

Homework 1 Solution

January 19, 2010

1) A global alignment of two sequences  $A=a_1a_2\dots a_M$  and  $B=b_1b_2\dots b_N$  can be represented by the set of index pairs  $P = \{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\}$ ,  $1 \leq i_1 \leq i_2 \leq \dots \leq i_k \leq M, 1 \leq j_1 \leq j_2 \leq \dots \leq j_k \leq N$ , where the index pairs  $(i_x, j_x)$  indicate that  $a_{i_x}$  is aligned with  $b_{j_x}$ .

a) Prove that the optimal score of a global alignment with end-gap penalties can be calculated as  $S_{MN}$ , where  $S_{ij}$  is derived recursively at each step as

$$\max \begin{cases} S_{i-1, j-1} + \sigma(a_i, b_j) \\ S_{i-1, j-1-p} + \sigma(a_i, b_{j-p}) + w(p) & p = 1, 2, \dots, j-1 \\ S_{i-1-q, j-1} + \sigma(a_{i-q}, b_j) + w(q) & q = 1, 2, \dots, i-1 \end{cases}$$

**Solution:**

$S_{ij}$  represents the maximal score of alignments of the prefixes  $a_1a_2\dots a_i$  and  $b_1b_2\dots b_j$ , as the maximization is over all possible ways of extending an alignment of shorter prefixes.

a-i) Indicate to what values  $S_{00}$ ,  $S_{0j}$ , and  $S_{i0}$  should be set for the recursion to work.

**Solution:**

$$S_{00} = 0; S_{0j} = w(j); S_{i0} = w(i)$$

Trace back: from the cell  $MN$  to cell  $00$ .

a-ii) How would you change the algorithm to calculate the optimal score for a global alignment without end-gap penalties?

**Solution:**

$$S_{00} = 0; S_{0j} = 0; S_{i0} = 0$$

Trace back: Start from the cell with the maximum score on the last row( $M$ ) or column( $N$ ), stop when column 0 or row 0 is reached.

a-iii) Give an algorithm to derive the number of all possible alignments for sequences of lengths  $M$  and  $N$ .

**Solution:**

When we fill the M x N matrix in the Needleman-Wunsch algorithm, for cell ij we accounted for 1 + (j-1) + (i-1) possible ways of one-step extentions of shorter alignments.

To derive the number of all possible alignments, we need fill another M x N matrix. Let  $N_{ij}$  represent the number of all possible alignments between sequence  $a_1 \dots a_i$  and  $b_1 \dots b_j$ . Then  $N_{ij} = N_{i-1,j-1} + \sum_{k=0}^{j-2} N_{i-1,k} + \sum_{k=0}^{i-2} N_{k,j-1}$ , where  $N_{i,0}$  and  $N_{0,j}$  are all equal to 1 ( $M \geq i \geq 0, N \geq j \geq 0$ ).  $N_{MN}$  is the total number of all possible alignments. (Please check the following partially filled table by enumerating the alignments for small M and N.)

		j								
		0	1	2	3	4	5	6	7	8
i	0	1	1	1	1	1				
	1	1	1	2	3					
	2	1	2	3	5					
	3	1	3	5	9					
	4	1								
	5									
	6									
	7									
	8									

b) Derive the algorithm to calculate the optimal score as in (a) but without the restriction of avoidance of double gaps.

**Solution:**

$$S_{00} = 0; S_{0j} = 0; S_{i0} = 0;$$

$$S_{ij} = \max \begin{cases} S_{i-1,j-1} + \sigma(a_i, b_j) \\ S_{i,j-p} + w(p) & p = 1, 2, \dots, j \\ S_{i-q,j} + w(q) & q = 1, 2, \dots, i \end{cases}$$

Trace back as before.

b-i) Determine the complexity of the algorithm: how many operations are required to calculate the optimal score?

**Solution:**

The number of additions is seen to be  $\sum_{i=1}^M \sum_{j=1}^N [1 + j + i] = MN + M \frac{N(N+1)}{2} + N \frac{M(M+1)}{2}$ . Thus, for M=N, the algorithm is of  $O(N^3)$ .

2) In class we showed that for an alignment of length N with per site match probability p, the length k of the longest common word is approximately  $\frac{(\ln N - \ln(\ln \frac{1}{\alpha}))}{\ln(\frac{1}{p})}$ , where  $\alpha$  is the probability of no occurrence of a k-word.

a) Re-derive the approximation.

**Solution:**

The match probability of each site:  $p$

The length of alignment:  $N$

The length of the longest common word:  $k$

$N$  trials with success probability:  $\mu = p^k$

Here  $N \rightarrow +\infty, \mu \rightarrow 0$

Poisson distribution:  $\text{Prob}\{x; N\mu\} = \frac{(N\mu)^x e^{-N\mu}}{x!}$

Want  $\text{Prob}\{x = 0\} = e^{-N\mu} = \alpha$

$$e^{-Np^k} = \alpha$$

$$-Np^k = \ln \alpha$$

$$\ln N + k \ln p = \ln \left( \ln \frac{1}{\alpha} \right)$$

$$k = \frac{\ln N - \ln \left( \ln \frac{1}{\alpha} \right)}{\ln \frac{1}{p}}$$

b) Set  $p=0.25$  and plot  $k$  as a function of  $\alpha$  for various choices of  $N$ . Interpret your graphs.

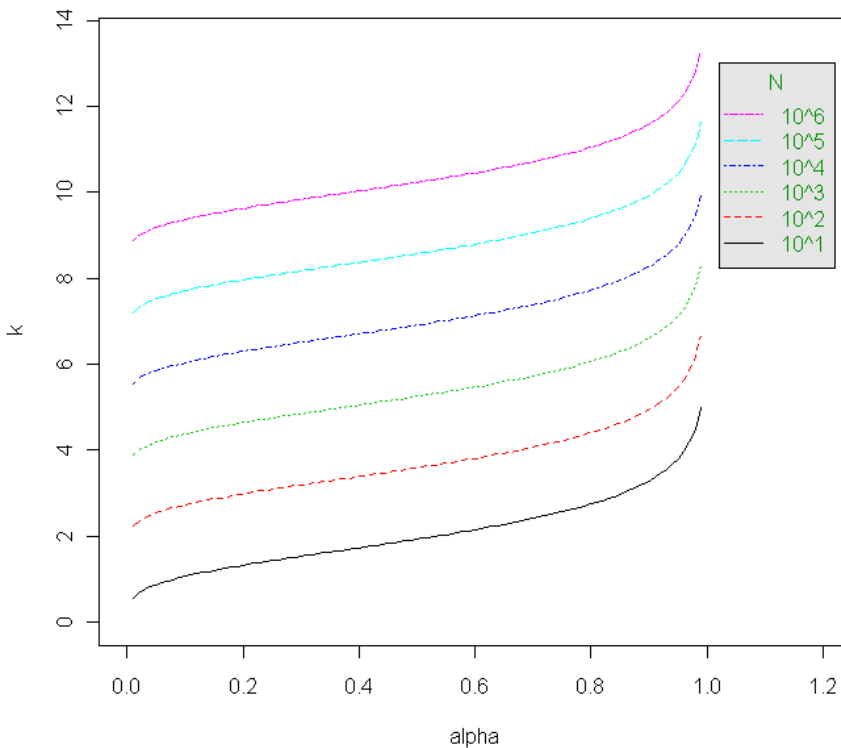


Figure 1: Expect common word length  $k$  as a function of  $\alpha$  for various  $N$ .

**Interpretation:**

1) For a constant  $N$ , increasing  $\alpha$  will get a larger  $k$ . In other words, in a sequence with  $N$  bps, it's harder to find a long conserved word than a short one.

2) For a constant  $\alpha$ , increasing  $N$  will get a larger  $k$ , with logarithmic dependence.  $k$  will increase by one for a sequence  $\frac{1}{p}$  times as long as the  $k$  calculated for  $N$ .