

Assignment 4

Due Wednesday, September 16 before midnight (California time)

1. Write a Python function that takes five inputs: [8pt]

- a. A DNA sequence string X,
- b. A DNA sequence string Y,
- c. A score S_{match} occurs for matching the characters of X and Y,
- d. A penalty $P_{mismatch}$ occurs for mismatching the characters of X and Y, and
- e. A penalty P_{gap} occurs if a gap is inserted into the string.

This function returns a Dynamic Programming scoring matrix for the global pairwise alignment.

For example, with the inputs ('ACGA', 'CACT', 3, -1, -2)

First, the function initializes a scoring matrix. The first row and column of the matrix are filled with gap scores.

		C	A	C	T
	0	-2	-4	-6	-8
A	-2				
C	-4				
G	-6				
A	-8				

Second, the function fills the matrix from the upper-left-hand corner of the matrix. To find the maximum score of each cell, it is required to know the neighboring scores (diagonal, left and up) of the current position. Compare three values: the value adding the match or mismatch score to the diagonal neighboring score, the value adding the gap score to the left neighboring score, and the value adding the gap score to the up neighboring score. Take the maximum among these three values and fill it into the i_{th} and j_{th} position. For instance, -1 (marked in red below) is the maximum among three values: $0-1=-1$ (the diagonal neighboring score plus a mismatch score), $-2-2=-4$ (the up neighboring score plus a gap penalty), $-2-2=-4$ (the left neighboring score plus a gap penalty). With this rule, fill up the scoring matrix completely.

		C	A	C	T
	0	-2	-4	-6	-8
A	-2	-1	1	-1	-3
C	-4	1	-1	4	2
G	-6	-1	0	2	3
A	-8	-3	2	0	1

2. [Bonus] Modify Function #1 to be able to store the back pointer to the cell from where the maximum score is obtained in each step. One way is that the function creates an extra matrix with the same size as the scoring matrix to store these pointers. This function traces back step

from the bottom-right-hand corner of the matrix. Finally, this function returns the optimal alignment (return one of the optimal alignments if there are many) [Bonus 2pt].