BMI/CS 576: Introduction to Bioinformatics
Homework 2

Due: 10/19/2017 by 11:59pm

The goal of this assignment is to become more familiar with the algorithms for pairwise sequence alignment. To turn in your assignment, copy all relevant files to the directory:
/u/medinfo/handin/bmi576/hw2/USERNAME
where USERNAME is your account name for the BMI network. Please note the homework policies posted at `http://www.biostat.wisc.edu/bmi576/hw.html`

Problem 1 (50 points)

Consider two DNA sequences $\mathbf{x}$ and $\mathbf{y}$. Suppose there are $P \geq 0$ position pairs $\{(i_p, j_p)\}_{p=1,\ldots,P}$, such that x_{i_p} must match to y_{j_p} for $p = 1, \ldots, P$. These pairs are called *anchor pairs*. Given the scoring scheme $S(\cdot, \cdot)$ as

$$S(x, y) = \begin{cases} \rho_1 & x = y \\ -\rho_2 & x \neq y \end{cases}, \tag{1}$$

and affine gap penalty as

$$w(k) = \begin{cases} -g - sk, & k \geq 1 \\ 0 & k = 0 \end{cases}, \tag{2}$$

your task is to find the best *global alignments* that achieve the highest score.

You should write a program, `anchor_alignment`, that takes a single filename as input. Inside the input file, the first line contains five non-negative integers, representing K, ρ_1, ρ_2, g and s, respectively. It is then followed by two lines of DNA sequences, representing $\mathbf{x}$ and $\mathbf{y}$, respectively. When $K > 0$, there will be K extra lines starting from the fourth line and each line will contain two integer numbers, indicating the anchor pair (i_p, j_p). You can assume that all anchor pairs are valid, in the sense that (1) they all range from 1 to the length of $\mathbf{x}$ and $\mathbf{y}$, respectively ; (2) there are no overlaps between any of them. Following are two sample input files:

Listing 1: input1.txt

```
0 5 3 1 2
ABCD
ABCD
```

Listing 2: input2.txt

```
1 2 1 0 2
TACGAGTACGA
ACTGACGACTGAC
6 7
```

Given an input, your program should output two lines to represent an optimal alignment between $\mathbf{x}$ and $\mathbf{y}$. A gap should be indicated by a underscore "_". To make the optimal alignment unique, whenever there are many multiple choices when maximizing either $M(i,j)$, $I_x(i,j)$ or $I_y(i,j)$, always choose the matrix in the following order: $M > I_x > I_y$. [1]

Similar to previous homework, your program should run via a wrapper shell script `anchor_alignment.sh`. A template wrapper script is available from: `http://www.biostat.wisc.edu/bmi576/hw/hw2/anchor_alignment.sh`. For sanity checking (but not sufficient for complete testing), example test cases are posted at: `http://www.biostat.wisc.edu/bmi576/hw/hw2/test_cases/`. Here are two examples of running the program from the command line, for the two inputs shown above:

```
$sh anchor_alignment.sh input1.txt
ABCD
ABCD

$sh anchor_alignment.sh input2.txt
TAC_GA_GTAC_GA_
_ACTGACG_ACTGAC
```

Problem 2 (25 points)

Again consider the case of two DNA sequence reads, $\mathbf{x}$ and $\mathbf{y}$, both of length n, that potentially overlap and that may contain substitution sequencing errors (but not insertion or deletion errors).

We wish to use a probabilistic model to help in determining whether or not the two reads truly overlap. We define the model with three random variables, $\mathbf{X}$, $\mathbf{Y}$, and O. Random variables $\mathbf{X}$ and $\mathbf{Y}$ represent the two read sequences. The random variable O represents the true "offset" of the second read $\mathbf{Y}$, with respect to the first read $\mathbf{X}$, where $O = i$ means that the first position of $\mathbf{Y}$ corresponds to the i-th position of $\mathbf{X}$. We will only consider positive integer values for O. Thus, $O = 1$ means that the reads completely overlap and $O > n$ means that the reads do not overlap. We define the joint probability distribution of these random variables as:

$$P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, O = o) = P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}|o)P(O = o) \tag{3}$$

with

$$P(O = o) = (1 - \theta)^{o-1}\theta \tag{4}$$

and

$$P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}|O = o) = \left(\prod_{i=1}^{\min(o-1,n)} q_{x_i}\right)\left(\prod_{i=o}^{n} p_{x_i,y_{i-o+1}}\right)\left(\prod_{j=\max(n-o+2,1)}^{n} q_{y_j}\right) \tag{5}$$

The q_c parameters represent the general frequencies of each base, c, whereas the $p_{a,b}$ parameters represent the probability of observing character a and character b at two read positions that correspond to each other. The θ parameter governs the geometric distribution for the offset random variable,O.

Suppose that we observe two read sequences $\mathbf{x} =$ `TATC` and $\mathbf{y} =$ `CTTC` and that the parameters of the model are $\theta = \frac{1}{3}, q_c = \frac{1}{4}, \forall c$, and $p_{a,b} = m$ for $a \neq b$ and $p_{a,b} = s$ for $a = b$. For this problem, consider $m = \frac{1}{84}$ and $s = \frac{3}{14}$.

(a) Compute $P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, O = 4)$. Show your work.

[1] Notations are followed from the course slide.

(b) Compute $P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$. Show your work.

(c) Compute $P(O = 4|\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$. Show your work.

(d) Compute $P(O \leq 4|\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$, the probability that the two reads truly overlap given that we observe these two specific read sequences. Show your work.

(e) What conditions do m and s have to satisfy for Eq.(5) to be a valid probability distribution. Show your work.

Problem 3 (25 points)

(a) In protein alignments, we do not just look at match/mismatches. We look at the similarities between amino acids. How are these represented ?

(b) What is the advantage of a seeded method like BLAST compared to a Needleman-Wunsch alignment ?

Deliverables

1. Solutions for written problems should be *typed* or *scanned* and saved as a pdf or Word document named as `solution.pdf` or `solution.doc(x)`, respectively.

2. Solutions for programming problems should include *all* source codes (*.py, *.java, *.R, etc), as well as the wrapper shell script `anchor_alignment.sh`. In case that a static programming language is used, e.g. C/C++ or JAVA, a shell script `compile.sh` for compiling source codes must also be provided.

3. Solutions for both written and programming problems should be uploaded to your folder on biostat server.

4. If you submit your homework late and want to use your free days, you need to submit a file named `freeday.txt` and put the number of free days you use in the file.