Assignment-3

2018101075

1. Given two nucleotide sequences

S = GGCTGCAACTAGCTC

T = GGGTAAGCTTGC

- After running DP for global alignment

$dp(i,0) = i \cdot d$ for $i = 0, \dots n$

$dp(0,j) = j \cdot d$ for $j = 0, \dots, m$

where d is gap penalty = -3,

n is length of S and m is length of T

$dp(i,j) = \max(dp(i-1,j-1) + s(x_i, y_j), dp(i-1,j) + d, dp(i,j-1) + d)$ for $\forall\ i,j > 0$, $i \le n$, $j \le m$

dp[n][m] = Similarity Score is : 23

S' = GGCTGCAACTAGCTC

T' = GGGTA_AGCTTG__C

- DP for local alignment

$dp(i,0) = 0$ for $i = 0, \dots n$

$dp(0,j) = 0$ for $j = 0, \dots m$

$dp(i,j) = \max(dp(i-1,j-1) + s(x_i, y_j), dp(i-1,j) + d, dp(i,j-1) + d, 0)$ $\forall\ i,j \ge 0$, $i \le n$, $j \le m$

Similarity Score = $\max_{i,j}(dp[i,j]) = 29$

S' = GGCTCACAACTAGC

T' = GGGCTA-AGCTTGC

- Algo is written in .py . Run it to see the results

3. Overlap sequences occur when one sequence contained in the other, or they have common overlaping regions e.g. when comparing fragments of genomic DNA sequence to each other, or to large chromosomal sequences, in sequence assembly.
- This is a special a case of global alignment that does not penalize overhanging ends, also called semi-global alignment.

- Boundary conditions are different from global alignment, $F(0,0) = 0$, $F(i,0) = 0$ $i = 1, \dots n$

$F(0,j) = 0$ for $j = 1, \dots m$

- Recurrence relation is same as for global alignment

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

- Fmax is the maximum on the bottom border (i, m) for $i = 1, \dots n$ or the right border (n, j), $j = 1, \dots m$ which is different from global alignment where we take $Fmax = dp[n][m]$.
- Traceback starts from Fmax and ends when $(i,0)$ or $(0,j)$ is reached, same as global alignment

4. • An affine score $\gamma(g) = -d - (g-1)e$
where d - gap open penalty, e - gap extension penalty.
• $e < d$ allows long insertions and deletions to be penalized less than they would be the linear gap cost.
An affine score assumes that consecutive deletions / insertions and hence should be penalized

• An affine score assumes that consecutive deletions and insertions are a single mutation event as opposed to multiple insertions / deletions and hence should be penalized less.
• The advantage of Affine Score gaps is that they provide the most sensitive sequence matching methods.

5. Time complexity = $O(nm)$
Space complexity = $O(nm)$
where n and m are the lengths of two sequences
• The time complexity would be an issue in database search where a query sequence of length n is searched in a database of size approx few Gbs.
• Space complexity would be issue, as large space is required for comparing sequences of length few mbs which is not feasible.

6. PAM stands for Point Accepted Mutations used to measure the amount of evolutionary distance between two protein sequence. For any specific pair (A_i, A_j) of amino acids the (i,j) entry in the PAM n matrix reflects the frequency at which A_i is expected to replace with A_j in two sequences that are n PAM units diverged where one PAM of evolution means that the total number of substitutions is 1% of the sequence length.

- In the first stage, statistics are collected from aligned sequences that were believed to be one PAM units diverged and PAM1 matrix could be computed based on this data. Let m_{ij} denote the observed frequency of Amino acid A_i mutating into Amino acids A_j during one PAM units of evolutionary change. Once M is known, M^n gives the probabilities of any amino acid mutating to any other during n PAM units. The (i,j) entry in the matrix is therefore:-

$$\log\left(\frac{f(j) \cdot M^n(i,j)}{f(i)\, f_j}\right) = \log\left(\frac{M^n(i,j)}{f_i}\right)$$

where $f(i)$, $f(j)$ are the observed frequencies of amino acids A_i, A_j respectively. This approach assumes the frequencies of amino acids remain constant over time and mutational processes causing substitutions during an interval of one PAM unit operate in the same manner for longer periods. We take log value of probability in order to allow computing the total score of all substitutions using summation rather than multiplication.

7. We get more significant matches for the protein search. The reason are as follows:-

- DNA is made of just 4 different characters. Thus even two unrelated DNA are expected to have appron 25% similarity.
- A protein sequence is composed of around 20 different amino acids. This definitely improves the sensitivity of the comparison. Thus matches happening with proteins usually come from homologues.
- When comparing DNA sequences, we get significantly more random matches than we get with proteins. The reason for the same is that
  - DNA databases are much larger and grow faster than protein databases and thus experience more random hits.
  - For DNAs, we usually use identity matrices, while for proteins, we use more sensitive matrices like PAM and BLOSUM. These usually result in better search results.
  - Proteins are rarely mutated during evolution. Due to their conservation, searching them reveals remote evolutionary relationships.

8. Blast:-

1. It's search is based on similarity matrix. It searches one or more nucleic acid or protein databases for sequences similar to one or more sequences of any type. We see that it uses similar instead of identical pairs.
2. It uses a scoring matrix to score aligned pairs. Only those pairs that score above a threshold

all considered for extension (which, in itself, is without gaps).

3. Uncapped extension of HSPs with scores greater than threshold identifies maximal segment pairs.

4. The extension continues until the score drops below a threshold drop off from the maximum score encountered. Thereafter, the highest scoring segment pair, the MSP is identified.

5. BLAST can produce gapped alignments for the matches it finds. This, it does using the same strategy as FASTA of joining segments on different diagonals.

PSI - BLAST

1. It iteratively searches one or more protein databases for sequences similar to one or more protein query sequences. It is designed to find remote homologues with 15% - 25% identity levels.

2. It constructs scoring matrices by multiple alignment of hits obtained.

3. It searches the database with the new scoring matrix for every iteration. This iteration continues till convergence is reached.

4. The idea behind constructing a scoring matrix from the hits is that the new scoring matrix is tailor-made to find sequences similar to the query.

5. PSI blast unlike BLAST (which uses scoring matrix) uses position specific scoring matrices derived during the search itself.

↓

Similarity based

9. i) Scoring matrices for match M/ mismatch N ratio. Relative magnitude of M & N determines the NO. of nucleic acid PAMs (point accepted mutations per 100 residues) for which they are most sensitive at finding homologs. Therefore the absolute reward/penalty ratio should be increased as one looks at more divergent sequences and hence match/mismatch ratio for comparing nucleotide sequences is chosen to be large for highly conserved sequences, while it is small for divergent sequences.

ii) A match M/ mismatch N ratio of 0.5 (1/-2) is best for sequence that are 95% conserved

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

10. There are different scores for identical residues, depending on the degree to which that residues is found to be highly conserved in most proteins. Tryptophan is often found at key positions in proteins where they play a critical role and cannot be substituted easily. Therefore, two aligned tryptophans get a high score (11) whereas two aligned leucines get a much lower score (4), because leucine residues can often be substituted quite readily by other amino acids.

2. Tandem repeats occur in dna when a pattern of one or more nucleotides is repeated and the repetitions are directly adjacent to each other. Given sequences TCGGACACTGACACCACACACAGAGTTA, we have identified Tandem repeat in the dna. By manually checking, CACACCACAC is the CA repeat region.

11. Check 11.png for scoring matrix.