Author: Nivedita Yadav

Comparing Sequence using Dynamic Programming Algorithm

ALGORITHM

- ✖ An algorithm is a sequence of instructions that one must perform in order to solve a well-formulated problem.
- ✖ We will specify problems in terms of their inputs and their outputs, and the algorithm will be the method of translating the inputs into the outputs.
- ✖ In order to solve a problem, some entity needs to carry out the steps specified by the algorithm.

DYNAMIC PROGRAMMING ALGORITHM

- ✖ Bellman pioneered the systematic study of dynamic programming algorithm in the 1950s.
- ✖ In mathematics, computer science, economics, and bioinformatics, dynamic programming algorithm is a method for solving complex problems by breaking them down into simpler sub problems.
- ✖ A dynamic programming algorithm will examine all possible ways to solve the problem and will pick the best solution.

DYNAMIC PROGRAMMING ALGORITHM IN BIOINFORMATICS

- ✖ Dynamic programming algorithm is widely used in bioinformatics for the tasks such as *sequence alignment, sequence comparison, protein folding, RNA structure prediction, nucleosome positioning*, transcription *factor binding* and *protein-DNA binding.*
- ✖ First dynamic programming algorithms for protein-DNA binding were developed in the 1970s independently by Charles Delisi in USA and George Gursk in USSR

DYNAMIC PROGRAMMING ALGORITHM APPLICATIONS

AREAS:

- ➕ Bioinformatics.
- ➕ Control theory.
- ➕ Information theory.
- ➕ Operations research.
- ➕ Computer science: theory, graphics, AI systems etc.
- ➕ Mathematics
- ➕ Economics

SOME FAMOUS DYNAMIC PROGRAMMING ALGORITHMS
- ✚ Viterbi for hidden Markov models.
- ✚ Unix diff for comparing two files.
- ✚ Smith-Waterman for sequence alignment.
- ✚ Bellman-Ford for shortest path routing in networks.
- ✚ CKY for parsing context free grammars.
- ✚ Needleman–Wunsch algorithms for sequence alignment.
- ✚ Recursive least squares method etc.

STEPS TO DESIGNING A DYNAMIC PROGRAMMING ALGORITHM
1. Characterize optimal substructure
2. Recursively define the value of an optimal solution
3. Compute the value bottom up
4. (If needed) Construct an optimal solution

SEQUENCE COMPARISON
- Comparing two (pair-wise alignment) or more multiple sequences by searching for a series of individual characters or patterns that are in the same order in the sequences and the procedure is known as sequence alignment
- There are two types of algorithms for comparing sequences: ***Needleman - wunsch algorithm and smith - watherman algorithm***
- If two sequences have approximately the same length and are quite similar, they are suitable for the global alignment.

COMPARING SEQUENCE USING DYNAMIC PROGRAMMING ALGORITHM
- ✖ The method compares every pair of characters in the two sequences and generates an alignment, which is the best or optimal.
- ✖ Global alignment program is based on Needleman-Wunsch algorithm and local alignment on Smith-Waterman. Both algorithms are derivates from the basic dynamic programming algorithm.
- ✖ The comparison procedure depends upon scoring system, which can be based on probability that
  - **1)** A particular amino acid pair is found in alignments of related proteins (p_{xy});
  - **2)** The same amino acid pair is aligned by chance ($p_x p_y$);
  - **3)** Introduction of a gap would be a better choice as it increases the score.

FORMAL DESCRIPTION OF DYNAMIC PROGRAMMING ALGORITHM
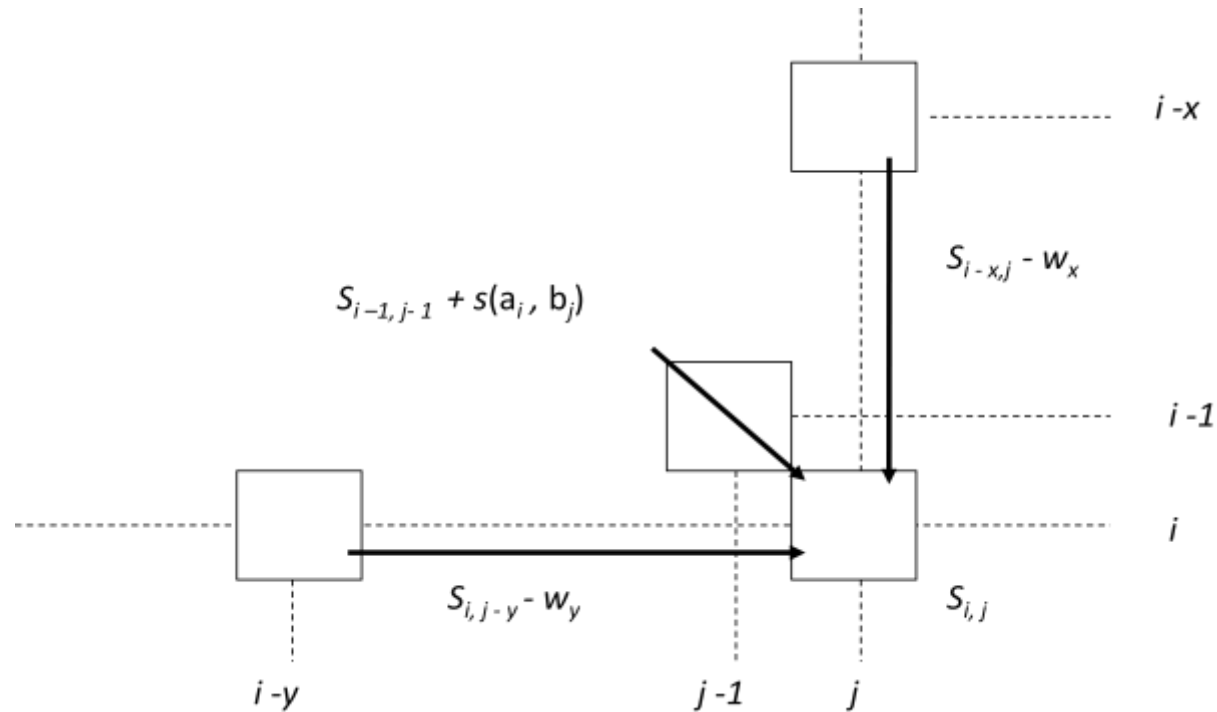
- This diagram indicates the moves that are possible to reach a certain position (*i,j*) starting from the previous row and column at position (*i -1, j-1*) or from any position in the same row or column
- Diagonal move with no gap penalties or move from any other position from column *j* or row *i*, with a gap penalty that depends on the size of the gap

FINDING THE BEST POSSIBLE ALIGNMENT BY COMPARING 2 SEQUENCES IS SOLVED BY THE NEEDLEMAN-WUNSCH ALGORITHM (GLOBAL ALINGMENT)

Suppose following are the two biological sequences which we want to compare for sequence alignment using dynamic programming algorithm

Seq1- **T T T C** (4+1=5)

Seq2- **T G C** (3+1=4)

SCORING SCHEAME:

      1. Reward for match (+1)

      2. Penalizing for mismatch (-1)

      3. Penalizing for gap or indel (-2)

STEP-1: INITILIZATION-Building a table *T* with *m*+1 column & *n*+ 1 row:

	T	G	C
0	**-2**	**-4**	**-6**
-2			
-4			
-6			
-8			

(rows labeled T, T, T, C)

Initializing first row and first column with multiple of gap penalty
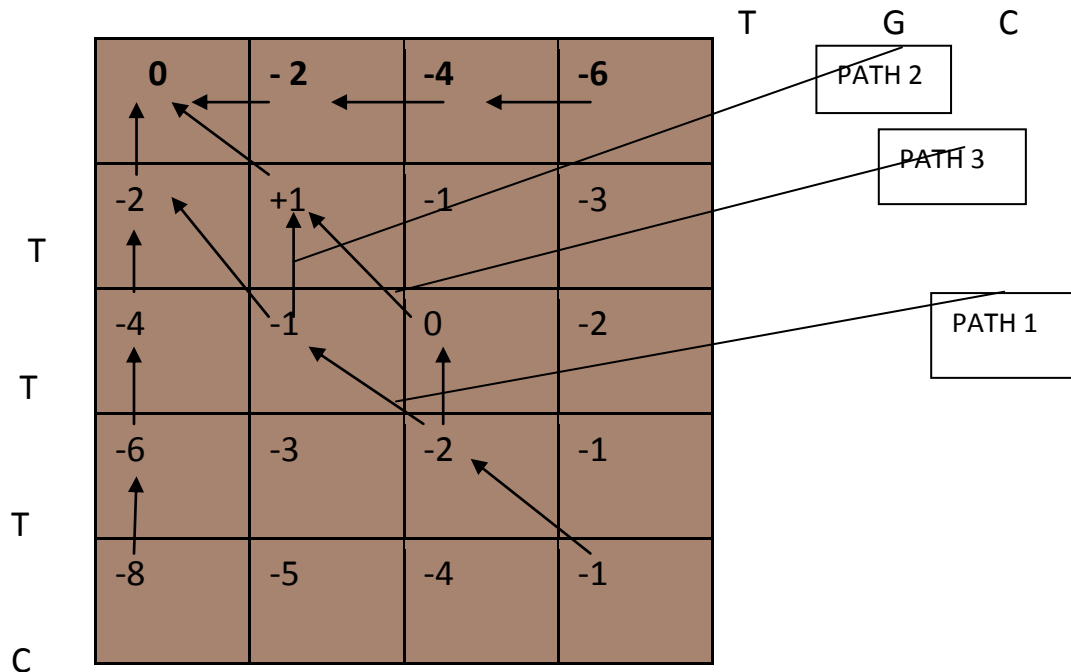
STEP-2: FILL REST OF THE CELL WITH THE Vmax VALUE

	T	G	C
0	**- 2**	**-4**	**-6**
-2	+1	-1	-3
-4	-1	0	-2
-6	-3	-2	-1
-8	-5	-4	-1

(rows labeled T, T, T, C)

USING FORMULA:

$$T(i, j) = \max \begin{cases} T(i\text{-}1, j\text{-}1) + \sigma(S_1(i), S_2(j)) \\ T(i\text{-}1, j) + \text{gap penalty} \\ T(i, j\text{-}1) + \text{gap penalty} \end{cases}$$

STEP 3: GENERATING OPTIMAL PATH FROM THE VMAX VALUE

Back tracking is done from right bottom cell toward the top cell in the table by putting the arrow towards the previous value of Vmax

Step 4: GENERATING OPTIMAL ALIGNMENT FROM THE OPTIMAL PATH
Rule1: if arrow moves in horizontal direction put the gap in the vertical sequence.
Rule 2: if arrow moves in vertical direction put the gap in the horizontal sequence.
Rule 3: If the arrow moves in diagonal direction, align the character between the horizontal direction and vertical direction.

OPTIMAL SOLUTIONS :
PATH 1-

Horizontal seq.	-	**T**	**G**	**C**
Vertical seq**.**	**T**	**T**	**T**	**C**

PATH 2-

Horizontal seq.	**T**	-	**G**	**C**
Vertical seq**.**	**T**	**T**	**T**	**C**

PATH 3-

Horizontal seq.	**T**	**G**	-	**C**
Vertical seq**.**	**T**	**T**	**T**	**C**

CONCLUSION

Hence, comparing two sequences by dynamic programming algorithm is used in alignment of sequence.

References (if any)

1. Introduction to **Bioinformatics, Attwood**. Publisher, Pearson Education, 1999.
2.

<mark>Terms</mark> - Do not remove or change this section   (It should be emailed back to us as is)

• This form is for genuine submissions related to biotechnology topics only.

• You should be the legal owner and author of this article and all its contents.

• If we find that your article is already present online or even containing sections of copied content then we treat as duplicate content - such submissions are quietly rejected.

• If your article is not published within 3-4 days of emailing, then we have not accepted your submission. Our decision is final therefore do not email us enquiring why your article was not published. We will not reply. We reserve all rights on this website.

• Do not violate copyright of others, you will be solely responsible if anyone raises a dispute regarding it.

• Similar to paper based magazines, we do not allow editing of articles once they are published. Therefore please revise and re-revise your article before sending it to us.

• Too short and too long articles are not accepted. Your article must be between 500 and 5000 words.

• We do not charge or pay for any submissions. We do not publish marketing only articles or inappropriate submissions.

• Full submission guidelines are located here: http://www.biotecharticles.com/submitguide.php

• Full Website terms of service are located here: http://www.biotecharticles.com/privacy.php

As I send my article to be published on BiotechArticles.com, I fully agree to all these terms and conditions.