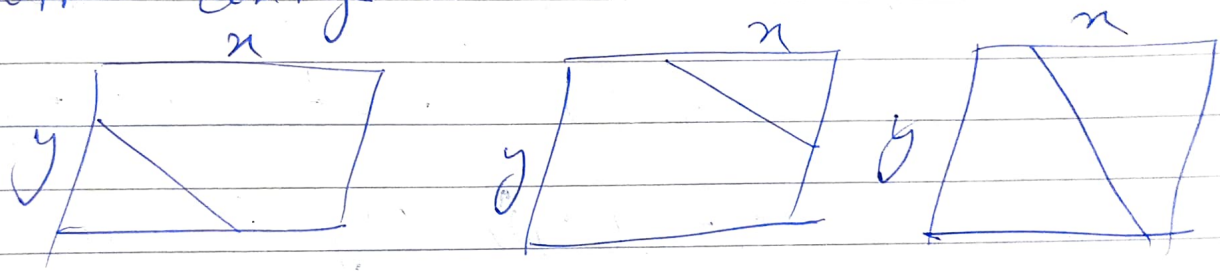
Nameesh Narayan Tiwari

20201010774

Q3 Overlap sequences are observed when one sequence is contained in another or the two sequences have some common overlapping regions.

These situations come forth when we are comparing fragments of genomic DNA sequence to each other or to large chromosomal sequences like in sequence assembly.

diff. configs :



initialization eqns :

$$F(0,0) = 0 \qquad F(i,0) = 0 \quad \forall \quad i = 1 \ldots n$$

$$F(0,j) = 0 \quad \forall \quad j = 1 \ldots m$$

rec. relations :

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

Boundary conditions:

start - $F(0,j)$ or $F(i,0)$ $i = 1 \ldots n$

end - $F(i,m)$ or $F(n,j) = j = 1 \ldots m$

traceback conditions:

traceback starts from Fmax, where Fmax is the max value on the bottom border (i,m) or on the ~~right~~ right border (n,j) & continues till top $(i,0)$ or left $(0,j)$ edge is reached

for any $i = 1 \ldots n$ & $j = 1 \ldots m$

Q1) → GGCTGCAACTAGCTC

→ GGGTAAGCTTGC

Global alignment (Needlemen - Wunsen algo)

		G	G	G	T	A	A	G	C	T	T	G	C
	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
G	-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	-9	-2	2	9	11	15	22	19	16	13
A	-33	-26	-19	-12	-5	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	24	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

Genius
Page:
Date: / /

$d = -3$ Boundary conditions: $F(i,0) = F(0,i) = -id$

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

Final similarity score = 23

Best alignment:

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

Local alignment (Smith waterman algo)

Q1

		G	G	G	F	A	A	G	C	T	T	G	C
	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	13	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

$d = -3$ boundary cond'n : $F(i,0) = F(0,i) = 0$

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

Final similarity score = 29

Best alignment :

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

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

(Q2) dinucleotide CA repeat region

TGG CACACT CACA CCACACA GACAGTTA

TGG [CA][CA] CT [CA][CA] C [CA][CA][CA] GA [CA] GTTA

Tandem repeat region : CACA CCACAC

with repeat element CACAC

score of CACAC = 20

(Q4) advantage of affine gap scores

Affine gap score : $\gamma(g) = -d-(g-1)e$

d :- gap open penalty

e :- gap extension penalty

g :- number of the consequent gaps

$e < d$ allows long insertions & deletions to be penalized less compared to that obtained by linear gap score.

Affine gap score provides more sensitive sequence matching methods as it assumes that consecutive deletions or insertions are a single mutation event as opposed to multiple insertions or deletions & so should be penalized less.

Q5 Time complexity of DP = $O(nm)$

Space complexity of DP = $O(nm)$

n, m be the length of sequences

Time complexity may create problems in database search where a query sequence of length 'n' is searched in a database of a few GBs in size (approx)

Space complexity -

when comparing complete genomes / chromosomes that are atleast a few MBs long.

Q10 BLOSUM 62 matrix

conserved Tryptophan position - score S(W,W) = 11

conserved Leucine position - score S(L,L) = 4

reasons

Leucine residues can be easily substituted by other amino residues whereas Tryptophan and cysteine can't be easily substituted as they are found at key positions in proteins.

Q9 (i) Relative magnitudes of the match score (M) & mismatch score (N) determines the no. of nucleic acid PAMs (point accepted Mutations per 100 residues) for which they are most sensitive at finding homologues.

∴ reward/penalty ratio should be increased as one observes more divergent sequences.

Hence M/N ratio for comparing nucleotide sequences is chosen to be large for highly conserved sequences while it is small for diverse sequences.

(ii) M/N ratio of 1/2 is the best suited for 95% conserved sequences

	A	C	G	T
A	1	-2	-2	-2
C	-2	1	-2	-2
G	-2	-2	1	-2
T	-2	-2	-2	1

(Q8) PSI-BLAST v/s BLAST programs

PSI-BLAST

→ allows user to build a position specific scoring matrix derived during the search itself unlike BLAST.

→ it constructs scoring matrices by multiple alignment of hits obtained.

→ first we search the database with the new scoring matrix for every iteration & iterate until convergence is reached.

BLAST

→ this is a heuristic algorithm designed to find high scoring local alignments between a query sequence & a target database.

→ we look for HSP by using a scoring matrix aligned pairs. Only those pairs which score above a threshold are considered for extension.

→ These HSPs are then extended in both directions to get MSP until score drops below a threshold drop-off from the maximum score encountered. Thus gives us the MSP b/w 2 sequences.

HSP - high scoring segment pairs MSP - maximal segment pairs

(Q7) we get more significant matches for protein :

reasons

→ DNA made of just 4 characters (A, T, G, C) so even 2 unrelated DNA are expected to have ~ 25 %.

Whereas protein sequences is composed of 20 amino acids hence high similarity will imply homology.

→ DNA search has more random matches compared to protein as very different DNAs can code for similar proteins.

→ DNA databases are much larger & grow faster hence going more random hits.

→ for DNA, we generally use identity matrices whereas for proteins we employ more sensitive matrices such as BLOSUM.

(Q6) construction of Nucleic acid PAM scoring matrices

A log-odds approach is followed where score is proportional to the log of ratio of target frequencies to background frequencies

i.e Score matrix for time t is given by:

$$S(a,b/t) = \log \frac{P(a/b,t)}{q_a \, q_b}$$

$P(a/b,t)$:- conditional probability that 'b' is substituted by 'a' in 't'.

q_a, q_b :- freq. of AAs in 'a', 'b' respectively.

first align closely related sequences (>85% identity) & then observe the probability of AA changes & compute the log-odds ratio.

Then normalize the matrix to give average change of 1% of all positions to obtain PAM-1 matrix.

to derive scoring matrices for distantly related sequences from data about closely related sequences, extrapolate PAM-1 to get the scoring matrices to any PAM distance as follows:

$$M_n = (M_1)^n$$

M_n – substitution prob. after n PAMs

M_1 – matrix reflecting 99% seq. conservation & one accepted point mutation (PAM-1) per 100 residues.