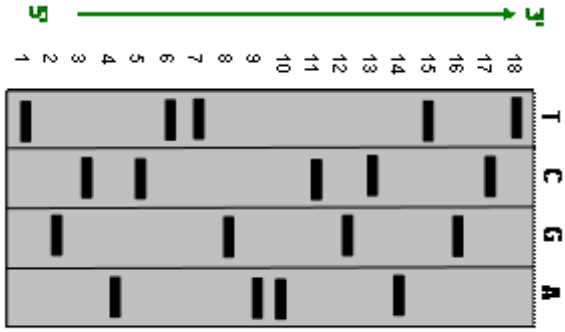
Exercise set 1 Solution

1. **What is the DNA sequence corresponding to the Sanger plot below?**



The sequence is read from 5' to 3'.
TGCACTTGAACGCATGCT

2. **Given two sequences, which value is larger: their local similarity or their global similarity? Why? How does their semi-global similarity compare with the other two values?**

Solution:

By definition:
Any global alignment is also a semi global alignment, but there could be better semi-global alignments (each semi-global alignment can be seen as a global alignment between a prefix of one string and a suffix of the other string). Thus, the semi-global best score could be larger than the global score. Also, any semi-global alignment is at the same time a local alignment, but there could be better local alignments (any local alignment could be seen as a semi-global alignment between a prefix of one string and a suffix of the other string; note that it can also be seen as a global alignment between two factors of the two strings). Thus, the local alignment score could be larger than the semi-global one.

The three scores are in general in the following relationship:

Global score ≤ semi-global score ≤ local score

4. **Find all best global alignments between sequences AAAC and AGC, where the scoring scheme is +1 for match, -1 for mismatch and -2 for an alignment with a gap.**

Solution:
Two sequences are given:
s : AAAC
t : AGC

For finding best alignments between *s* and *t*, first create a score matrix D filled with maximum alignments score and right most cell in the last raw gives the best alignment score.

This is calculated using following formula.

D(i,j) = Max { $D(i,j-1) + g$, $D(i-1,j) + g$, $D(i-1,j-1) + f(s[i],t[j])$ }

Here $f(s[i],t[j])$ gives the mismatch/match score for characters s[i] and t[j].

$f(s[i],t[j])$ = 1 , if s[i] = t[j]

= -1 otherwise.

D =

		t			
			A	G	C
		0	1	2	3
s	0	0	← -2	← -4	← -6
A	1	↑-2	↖ 1		
A	2	↑-4			
A	3	↑-6			
C	4	↑-8			

D(1,1) = Max { $D(1,0) + (-2)$, $D(0,1) + (-2)$, $D(0,0) + f(s[1],t[1])$ }

= Max { $-2-2$, $-2-2$, $0+1$ }

= 1 (It is calculated from the cell D(0,0). Cell D(1,1) would have diagonal arrow pointing D(0,0).)

Arrows in the table indicate from which cell the maximum score is calculated. We continue filling entries and tracing arrows.

D =

		t			
			A	G	C
		0	1	2	3
s	0	0	← -2	← -4	← -6
A	1	↑-2	↖ 1	←-1	← -3
A	2	↑-4	↖ ↑ -1	↖ 0	↖ ← -2
A	3	↑-6	↖ ↑ -3	↖ ↑ -2	↖ -1
C	4	↑-8	↑ -5	↖ ↑ -4	↖ -1

Best alignment score is -1.

Optimal alignments could be found by walking on the traced arrow path from D(m,n) to D(0,0).

3 possible moves:

- Diag: the letters from two sequences are aligned
- Left: gap is introduced into the left sequence
- Up: a gap is introduced into the top sequence

Resulting alignments are as follows:

Alignments	AAAC _ AGC	A A A C A G _ C	A A A C A _ G C
Score	(-2)+1+(-1)+1 = -1	1+(-1)+(-2)+1 = -1	1+ (-2)+(-1)+1 = -1
Solution	Best Alignment	Best Alignment	Best Alignment

Alignments listed above are the best alignments.

5. Find all best global alignments between sequences ATAG and TTCG, where the scoring scheme is +1 for match, -1 for mismatch and -1 for an alignment with a gap.

Solution:

D =

			t			
			T	T	C	G
		0	1	2	3	4
	0	0	← -1	← -2	← -3	← -4
A	1	↑-1	↖ -1	↖← -2	↖← -3	↖← -4
T	2	↑-2	↖ 0	↖ 0	← -1	← -2
A	3	↑-3	↑ -1	↖↑-1	↖ -1	↖ -2

s

G	4	↑-4	↑ -2	↖↑-2	↖↑ -2	↖ 0

Start tracing a path from D(4,4) to D(0,0).

Alignments	A T A G T T C G
Score	(-1)+1+(-1)+1= 0

6. Find all best local alignments between sequences ATACTGGG and TGACTGAG, using the same scoring scheme as in exercise 2.

Solution:

Two sequences are given

s : ATACTGGG

t: TGACTGAG

Method : Smith Waterman method

Entries in the table are calculated using following formula.

$$L(i,j) = Max\{0, L(i-1,j-1) + f(s[i],t[j]), L(i-1,j) + g, L(i,j-1) + g\}$$

L =

			t							
			T	G	A	C	T	G	A	G
		0	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0	0
A	1	0	0	0	↖1	0	0	0	↖1	0
T	2	0	↖1	0	0	↖1	↖1	0	0	↖0
A	3	0	0	↖0	↖1	0	↖0	↖0	↖1	0
C	4	0	0	0	0	↖2	←0	0	0	↖0
T	5	0	↖1	0	0	0	↖3	←1	0	0
G	6	0	0	↖2	←0	0	↑1	↖4	←2	↖1
G	7	0	0	↑0	0	0	0	↖2	↖3	↖3
G	8	0	0	↖1	0	0	0	↖1	↖ 1↑	↖4

s

From the above matrix we find the highest score and trace the path until we come to a cell with score zero. This cell is not included in the alignment.

Alignments	ACTGGG ACTGAG	ACTG ACTG
Score	1+1+1+1+(-1)+1 = 4	1+1+1+1 = 4

Above alignments are the best local alignments.

7. What scoring schemes should you use to determine the longest common substring and the longest common subsequence for two given strings, using the algorithm for best global alignment?

Solution:

For the longest common substring we cannot allow any mismatches and any alignments with gaps. We also re-use the idea from the local alignment: any common substring (even the empty one is allowed), ending at position i in string u, and at position j in string v, is a suffix of u[1…i] and a suffix of v[1…j]. The (i,j) entry of the matrix will give the length of the longest common suffix ending at position i in u and at position j in v. So the suggestion is to use the local alignment algorithm with the following scoring scheme:

- Mismatch: minus infinity
- Alignment with gap: minus infinity
- Match: 1

As in the local alignment algorithm, the option of taking the empty substring/suffix is allowed, hence the matrix will only have non-negative entries. The initialization of the first row/column is as in the local alignment algorithm, i.e., with 0. Simply choose the maximum value in the matrix and walk on the arrows (in this case all are diagonals) until just before the first 0 is reached.

For the longest common subsequence we want to count only the matches. Alignment with the gaps is alright (because in subsequences the letters need not be consecutive), but mismatches are not. The search is over the entirety of the two strings. The (i,j) entry of the matrix will give the length of the longest common subsequence of u[1…i] and of v[1…j]. Hence the idea that this is a version of the global alignment algorithm with the following scoring scheme:

- Mismatch: minus infinity
- Alignment with gap: 0
- Match: 1

Note that the matrix will consist of nonnegative values only because a mismatch can always be avoided through two alignments with gaps. The initialization of the first row/column is as in the global alignment algorithm with multiples of the gap, i.e., with 0. As in the global alignment algorithm, walk on the arrows from the bottom right corner to the top left corner and write down the alignment. All the aligned letters from the two strings (carrying a match, never a mismatch) will give the longest common subsequence.

8. Apply the scoring scheme you indicated in exercise 7 to find all longest common substrings for strings ATACTGGG and TGACTGGT.

Solution:

The longest common substring is **ACTGG**, as seen from the calculations below.

A T A C T G G G
T G A C T G G T

		A	T	A	C	T	G	G	G
	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	1	0	0	0
G	0	0	0	0	0	0	2	1	0
A	0	1	0	1	0	0	0	0	0
C	0	0	0	0	2	0	0	0	0
T	0	0	1	0	0	3	0	0	0
G	0	0	0	0	0	0	4	1	2
G	0	0	0	0	0	0	1	5	1
T	0	0	1	0	0	1	0	0	0