CAROLINA CARAMELO 89790 - DIOGO ANTUNES 86797 – DULCE CANHA 86979 – MARIANA MOURÃO 98473
2020/2021

# LAB#2 – SEQUENCE ALIGNMENT

## Group I

To determine the optimal global alignment of the two sequences under analysis - S1: GGATCC and S2: GGCCG - the Needleman-Wunsch algorithm was used, with the following scoring model:

*match = +3*

*mismatch = -2*

*gap = -4*

The Needleman-Wunsch algorithm is a dynamic programming application used in Bioinformatics that consists in the construction of an array of indexes i and j, F(i,j), which stores the scores of the best alignment between a prefix of $S1_{(1,\dots,i)}$ and of $S2_{(1,\dots,j)}$, computed recursively by considering the following conditions:

$$F(0,0) = 0$$

$$\mathrm{F}(\mathrm{i},0) = -id$$

$$\mathrm{F}(0,\mathrm{j}) = -jd$$

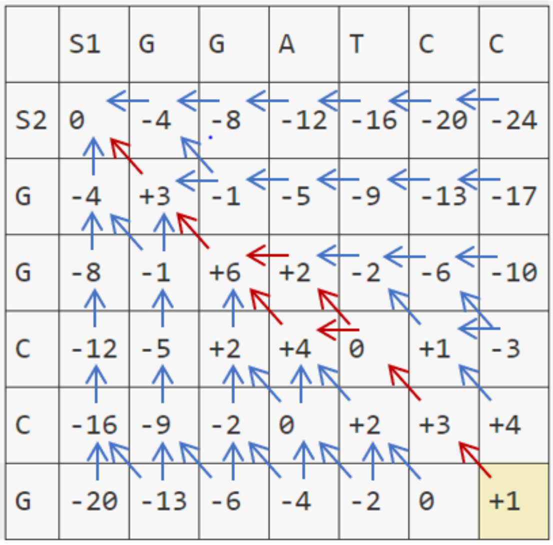
$$F(i,j) = max\left\{\begin{matrix} F(i-1,j-1) + s(x_i,y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{matrix}\right\}$$

where $s(x_i,y_j)$ corresponds to the score that is assigned according to the match or mismatch between the nucleotides and d takes the value of 4 (module of the gap penalty).

Figure 1 shows the matrix obtained according to the mentioned conditions. The entry marked in red corresponds to the best score for the alignment and the arrows in red mark the paths with the optimal global alignments.

By performing traceback from the bottom-right cell to the upper-left, two optimal global alignments were obtained (bifurcation on the traceback path), both with a score of +1:

| | | |
|---|---|---|
| S1: | G G A T C C | G G A T C C |
| S2: | G G – C C G | G G C – C G |



| | S1 | G | G | A | T | C | C |
|---|---|---|---|---|---|---|---|
| S2 | 0 | -4 | -8 | -12 | -16 | -20 | -24 |
| G | -4 | +3 | -1 | -5 | -9 | -13 | -17 |
| G | -8 | -1 | +6 | +2 | -2 | -6 | -10 |
| C | -12 | -5 | +2 | +4 | 0 | +1 | -3 |
| C | -16 | -9 | -2 | 0 | +2 | +3 | +4 |
| G | -20 | -13 | -6 | -4 | -2 | 0 | +1 |

*Figure 1 - Dynamic Programming Matrix obtained by applying the Needleman-Wunsch algorithm. The entry marked in yellow corresponds to the best score for the alignment, the red arrows mark the traceback paths which obtain the optimal global alignments and the blue arrows indicate the origin of the entry value according to the mentioned conditions.*

To validate the score obtained from the matrix, the same was calculated by summing the score terms for each aligned pair of residues, plus the score terms for each gap:

Global score 1 = 3+3−4−2+3−2=+1

Global score 2 = 3+3−2−4+3−2=+1

The two optimal local alignments obtained correspond to the maximization of the number of identical or conserved residue pairs and minimization of the number of unconserved residue pairs or gaps (Durbin et al. 1998).

## Group II

For this exercise, given two amino acid sequences – S1: WPIWPC and S2: IIWPI –, it's intended to determine all the optimal local alignments between them, consisting of identifying the local regions with the highest level of similarity. For that, it's considered the Smith-Waterman algorithm, as well as a scoring system with the following specifications:

*scoring matrix* = BLOSUM 50

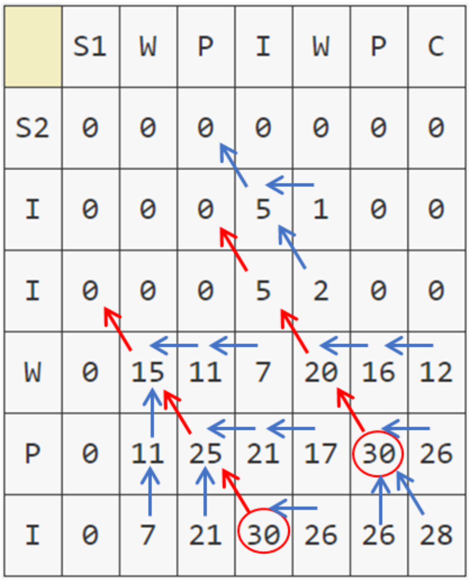
*gap* = -4

The Smith-Waterman algorithm also follows the dynamic programming paradigm, modifying the recurrence formula for obtaining the dynamic programming matrix F as follows:

$$F(i,j) \; = \; max \begin{cases} 0, \\ F(i-1,j-1)+s(x_i,y_j), \\ F(i-1,j) \; -d, \\ F(i,j-1)-d. \end{cases}$$

, where *i* a *j* correspond respectively to one sequence indexing, being *s(*$x_i$*, *$y_i$*)* the score for the aligned pair of amino acids stored in the scoring matrix. By imposing the minimum score to 0, the boundary conditions (F(i,0) and F(0,j)) are set to 0, initializing as well F(0,0) = 0.

Figure 2 shows the matrix obtained according to the mentioned conditions. The entries marked in red corresponds to the best scores for the alignment and the arrows in red mark the paths with the optimal local alignments.



| | S1 | W | P | I | W | P | C |
|---|---|---|---|---|---|---|---|
| S2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 5 | 1 | 0 | 0 |
| I | 0 | 0 | 0 | 5 | 2 | 0 | 0 |
| W | 0 | 15 | 11 | 7 | 20 | 16 | 12 |
| P | 0 | 11 | 25 | 21 | 17 | 30 | 26 |
| I | 0 | 7 | 21 | 30 | 26 | 26 | 28 |

*Figure 2 - Dynamic programming matrix, computed by the Smith-Waterman algorithm. The cells marked with a red circle identify the maximum score for the local alignments between S1 and S2, the red arrows correspond to the traceback paths which obtain the optimal local alignments, and the blue arrows encode from where the scores came from.*

Since the considered algorithm measures the log-odds ratio, i.e., the logarithm of the relative likelihood that the sequences are related (match model), as opposed to being unrelated (random model) (Durbin et al. 1998), the optimal local alignments are obtained by seeking the maximum score on the matrix (+30) and building it up in reverse, by performing traceback from the cells with the highest score until reaching 0, which demarks the start of an alignment.

By doing the previously described, two optimal local alignments were obtained, both with a score of +30:

| | | |
|---|---|---|
| S1: | I W P | W P I |
| S2: | I W P | W P I |

To validate the score obtained from the matrix, the same can be derived by summing the score terms for each aligned pair of residues, plus the score terms for each gap (there aren't):

$$S = s(I, I) + s(W,W) + s(P,P) = 5 + 15 + 10 = 30$$

As seen, the optimal local alignments obtained are composed of pairs of identical amino acid residues, being more likely to occur in a real alignment than by chance, thus contributing with positive scoring terms. The subsequences obtained may indicate some shared domain between the amino acid sequences being studied, which, if being the case, has been under enough selection to preserve detectable similarity, with the rest of both sequences having accumulated noise through mutations that are no longer alignable (Durbin et al. 1998).

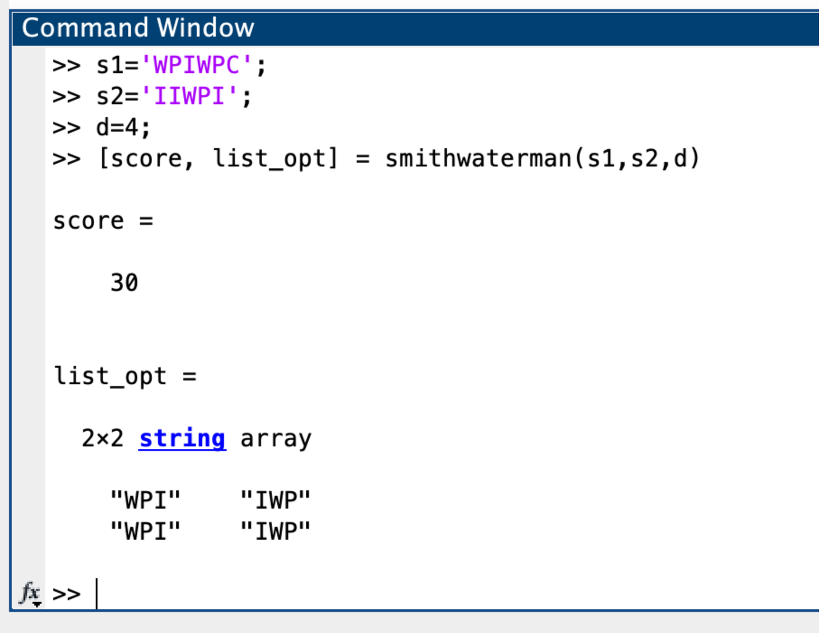
## Group III

For the implementation of the Smith-Waterman algorithm, the *Matlab* computing environment was utilized. The goal was to develop a function that, given two amino acid sequences (*s1* and *s2*) and a linear gap penalty (*d*), would compute all the optimal local alignments between these sequences and also return their score, using the BLOSUM50 scoring matrix.

Two functions were implemented. The main one, *smithwaterman*, receives *s1*, *s2* and *d* as inputs, and returns two outputs: *score* (the maximum score found in the computed scoring matrix, which corresponds to the score of the optimal alignment(s)) and *list_opt* (a 2xn matrix containing n optimal local alignment(s), each corresponding to a column).

The second function, *path_find*, is an auxiliary to the main and its role is to traceback all the local alignments starting from the cells containing the maximum score, accounting for bifurcations.

In Figure 3 is an example of the execution of the program given the same inputs as the problem stated in group II. The output was consistent with the previously obtained results.



Command Window

```
>> s1='WPIWPC';
>> s2='IIWPI';
>> d=4;
>> [score, list_opt] = smithwaterman(s1,s2,d)

score =

    30


list_opt =

  2×2 string array

    "WPI"    "IWP"
    "WPI"    "IWP"

>> 
```

*Figure 3 - Output produced by the algorithm for same input as Group II*

To verify the algorithm's time complexity, another function was designed: *sm_performance*. It generates two random sequences of amino acids, starting with a length of 5 and going up to 100. For each length, the same for both sequences, it registers the time that the algorithm takes to run. As Figure 4 illustrates, the empirical running times are approximately fitted by a growth tendency with the quadratic of the size of the input sequences. The oscillations observed correspond to the noise derived from the randomness of the generated sequences, which may have associated computational complexities lower than O($N^2$), which corresponds to the upper limit.

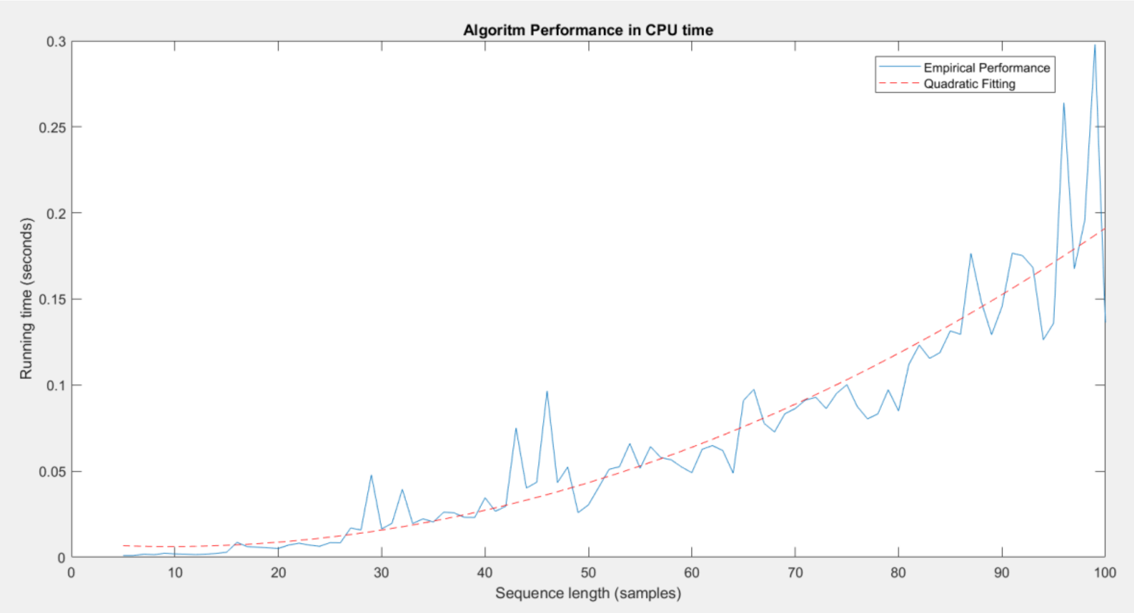


*Figure 4 - Graphical representation of SW algorithm's performance in terms of CPU time. The horizontal scale represents the size of the input sequences (in samples) and the vertical scale represents the running time (in seconds).*

## Group IV

In this group, given the four nucleotide sequences under analysis – S1: AACGTC, S2: AGCGCC, S3: CCCGT and S4: ACAT –, it's intended to obtain the best multiple sequence alignment (MSA), taking into consideration the following scoring model:

*match = +2*

*mismatch = -1*

*gap = -3*

In order to reduce the volume of the multidimensional dynamic programming matrix to be computed (contained in a hyperspace – 4D), it was adopted the progressive alignment algorithm (heuristic algorithm), restricting the analysis to 2D (as done for the previous exercises).

Firstly, the optimal global pairwise alignments for all possible pair of sequences (given by ${}^{k}C_2$, with k being the number of sequences, hence ${}^{4}C_2 = 6$) were constructed, through the application of the Needleman-Wunsch algorithm. The respective matrices are presented in the annex.

**S1 and S2 optimal global alignment**

| | |
|---|---|
| S1: | A A C G T C |
| S2: | A G C G C C |

**S1 and S3 optimal global alignment**

| | |
|---|---|
| S1: | A A C G T C |
| S3: | C C C G T – |

**S1 and S4 optimal global alignment**

For this pair of sequences, two optimal global alignments were obtained.

| | | |
|---|---|---|
| S1: | A A C G T C | A A C G T C |
| S4: | – A C A T – | A – C A T – |

**S2 and S3 optimal global alignment**

For this pair of sequences, two optimal global alignments were obtained.

| | | |
|---|---|---|
| S2: | A G C G C C | A G C G C C |
| S3: | C C C G – T | C C C G T – |

**S2 and S4 optimal global alignment**

For this pair of sequences, three optimal global alignments were obtained.

| | | | |
|---|---|---|---|
| S2: | A G C G C C | A G C G C C | A G C G C C |
| S4: | A – C – A T | A – C A – T | A – C A T – |

**S3 and S4 optimal global alignment**

For this pair of sequences, four optimal global alignments were obtained.

| | | | | |
|---|---|---|---|---|
| S3: | C C C G T | C C C G T | C C C G T | C C C G T |
| S4: | A C – A T | – A C A T | A – C A T | A C A - T |

The respective scores of the obtained alignments are displayed in table 1.

*Table 1 - Table with the global optimal alignment* scores *between each pair of nucleotide sequences, obtained through the Needleman-Wunsch algorithm.*

| Pair of sequences | Score |
|---|---|
| S1-S2 | +6 |
| S1-S3 | +1 |
| S1-S4 | -1 |

| | |
|---|---|
| S3-S4 | -1 |
| S2-S3 | -2 |
| S2-S4 | -4 |

From the scores obtained, a guide tree was created (Figure 5), where the most similar pairs of sequences (higher alignment scores) are merged into a node first, with each node representing an alignment, being the final root node the complete multiple sequence alignment (Durbin et al. 1998).

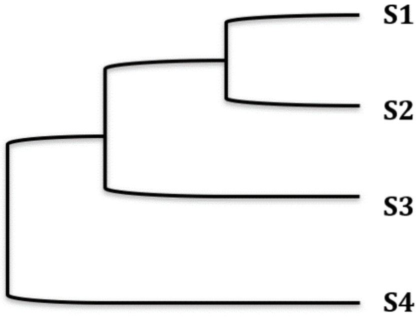


*Figure 5 - Guide tree created based on the global alignment scores between each pair of sequences.*

Following the estimated guide tree from outward to inward, it's obtained the order in which to progressively align the sequences, being iteratively performed by the Needleman-Wunsch algorithm. The alignment of the most similar sequences (S1 and S2) was already performed, remaining the alignments between S1 and S2 with S3 (Figure 6), and subsequently this later alignment with S4 (Figure 7).

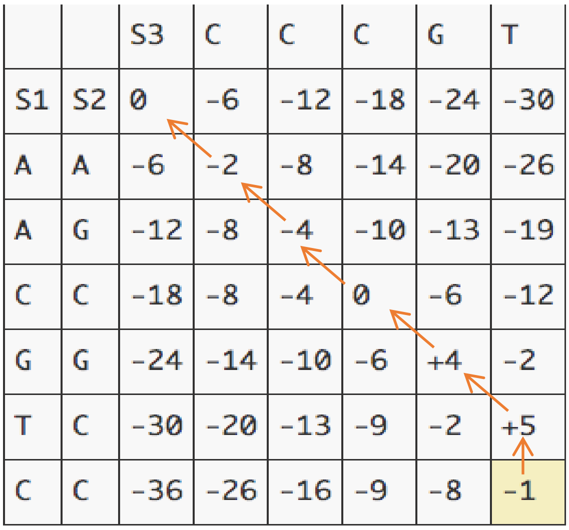
Thus, the optimal multiple alignment obtained for the four nucleotide sequences under study is the following:

S1: A A C G T C

S2: A G C G C C

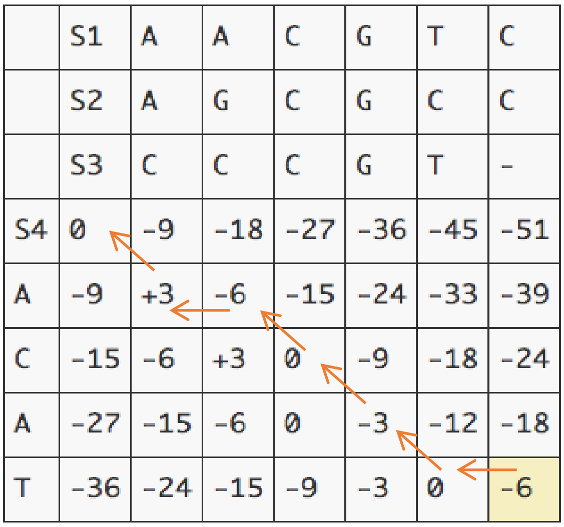
S3: C C C G T –

S4: A – C A T –



|  |  | S3 | C | C | C | G | T |
|---|---|---|---|---|---|---|---|
| S1 | S2 | 0 | -6 | -12 | -18 | -24 | -30 |
| A | A | -6 | -2 | -8 | -14 | -20 | -26 |
| A | G | -12 | -8 | -4 | -10 | -13 | -19 |
| C | C | -18 | -8 | -4 | 0 | -6 | -12 |
| G | G | -24 | -14 | -10 | -6 | +4 | -2 |
| T | C | -30 | -20 | -13 | -9 | -2 | +5 |
| C | C | -36 | -26 | -16 | -9 | -8 | -1 |

*Figure 6 - Resulting dynamic programming matrix of the global alignment between the previous alignment of S1 and S2 with the S3 sequence.*



|  | S1 | A | A | C | G | T | C |
|---|---|---|---|---|---|---|---|
|  | S2 | A | G | C | G | C | C |
|  | S3 | C | C | C | G | T | - |
| S4 | 0 | -9 | -18 | -27 | -36 | -45 | -51 |
| A | -9 | +3 | -6 | -15 | -24 | -33 | -39 |
| C | -15 | -6 | +3 | 0 | -9 | -18 | -24 |
| A | -27 | -15 | -6 | 0 | -3 | -12 | -18 |
| T | -36 | -24 | -15 | -9 | -3 | 0 | -6 |

*Figure 7 - Resulting dynamic programming matrix of the global alignment between the previous alignment of S1, S2 and S3 with the S4 sequence.*

Finally, the SP score of the obtained multiple alignment is computed as the sum over all column scores, which in turn are obtained by summing the scores for each combination of nucleotides within the considered column.

**First position:** $3 \times 2 + 3 \times (-1) = 3$

**Second position:** $3 \times (-1) + 3 \times (-3) = -12$

**Third position:** $6 \times 2 = 12$

**Fourth position:** $3 \times 2 + 3 \times (-1) = 3$

**Fifth position:** $3 \times 2 + 3 \times (-1) = 3$

**Sixth position** $1 \times 2 + 1 \times 0 + 4 \times (-3) = -10$

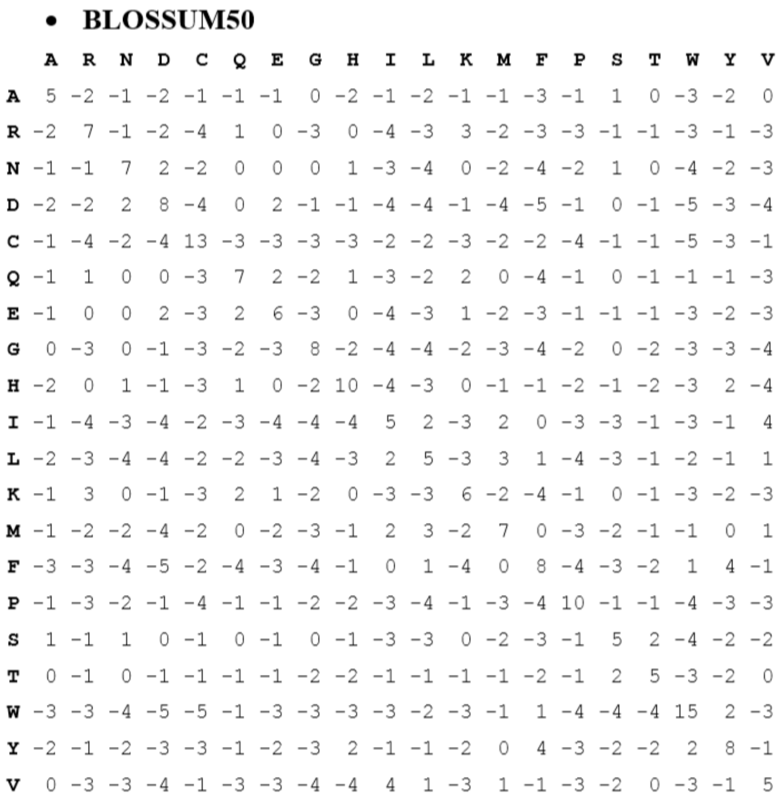
**Final score:** $3 - 12 + 12 + 3 + 3 - 10 = \boxed{-1}$

The obtained score equals the sum of the scores of all pairwise alignments defined by the multiple alignment, which is traduced by +6 – 1 – 6 = -1.

## Bibliografia

Durbin, Richard, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. 1998. “Biological SequenceAnalysis.” *Biological Sequence Analysis*.

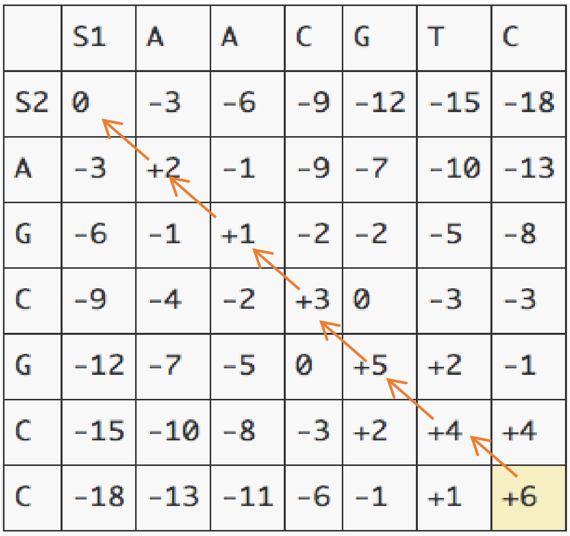
# Annex



- **BLOSSUM50**

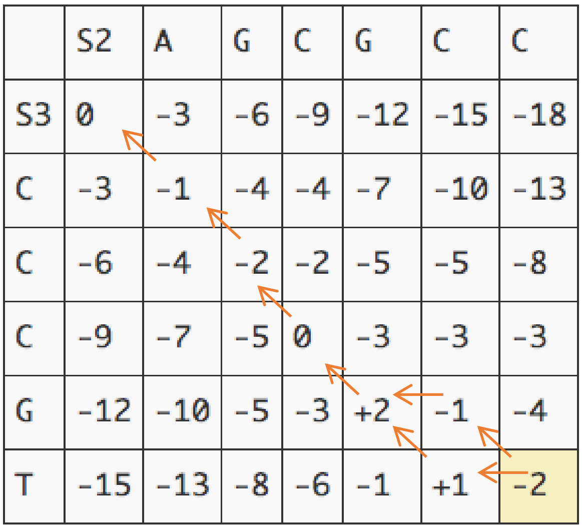
| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | 5 | -2 | -1 | -2 | -1 | -1 | -1 | 0 | -2 | -1 | -2 | -1 | -1 | -3 | -1 | 1 | 0 | -3 | -2 | 0 |
| **R** | -2 | 7 | -1 | -2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

*Figure 8 - BLOSUM 50 matrix, from which the alignment scores were taken*



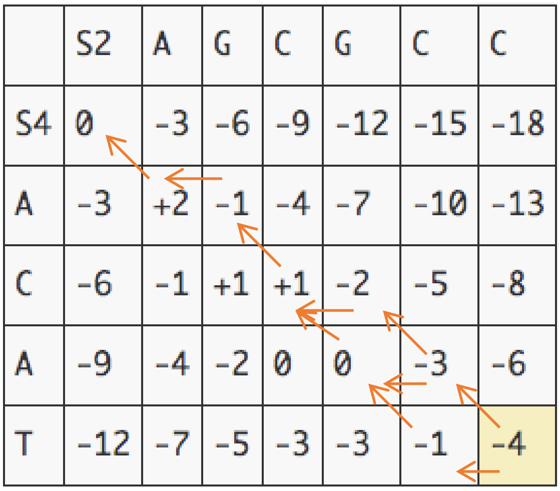
| | S1 | A | A | C | G | T | C |
|---|---|---|---|---|---|---|---|
| S2 | 0 | -3 | -6 | -9 | -12 | -15 | -18 |
| A | -3 | +2 | -1 | -9 | -7 | -10 | -13 |
| G | -6 | -1 | +1 | -2 | -2 | -5 | -8 |
| C | -9 | -4 | -2 | +3 | 0 | -3 | -3 |
| G | -12 | -7 | -5 | 0 | +5 | +2 | -1 |
| C | -15 | -10 | -8 | -3 | +2 | +4 | +4 |
| C | -18 | -13 | -11 | -6 | -1 | +1 | +6 |

*Figure 9 - Dynamic programming matrix obtained for the optimal global alignment of S1 with S2, through the application of the Needleman-Wunsch algorithm.*



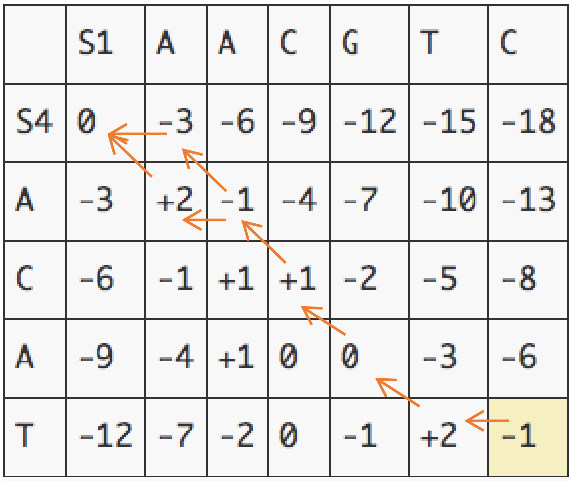
| | S2 | A | G | C | G | C | C |
|---|---|---|---|---|---|---|---|
| S3 | 0 | -3 | -6 | -9 | -12 | -15 | -18 |
| C | -3 | -1 | -4 | -4 | -7 | -10 | -13 |
| C | -6 | -4 | -2 | -2 | -5 | -5 | -8 |
| C | -9 | -7 | -5 | 0 | -3 | -3 | -3 |
| G | -12 | -10 | -5 | -3 | +2 | -1 | -4 |
| T | -15 | -13 | -8 | -6 | -1 | +1 | -2 |

*Figure 10 - Dynamic programming matrix obtained for the optimal global alignment of S2 with S3, through the application of the Needleman-Wunsch algorithm.*



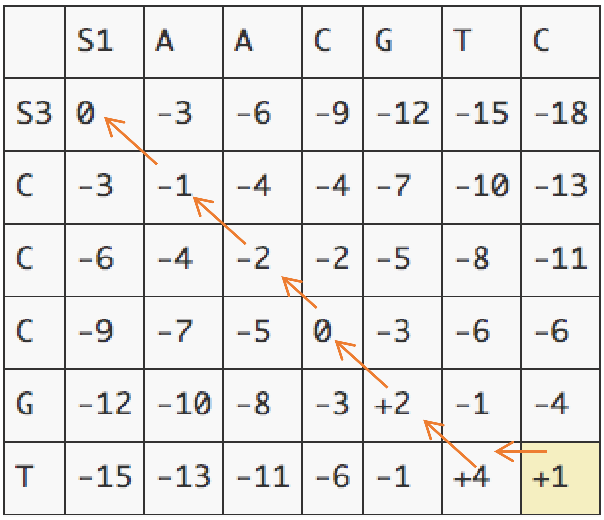
| | S2 | A | G | C | G | C | C |
|---|---|---|---|---|---|---|---|
| S4 | 0 | -3 | -6 | -9 | -12 | -15 | -18 |
| A | -3 | +2 | -1 | -4 | -7 | -10 | -13 |
| C | -6 | -1 | +1 | +1 | -2 | -5 | -8 |
| A | -9 | -4 | -2 | 0 | 0 | -3 | -6 |
| T | -12 | -7 | -5 | -3 | -3 | -1 | -4 |

*Figure 11 - Dynamic programming matrix obtained for the optimal global alignment of S2 with S4, through the application of the Needleman-Wunsch algorithm.*



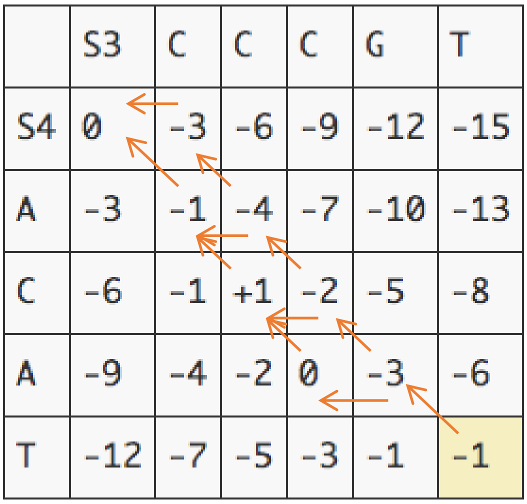
| | S1 | A | A | C | G | T | C |
|---|---|---|---|---|---|---|---|
| S4 | 0 | -3 | -6 | -9 | -12 | -15 | -18 |
| A | -3 | +2 | -1 | -4 | -7 | -10 | -13 |
| C | -6 | -1 | +1 | +1 | -2 | -5 | -8 |
| A | -9 | -4 | +1 | 0 | 0 | -3 | -6 |
| T | -12 | -7 | -2 | 0 | -1 | +2 | -1 |

*Figure 12 - Dynamic programming matrix obtained for the optimal global alignment of S1 with S4, through the application of the Needleman-Wunsch algorithm.*



| | S1 | A | A | C | G | T | C |
|---|---|---|---|---|---|---|---|
| S3 | 0 | -3 | -6 | -9 | -12 | -15 | -18 |
| C | -3 | -1 | -4 | -4 | -7 | -10 | -13 |
| C | -6 | -4 | -2 | -2 | -5 | -8 | -11 |
| C | -9 | -7 | -5 | 0 | -3 | -6 | -6 |
| G | -12 | -10 | -8 | -3 | +2 | -1 | -4 |
| T | -15 | -13 | -11 | -6 | -1 | +4 | +1 |

*Figure 13 - Dynamic programming matrix obtained for the optimal global alignment of S1 with S3, through the application of the Needleman-Wunsch algorithm.*



| | S3 | C | C | C | G | T |
|---|---|---|---|---|---|---|
| S4 | 0 | -3 | -6 | -9 | -12 | -15 |
| A | -3 | -1 | -4 | -7 | -10 | -13 |
| C | -6 | -1 | +1 | -2 | -5 | -8 |
| A | -9 | -4 | -2 | 0 | -3 | -6 |
| T | -12 | -7 | -5 | -3 | -1 | -1 |

*Figure 14 - Dynamic programming matrix obtained for the optimal global alignment of S3 with S4, through the application of the Needleman-Wunsch algorithm.*