Science-2 - Part 2

Assignment-2:-

NAGA. MANOHAR
2021101128

(i) consider a DP matrix of ~~length~~ dimensions

$((\text{length of sequence } 1)+1) \times ((\text{length of sequence } 2)+1)$

rows × cols

Here, rows depict S_1 & cols ~~are~~ depicts S_2

$S_1 \overset{1st}{-}$ sequence1
S_2 - sequence 2

Now, we have to build the matrix using following recursive relation / eqn - for 2 alignments

$$① \; dp[i][j] = \max \begin{cases} dp[i-1][j-1] + score(S_1[i-1], S_2[j-1]), \\ dp[i-][j] \cancel{+} d, \\ dp[i][j-] - d \end{cases}$$

Boundary : $dp[i][0] = dp[0][j] = -i * d$

$dp[0][0] = 0$

d = gap-open penalty

– Global Alignment

(2)

$$dp[i][j] = \max \begin{cases} 0, \\ dp[i-1][j-1] + score(\cancel{dp}\, s_1[i-1], s_2[j-1]), \\ dp[i-1][j] - d, \\ dp[i][j-1] - d \end{cases}$$

Boundary Conditions: $dp[i][0] = 0$, $dp[0][0] = 0$

$dp[0][j] = 0$,

Global alignment

① Needlemann-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	11	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	7	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	-7	-2	2	9	11	15	22	19	16	13
A	-33	-26	-19	-12	-9	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	13

Final similarity ~~So~~ Score = 23

Best alignment :-

GGCTGCAACTAGCTC

GGGTA-AGCTTG --C

②

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	13	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	19
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

Final similarity score = 23

Best Alignment:

GGCTGCAACTAGCTC
GGGTA-AGC TTGC--

② Given score (match, mismatch, indel) = (1, 0, −1)

there, we have to consider the CA sequence like "CACACA ——

comparison sequence is

$-TGG\boxed{CACA}CT\boxed{CACA}-C\boxed{CACACA}GA\boxed{CA}GTTA$

~~given score~~ =

$1+1+(-1))+3+0+1+0+0$

$\therefore -1+0+1+1+0+$

$=6$ is the required score

(3)

DP for overlap regions

DP for overlap regions can be used for sequence alignments where we need to find the best alignment between 2 sequences of DNA (or) protein.

Thus its used when you need to compute optimal solutions for subproblems that share common Regions. (as in 2 DNA / Protein sequences)

Boundary Conditions:-

It involves determining the optimal solution for the smallest subproblems (or) base cases.

For sequence alignment, the base case is when one sequence becomes empty, resulting in score of zero, as there is no overlap between sequences

Recursive Relations:-

let
S_1, S_2 be the sequences

$M[i][j]$ be the maximum alignment score

for substrings $S_1[0:i]$ and $S_2[0:j]$

$s[0:x] = [s_0 s_1 \ldots s_{x-1}]$

initial substr of length x

$$M[i][j] = \max \left\{ \begin{array}{l} M[i-1][j-1] + \text{Score}(S_1[i-1], S_2[j-1]) \\ M[i-1][j] - \text{gap-penalty}, \\ M[i][j-1] - \text{gap-penalty} \end{array} \right\}$$

$d = \overset{\text{open}}{\text{gap-penalty}}$

Boundary conditions are when either $i == 0$ (or) $j == 0$

and

the final answer would be $= M[\text{length}(S_1)][\text{length}(S_2)]$

- Score function :→ gives the respective match (or) mismatch score

4 Advantages of using affine gap scores?

They offer advantages over- linear gap penalties in sequence alignment by accounting for stretches of gaps and avoiding over-penalisation of long gaps.

~~They use~~ affine score ⇒ $\delta(g) = -d-(g-1)e$

linear score ⇒ $\delta(g) = -g.d$

where

g = gap length e = gap-extension penalty

d = gap-open penalty

$e<d$ - allows long insertions and deletions to be penalised less than they would be by the linear gap cost.

- d/e - provide more flexibility in gap penalisation.

~~Affine gap scores can avoid excessive penalisation of long gaps~~

- Using affine gap scores can lead to more accurate and biologically meaningful sequence alignments, particularly when dealing with larger gaps (or) longer sequences

They are useful in detecting homologous regions between different genes and proteins.

Many popular alignment tools, like BLAST use affine gap scores by default.

Complexity of DP

let $n, m \rightarrow$ length of ~~the~~ the 2 sequences

then

Time complexity is $O(m \times n) = O(mn)$
$= O(nm)$

Space complexity is $O(nm)$

space complexity issue :- Not feasible for comparing complete genomes or chromosomes ~ a few Mbs long

Time complexity issue :- In database search, a query sequence of length n is searched in a database of size ~ few Gbs.

Thus for long sequences both time & space are an issue.

In addition to this, if the scoring function is complex, such as when using more advanced substitution matrices, can increase the computational complexity of the algorithm

Techniques like heuristics, parallelisation and divide & conquer can reduce complexity and make it feasible to compare larger sequences (or) use more complex scoring functions.

$O(nm)$ - is because we need to store the alignment scores for each pair of substrings in the 2 sequences.

⑥

PSI-BLAST vs blastp

PSI-BLAST (Position Specific Iterated BLAST) is a variant of ~~Bla~~ BLAST algorithm that uses query profile instead of a single query sequence.

It generates a position-specific scoring matrix (PSSM) based on multiple iterations of a preliminary BLAST search, which is used to score the alignments in subsequent searches

PSI-BLAST is more sensitive than blastp in identifying distantly related homologs and refining the alignment and scoring od these homologs, but can be more computationally intensive due to its iterative nature and the need to generate and update the PSSM in each iteration

The choice between PSI-BLAST and blastp depends on the specific needs of the user, the computational resources available and on the nature of the protein sequence analyses taste

For highly conserved sequences, the goal is to identify subtle differences between sequences that may be indicative of functional differences.

A small match/mismatch ratio will make it easier to identify these subtle differences, as even small differences in sequence can result in a lower alignment score

For divergent sequences, a larger match/mismatch ratio is needed to account for the larger number of differences between sequences.

A larger ratio can help ensure that even if there are many mismatches between the query sequence and database sequences the alignment score will still be significant enough to indicate a potential notch

Thus, by adjusting the ratio based on the level ~~x~~ of sequence conservation, the algorithm can optimise the alignment score to identify potential matches while minimizing false positives

8

~~Given that,~~

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 based on observed frequencies of amino acid substitutions in protein alignments.

Tryptophan is a rare amino acid, occurring in only about 1% of amino acid residues, while Leucine is more common, occurring in about 9%.

The higher score for Tryptophan compared to Leucine in the BLOSUM62 matrix is to reflect its lower frequency of occurrence and higher degree of conservation in protein sequences

The BLOSUM62 matrix is designed to reflect the relative frequency of substitutions that occur in real proteins. As a result, amino acids that are less common (or) occur in more conserved positions are generally assigned higher scores to reflect their higher degree of conservation

(9)

To find the Self-complementary regions in the given RNA sequence

AUGWGGCAUGCCAGG, we first create a matrix with RNA sequences on both axes

A dot is placed at each position of the matrix where the RNA forms a base pair with itself

	A	U	G	U	G	G	C	A	U	G	C	C	A	G	G
A		●		●					●						
U	●							●					●		
G							●				●	●			
U	●							●					●		
G							●				●	●			
G							●				●	●			
C			●		●	●				●				●	●
A		●		●					●						
U	●							●					●		
G							●				●	●			
C			●		●	●				●				●	●
C			●		●	●				●				●	●
A		●		●					●						
G							●				●	●			
G							●								

Thus,

the diagonals in the plot show RNA self-complementary regions in the given sequence. ~~They are~~

Longest self-complementary region from the plot :-

= UGGCAUGCCA

other small regions are :-

AUG, CAU (length-3),

and there are many length-2 regions as well.