AIN411 - Fundamentals of (Introduction to) Bioinformatics

Assignment 1

Burak Kurt

2200765010

Question 1 (10 points)

Please carefully explain each question below (in a total of 2-3 sentences for each item)

a) What is central dogma? What does it say about the information flow at the molecular level in living organisms?

Answer: The central dogma states information flow between DNA, RNA and protein. Information pass through DNA to RNA (transcription), then RNA to proteins (translation). Central dogma also states that this process can not be reversed.

b) What is the significance of homology in bioinformatics??

Answer: Homology is the concept of similarity between biological sequences or structures that are derived from a common ancestor. Homology can be used to infer evolutionary relationships, functional annotations, and structural predictions of unknown proteins

c) Briefly describe the purpose of a multiple sequence alignment in bioinformatics.

Answer: Main purposes of a multiple sequence alignment are:

1) Finding conserved residuals,patterns or regions on sequences to identify important functional regions which is common on both sequences.
2) Estimating how far away each sequence from another in evolutionary time steps.
3) Prediction of protein structures.

d) Explain how BLAST manages to run faster than the optimal sequence alignment. algorithm Does BLAST perform the same as the optimal alignment regarding accuracy?

Answer: Optimal sequence alignment uses Dynamic Programming to achieve optimal score but it is computationally expensive when we search database for given query. BLAST divides given query into parts and search database for exact matches for this parts. When match is found, algorithm extends both sequences to left and right until score down below given threshold value. This process is much faster than optimal sequence alignment algorithm, although it may fail to find best matches which lowers accuracy.

Question 2 (55 points)

Implement the pairwise sequence alignment of amino acid (protein) sequences via dynamic programming (you should select the correct alignment algorithm for the sequences given below and implement only that algorithm, either local / Smith-Waterman or global / Needleman-Wunch). Your implementation should take 2 sequences of any length (written in 2 different lines of the same text file) as input (text file should be accepted as a command line argument), include the following additional input arguments:

- a scoring matrix (should be able to take any scoring scheme in the format of square a matrix),
- a gap opening penalty value (a negative integer), and
- a gap extension penalty value (a negative integer).

In the first part of the output, include the aligned sequences in the classical 3-line notation (first line for the first sequence, second line for the '|' characters for the positions where there is a match between 2 sequences and space characters when there is no match, and third line for the second sequence) including '–' character for gaps in the aligned sequences. This should either be printed on screen or written in an output text file. The second part of the output should be the raw alignment score. The third part of the output should be the percent identity between the two aligned sequences, which can be calculated by multiplying the number of matches in the pair by 100 and dividing by the length of the aligned region, including gaps. You may use any programming language (Python is preferred), but your script should run on a basic Unix/Linux shell (such as bash) without any external dependencies. It is not okay to use specialized libraries such as the ones related to bioinformatics. Sample input-output is given below:

Global sequence alignment (blosum62, gap open: -10, gap extend: -5):

input:

```
MVSPADKTNVKAAWGVG
MVLSGEDKSNIKWGKV
```

output:

```
MV-SPADKTNVKAAWGVG
||  |  ||  |  |  ||
MVLSGEDKSNIK--WGKV
Alignment score: 23.0
Identity value: 9/18 (50.0%)
```

Local sequence alignment (blosum62, gap open: -10, gap extend: -5):

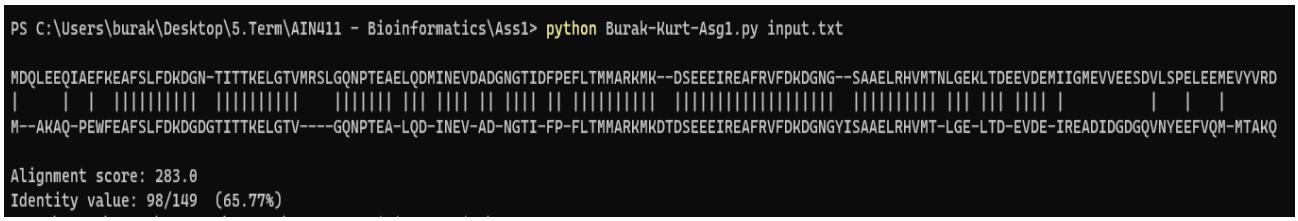
input:

```
MVSPADKTNVKAAWGVG
MVLSGEDKSNIKWGKV
```

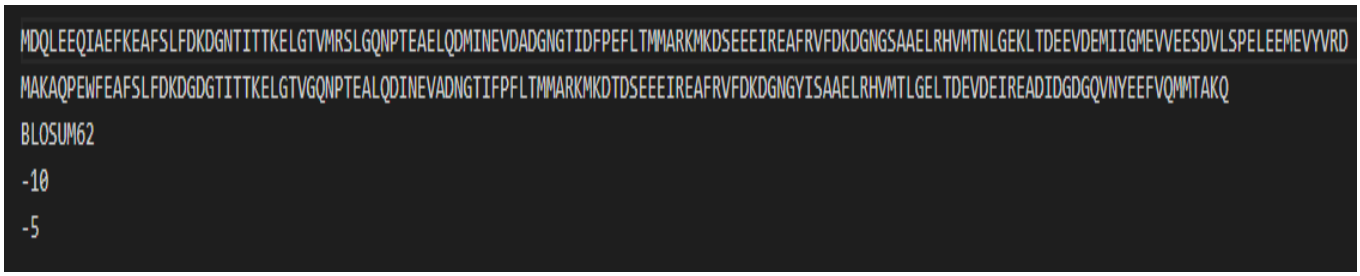
output:

```
MVSPADKTNVKAAWG
   |   ||  |  |   ||
VLSGEDKSNIK--WG
Alignment score: 31.0
Identity value: 7/15 (46.7%)
```

a) Explain how your code runs and show the run command over an example (submit your script file as part of your assignment submission for testing).



1: Run command and output



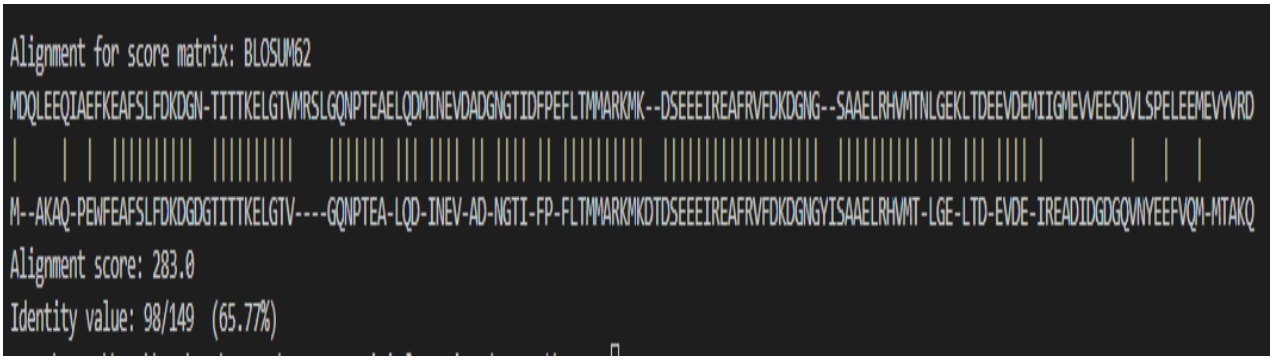
2 Input file example

To be able to run code successfully, prompt “python Burak-Kurt-Asg1.py [input file].txt “ is enough. Content of the input file is shown in image 2. In the first two lines, sequences are given, next line contains scoring matrix, which is only accepts BLOSUM matrices with different scores (62,90,etc.). Gap open and gap extension penalties should be written in the last two lines.

b) Align the sequences of Protein A and Protein B given below, with the alignment algorithm of your choice (using your own implementation), paste the alignment output and percent

identities in your answer sheet (parameters: BLOSUM62, gap open= -10, gap extend= -5). Discuss why did you chose this particular algorithm? Was the algorithm of your choice successful in the end?

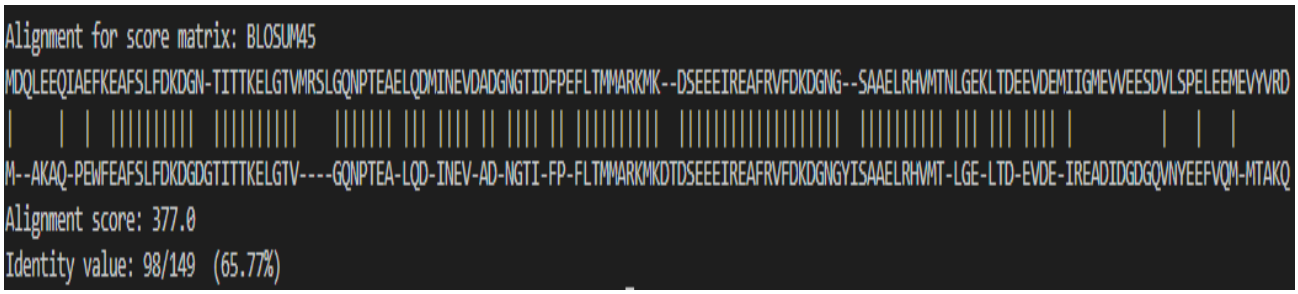
Since lengths of protein A and protein B are similar, global alignment algorithm is preferred for pairwise alignment. Local alignment would be useful if difference between length of two proteins are high or one of them is sub-part of another.



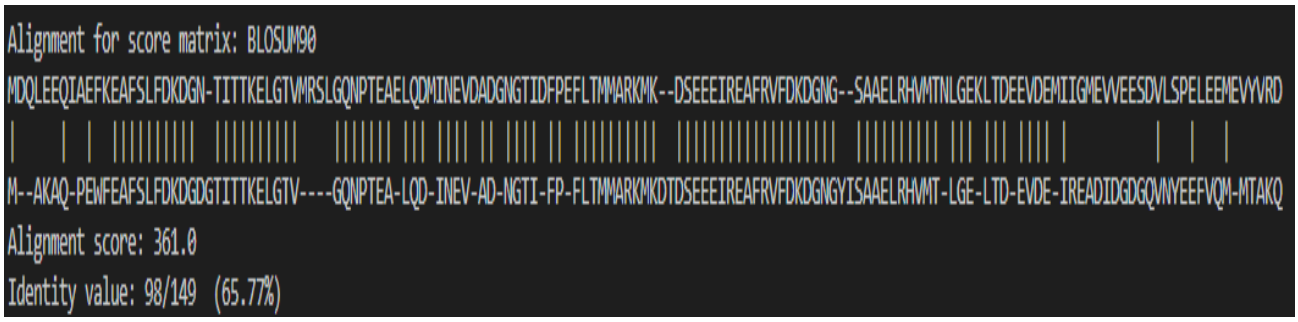
3 BLOSUM62 output

Alignment result and scores shown above. Global alignment algorithm works well for these two sequences, so identity value is 65.77%.

c) Investigate the impact of changing the scoring matrix on the alignment. Choose a different scoring matrix (e.g., BLOSUM45, BLOSUM90, etc.) and discuss how they affect the alignment result. Include specific changes in the alignment score and any variations in the aligned sequences. Discuss your results.



4 BLOSUM45 output



5 BLOSUM90 output

For each matrix score setting, nothing changed but alignment scores. Since score of each amino acid pair differs in each scoring matrix, different alignment scores are obtained. In some cases, alignments would be different when backtracking is applied on the scoring table but for these two proteins, alignments are same for each scoring matrix.

d) Suppose we are looking for a region of functional importance that is similar between these two sequences. This region spans the whole of the shorter sequence but a subset of the longer one. Which algorithm would you choose, and what is the reason behind it? Why could the other algorithms not correctly identify this region?

The Smith-Waterman algorithm is particularly useful in this scenario because it is designed to identify regions of similarity within larger sequences. It compares segments of all possible lengths and optimizes the similarity measure, making it ideal for finding a region of functional importance that spans the whole of the shorter sequence but only a subset of the longer one. The reason why other algorithms may not correctly identify this region is because many of them, such as BLAST, use heuristic methods that do not guarantee finding the optimal alignment. These algorithms are designed to be faster and use less computational resources, but they may miss the best alignment, especially in cases where the region of similarity is small compared to the length of the sequence.

Question 3

S1: ATCGATCGA

S2: ATCGATCGT

S3: ATCGATCGAT

S4: ATCATCGTAA

S5: ACCGGTATG

$\binom{5}{2} = 10$ pairwise alignment

S1-S2:

	-	A	T	C	G	A	T	C	G	A
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	1	2	1	2	3	4	3
T	-2	0	2	1	0	-1	0	1	2	3
C	-3	-1	1	3	2	1	0	1	0	1
G	-4	-2	0	2	4	3	2	1	2	1
A	-5	-1	-1	1	3	5	4	3	2	3
T	-6	-2	0	0	2	4	6	5	4	3
C	-7	-3	-1	1	1	3	5	7	6	5
G	-8	-4	0	0	2	2	4	6	8	7
T	-9	-5	1	-1	1	1	3	5	7	7

S1: ATCGATCGA

S2: ATCGATCGT

Similarity: 8/9 = 0.88

SARP HAVACILIK LOJİSTİK TURİZM SAN. VE TİC. A.Ş.

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

...../...../.....

S1-S3:

		A	T	C	G	A	T	C	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	-1	-2	-1	-2	-3	-4	-3
T	-2	0	2	1	0	-1	0	-1	-2	-3
C	-3	-1	1	3	2	1	0	1	0	-1
G	-4	-2	0	2	4	3	2	1	2	1
A	-5	-1	-1	1	3	5	4	3	2	3
T	-6	-2	0	0	2	4	6	5	4	3
C	-7	-3	-1	1	1	3	5	7	6	5
G	-8	-4	-2	0	2	2	4	6	8	7
A	-9	-3	-3	-1	1	3	3	5	7	9
T	-10	-2	-4	-2	0	2	4	4	6	8

Similarity:

9/10 = 0.9

S1-S4:

		A	T	C	G	A	T	C	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	-1	-2	-1	-2	-3	-4	-5
T	-2	0	2	1	0	-1	0	-1	-2	-3
C	-3	-1	1	3	2	1	0	1	0	-1
A	-4	0	0	2	2	3	2	1	0	1
T	-5	-1	1	1	1	2	4	3	2	1
C	-6	-2	0	2	1	1	3	5	4	3
G	-7	-3	-1	1	3	2	2	4	6	5
T	-8	-4	0	0	2	2	3	3	5	5
A	-9	-3	-1	-1	1	3	2	2	4	6
A	-10	-2	-2	-2	0	4	3	2	3	7

S1: A T C G A T C G _ A _
S4: A T C – A T C G T A A

Similarity: 8/11 = 0.72

SARP
SARP HAVACILIK LOJİSTİK
TURİZM SAN. VE TİC. A.Ş.

ATC_G__ATCGA = S1
A_CCGGTAT_G_ = S5

...../...../.....

S1-S5:

		A	T	C	G	A	T	C	G	A
	(0)	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	(1)	(0)	-1	-2	-1	-2	-3	-4	-3
C	-2	0	0	(1)	0	-1	-2	-1	-2	-3
C	-3	-1	-1	(2)	1	0	-1	0	-1	-2
G	-4	-2	-2	1	(3)	2	1	0	1	0
G	-5	-3	-3	0	(4)	3	2	1	2	1
T	-6	-4	-2	-1	3	(3)	4	3	2	1
A	-7	-3	-3	-2	2	(4)	3	3	2	3
T	-8	-4	-2	-2	1	3	(5)	(4)	3	2
G	-9	-5	-3	-3	2	2	4	4	(5)	(4)

Similarity:
6/12 = 0.5

S2-S3:

		A	T	C	G	A	T	C	G	T
	(0)	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	(1)	0	-1	-2	-1	-2	-3	-4	-5
T	-2	0	(2)	1	0	-1	0	-1	-2	-1
C	-3	-1	1	(3)	2	1	0	1	0	-1
G	-4	-2	0	2	(4)	3	2	1	2	1
A	-5	-1	-1	1	3	(5)	4	3	2	1
T	-6	-2	0	0	2	4	(6)	5	4	5
C	-7	-3	-1	1	1	3	5	(7)	6	5
G	-8	-4	-2	0	2	2	4	6	(8)	7
A	-9	-3	-3	-1	1	3	3	5	7	(7)
T	-10	-4	-4	-2	0	2	4	4	6	(8)

Similarity:
8/10 = 0.8

ATCGATCG_T = S2
ATCGATCGAT = S3

SARP HAVACILIK LOJİSTİK
TURİZM SAN. VE TİC. A.Ş.

ATCGATCGT =S2
ATC-ATCGTAA =S4

...../...../.....

S2-S4:

	-	A	T	C	G	A	T	C	G	T
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	-1	-2	-1	-2	-3	-4	-5
T	-2	0	2	1	0	-1	0	-1	-2	-1
C	-3	-1	1	3	2	1	0	1	0	-1
A	-4	0	0	2	2	3	2	1	0	-1
T	-5	-1	1	1	1	2	4	3	2	3
C	-6	-2	0	2	1	1	3	5	4	3
G	-7	-3	-1	1	3	2	2	4	6	5
T	-8	-4	0	0	2	2	3	3	5	7
A	-9	-3	-1	-1	1	3	2	2	4	6
A	-10	-2	-2	-2	0	4	3	2	3	5

Similarity:
8/11 = 0.72

S2-S5:

	-	A	T	C	G	A	T	C	G	T
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	1	0	-1	-2	-1	-2	-3	-4	-5
C	-2	0	0	2	1	0	-1	-2	-1	-2
C	-3	-1	-1	1	1	0	-1	0	-1	-2
G	-4	-2	-2	1	3	2	1	0	1	0
G	-5	-3	-3	0	2	4	3	2	1	1
T	-6	-4	-2	-1	2	3	4	3	2	3
A	-7	-3	-3	-2	1	4	3	3	2	2
T	-8	-4	-2	-3	1	3	5	4	3	4
G	-9	-5	-3	-3	2	2	4	4	5	4

similarity:
5/12 = 0.41

ATC-G--ATCCT=S2
A-CCGGTATG---=S5

SARP HAVACILIK LOJİSTİK TURİZM SAN. VE TİC. A.Ş.

```
  1 1 1  11 11  1
ATCCATCG-A-T =S3
ATC-ATCGTAA- =S4
```

...../...../.....

S3-S4:

		A	T	C	G	A	T	C	G	A	T
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
A	-1	1	0	-1	-2	-1	-2	-3	-4	-3	-4
T	-2	0	2	1	0	-1	0	-1	-2	-3	-2
C	-3	-1	1	3	2	1	0	1	0	-1	-2
A	-4	0	0	2	2	3	2	1	0	1	0
T	-5	-1	1	1	1	2	4	3	2	1	2
C	-6	-2	0	2	1	1	3	5	4	3	2
G	-7	-3	-1	1	3	2	2	4	6	5	4
T	-8	-4	0	0	2	2	3	3	5	5	6
A	-9	-3	-1	-1	1	3	2	2	4	6	5
A	-10	-2	-2	-2	0	4	3	2	3	7	6

Similarity:
8/12 = 0.66

S3-S5:

		A	T	C	G	A	T	C	G	A	T
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
A	-1	1	0	-1	-2	-1	-2	-3	-4	-3	-4
C	-2	0	1	1	0	-1	-2	-1	-2	-3	-4
C	-3	-1	-1	2	1	0	-1	0	-1	-2	-3
G	-4	-2	-2	0	3	2	1	0	1	0	-1
G	-5	-3	-3	-1	2	2	1	0	2	1	0
T	-6	-4	-2	-2	1	2	3	2	1	1	2
A	-7	-3	-3	-3	1	3	2	2	1	2	1
T	-8	-4	-2	-3	0	2	4	3	2	1	3
G	-9	-5	-3	-3	1	1	3	3	4	3	2

Similarity:
6/12 = 0.5

```
 1 1  1  1  11
ATCG-ATCGAT- =S3
ACCGG-T--ATG =S5
```

SARP HAVACILIK LOJİSTİK TURİZM SAN. VE TİC. A.Ş.

ATCATCGTAA
A-CGG-ITATG

...../...../.....

S4-S5:

		-	A	T	C	A	T	C	G	T	A	A
	-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
A	-1	-1	1	0	-1	0	-1	-2	-3	-4	-3	-2
C	-2	0	0	1	0	1	0	-1	-2	-3	-4	
C	-3	-1	-1	1	0	-1	0	-1	-2	-3	-4	
G	-4	-2	-2	0	0	-1	-1	1	0	-1	-2	
G	-5	-3	-3	-1	-1	-1	-2	0	1	0	-1	
T	-6	-4	-2	-2	-2	-2	-2	1	3	2	1	
A	-7	-3	-3	-3	-1	-2	-3	-1	2	4	5	
T	-8	-4	-2	-3	-2	0	-1	-2	3	3	5	
G	-9	-5	-3	-3	-3	-1	-2	0	2	2	4	

Similarity:

2/10 = 0.2

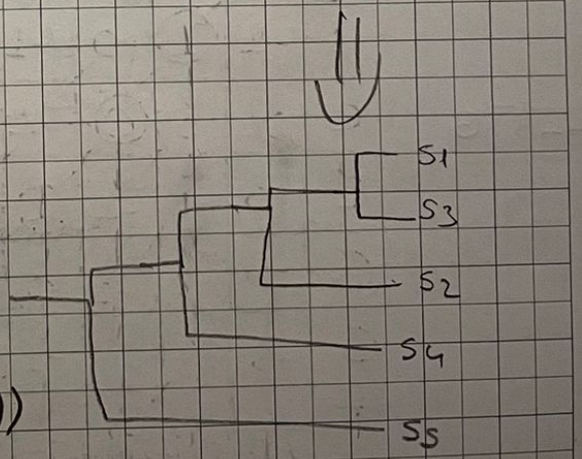
a) Similarity Matrix:

	S1	S2	S3	S4	S5
S1	-	0.88	0.9	0.72	0.5
S2		-	0.8	0.72	0.41
S3			-	0.66	0.5
S4				-	0.2
S5					-

b) Guide Tree

Most Similar genes

S1 - S3 => 0.9
S1 - S2 => 0.88
S1 - S4 => 0.72
S1 - S5 => 0.5



Align(S5, Align(S4, Align(S2, Align(S1, S3))))

Align (S_1, S_3) => Already written: ATCGATCGA_
ATCGATCGAT
⇓
Profile: ATCGATCGAT => S

Align (S_2, S) => Since profile S is same with S_3, we can use alignment of S_2 and S_3.

ATCGATCGA_
ATCGATCGAT
ATCGATCG_T
⇓
Profile: ATCGATCGAT => $S = S_3$

Align (S_4, S) => We can take Align (S_4, S_3) as a result.

ATCGATCGA_
ATCGATCGAT
ATCGATCG_T
ATC_ATCGTAA_
⇓
Profile: ATCGATCGATA => S

Align (S_5, S)

		-	1	-	A	T	C	G	A	T	C	G	A	T	A
		-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
	A	-1	1	0	-1	-2	-1	-2	-3	-4	-3	-4	-3	
	C	-2	0	0	1	0	-1	-2	-1	-2	-3	-4	-5	
	C	-3	-1	-1	1	0	1	-2	-1	-2	-3	-4	-5	
	G	-4	-2	-2	0	2	1	0	-1	0	-1	-2	-3	
	G	-5	-3	-3	-1	1	2	1	0	1	0	-1	-2	
	T	-6	-4	-2	-2	2	2	3	2	1	0	1	0	
	A	-7	-3	-3	-3	1	3	2	2	1	2	1	2	
	T	-8	-4	-2	-3	0	2	4	3	2	1	3	2	
	G	-9	-5	-3	-3	1	1	3	3	4	3	2	2	

...al SS:
...GG_T_-ATG

Final MSA result:

```
ATCGATCGA___   => S1
ATCGATCGATT_   => S3
ATCGATCG_T__   => S2
ATC-ATCGTAA_   => S4
ACCGG-T__-ATG  => S5
1 2 3 4 5 6 7 8 9 10 11 12
```

c) Sum of Pairs Scoring:

Column 1: $S(S_1,S_3)=1 + S(S_1,S_5)=1 + S(S_3,S_4)=1 + S(S_2,S_5)=1$
$S(S_1,S_2)=1 + S(S_2,S_3)=1 + S(S_3,S_5)=1$
$S(S_1,S_4)=1 + S(S_2,S_4)=1 + S(S_4,S_5)=1$
$= 10$

Column 2: $S(S_1,S_2)=1 + S(S_2,S_3)=1 + S(S_3,S_5)=-1$
$S(S_1,S_3)=1 + S(S_2,S_4)=1 + S(S_4,S_5)=-1$
$S(S_1,S_4)=1 + S(S_2,S_5)=-1$
$S(S_1,S_5)=-1 + S(S_3,S_4)=1$
$= 2$

Column 3: Score of 1 comes from each sequence pair $= 10$.

Column 4: +1 from every match except $S(S_1,S_4)=-1$, $S(S_2,S_4)=-1$, $S(S_3,S_4)=-1$, $S(S_5,S_4)=-1$ $= 6$

Column 5: +1 from every match except $S(S_1,S_5)=-1$, $S(S_2,S_5)=-1$, $S(S_3,S_5)=-1$, $S(S_4,S_5)=-1$ $= 6$

Column 6: +1 from every match except matches with S_5 $= 6$

Column 7: +1 from every match except matches with S_5 $= 6$

Column 8: +1 from every match except matches with S_5 $= 6$

Column 9: $S(S_1,S_3)=1$ other 8 match is -1 $= -7$
$S(S_2,S_5)=0$

Column 10: $s(s_3, s_2) = 1$ other 8 match is -1 $= -6$
$s(s_4, s_5) = 1$

Column 11: $s(s_1, s_3) = 0$ other 8 match is -1 $= -6$
$s(s_1, s_2) = 0$

Column 12: $s(s_1, s_5) = -1$
$s(s_2, s_5) = -1$ other 6 match is 0 $= -4$
$s(s_3, s_5) = -1$
$s(s_4, s_5) = -1$

Final Result: $10 + 2 + 10 + 6 + 6 + 6 + 6 + 6 - 7 - 6 - 6 - 4 = 28$

..../..../.....

d) Conserved Residuals → Conserved Region

A	T	C	G	A	T	C	G	A	_	_	_	⇒ human
A	T	C	G	A	T	C	G	A	T	_	_	⇒ monkey
A	T	C	G	A	T	C	G	-	T	_	_	⇒ mouse
A	T	C	-	A	T	C	G	T	A	A	_	⇒ frog
A	C	C	G	G	_	T	_	_	A	T	G	⇒ bacteria

e) According to similarity matrix, bacteria is the most distantly related organism to human because it has the lowest similarity score with human Gene X, it means that, bacteria has the farthest evolutionary distance between human other than 4 organisms. Genes of similar (evolutionary close) species tend to be similar than other species, so if ~~we~~ change Gene X to another gene, results will be similar.