Dhirubhai Ambani Institute of
Information and Communication Technology

SC205 - Discrete Mathematics Project

Sequence Alignment Algorithms

Srushti Makwana - 202201458
Isha Bhanushali - 202201429
Shruti Ranjit Choudhary - 202201020
Dharmi Patel - 202201467
Darpan Lunagariya - 202201462
Ayush Chaudhari - 202201517

Assigned by
Dr. Manish K Gupta

June 22, 2024

Contents

List of Algorithms

Abstract

This project report offers a thorough examination of sequence alignment methods and algorithms, with an emphasis on their efficiency and applicability to various kinds of sequence data. It starts with a brief introduction of sequence alignment. Then it explains pairwise alignment, and further it introduces global and local alignment using Needleman-Wunsch Algorithm and Smith-Waterman Algorithm respectively.

1 Introduction

Sequence alignment is used for comparing and detecting similarities between biological sequences like DNA, RNA, protein etc. Finding the best match or alignment between two or more sequences while accounting for both their similarities and differences is the main objective of sequence alignment. It includes organising the sequences in such a way that the number of matching characters is maximised while taking into account any insertions, deletions, and substitutions (also known as gaps) required to get the optimal alignment. It is a way of arranging biological sequences in order to identify functional, structural or evolutionary relationship between sequences. It can be used for developing phylogenetic trees, identifying sequence similarity, producing homology models of protein structures, predicting higher order structures of proteins and RNAs.

Types of Sequence Alignment

There are main two types of sequence alignment, which are as follows:

1. **Pairwise Alignment** Pairwise alignment involves aligning two sequence at a time.It is generally used to establish the best alignment between two sequences by highlighting similarities and contrasts between them. It is frequently used as the first stage of sequence analysis.
2. **Multiple Sequence Alignment** Multiple sequence alignment involves aligning three or more sequences simultaneously. It is used to align and compare various sequences in order to find conserved sections, motifs, and patterns among all the sequences. It aids in the investigation of evolutionary links, the identification of functional areas, and the evaluation of the structural and functional effects of changes.

However, we will be only discussing pairwise alignment.

2 Pairwise Alignment

We can further categorize pairwise alignment into 2 main types:

- Global Alignment
- Local Alignment

1. **Global Alignment**
   - Global alignment maximize the overall similarity between two sequences by aligning them along their whole length from beginning to end.
   - This type of alignment is appropriate when the sequences being compared are expected to have significant similarity across their entire lengths.
   - Global alignments include gaps at the beginning or end of sequences to achieve the best alignment across their entire lengths.
   - One famous algorithm which is based on dynamic programming for global alignment is Needleman-Wunsch algorithm.
2. **Local Alignment**
   - Local alignment seeks to find regions of similarity or conserved patterns between two sequences, allowing for mismatches or gaps in low-similarity regions.
   - This sort of alignment is appropriate, when the sequences being compared have significant differences in overall length or contain sections with distinct functional or structural value.
   - Local alignments detect the most significant matching regions while ignoring non-matching sequence segments.
   - One famous algorithm for local alignment is Smith-Waterman algorithm which is also based on dynamic programming.

We will be discussing both algorithms of global and local alignment, but first, let us give you an introduction to the mathematical terms and concepts that are necessary to understand the algorithms.

3 Formulating the Mathematics

Let us give you some terminologies which are necessary to understand sequence alignment algorithms:

1. **Alphabet** - finite set of letters.
2. **Sequence** - finite string of letters, each belongs to alphabet.
3. **Null Character** - represented by '-', it signifies absence of letter.
4. **Expanded Sequence** - An expanded sequence S' is a sequence S with arbitrary number of null characters put at the beginning,end or between any two of its characters.

5. **Global Pairwise Alignment** - A global pairwise alignment of sequences S and T is a one-to-one co-linear correspondence of extended sequences S' and T' that contains no nulls from S' and T'.

   Example:- Global alignment of protein sequences VHLTDSEKTAVTALFG and VALTMKYEALVSDSLIAF is

   VHLT-DSEKTAVTAL-FG
   VALTMKYEALVSDSLIAF

6. **Local Pairwise Alignment** - A local pairwise alignment of sequences S and T is a one-to-one co-linear correspondence of equal length segments of expanded sequences S' and T', in such a way that no nulls from the segments correspond.

   Example:- One possible Local alignment of protein sequences VHLTDSEKTAVTALFG and VALTMKYEALVSDSLIAF is

   LT
   LT

7. **Column of Pairwise Alignment** - One-to-one correspondence of a single letter (or null character) in one sequence with a single letter (or null character) in the other.

8. **Substitution** - Column that aligns two letters which belongs to alphabet.
   Substitution Score - Score defined for substitution.

9. **Indel** - Column which aligns the letter that belong to alphabet with null.
   Indel Score - Score defined for indel.

10. **Alignment Score** - Addition of indel score and substitution score.

11. **Optimal Alignment** - Alignment with maximum alignment score.

12. **Path Graph** - Assume that we have two sequences with length x and y respectively. It is a $(x+1)*(y+1)$ rectangular array of nodes, with directed horizontal, vertical or diagonal edges between adjacent nodes.

After these terminologies, let us discuss widely used global and local sequence alignment algorithms.

4 Needleman-Wunsch Algorithm

It is a dynamic programming algorithm used for global alignment.The dynamic programming method used by the Needleman-Wunsch algorithm divides the alignment problem into smaller subproblems and solves each one iteratively. It uses a scoring system to give values to sequence gaps, matches, and mismatches.

Algorithm

```
Input: Two sequences of length $x$ and $y$ respectively.
Output: Optimal global alignment score and optimal global alignments.
1. Create the corresponding path graph G with $(x+1)*(y+1)$ nodes,
   indexed from 0 to $x$ and 0 to $y$.
2. Initialise upper left node of G with 0.
3. Initialise the rest of first column and first row nodes with indel score.
4. For every reamining node at location $(i, j)$, compute
   $V_{score} \leftarrow G(i-1, j) + score\ of\ edge\ from\ (i-1, j)\ to\ (i, j)$
   $H_{score} \leftarrow G(i, j-1) + score\ of\ edge\ from\ (i, j-1)\ to\ (i, j)$
   $D_{score} \leftarrow G(i-1, j-1) + score\ of\ edge\ from\ (i-1, j-1)\ to\ (i, j)$
   $G(i, j) \leftarrow max\ \{V_{score},\ H_{score},\ D_{score}\}$
5. Mark en edge to the node from where we are getting this maximum value.
6. The value at node $(x, y)$ is optimal global alignment score.
7. Starting from top-left corner, head to bottom-right corner by following
   these edges and retrive optimal global alignment.
```

Algorithm 1: Needleman-Wunsch Algortihm

$G(x, y)$ shows the optimal global alignment scores and marked edges can be utilized to get aligned sequences achieving this maximum score. Time and space complexity of this algorithm is $O(x * y)$.

In short, we can say that this algorithm works on three steps given below:

1. Initialisation of the path graph.
2. Filling values in node of the path graph.
3. Traceback to identify the optimal global aligned sequence.

Example

Let us take one example to better understand Needleman-Wunsch algorithm. Suppose we want to align these 2 sequences: S_1 = ATCT, S_2 = GTATAC.

Step 1

Length of sequence $S1$, $x = 4$.
Length of sequence $S2$, $y = 6$.
Initialise a path graph of $(x+1)*(y+1) = 5*7 = 35$ nodes.

Scoring system for this example is as follow:

Match score = 1
Mismatch score = -1
Indel score = -2

	S_2	G	T	A	T	A	C
S_1							
A							
T							
C							
T							

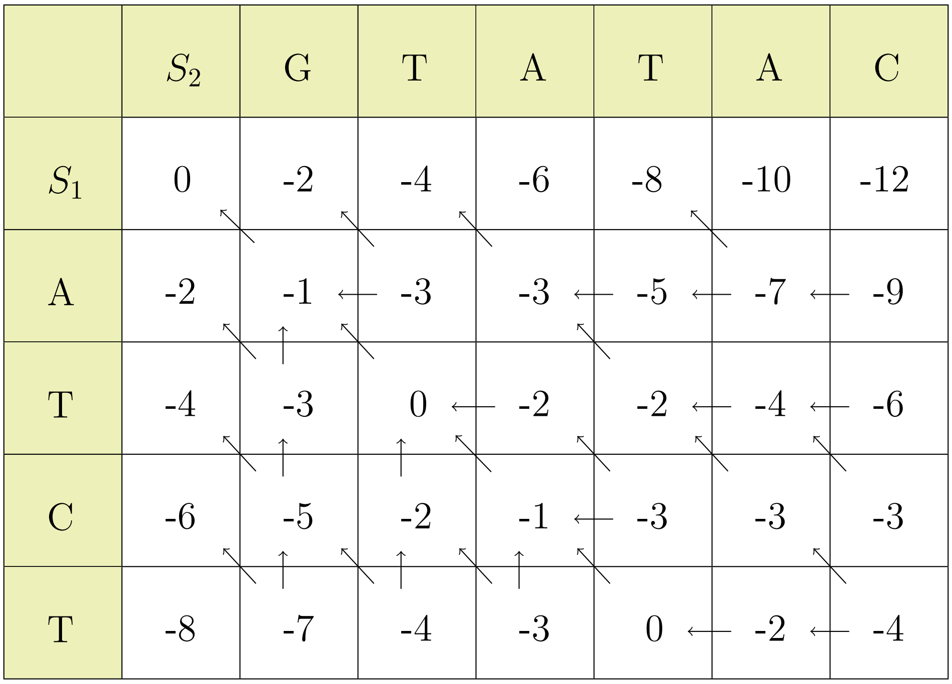
Step 2

Initialize top let node to zero and other nodes of first row and column with indel score.

	S_2	G	T	A	T	A	C
S_1	0	-2	-4	-6	-8	-10	-12
A	-2						
T	-4						
C	-6						
T	-8						

Step 3

The procedure begins at the upper left corner of the matrix and moves one row at a time towards the lower right corner to determine the alignment scores. By aligning the matching residues, the algorithm fills each cell in the matrix with the highest score possible. Also, in this procedure, we have to mark an edge to the side from which we are achieving this maximum value.

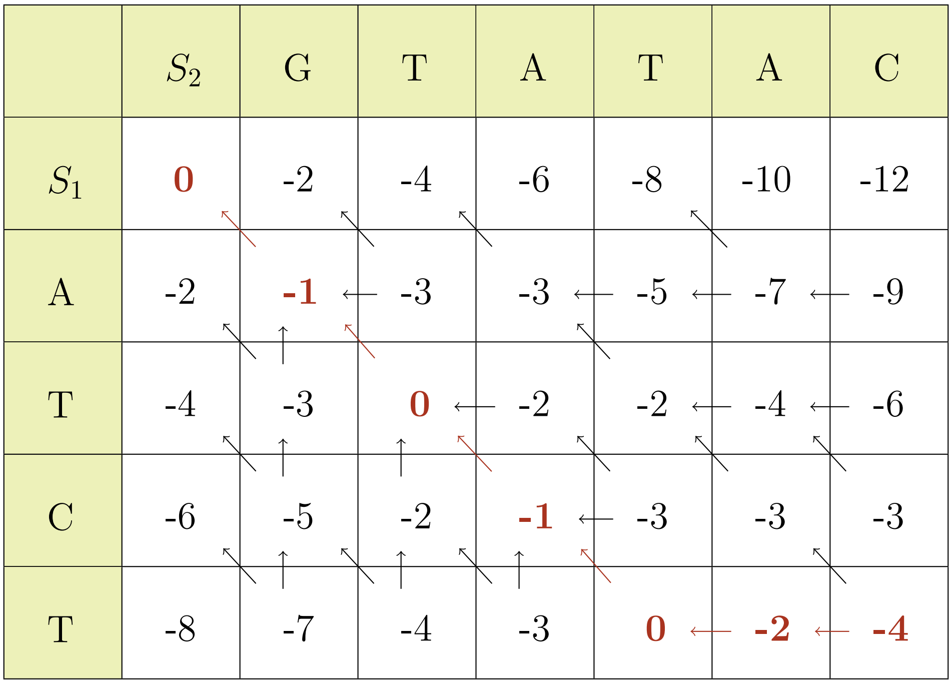


	S_2	G	T	A	T	A	C
S_1	0	-2	-4	-6	-8	-10	-12
A	-2	↖ -1	↖ ← -3	↖ -3	← -5	↖ ← -7	← -9
T	-4	↖ ↑ -3	↖ 0	← -2	↖ -2	← -4	← -6
C	-6	↖ ↑ -5	↑ -2	↖ -1	↖ ← -3	↖ -3	↖ -3
T	-8	↖ ↑ -7	↖ ↑ -4	↖ ↑ -3	↖ 0	← -2	↖ ← -4

The value on the node (x, y) represents the optimal global alignment score. Here, the optimal global alignment score is -4.

Step 4

After filling the path graph, we can perform a traceback to find the optimal alignment path. Starting from bottom-right node and moving towards top-left node, adjacent cells are examined in reverse order to determine the best path with the maximum total score.



	S_2	G	T	A	T	A	C
S_1	**0**	-2	-4	-6	-8	-10	-12
A	-2	**-1**	← -3	-3	← -5	← -7	← -9
T	-4	-3	**0**	← -2	-2	← -4	← -6
C	-6	-5	-2	**-1**	← -3	-3	-3
T	-8	-7	-4	-3	**0**	← **-2**	← **-4**

Starting from the top-left node, follow the arrows which are represented by different color to bottom-right node, to get the aligned sequences.

In this exapmle, optimal alignment score is -4 and aligned sequences are:

S_1 = ATCT−−
S_2 = GTATAC

This algorithm considered all possible alignments, calculated scores based on matches, mismatches and gaps, and identified alignment with highest score.

5 Smith-Waterman Algorithm

Smith-Waterman algorithm is a dynamic programming algortihm used for local sequence alignment. It is very similar to Needleman-Wunsch Algorithm of global sequence alignment.But unlike the Needleman-Wunsch algorithm, which aims to align whole length of sequences, this algorithm focuses on finding best alignment within local region.

Many biological sequences are not related on their entire length, but they are related across subsequences. This algorithm helps us to find this type of relation between two biological sequences.

Algorithm

We can make these modifications in Needleman-Wunsch algorithm, to get Smith-Waterman algorithm:

1. If any of the node have negative score, update it zero. Also, allow a path to start from any node which have score 0, not just top-left node.
2. Record the node which have maximum score and start tracing back from that node.
3. Terminate the traceback when the node with score zero is reached.

```
Input: Two sequences of length $x$ and $y$ respectively.
Output: Optimal local alignment score and optimal local alignments.
1. Create the corresponding path graph G with $(x+1)*(y+1)$ nodes, indexed from 0 to $x$ and 0 to $y$.
2. Initialise nodes of first row and columns to zero.
3. To find the score of node $(i, j)$, compute
    $V_{score} \leftarrow G(i-1, j) + score\ of\ edge\ from\ (i-1, j)\ to\ (i, j)$
    $H_{score} \leftarrow G(i, j-1) + score\ of\ edge\ from\ (i, j-1)\ to\ (i, j)$
    $D_{score} \leftarrow G(i-1, j-1) + score\ of\ edge\ from\ (i-1, j-1)\ to\ (i, j)$
    $G(i, j) \leftarrow max\ \{0,\ V_{score},\ H_{score},\ D_{score}\}$
4. If the score is greater than zero, mark an edge to the node from where we are achieving this maximum value.
5. Record the node with maximum score. It is optimal local alignment score.
6. Start tracing back from the node which have maximum score to get optimal local alignment.
```

Algorithm 2: Smith-Waterman Algortihm

Time and space complexity of this algorithm is $O(x * y)$. Smith-Waterman algorithm's dynamic programming approach and traceback mechanism allows to consider all possible alignments and select best local alignment based on the scores.

In short, this algorithm works on the steps below:

1. Initialization of the path graph.
2. Filling scores in node of path graph.
3. Tracing back from the highest score node to the node which have 0 score, to find the optimal local aligned sequence.

Example

Let us take an example to understand Smith-Waterman algorithm. We will find local alignment of previously discussed sequences,

S_1 = ATCT and S_2 = GTATAC

Length of sequence S_1, $x = 4$.
Length of sequence S_2, $y = 6$.

The scoring mechanism is as follow:

Match Score = 1
Mismatch Score = -1
Indel Score = -2

Step 1

Initialize a path graph of $(x + 1) * (y + 1) = 5 * 7 = 35$ nodes.

	S_2	G	T	A	T	A	C
S_1							
A							
T							
C							
T							

Step 2

Initialize the nodes of first row and columns with score zero.

	S_2	G	T	A	T	A	C
S_1	0	0	0	0	0	0	0
A	0						
T	0						
C	0						
T	0						

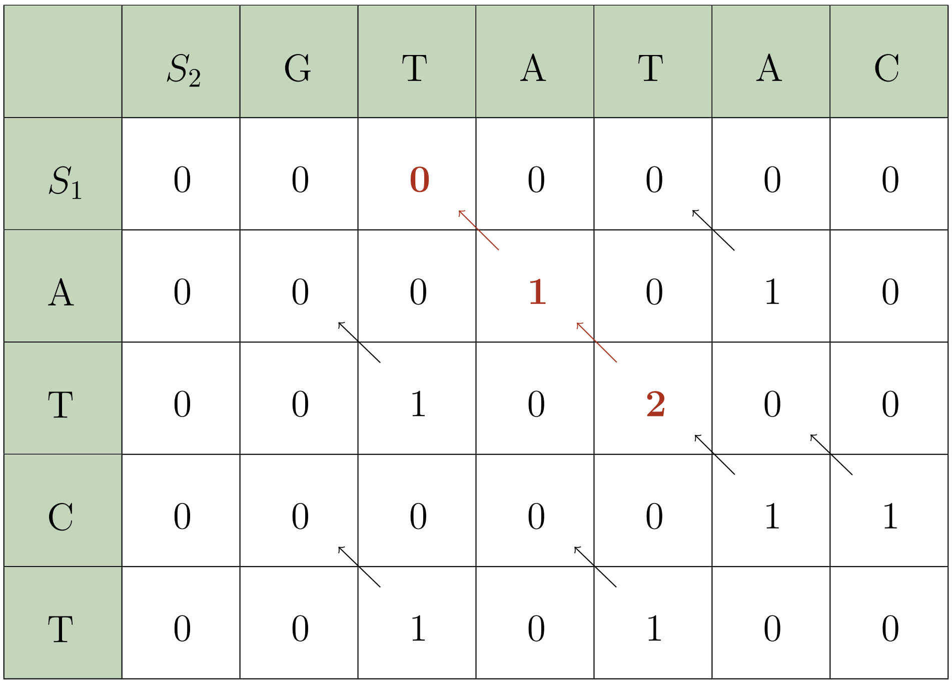
Step 3

Fill scores into remaining nodes by the procedure given in the algorithm. Also, if the score is grater than zero, mark an edge to the node from where we are getting the maximum score.

	S_2	G	T	A	T	A	C
S_1	0	0	0	0	0	0	0
A	0	0	0	↖ 1	0	↖ 1	0
T	0	0	↖ 1	0	↖ 2	0	0
C	0	0	0	0	0	↖ 1	↖ 1
T	0	0	↖ 1	0	↖ 1	0	0

Step 4

Determine the node with highest score and start tracing back from that node unless you reach the node with score 0.



	S_2	G	T	A	T	A	C
S_1	0	0	**0**	0	0	0	0
A	0	0	0	**1**	0	1	0
T	0	0	1	0	**2**	0	0
C	0	0	0	0	0	1	1
T	0	0	1	0	1	0	0

Maximum score in the path graph is called optimal local alignment score. In this example, optimal local alignment score is 2.

To retrieve the locally aligned sequence, follow the coloured path from the node with zero score to node with maximum score. Locally aligned sequences are as follow:

S_1 = AT
S_2 = AT

This algorithm checks all the possibilities, and identifies local aligned sequence with highest local alignment score.

6 Code

We have written code of Global Alignmment using Needleman-Wuncsh Algorithm and Local Alignment using Smith-Waterman Algorithm in C++ language. You can check the code by clicking here.
To know how to run this code, click here.

7 Commercialization

We can make a user-friendly website application to perform pairwise alignment. We can develope the functionality which can accept sequence input from the user. These input sequences can be aligned using the algorithms which we have discussed above. Also, we can include path graphs and directed edges between adjacent nodes of path graphs. We can include the functionality to choose custom paths and calculate the alignment score respective to the custom path.

References

[1] National Center for Biotechnology. Sequence alignment, chapter 20.1, handbook of discrete and combinatorial mathematics. 2nd edition. https://www.ncbi.nlm.nih.gov/books/NBK464187/.

[2] The Biology Notes. Sequence alignment- definition, types, methods, uses. https://thebiologynotes.com/local-global-multiple-sequence-alignment/.

[3] Science-Direct. Sequence alignment. https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/sequence-alignment/.

[4] McGill University. Sequence alignment. https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/s/Sequence_alignment.htm.

[5] Amrita Vishwa Vidyapeetham. Global alignment of two sequences - needleman-wunsch algorithm. https://vlab.amrita.edu/?sub=3&brch=274&sim=1431&cnt=1.

[6] Wikipedia. Needleman-wunsch algorithm. https://en.wikipedia.org/wiki/Needleman-Wunsch_algorithm.

[7] Wikipedia. Smith-waterman algorithm. https://en.wikipedia.org/wiki/Smith-Waterman_algorithm.