BMI/CS 576 Fall 2015
Homework #2

Prof. Colin Dewey

Due Friday, October 16th, 2015 by 11:59pm

The goal of this assignment is to become more familiar with the algorithms for 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`

1. (20 points) Suppose we wish to partition a DNA sequence into segments, such that each segment is predominantly composed of a single base. Let p denote a partition of a sequence and $\ell(p)$ denote a labeling of a partition, i.e., an assignment of a label (one of A, C, T, or G) to each segment of that partition. We decide to determine a "good" partition by maximizing an objective function

$$f(p) = \max_{\ell(p)} \left(a \cdot n(p) + b \cdot m(p, \ell(p)) + c \cdot x(p, \ell(p)) \right),$$

   where $n(p)$ is the number of segments created by p, $m(p)$ is the number of positions that match the label of their segment, and $x(p)$ is the number of positions that do not match the label of their segment. In general, b will be a positive value and a and c will be negative values. For example, the score of the partition

   AATGTACA|CGCC|TTGTT|CCC|GAGGGT

   would be $5a + 18b + 8c$ (the segments in this partition would be labeled, from left to right, as A, C, T, C, G).

   (a) How many possible partitions are there for a sequence of length L? (Ignore the labels of each segment)

   (b) Give an optimal partition and its score for the sequence

   CGCCATTAT

   with $a = -2, b = 2$, and $c = -1$.

(c) Describe a *dynamic programming* algorithm for computing the optimal partition score of a DNA sequence of length L, given values for a, b, and c. You do not need to describe how to find a partition that gives the optimal score. (Hints: (i) break the problem into $4L$ subproblems, four subproblems for each prefix of the sequence (ii) it will be helpful to consider subproblems in terms of *labeled* partitions) For your algorithm, please be sure to describe the following aspects:

   i. the meaning of the subproblem,

   ii. the recurrence used to solve a subproblem,

   iii. how to compute the overall solution.

2. (60 points) Recall that for the greedy approach to sequence assembly, we need to compute the length of longest overlap between the end of one read, x, and the beginning of a second read, y, for all ordered pairs of reads. In the lecture notes, and in HW1, we assumed error-free reads.

Suppose now that we are given reads that may contain sequencing errors, which could be either substitutions, insertions, or deletions. Instead of computing the length of a perfectly matching overlap between the end of one read and the beginning of a second read we might instead compute the "best" alignment between the suffix of the first read and the prefix of the second read, and use the score of that alignment in our assembly algorithm (instead of overlap length). Suppose that we score an alignment, a, with the affine-gap penalty function

$$f(a) = m \cdot \mathit{matches}(a) + h \cdot \mathit{mismatches}(a) + g \cdot \mathit{gaps}(a) + s \cdot \mathit{spaces}(a),$$

where $\mathit{matches}(a)$, $\mathit{mismatches}(a)$, $\mathit{gaps}(a)$, and $\mathit{spaces}(a)$ are the number of matches, mismatches, gaps, and spaces in a, respectively.

With this objective function, the score of the best alignment between a *suffix* of sequence x with a *prefix* of sequence y, may be computed with the same recurrence as for global alignment with affine gap penalties (see slide 50 of the pairwise alignment lecture), but with the initialization for $I_x(i, 0)$ changed to $I_x(i, 0) = 0$, for all i. The score of the best overlap alignment can then be obtained by finding the largest value in the last rows of the M, I_x, and I_y matrices, i.e.,

$$\max_j \max(M(m, j), I_x(m, j), I_y(m, j))$$

The traceback step involves starting from the maximizing entry in the above equation and tracing back through the matrices until an entry in the first column in reached. For example, if the maximizing entry in the last row is $M(m, j)$ and the traceback from that entry reaches the first column at entry $M(i, 0)$, then an alignment of the last $m - i$ characters of x with the first j characters of y achieves the optimal score.

Write a program, `overlap_align`, that takes as input two DNA sequences, match, mismatch, gap, and space scores, and outputs the optimal overlap alignment between a suffix of the first sequence and a prefix of the second sequence. Your program should be run via a simple shell script `overlap_align.sh` that calls your program in the appropriate manner. A template script is available from:
`http://www.biostat.wisc.edu/bmi576/hw/hw2/overlap_align.sh`
The script will take five command-line arguments: the name of a file containing two DNA sequences (one sequence per line), match score, mismatch score, gap score, and space score. Your program should print to the standard output stream an optimal overlap alignment (one sequence per line, with hyphens used as space characters), followed by its score on a separate line. For example, below is a command we should be able to run with its expected output:

```
$ sh overlap_align.sh sequences.txt 1 -1 -2 -1
ACTCGATCCACG
   CGAT--ACGTT
3
```

where `sequences.txt` has content:

```
ACTCGATCCACG
CGATACGTT
```

Your program should print out only one optimal alignment in cases where there are multiple optima. When there are ties for the maximizing entry in the last rows of the three matrices, choose the entry (m, j) that corresponds to the longest prefix of the second sequence (i.e., the entry with the largest j). If multiple matrices give the same maximizing value for entry (m, j), prioritize the matrices by the order M, I_x, I_y. During the traceback, if there are ties, prioritize the matrix to which you traceback by the same order, M, I_x, I_y.

Sample input and output files, are provided at: `http://www.biostat.wisc.edu/bmi576/hw/hw2/`

3. (20 points) Again consider the case of two DNA sequence reads, x and 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, X, Y, and O. Random variables X and Y represent the two read sequences. The random variable O represents the true "offset" of the second read Y, with respect to the first read X, where $O = i$ means that the first position of Y corresponds to the ith position of 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:

$$P(x, y, o) = P(x, y|o)P(o)$$

with

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

and

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

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 $x = GCG$ and $y = CCG$ and that the parameters of the model are $\theta = \frac{1}{2}$, $q_c = \frac{1}{4}, \forall c$, and $p_{ab} = \frac{1}{48}$ for $a \neq b$ and $p_{ab} = \frac{3}{16}$ for $a = b$.

(a) Compute $P(x, y, O = 3)$. Show your work.

(b) Compute $P(x, y)$. Show your work.

(c) Compute $P(O = 3|x, y)$. Show your work.

(d) Compute $P(O \leq 3|x, y)$, the probability that the two reads truly overlap given that we observe these two specific read sequences. Show your work.