Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-7, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Parallel Approach for Global Sequence Alignment

Mansi Sethi & Sandeep Singh
NORTHCAP University, Gurgaon, India

Abstract: *Through this study, we are trying to give fast computation solution to align DNA sequences or other biological sequences which are really long and computing them serially takes a long time. By this technique DNA sequences may be aligned faster and provide same and accurate results same as serial implementation of algorithm.*

Introduction

Before introduction of main topic, we should have an idea of basic terms used in this thesis:

Bio-informatics: is a branch of science that combines statistics, mathematics, science and engineering to interpret bio-logical data.
Sequence Alignment: is a way of arranging biological sequences so as to identify similarities between them.
Alignment can be categorized as follows:

Global Alignment: In global alignment the 2 biological sequences are considered similar over their entire lengths and an attempt is made to match them from end to end.
Eg. NGHUFFTYVBBNNJOLPLKJJN
QGMKLOOOBGMKLLLMKLLN

Local Alignment: In local alignment only small parts of sequences that have similarity are aligned. The entire sequence is not matched from end to end.
Eg. NGHUFFTYVBBNNJOLPLKJJN
QBMKLOOOBBNNLLLMPLLM

1.1 Needleman Wunsch Algorithm

In 1970, Saul B. Needleman and Christian D. Wunsch developed Needleman-Wunsch algorithm for global sequence alignment.

1.1.1 Working of Needleman Wunsch Algorithm

To understand the algorithm, let us consider 2 sequences
Sequence 1 - CGTGATTCAT
Sequence 2 - GACTTAC
The length of sequence 1 and sequence 2 is seen to be 11 and 7 respectively. Let us create matrix with 11+1 columns and 7+1 rows as below. The extra row and column is added to align the gap.

		C	G	T	G	A	A	T	T	C	A	T

G												
A												
C												
T												
T												
A												
C												

Once the initial matrix has been designed, a scoring scheme may be selected. Let us assume if residues at i and j positions are same, matching score is 1 (S(I,j)=1). If residues at i and j positions are not same, mismatch score is -1 (S(I,j)=-1). Let us assume the gap score (w) is -1.
#Note – Whenever we have an insertion or deletion, gap score is a penalty given to alignment.

Steps of programming:

Initialization step – At first all rows and columns of designed matrix are filled with zeroes.

Filling up matrix – Each cell value is calculated by using its previous neighboring cells (diagonal, left and upper) of current position. The way this is done may be seen below:
The cell may be filled up with 3 choices:
Choice 1 – a[i-1][j-1] + S[i,j], where S[i,j] is substitution score.
Choice 2 – a[i-1][j] + gap score
Choice 3 – a[i][j-1] + gap score
a[i][j] = Maximum (Choice 1 , Choice 2, Choice 3)

Trace Back Step - This is the most crucial step. The trace back step is used to find the best alignment. We start from the last cell of matrix. At first, we determine if current cell value is obtained from previous diagonal, left or upper cell. In case, value of cell is obtained from previous diagonal cell the last characters of both biological sequences are placed one below another. In case, value is obtained from immediate upper cell, gap at that place is introduced in first sequence. Similarly, if value is obtained from immediate left cell, gap at that place is introduced in the 2nd sequence.

#Note – The trace back is complete once we reach the top-left cell.

Eg. Let us take 2 sequences 'SEND 'and 'AND':

		S	E	N	D
	0	-10	-20	-30	-40
A	-10	1	-9	-19	-29
N	-20	-9	-1	-3	-13
D	-30	-19	-11	2	3

	S	E	N	D
done	left	left	left	left
up	diag	left	left	left
up	diag	diag	diag	left
up	up	diag	diag	diag

N

Now, the sequence obtained may be seen as:

The best alignment obtained this way is:

1) Starting from last cell we get-
D
D

2) Tracing backwards again diag is encountered so we put the sequences as:
ND
ND

3) Next cell on tracing back indicates 'left' so gap is introduced in the second sequence:
END
_ND

4) The fourth cell from trace back is 'diag' so, the sequence obtained is:
SEND
A_ND

2. Modified Needleman Wunsch Algorithm:

Our algorithm is based on Needleman Wunsch Algorithm. Here we have executed Needleman Wunsch Algorithm in parallel to improve its efficiency. Implementation may be seen below:

Implementation
Language used: **Python**
Parallel Implementation is achieved through: **Threads**

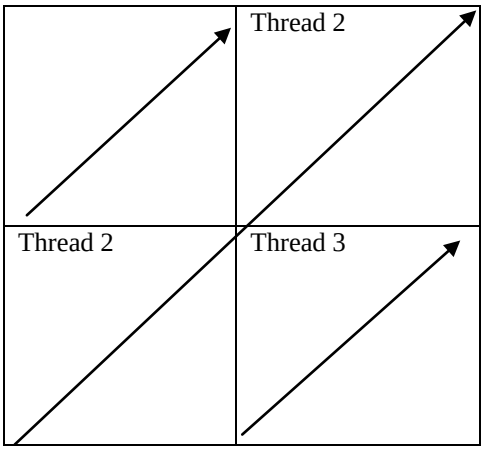
2.1 Steps of Programming:

1) Take 2 sequences you wish to align.
2) Initialize an array with number of columns = length of sequence1 + 1 and number of rows = length of sequence2 + 1.

3) Decide a scoring scheme with
a) Match score = 1
b) Mismatch score = -1
c) Gap penalty = -1

4) Fill up array with all elements as '0'.

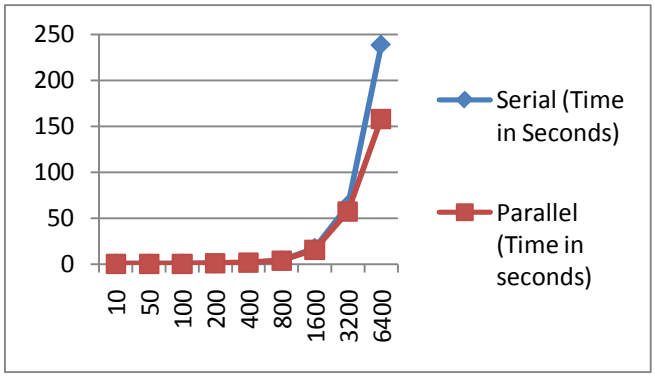
5) The way we are going to fill up the array may be seen below:



6) At first using thread 1 we fill up one fourth of array. Using next thread, we fill up the 2nd and 3rd square portions of the array. The 3rd thread is used to fill up the last square as shown in above figure. The threads have to be executed in order as thread1->thread2->thread3. For this prioritizing threads is important.

7) Next, we use backtracking as stated earlier to find out the best alignment.

3. Comparison between Serial & Parallel Implementation of Algorithm:



Input Size

In the upper graph, it may be seen that initially the serial implementation of algorithm is taking less time as compared to parallel implementation. It may be attributed to the fact that a lot of time is spent in formation of threads. However as Input Size increases, it is seen that time taken for execution of

parallel implementation is less than serial implementation.

4. Conclusion

By doing a slight change in algorithm, we may reduce the computation time. But threads have to be sequenced in a way that they always execute in order as thread1->thread2->thread3.

References

[1]. Parallel Needleman-Wunsch Algorithm for Grid, Tahir Naveed, Imtiaz Saeed Siddiqui, Shaftab Ahmed, Department of Computer Sciences and Engineering, Bahria University Islamabad, Pakistan.
[2]. The Needleman-Wunsch Algorithm for sequence alignment, 7th Melbourne Bioinformatics Course, The University of Melbourne.
[3]. Python Programming Language, www.tutorialpoint.com
[4]. https://en.m.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm
[5]. Comparison of Bio-Sequences, Northern Michigan University.
[6]. Protein multiple sequence alignment by hybrid bio-inspired algorithm, University of Catania, Italy.