CSE 494/598: Algorithms in Computational Biology - Assignment 1

Fall 2022

Instructor: Heewook Lee **Due Date:** **09/26/2022 11:59pm**

Exact Pattern Matching

Exact Pattern Matching

1. Implement Z-algorithm for Exact Pattern Matching problem discussed in class for two input DNA strings p and t where $\Sigma = \{A, C, G, T\}$ and $p, t \in \Sigma^+$. You may assume $|p| \leq |t|$.

   Input: a plain textfile containing two lines, where text t is given in the first line and pattern p is given in the second line

   Output: your program will output positions of all occurrences of p in t and use 1-based index for positions (1st character position is 1 and the last character position is n if $|t| = n$). Each position must be printed as standard out in separate line starting from the lowest position to the highest position.

   Sample input file content:
   ```
   ACAGTATCAGTACAG
   CAG
   ```
   Sample output:
   ```
   2
   8
   13
   ```
   Note: You must implement Z-algorithm from scratch to receive credit [**40 pts for 494, 30pts for 598**].

2. **Z-algorithm** Biological sequences are often circular as in many bacterial genomes. Scientists simply cut the genome sequence at either an arbitrary point or at a origin of replication. Genome assemblers (a program that can stitch short sequences into a large genome sequence) also do this by breaking the circular sequence at a random location and report it as a linear sequence. If an identical circular sequence is broken at a different positions, they can often result in different linear sequences. Given two linear sequences X and Y, explain how we can use Z-algorithm to check if X and Y come from an identical circular sequence [**20 pts**].

3. Read the uploaded lecture notes on Exact Pattern Matching and study the KMP algorithm. Given a pattern P, we defined $lps[i]$ as the length of the longest nontrivial suffix of P[1..i] that matches a prefix of P. Here is a pattern $P =$ `ATCATCT` and its lps array:

i	1	2	3	4	5	6	7
P[i]	A	T	C	A	T	C	T
lps[i]	0	0	0	1	2	3	0

   Dr. Wiz claims that s/he has a better idea for a modified KMP pattern matching and newly defines a lps' array for a given pattern, where $lps'[i]$ is the length of the longest nontrivial suffix of P[1..i] that matches a prefix of P such that $P[lps[i] + 1] \neq P[i + 1]$. That is the character following the matched longest suffix/prefix is not same. If those two characters are equal, $lps'[i] = 0$. For example, for the above example of $P =$ `ATCATCT`, lps' values are `0 0 0 0 0 3 0` instead of `0 0 0 1 2 3 0`. Dr. Wiz claims that KMP search routing can directly use lps' values instead of lps values to make the search process a bit more efficient. Is this a valid claim? Test it out by running KMP algorithm by hand on $P =$ `ATCATCT` and $S =$ `TCATCATGATGATCATCT`, and explain whether this is a valid claim or not [**20 pts**].

4. **(Optional for 494 students)** It may have been obvious that there are lots of similarities between Z array and lps array. Pre-computed Z values can be used to compute lps instead of using the approach/method we discussed in lectures. Write out the algorithm for computing the modified lps' values using the Z values. The inputs to algorithms are a string S over the alphabet (ex: $\Sigma = \{$`A,C,G,T`$\}$) and Z array values for S. The output is the modified lps' array (Providing pseudocode is sufficient) [**10 pts for 598**].