ans 1)

ASSIGNMENT-IV

ROMICA RAISINGHANI
2021101053

Given :

Sequence 1 : GGCTGCAACTAGCTC (length 15)

Sequence 2 : GGGTAAAGCTTGC (length 12)

The transition - transversion scoring matrix is :

	A	C	G	T
A	4	-1	1	-1
C	-1	4	-1	1
G	1	-1	4	-1
T	-1	1	-1	4

Gap penalty = 3

Global alignment: Needleman-Wunsch Algorithm

	X	G	G	G	T	A	A	G	C	T	T	G	C
X	0	-3	-6	-9	-12	-15	-18	-21	-24	-27	-30	-33	-36
G	-3	4	1	-2	-5	-8	-11	-14	-17	-20	-23	-26	-29
G	-6	1	8	5	-2	-1	-4	-7	-10	-13	-16	-19	-22
C	-9	-2	5	7	6	3	0	-3	-3	-6	-9	-12	-15
T	-12	-5	2	4	8	8	5	2	-1	1	-2	-5	-8
G	-15	-8	-1	6	8	12	9	9	6	3	0	2	-1
C	-18	-11	-4	3	4	9	11	8	13	10	7	4	6
A	-21	-14	-7	0	4	11	13	12	10	12	9	8	5
A	-24	-17	-10	-3	1	8	15	14	11	9	11	10	7
C	-27	-20	-13	-6	-2	5	12	14	18	15	12	10	14
T	-30	-23	-16	-9	-5	2	9	11	15	22	19	16	13
A	-33	-26	-19	-12	-8	2	6	10	12	19	21	20	17
G	-36	-29	-22	-15	-8	-1	3	10	9	16	18	25	22
C	-39	-32	-25	-18	-11	-4	0	7	14	13	17	22	29
T	-42	-35	-28	-21	-14	-7	-3	4	11	18	17	19	26
C	-45	-38	-31	-24	-17	-10	-6	1	8	15	19	16	23

Boundary : $F(i,0) = F(0,1) = -id$

$F(0,0) = 0$

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

Final similarity score = 23

Best alignment :

GGCTGCAACTAGCTC

GGGTA-AGCTTG--C

Local Alignment: Smith-Waterman Algorithm

	X	G	G	G	T	A	A	G	C	T	T	G	C
X	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	4	4	4	1	1	1	4	1	0	0	4	1
G	0	4	8	8	5	2	2	5	3	0	0	4	3
C	0	1	5	7	9	6	3	2	9	6	3	1	8
T	0	0	2	4	11	8	5	2	6	13	10	7	5
G	0	4	4	6	8	12	9	9	6	10	12	14	11
C	0	1	3	3	7	9	11	8	13	10	11	11	18
A	0	1	2	4	4	11	12	12	10	12	9	12	15
A	0	1	2	3	3	8	15	14	11	9	11	10	12
C	0	0	0	1	4	5	12	14	18	15	12	10	14
T	0	0	0	0	5	3	9	11	15	22	19	16	13
A	0	1	1	1	2	9	7	10	12	19	21	20	17
G	0	4	5	5	2	6	10	11	9	16	18	25	22
C	0	1	3	4	6	3	7	9	15	13	17	22	29
T	0	0	0	2	8	5	4	6	12	19	17	19	26
C	0	0	0	0	5	7	4	3	10	16	20	17	23

Boundary conditions: $F(i,0) = 0$

$F(0,j) = 0$

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

Final similarity score = 29

Best alignment:

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

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

Ans2) -

The given sequence is :

TGGCACACTCACACCACACAGACAGTTA

The scoring scheme is (1, 0, -1) for match, mismatch and indel, respectively.

We start by comparing each nucleotide in the sequence to the reference dinucleotide "CA" to determine if it's a match or a mismatch.

If it's a match, we assign a score of 1.

If it's a mismatch, we assign a score of 0

T G G C A C A C T C A C A C C A C A C A G A C A G T T A
0 -1 -1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 -1 1 1 1 1

As we go through the sequence, we add up the scores for each position to get the total score. In this case, the dinucleotide "CA" is repeated multiple times in the sequence "CACACACACAGA" and all the comparisons result in matches, so we add 1 for each position where "CA" is matched.

Adding up the scores, we get = 18

So, the score for the CA repeat region is 18.

Ans3) - Dynamic Programming (DP) is a technique used in various computational algorithms for solving optimization problems.

In case of overlap regions, DP can be used when we want to find the best alignment of two sequences such that one sequence is allowed to overlap with the other. This can occur in scenarios such as finding overlapping regions in DNA sequences, identifying overlapping regions in image processing, or determining overlapping regions in speech or audio signals.

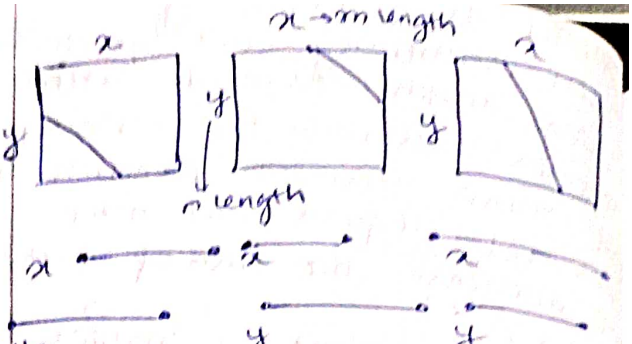
The overlap regions are observed when we are comparing large chromosomal sequences in sequence assembly or fragments of genomic DNA sequences.

Boundary conditions :

We need to define boundary conditions to determine the starting and ending points of the alignment. The boundary conditions are usually dependent on the specific problem and the desired outcomes. For example, in the case of DNA sequence overlap, the boundary conditions could be the start and end positions of the overlapping regions in the two sequences.

Recursive relations :

Recursive relations in DP for overlap regions are similar to regular DP algorithms, but with modifications to account for the overlapping regions. The recursive relations depend on the scoring system used for alignment and the specific constraints of the problem.



Above shown is a special case of global alignment which does not penalize overhanging ends.

We will initialize the DP in the same way as we do for local alignment.

Hence,

$F(0,0) = 0$

$F(i,0) = 0$

$F(0,j) = 0$

for $i = 1, 2, \ldots, m$

$j = 1, 2, \ldots, n$

Also,

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

Here, Boundary start : $F(0,j)$ or $F(i,0)$

Boundary end : $F(i,n)$ or $F(m,j)$

Here, recurrence in DP follows that of global alignment.

Now,

Traceback conditions :

The traceback starts from $f_{max} = \max\{F(i,n), F(m,j)\}$ and continues till top $F(i,0)$ or left $F(0,j)$ edge is received for any i and j.

Ans4) – Following are the advantages of using affine gap scores in sequence alignment:

1) Flexibility in modeling gaps:
Affine gap scores allow for a more flexible modeling of gaps or insertions in sequences compared to simple gap penalities. In simple gap penalities, a fixed penality is applied for each gap regardless of its length or position in the sequence. On the other hand, affine gap scores consider both the opening and extension penalities for gaps separately. This allows for a more nuanced representation of the cost of gaps, as the opening penalty represents the cost of initiating a gap, and the extension penalty represents the cost of extending an existing gap. This flexibility allows for a more accurate representation of the potential variation in the cost of gaps in real biological sequences.

2) Consideration of local context:
Affine gap scores take into account the local context of gaps in the sequences being aligned. In many biological sequences, gaps tend to occur in specific regions with distinct characteristics. For example, in protein sequences gaps often occur in loops or other structurally flexible regions. Affine gap scores can capture their local context by allowing for different penalties for opening and extending gaps in different regions of the sequences. This can lead to more biologically meaningful alignments that reflect the characteristics of the sequences being aligned.

3. Improved alignment accuracy:
The use of affine gap scores can potentially result in improved alignment accuracy. By considering both the opening and extension penalties for gaps separately, affine gap scores can more accurately reflect the actual costs associated with gaps in biological sequences. This can result in alignments that better capture the evolutionary relationships or functional similarity between sequences. In contrast, simple gap penalties may not accurately capture the true costs of gaps and may lead to less accurate alignments.

4. Ability to fine-tune gap penalties:
Affine gap scores provide the flexibility to fine-tune the gap penalties to better suit the specific characteristics of the sequences being aligned.

Affine gap score is defined as:

$$\gamma(g) = -d - (g-1)e$$

where

$d \rightarrow$ gap open penalty
$g \rightarrow$ number of consequent gaps.
$e \rightarrow$ gap extension penalty

Affine gap score provides more sequence methods as consecutive deletions or insertions are a single mutation event as compared to multiple insertions and deletions and hence should be penalised less.

Ans5) - consider two sequences of lengths 'm' and 'n' respectively.
Then,

Time complexity = $O(nm)$

space complexity = $O(nm)$

These are the time and space complexities of DP algorithm for sequence comparison.

Time complexity becomes an issue in database search, where a query sequence of length 'n' is searched in a database of few GigaBytes in size.

Space complexity becomes an issue when comparing complete genoms / chromosomes that are atleast few MegaBytes only.
Moreover,

① Large input sequences:
Here, the time complexity results in longer computation time.
Similarly, space complexity leads to significant memory usage, especially when comparing multiple sequences simultaneously or when working with large datasets.

② High computational demands:
Some applications of sequence comparison, such as large-scale genomic or proteomic analyses, may involve comparing multiple sequences or performing iterative comparisons. In such cases, cumulative time and space requirements of DP algorithm can become substantial and impact overall computational demands of analysis.

③ Limited computational resources:
In situations where computational resources are constrained, such as on low-end hardware or in resource-limited environments, the time and space requirements of DP algorithm can pose challenges. Limited processing power or memory capacity may result in longer computation times or insufficient memory for storing large matrices, leading to performance issues or even program failures.

Ans6) - PSI-BLAST (Position-Specific Iterated BLAST) and BLASTP (Basic Local Alignment Search Tool for Proteins) are two related programs used for protein sequence comparison and identification. for homologous sequences, but they have some key differences:

① Iterative search: PSI-BLAST is an iterative search algorithm while BLAST-P is a single round search algorithm.

② Position-Specific scoring matrix (PSSM): PSI-BLAST generates a PSM during the iterative search process, whereas BLAST-P uses a fixed scoring matrix chosen by the user.

③ Sensitivity: PSI-BLAST is generally more sensitive than BLASTp in detecting distant distantly related sequences or sequences with low sequence similarity.

④ Computational time: PSI-BLAST may require more computational time compared to BLASTp due to its iterative nature and generation of a refined PSSM during the search process.

(5) False positives: PSI-BLAST may have a higher likelihood of generating false positives compared to BLASTp due to the iterative search process and use of PSSM.

(6) Output format ~~Search strategy~~: PSI-BLAST typically generates a PSSM as part of the output, which can be used for subsequent searches, while BLASTp generates standard pairwise sequence alignments.

(7) Search strategy: PSI-BLAST uses a profile-profile comparison approach, while BLASTp uses a pairwise sequence comparison approach.

(8) User-input: PSI-BLAST requires a multiple sequence alignment (MSA) or a sequence database as input in addition to the query sequence, while BLASTp only requires a query sequence and a protein database as input.

(9) Algorithmic complexity: PSI-BLAST is generally more complex than BLASTp due to its iterative nature, use of PSSM, and profile-profile comparison approach.

Ans 7)- Reasons for choosing a large match/mismatch ratio for highly conserved sequences and a small match/mismatch ratio for divergent sequences in the BLAST database search algorithm are:

(1) Emphasis on conserved regions: Highly conserved sequences are functionally important and biologically relevant. By choosing a large match/mismatch ratio for highly conserved sequences, BLAST places more emphasis on accurately detecting and aligning these regions to identify functional domains, motifs, or important regions in nucleotide sequences.

(2) Increased sensitivity for conserved sequences: Conserved sequences are expected to have a higher similarity. A larger match/mismatch ratio allows BLAST to more sensitively detect and align these sequences, even when the similarities are subtle. This increases the sensitivity of BLAST in identifying homologous sequences, especially when searching for closely related sequences.

(3) Penalty for divergent sequences: Divergent sequences are expected to have a lower similarity and may contain more mismatches. By using a smaller match/mismatch ratio for divergent sequences, BLAST penalizes mismatches less, allowing for a higher number of mismatches without significantly affecting the alignment score. This allows BLAST to tolerate more sequence divergence and align sequences with lower similarity or identity.

(4) Balance between sensitivity and specificity: The choice of match/mismatch ratio

in BLAST is a trade-off between sensitivity and specificity. A large match/mismatch ratio increases sensitivity by allowing for more accurate detection of conserved regions, but may also increase the risk of false positives.
On the other hand, a smaller match/mismatch ratio increases specificity by penalizing mismatches more, but may reduce sensitivity by making it harder to detect distantly related or divergent sequences.

(5) Optimization for different sequence types: Different sequences have varying levels of conservation and divergence. Choosing a large match/mismatch ratio for highly conserved sequences and a small match/mismatch ratio for divergent sequences allows BLAST to optimize the alignment results for different types of nucleotide sequences, considering their unique characteristics.

Ans 8) In BLOSUM62 matrix, a conserved tryptophan position has score $S(W,W) = 11$ but a conserved leucine position has score $S(L,L) = 4$.

The BLOSUM62 matrix is a substitution matrix used to score amino acid alignments. A pair of amino acids' score is determined by the frequency of their replacement in a database of aligned sequences.
A pair's score is also determined by the likelihood of their occurrence in random sequences.

The physiochemical qualities of the amino acids are also used to calculate the score of a pair of amino acids.

Now,
Tryptophan (W) is a large, hydrophobic, aromatic amino acid that is uncommon in proteins.
Also, Leucine (L) is a short, hydrophobic, aliphatic amino acid found in many proteins

Tryptophan's scarcity and the conservation of its location in a protein sequence imply that it plays a vital part in the protein's structure or function.

The presence of Leucine and the conservation of its location in a protein sequence indicate that it is a vital for protein stability or packing.

Because the physiochemical qualities of ~~the~~ Tryptophan and leucine differ, their substitution frequencies and probabilities may differ.

As a result, the BLOSUM62 matrix score of $S(W,W) = 11$ and $S(L,L) = 4$ may represent the various functions and features of Tryptophan and Leucine in proteins.

Ans 9) – Given:

RNA sequence:

AUGUGGCAUGCCAGG

The dotplot approach is a graphical method used to identify self complementary regions in a nucleotide sequence.

It involves creating a matrix, where each cell represents a pair of nucleotides in the sequence, and marking a dot in the cell if the corresponding pair of nucleotides are complementary to each other.

5' to 3' forward sequence

AUGUGGCAUGCCAGG

5' to 3' forward form of reverse complement

CCUGGCAUGCCACAU

Here,

In the dotplot, the cross represents complementary nucleotide pairs, which are aligned along the diagonal.

Based on the dotplot, we can identify the self-complementary regions in the given RNA sequence.

	A	U	G	U	G	G	C	A	U	G	C	C	A	G	G
C							X				X	X			
C							X				X	X			
U		X		X					X						
G			X		X	X				X				X	X
G			X		X	X				X				X	X
C							X				X	X			
A	X							X							
U		X		X					X						
G			X		X	X				X				X	X
C							X				X	X			
C							X				X	X			
A	X							X					X		
C							X				X	X			
A	X							X					X		
U		X							X						

The length of the longest self complementary region is 10

The longest self complementary region is

UGGCAUGCCA

Other small self complementary regions are:

AUG and CAU (length 3 each)