LAB#2 – Sequence alignment

The goal of this Lab is to learn about sequence alignment algorithms (Needleman–Wunsch, Smith-Waterman, progressive multiple sequence alignment) and improve your programming skills.

Group I (5 points)

Consider the following two **nucleotide sequences**:

S1: G G A T C C **S2: G G C C G**

and the following *scoring model* (use a gap penalty function linear in the number of gaps):

match **= +3**
mismatch **= -2**
gap **= -4**

Use the **Needleman-Wunsch** algorithm and the scoring model above to find **all** optimal **global alignments** for **S1** and **S2**. Present the dynamic programming matrix used by the algorithm, all the optimal global alignments obtained, and their score.

Group II (4 points)

Consider the following two **amino acid** sequences

S1: W P I W P C **S2: I I W P I**

and the following *scoring model* (use a gap penalty function linear in the number of gaps)

scoring matrix **= BLOSUM 50 (see below)**
gap **= -4**

Use the **Smith-Waterman** algorithm and the scoring model above to find all optimal **local alignments** for the sequences **S1** and **S2**. Present the dynamic programming matrix used by the algorithm, all the optimal local alignments obtained and the respective score.

Group III (7 points)

Use the **programming language of your choice** to program the **Smith-Waterman** algorithm to compute the local alignment(s) between two **amino acid sequences** using the BLOSUM 50 **scoring matrix.** The gap penalty function should be linear in the number of gaps.

The **input** and **output** of the program should be as follows:

INPUT

1) Two amino acid sequences: **S1** and **S2 (two strings).**

2) The **gap penalty** (integer)

OUTPUT

1) Score of the optimal local alignment(s) between **S1** and **S2.**

2) Print *all* possible optimal alignments between **S1** and **S2.**

Present the **source code** and the **output** produced by your program when the input is the problem described in question **II**.

Present performance results that show that the algorithm running time increases with the quadratic of the size of the input sequences.

Group IV (4 points)

Consider the following set of **nucleotide sequences**

S1: A A C G T C
S2: A G C G C C
S3: C C C G T
S4: A C A T

and the following *scoring model* (use a gap penalty function linear in the number of gaps):

***match* = +2**
***mismatch* = -1**
***gap* = -3**

Use **Multiple Sequence Alignment** and the scoring model above to obtain the the **best multiple alignment**.

Use the Sum of Pairs (SP) scores model to compute the final score of the obtained multiple alignment.

- **BLOSUM50**

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-4	15	2	-3
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5