Understanding pairwise alignment algorithms for DNA sequencing from a biologists perspective

Harsh Thakkar, Prasenit Majumder

November 30, 2014

1 Introduction

1.1 Biology meets Computer Science

The nature of problems found in biology are analogous to those encountered in computer science discipline. Rather, one can say that computer science is a motivated from biology. For instance, Information is digitally encoded using binary encoding scheme, i.e. o and 1 (two bit scheme). Similarly every organism found in nature, in its root, is encoded using DNA (encoding scheme used by nature is of 4 bits say A-T-C-G).

DNA can be viewed as enormously long repetitive strings of a variety of combinations derived from A-T-C-G. To understand the process of human evolution, there is a need to understand the biological and physical phenomena governing these variations in DNA of various organisms. This study of tracing the root/origin of a particular species or living organisms in general is called **Phylogeny**. This study emerged as a result of substantial efforts put forward by the very famous Human Genome project[citation needed] which was started in 1990 with a goal to study the processes such as: human evolution, cause of diseases and how they spread.

Phylogenetics can be viewed as an interdisciplinary effort to understand and develop the field of phylogeny. This consists of developing various algorithms to automate the process of identifying matching and analyzing the variances/mutations found in the basic building blocks of organisms aka DNA. Mutations in biology are termed as variants or alternate sequence or more commonly subsequences derived from a parent or ancestor sequence.

1.1.1 Motivation

A simple strand or piece of DNA consists of a very long string of alternating sequences of A-T-C-G. in order to have an deeper insight of the natural biological phenomena, we need to be able to ”mimic” the same functionality found in proteins or DNA. It is not possible to manually analyze this gigantic amount of data. For instance, Homo sapiens hemoglobin, beta (HBB), mRNA strand obtained from the NCBI website[1] is illustrated below:

ACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGACACCATGGTGCATCTGACTCCTGA
GGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGTTGG
TATCAAGGTTACAAGACAGGTTTAAGGAGACCAATAGAAACTGGGCATGTGGAGACAGAGAAGACTCTTGGGTTTCT
GATAGGCACTGACTCTCTCTGCCTATTGGTCTATTTTCCCACCCTTAGGCTGCTGGTGGTCTACCCTTGGACCCAGA
GGTTCTTTGAGTCCTTTGGGGATCTGTCCACTCCTGATGCTGTTATGGGCAACCCTAAGGTGAAGGCTCATGGCAAG
AAAGTGCTCGGTGCCTTTAGTGATGGCCTGGCTCACCTGGACAACCTCAAGGGCACCTTTGCCACACTGAGTGAGCT
GCACTGTGACAAGCTGCACGTGGATCCTGAGAACTTCAGGGTGAGTCTATGGGACGCTTGATGTTTTCTTTCCCCTT
CTTTTCTATGGTTAAGTTCATGTCATAGGAAGGGGATAAGTAACAGGGTACAGTTTAGAATGGGAAACAGACGAATG
ATTGCATCAGTGTGGAAGTCTCAGGATCGTTTTAGTTTCTTTTATTTGCTGTTCATAACAATTGTTTTCTTTTGTTTA
ATTCTTGCTTTCTTTTTTTTTCTTCTCCGCAATTTTTACTATTATACTTAATGCCTTAACATTGTGTATAACAAAAGG
AAATATCTCTGAGATACATTAAGTAACTTAAAAAAAAACTTTACACAGTCTGCCTAGTACATTACTATTTGGAATAT
ATGTGTGCTTATTTGCATATTCATAATCTCCCTACTTTATTTTCTTTTATTTTTAATTGATACATAATCATTATACA
TATTTATGGGTTAAAGTGTAATGTTTTAATATGTGTACACATATTGACCAAATCAGGGTAATTTTGCATTTGTAATT
TTAAAAAATGCTTTCTTCTTTTAATATACTTTTTTGTTTATCTTATTTCTAATACTTTCCCTAATCTCTTTCTTTCA
GGGCAATAATGATACAATGTATCATGCCTCTTTGCACCATTCTAAAGAATAACAGTGATAATTTCTGGGTTAAGGCA
ATAGCAATATCTCTGCATATAAATATTTCTGCATATAAATTGTAACTGATGTAAGAGGTTTCATATTGCTAATAGCA
GCTACAATCCAGCTACCATTCTGCTTTTATTTTATGGTTGGGATAAGGCTGGATTATTCTGAGTCCAAGCTAGGCCC
TTTTGCTAATCATGTTCATACCTCTTATCTTCCTCCCACAGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCAT
CACTTTGGCAAAGAATTCACCCCACCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGCCCTGGC
CCACAAGTATCACTAAGCTCGCTTTCTTGCTGTCCAATTTCTATTAAAGGTTCCTTTGTTCCCTAAGTCCAACTACT
AAACTGGGGGATATTATGAAGGGCCTTGAGCATCTGGATTCTGCCTAATAAAAAACATTTATTTTCATTGC

This strand is typically of length 1606 characters. Other than just for the process of matching such huge sequences of nucleotide strands, there are various motivations such as: constructing sequences from other smaller overlapping subsequences, searching the genome database from possible matches (strict match or other-wise partial matches), comparing multiple segments of nucleotides for similarities, mining very frequent patterns, and other statistical inferences that lead to overall knowledge from just this long repetitive data. Thus, there is where computer scientists meets biology, since all this tedious work cannot be done manually!

1.2 Popular DNA sequencing methods

In general we can classify the sequence alignment methods in two broad categories: **Pairwise alignment** and **Multiple Sequence Alignment (MSA)**
In **Pairwise alignment**, we have the following sequencing methods:

- **Sanger method** It is a chain termination method also known as dideoxy method. It is best suitable for sequencing small fragment of DNA.

[1]Homo sapiens hemoglobin mRNA from NCBI website http://www.ncbi.nlm.nih.gov/nuccore/NC_000011.10?report=fasta&from=5225466&to=5227071&strand=true online

- **Maxam Gilbert method** Its out-dated method, no longer used for DNA sequencing. This method a bio-chemical process involving a chain of chemical reactions to be performed on DNA segments.

In **Multiple Sequence Alignment (MSA)**, we have the following sequencing methods:

- **Shotgun sequence method** This sequencing method is used to sequence the whole genome, rather than small fragments of DNA as in Sanger method (pairwise). It is capable of sequencing large pieces of DNA.
- **Pyrosequencing method** (2^{nd} generation method) This a synthesis based DNA sequencing method. It is accurate and fast as compared to other methods.
- **Illumina sequencing** (Next generation sequencing method): This method uses a series of micro-array of reversible dye terminators which recognize the base pairs in DNA strands. It is fast, and allows sequencing of multiple strands simultaneously.

In this report we focus to study the pairwise alignment algorithms in detail.

1.3 Understanding the basic terminology

Before we begin, first let us first get familiar with the terminology used in the complicated process of this method.

- **Sequence**: Sequence is **string** or **word** of alternating characters in this case the DNA sequence (or the query string **Q**)
- **Segment**: Is a part or a piece of the DNA sequence which is to be matched or aligned to sequences in the genome database. In pairwise alignment we generate various combinations of subsequences from the input query (Q).
- **Alignment**: Is a process of ordering the of characters in the string to find the maximum common characters between them.
- **Hit or match**: If the two characters are same, we call it a hit or match if they are different its a miss-match. For instance, consider the two strings {ATCGA}, {CGCGAT} call them S1 and S2 respectively. Here, the three characters C-G-A of both the strings match, thus we have 3 hits while initial two miss matches and one (last character) is considered as a gap.
  A-T-C-G-A-
  C-G-C-G-A-T
- **Gap**: A gap is the absence of a character or it can be considered as empty space in the string. Gaps generally occur when the two strings are not of the same length. It can be denoted by a space or - (dash)
- High scoring-segment Pairs (HSP):

2 Pairwise alignment vs Multiple Sequence alignment

Pairwise alignment is used to detect homologies between different protein or DNA sequences, either as global or local alignments. This can be solved using dynamic programming in time proportional to the product of the lengths of the two sequences being compared. However, this is too slow for searching current databases and in practice algorithms are used that run much faster, at the expense of possibly missing some significant hits due to the heuristics employed. Such algorithms are usually on seed and extend approaches in which first small exact matches are found, which are then extended to obtain long inexact ones.

2.1 Types of alignments and their significance

2.1.1 Local alignment vs Global alignment

Local alignments focuses on the maximum number of common subsequences or substrings generated by the two input string S1 and S1. While global alignment focuses on the overall segment of DNA, rather the finding the maximum number of exact matches from its subsequences. It searches for the best alignment or the best matching string for the input against the genome database. Global alignment typically aligns the entire DNA sequence which will miss out the opportunity of finding the high matching subsequences.

There are many applications where the two sequences may not be very similar as-is, but consist of high number of matching subsequences. The task is to find, extract and align a pair of subsequences, one from each of the two given sequences, that exhibit the highest similarity. For instance, Proteins of different kind and of different species, often exhibit local similarities called homeoboxes. These represent the information of the functioning of the sup-parts of proteins. To target this problem we use local alignment between different sequences of DNA.

We will see the illustration of both the alignment schemes and their difference in the later section.

3 Algorithms for pairwise alignment

Blast-N[1] and FASTA[3] are an example of 1^{st} generation pairwise alignment algorithms. These algorithms have been developed to target the the sensitive issue of searching huge datasets of genome data with relatively shorter queries to identify remote homologies. This homology matching of a gene function is typically used to derive the hypothesis about the function of the sequence of query. These algorithms have to satisfy two fundamental requirements of the nature of problem: the search should be rapid and sensitive to matches in enormous datasets. In order to speed-up the process of searching, these algorithms incorporate preprocessing of the query. global vs local alignment methods!

3.1 BLAST with local alignment

BLAST stands for *Basic Local Alignment Search Tool*. Blast-n (where n is the word-span, discussed later) is an advanced version of FASTA algorithm, which is specifically used for DNA sequence matching. It is faster, accurate as it doesn't perform an exhaustive search of entire subsequence patterns of all of the queried dna segment, like in Smith-Waterman algorithm. In other words, blast-n algorithm is customized to return match results for segments which are above a minimum threshold level as input by a user. There are many variants [2] of this algorithm namely, gapped BLAST, blast-p, blast-x and etc. These are used for specific nature of query searches. For instance, the blast-p algorithm searches for matching patterns from the query of amino acid sequence against the protein sequence database. BLAST-N is built on an assumption that small fractions or segments of better matching pairs are superior than long segments with low matching pairs. In this section we will see the functioning of the blast-n algorithm in detail.

BLAST makes use of the SMITH-WATERMAN[citation needed] matrix to score the sequences DNA strands and BLOSUM[cite here] or PAM[cite here] based scheme for matching of protein amino acids mutations with the genome sequence database respectively. BLOSUM is typically a 20x20 substitution matrix. An illustration of how this algorithm works is discussed in detail later in this section.

The overall functioning of the BLAST-N algorithm are shown in the figure below:

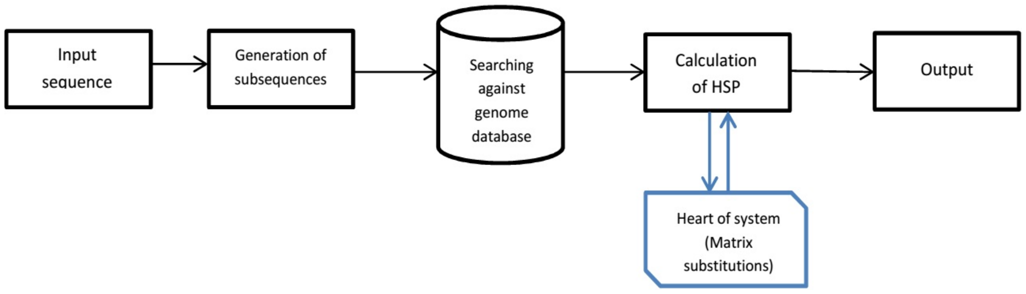

Input sequence
Generation of subsequences
Searching against genome database
Calculation of HSP
Output
Heart of system (Matrix substitutions)

Figure 1: The figure represents a systematical block diagram of the working of BLAST algorithm for DNA sequence alignment and matching.

3.1.1 Matrix generation and population

One might think why do we use matrices to store/calculate information from DNA data? Well, this could be explained by the answer which is the question itself! As we know, DNA is an extensively long sequence of repetitive genetic information (A-T-C-G). This information is stored in a very beautiful way as a double helix structure in the nucleotide, thus, this information is naturally folded with very high degree of compactness. In a similar manner, taking inspiration from nature, in computer science we use matrices. DNA is unfolded, turn out to be a few kilometers long!. It is not possible to match this extremely long information due to technical and infrastructural limitation. Thus, a matrix can be understood as two dimensional fold, m number of folds for a string of length n. Thus, the matrix is of size mxn. We show the working of the heart of the algorithm, i.e. the smith-waterman matrix, in the figures later in this section.

Smith-Waterman matrix[cite here] is used for calculating the score of matching DNA subsequences in BLAST. The subsequences returning the highest scores are referred to as *High Scoring-segment Pairs* (HSP). There are some rules for populating the matrix and for calculating the scores, which are described below:

A matrix is created as shown below, such that it has an additional row and column with respect to the lengths of the sequences to be matched. S1= {TCG} and S2= {ATCG}, thus a matrix of 4x5 or 5x4 (any can be used) is created. An extra column and row is added to place the Gap (space) which denoted the start of the process. Since in local alignment, we consider segments of the whole DNA sequence, we start with gap since these are not the actual starting and ending segments. More segments can be aligned later as the process continues.

Scoring scheme: We need a scoring scheme to tell us which subsequences are doing well and which are not. Just as in an exam, answering a question correctly fetches you marks, wrong answer means deduction in marks and an un-attempted question will be awarded 0 or which ever score is decided previously. Let us consider the following scoring scheme:
Match = **+1**, Miss-match= **-1**, Gap= **-2**.

Rules for populating scores in smith-waterman matrix:

1. For each entry we can have values from three directions. i.e. the value of a cell will be dependent on the cells which are (left, below and bottom-left diagonal). *We will use arrows to depict from which cell we obtained the value.*

2. We choose only the maximum value from all the values obtained from the three cells. The formula is:

   (a) The value coming from the cell below is calculated by subtracting the gap score from the value of that cell and then add it to the value in the current cell.

   (b) The value coming of the bottom-left diagonal is calculated as:

   - if it is a match or a hit: we add the Match score to the value of the current cell.
   - If it is a miss we simply subtract the miss-match score.

(c) **The most important rule for LOCAL ALIGNMENT IS: All the negative values are represented by 0. If the value of any cell is calculated to be less than zero, we enter a zero.**

3. To obtain the common subsequences, we use the **traceback** method. This is main reason of putting arrows in the matrix. the arrows will guide us back to the origin of the process, and thus we will obtain the matching sequences. We start from the Largest value in the matrix. It can be anywhere from the last column, or last row. This is because we are progressing upwards, so the values are added while going upwards filling the matrix. The more the matches, the more positive values we obtain.

   - If the value of the current cell is obtained from a diagonal cell then we write both the characters.
   - If the value of the current cell is obtained from either bottom or left cell, then we write a gap for the character towards the sequence which the arrow is pointing and keep the pther character as it is.

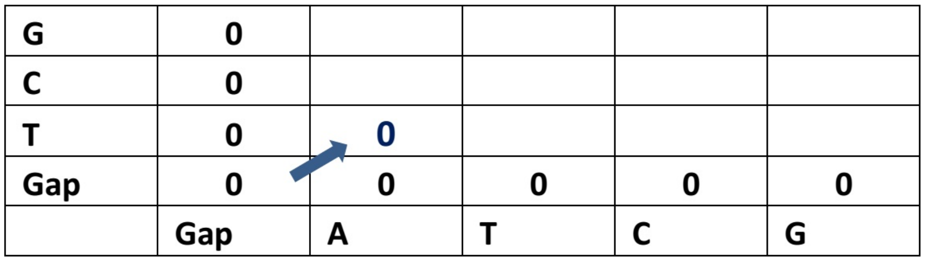
Let us start with the above example of two sequences. Consider the sequences S1= {TCG} and S2= {ATCG}. We demonstrate a stepwise processing of matrix values based on the above rules.

G					
C					
T					
Gap					
	Gap	**A**	**T**	**C**	**G**

Figure 2: The figure represents the initial construction of the smith-water matrix for calculating the scores of subsequences.

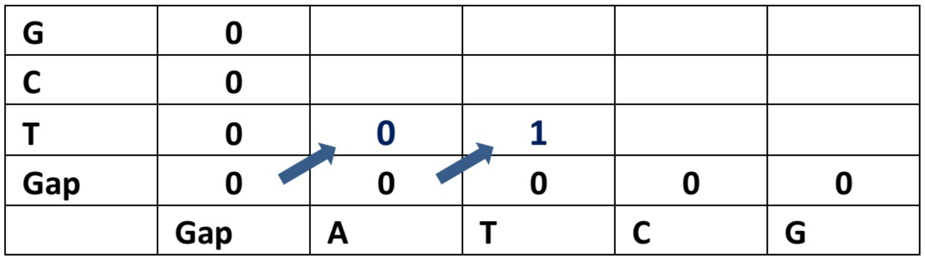
G	**0**				
C	**0**				
T	**0**				
Gap	**0**	**0**	**0**	**0**	**0**
	Gap	**A**	**T**	**C**	**G**

Figure 3: The figure represents the population of matrix with zero's in the first column and rows of the sequences.



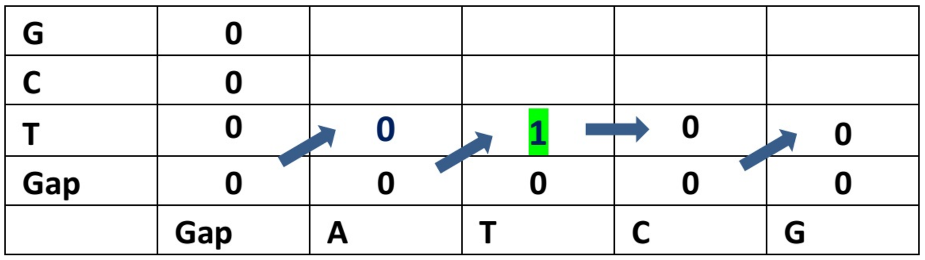
G	0				
C	0				
T	0	0			
Gap	0	0	0	0	0
	Gap	A	T	C	G

Figure 4: The figure represents the choice of the cell from which the value of the current cell in governed i.e. rule 2. Here the values of coming from the adjacent cells for characters A and T will be maximum of (-2,-2 and -1)= -1, for the current cell. Since, we represent negative values as zero (rule 2c), we put a zero instead of -1.



G	0				
C	0				
T	0	0	1		
Gap	0	0	0	0	0
	Gap	A	T	C	G

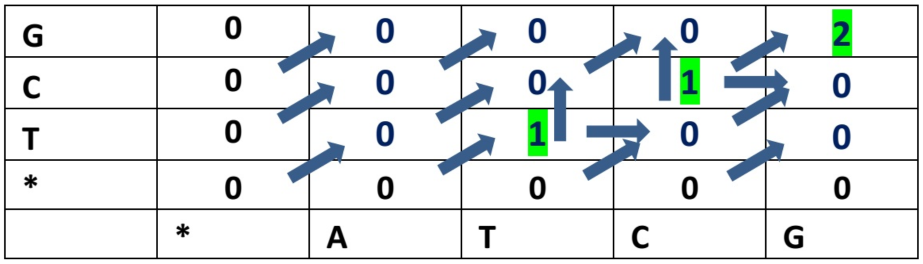
Figure 5: The figure represents the next iteration of the process. We fill every cell according to the matrix population rules with arrows denoting the direction of the cell from which it is obtained.



G	0				
C	0				
T	0	0	1	0	0
Gap	0	0	0	0	0
	Gap	A	T	C	G

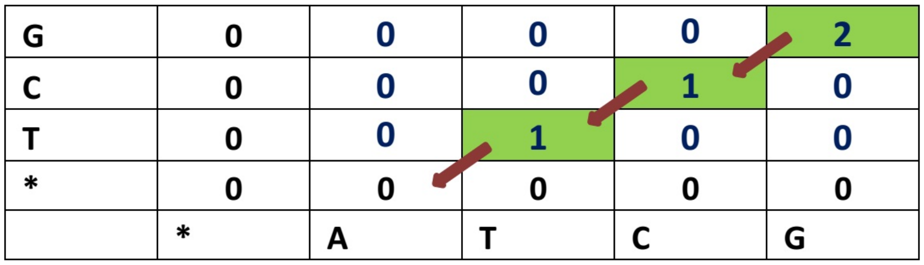
Figure 6: In this figure we represent a green color on 1, emphasizing the Match or a Hit. Thus, here we follow the rule 2(b) to calculate the value of current cell. Since, T-T is a match we add the value of the bottom-left diagonal cell with the hit score (+1). Thus we have, 0 + (+1)=1.

Thus, we get the alignment of the sequence pair as follows:
S1= *–T-C-G
S2= A-T-C-G



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

Figure 7: The figure represents the fully populated matrix.



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

Figure 8: The **Traceback** step [rule 3], this figure represents the process of tracbacking with the help of arrows which were marked during the process of populating the matrix. We find the highest element of the matrix and then trace the cell from which its value was governed. Doing this in a repetitive manner we reach zero or the start. All the values of the traceback are written using the traceback rules. Hence, the matching segments are obtained.

The HSP of this subsequence is the value of the maximum element of the matrix. Thus the HSP of S1,S2 pair is 2. This can be confirmed by calculating the total of the number of matches, miss-matches and gaps in the alignment. Here, form the above alignment we can see that there are 3 matches and one Gap, i.e. 3x(+1) + 1*(-2) = **+2**.

3.2 BLAST with global alignment

As discussed earlier in section 2.1.1, the prime difference between these two alignments is that in local the many subsequences generated from the input DNA sequence are aligned with all possible subsequences from the genome database. While in global the whole sequence (input) is aligned with the possible choices without generating subsequences.

Other than this, there is a slight change in the scoring scheme. We start the initial population of the matrix by adding gap scores (figure 8). Also, we do not represent the negative numbers by zero. The score which are obtained in the matrix are populated as is. We demonstrate the same example as in earlier section using a global align method as shown below.

G					
C					
T					
Gap					
	Gap	**A**	**T**	**C**	**G**

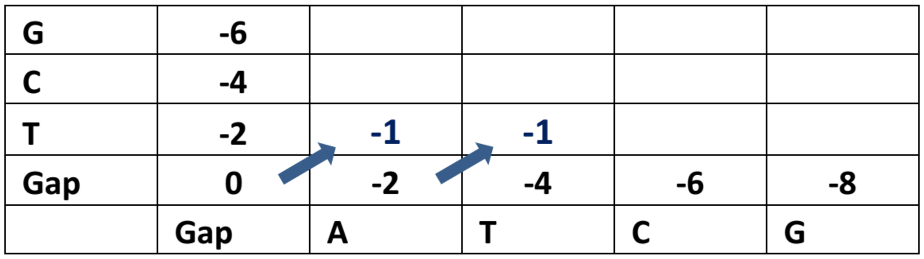
Figure 9: The figure represents matrix generation. The generation rules are the same as in local alignment studied earlier.

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

Figure 10: The figure represents the initial state of matrix in a global alignment algorithm.

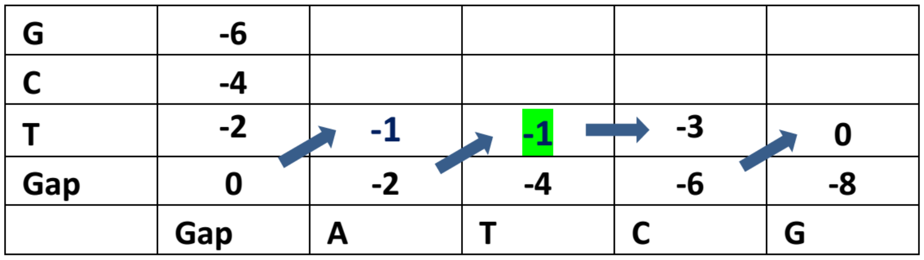
G	**-6**				
C	**-4**				
T	**-2**	**-1**			
Gap	**0**	**-2**	**-4**	**-6**	**-8**
	Gap	**A**	**T**	**C**	**G**

Figure 11: The figure represents the first iteration of the algorithm, same as in local alignment.



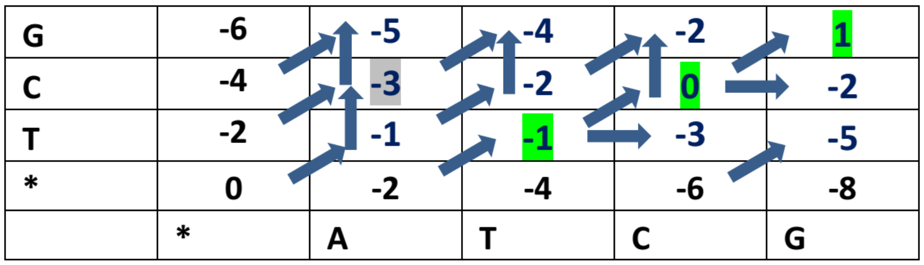
G	-6				
C	-4				
T	-2	-1	-1		
Gap	0	-2	-4	-6	-8
	Gap	A	T	C	G

Figure 12: The figure represents the next iteration of the process. We fill every cell according to the matrix population rules with arrows denoting the direction of the cell from which it is obtained.



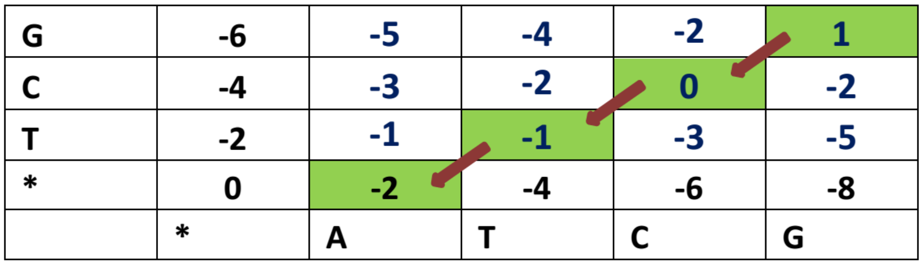
G	-6				
C	-4				
T	-2	-1	-1	-3	0
Gap	0	-2	-4	-6	-8
	Gap	A	T	C	G

Figure 13: The figure represents the first iteration of the algorithm. We now fill the second row of the matrix.



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

Figure 14: In this figure we represent a completely populated matrix using global alignment.



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

Figure 15: This figure represents the traceback procedure of the filled matrix which is similar to the one in local alignment.

References

[1] Altschul, S. F., et al., "Basic Local Alignment Search Tool," J. Mol. Biol., Vol. 215, No. 3, (1990), pp. 403–410.

[2] Altschul, S. F., et al., "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs," Nucleic Acids Research, Vol. 25, No. 17, (1997), pp. 3389–3402.

[3] Pearson, William R. "[5] Rapid and sensitive sequence comparison with FASTP and FASTA." Methods in enzymology 183 (1990): pp. 63-98.