0.1, rainy -> warm = 0.3, rainy -> cool = 0.6.

In particular you need

a) Draw the HMM diagrams for the transition and emission probabilities.
b) Draw the Viterbi graph and write the pseudo code of the Viterbi algorithm (Write the recursion formula for Viterbi algorithm).
c) Write the recursion formula for the forward algorithm.
d) Write the pseudo code for Viterbi learning and Baum-Welch algorithm in the context of this problem.

Q4: Given an MSA, construct the HMM profile and estimate the initial parameters given the following DNA sequences:

Sequence 1: AGCTAGTA
Sequence 2: AG-TCGTT
Sequence 3: AT-GCGTA
Sequence 4: AGCTCGGA

Q5: De Bruijn graph

a) Report all the 3-mers of ACGTTTGGCGGG
b) Create a de Bruijn graph for the 3-mers from part a. Note that the 3-mers will be the label of the edges and 2-mers will be the label of the nodes.
c) Find the Eulerian path in the graph from b.