Parallel Fast Dynamic Algorithm for Sequence Alignment Using OpenMP and Partationing Scheme

Sara A.Shehab
Faculty of Computers and Artificial Intellgence
Sadat City University
{Sara.shehab@fcai.usc.edu.eg}

Abstract—**Sequence alignment process considered one of the most important tasks in bioinformatic. There are two types of alignment pairwise and multiple sequence alignment. Many algorithms proposed to complete this task. The key parameter in these algorithms is alignment and its scoring value. If the proposed algorithm maximizes the score, so it will be optimal. The algorithms used to align sequences have two main drawbacks. The first is the sensitivity when the data used is very large, the output score is not optimal and has a bad sensitivity. The second is the execution time when the data is large. To overcome these two problems the parallel version of Fast Dynamic Algorithm for Pairwise sequence Alignment is proposed. The first problem solved by partitioning scheme and the second solved by using OpenMP to distribute tasks on available threads. The results indicated that the proposed parallel Algorithms achieve high level of accuracy and sensitivity and improve the execution time.**

Keywords—Sequence alignment, optimal score, Fast Dynamic Algorithm for Pairwise Sequence Alignment, OpenMP

I. INTRODUCTION

The way of arranging two or more sequences to identify the region of similarity between them called sequence alignment. When aligning two sequences this known as pairwise sequence alignment [1]. When aligning more than two sequences it is known as multiple sequence alignment. Many algorithms used in pairwise alignment such as Needleman [2], smith [3] and FDASA [4]. Whereas there are many algorithms used in multiple sequence alignment like Clustal Omega [5], MAFTT [6], and MUSCLE [7]. In the alignment process there are three types of operations trying to get the optimal solution, substitution, inserting or deleting. Gap inserted between bases to get high level of matching between sequences. There are two types of alignment local alignment and global alignment. In global alignment the sequences are aligned from the end to the end even though there are differ in some parts [8]. Local alignment aligns the parts that have more similarity. Local alignment is better as it gets high match between sequences Fig.1.There are many Pair-Wise algorithms used to compare two sequences. Fig.2 illustrates an alignment between the sequences A=ACAAGACAGCGT and B=AGAACAAGGCGT [9]. In the alignment two gaps are used to get maximum match between sequence A and sequence B. the score of the alignment can be calculated from the number of matches and number of mis matches and number of gaped used in alignment.

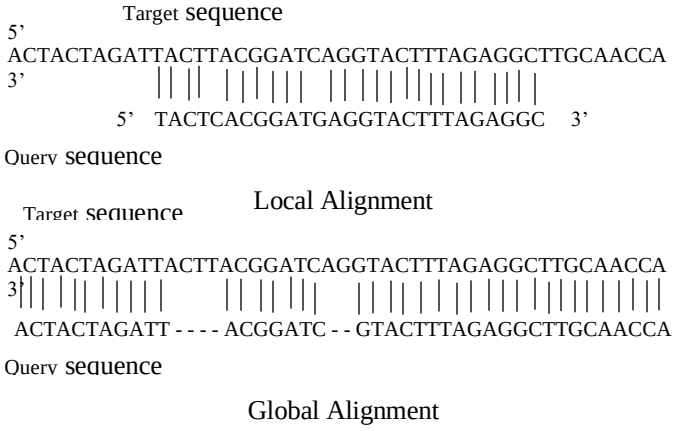

Fig. 1. local Vs Global Alignment

Fig. 2. One of the Optimal Alignment

The previous work in alignment based on dynamic algorithms. like Needleman and Smith their algorithm depend on creating matrix of M *N whereas M is the length of first sequence and N the length of second sequence. Many algorithms used to fill this matrix [10-12]. FDASA Algorithm detect that there is no need to fill all cells in matrix it just fills the three main diagonals to get optimal solution. The problems in both algorithms when using large data set, the solution is not optimal, and the sensitivity is very low, and the time is very high to align large sequences. The proposed algorithm overcome this problem by using partitioning scheme and paralyzed algorithm with Open MP.

II. RELATED WORK

A. FDASA Algorithm

As we mentioned before FDASA algorithm depends on creating matrix of M*N where M is the length of first sequence and N is the length of the second sequence. After that the filling matrix operation focus on the three main diagonals only and ignore all other cells. This calculation reduces memory location used and decrease execution time and at the same time get the

optimal solution of the two sequences. The flow chart of the FDASA Algorithm in Fig.3.

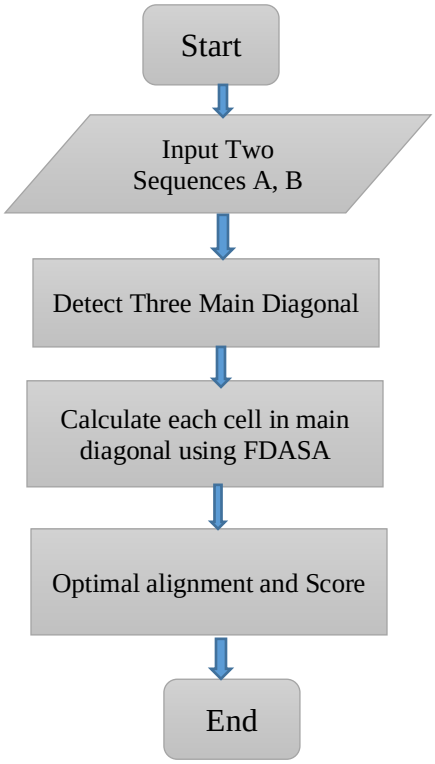

Fig. 3. FDASA Flow Chart

The first step in flowchart input two sequences. secondly detect the three main diagonal. thirdly calculate the value of each cell depends on the calculation as follows: -

1. Create for loop for each cell in main diagonals.
2. For each cell in column
3. For each cell in row
4. If the two column and row have the same value
5. The current cell value = the diagonal value + match
6. The direction of the pointer is diagonal
7. If the column in the current cell no the same as row
8. Current cell = max

{Diagonal cell + mismatch, case 1

Below cell + gap, case2

Above cell + gap, case3}

Hint: to get the current cell value, only one value from diagonal the below or above value can do this task.

Direction of arrow =

{DIAG, if case 1

LEFT, if case 2

UP, if case 3}

Finally, the output is the optimal alignment with max score and traceback of the alignment.

B. FDASA CaseStudy

Given two sequence A=AGTA and B=ATA, the length of two sequences differs. First the matrix length is M*N that is 4*3. The main steps of FDASA algorithm are as follow: -

1. Initialization

Gap value= -1
Match value= +1
Mismatch value= -1
Arrow direction= this arrow used to trace back the alignment.
M (0, 0) = 0
M (0, rows) = C (0,0) +Gap value = -1
M (j, columns) = C (0,0) +Gap value = -1

2. Find the importatnt valuse in three diagonal.

3. Main Iteration

Evaluate each cell in diagonals
If two values of column and row same
M (1,1) =M (0,0) +match value =0+1=1
Direction of arrow =diagonal.
If the two values of row and column not the same
M (1,2) = max (M (0,1), M (1,1), M (0,2)) +mismatch value
M (0,2) is empty so,
M (1,2) =max (M (0,1), M (1,1)) +mismatch value
= max (-1,1) +(-1) =1-1=0

Direction of arrow = the highest value =left
After calculating all cells in diagonal, the matrix will be as (TABLE 1):

TABLE I. FILL 3 DIAGONAL VALUES AND TRACE BACK POINTER

	-	A	G	T	A
-	0	-1			
A	-1	1	0		
T		0	0	+1	
A			-1	0	2

4. Termination

The optimal score is in the last cell of matric M (row, column)
The optimal score in matrix =2
Using the direction arrow to get the alignment it will be as follows: -

A G T A
| | | |
A - T A

The Score value can be calculated from equation

SP Score= P *MV + S *MSV+ G *GP (1)

Where P is number of match, MV match value, S number of mismatch, MSV mismatch value, G number of gap inserted in alignment and GP is the gap value.

In the previous example SP= (3) *1+(0) *-1+(1) *-1=3-1=2

5. Performance

Time: O (3(column length) +1) if length of row = column
O (3(column length) +2) if length of row! = column
= O (3*(column length) +2) = O (11)

Space: O (3(column length) +1) if length of row = column
O (3(column length) +2 if length of row! = column
O (3*(column length) +2) = O (11)

In 2018 [13], Suzuki reduce the size used to store matrix used in dynamic programming algorithms by differential encoding. This method works effectively with storing each DP cell to exploits SIMD operation. The score is also maximized, and the package of Gaba Library is developed [14].

Recently, in 2018 [15] Rahn proposed a parallel version of SWG among its many algorithms to increase performance. the proposed algorithm based on integer saturation strategy. The result shows that this algorithm need short time to align sequences.

In 2020, Santiago Marco-Sola proposes WFA a wave front alignment algorithm [16]. the proposed algorithm depends on using homologous regions between sequences to improve the alignment. WFA algorithm can take O(ns) time to complete alignment where n is the length of alignment and s is the score. The proposed WFA also can improve the performance up to 20-300x faster than previous algorithms.

III. Proposed Parallel FDASA

In the proposed algorithm two scheme used the first is partitioning scheme the second is parallel FDASA using OpenMP. In the partitioning scheme the input data set is partitioning to equivalents partitions and each partition aligned with FDASA. The partitioning scheme overcome the problem of using large data set and the problem of sensitivity. As it is now easily to evaluate the optimal alignment and optimal score for each partition separately and finally merge the output from each partition to produce the whole alignment for input. In the second part the parallel FDASA use available threads to align partitions. Each thread takes one partition and apply the FDASA algorithm on it. The parallel algorithm solves the high execution time problem and improve the performance. The proposed parallel FDASA is shown in Fig.4. in the figure the first step input data set, after that the data split to many partitions equivalent in size. The next step using Open MP to paralyze the FDASA as assign each partition to thread and apply the FDASA algorithm on it. Finally, the output from each partition will merged and output the final alignment and score for whole data set.

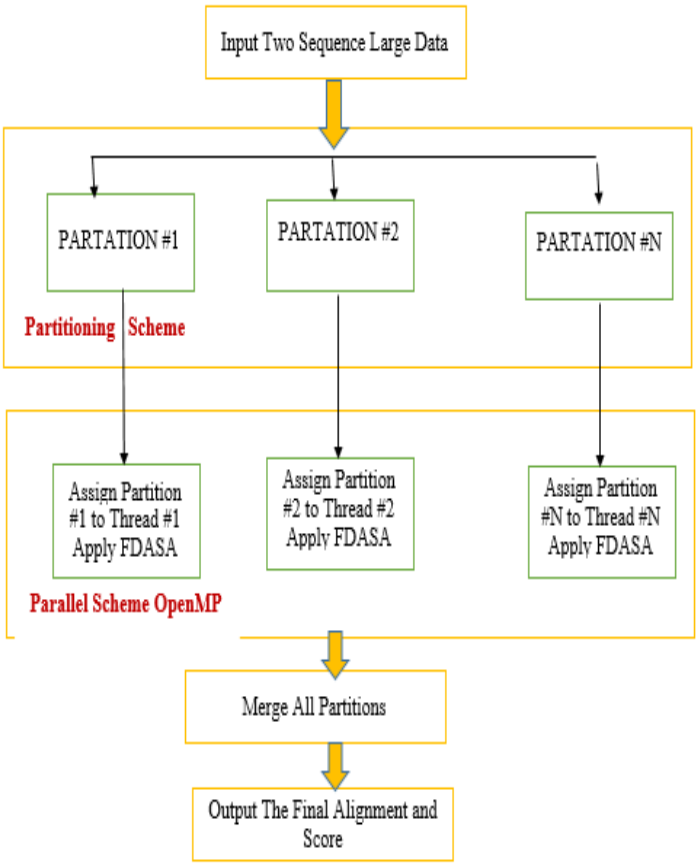

Fig. 4. Parallel FDASA Diagram

IV. Experiential Results

The Parallel version of FDASA is implemented with C++ with Open MP under windows 10. In the implementation side the partitioning scheme is implemented firstly and after that the parallel version. This version of code run under CPU environment.

A. The Impact of Using Partationing Scheme

In the partitioning scheme the input data is partitioned into equivalent parts and apply FDASA algorithm on it. The data used in implementation is BALIBASE data set, OXBENCH and SMART data set. TABLE II discuss the SP score for BALIBASE data set in the partitioning scheme and non-partitioning scheme. TABLE III list the SP score for OXBENCH data set with and without partitioning scheme, whereas TABLE IV list the SP score for SMART data set. the results indicate that in partitioning scheme the SP score will maximized when compared with the non-Partitioning. this conduct that the proposed algorithm maximizes the SP score and Sensitivity when compared with other pairwise sequence alignment algorithms.

TABLE II. SP SCORE IN PARTATIONING AND WITHOUT PARTATIONING FOR BALIBASE DATA SET

Data Set	SP Score	
	Non-Partitioning	***Partitioning***
RV11_BB11025	175	181
RV11_BBS11002	94	122
RV11_BBS11008	153	304
RV11_BBS11025	149	122
RV12_BBS12039	175	185

TABLE III. SP SCORE IN PARTATIONING AND WITHOUT PARTATIONING FOR OXBENCH DATA SET

Data Set	SP Score	
	Non-Partitioning	***Partitioning***
4t2	173	188
22s33	176	407
44t42	204	218
139s4	532	402
512	285	314

TABLE IV. SP SCORE IN PARTATIONING AND WITHOUT PARTATIONING FOR SMART DATA SET

Data Set	SP Score	
	Non-Partitioning	***Partitioning***
AXH	312	318
LCCL	259	265
POU	217	251
SAND	163	163
VWC_out	336	336
WH1	177	209
WIF	248	297

In the figures Fig.5, Fig.6 and Fig.7 the diagram of the SP score for the three used data set BALIBASE, OXBENCH and SMART data set in the partitioning scheme and non-partitioning scheme. The graphs show that with partitioning scheme the SP score will increased, and the sensitivity also increased that is the main problem in previous algorithms.

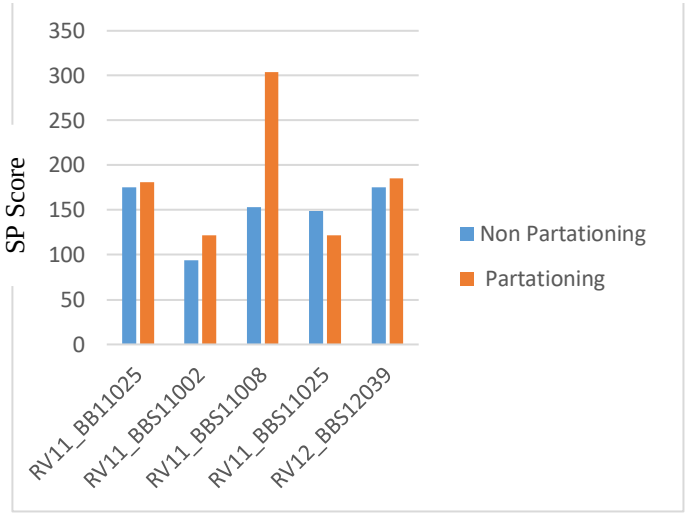

Fig. 5. SP score with and without partationing in BALIBASE data set.

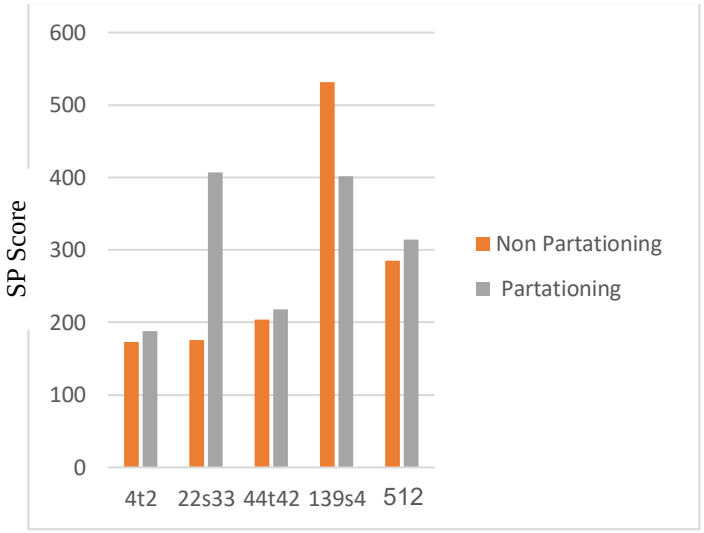

Fig. 6. SP score with and without partationing in OXBENCH data set.

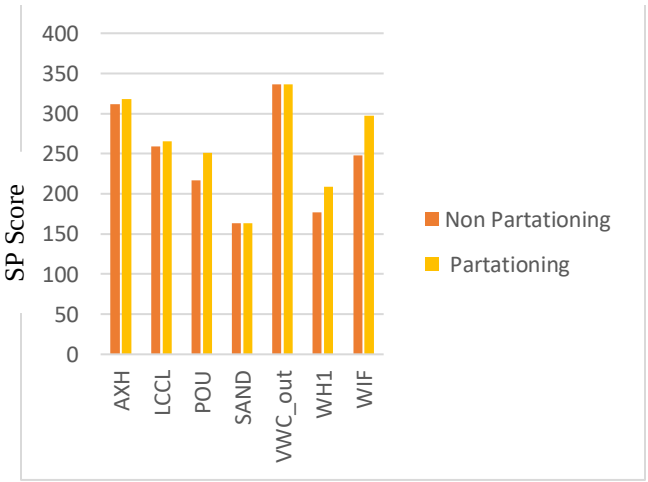

Fig. 7. SP score with and without partationing in SMART data set.

B. *The Impact of Parallel FDASA using Open MP*

In the sequential version of FDASA the whole data set is aligned at same time. The sequential is good with small data set but when the data set is very large the sequential version of FDASA and other pairwise algorithms the output is not the optimal and the SP score is not the max and the time to align sequences is high. For these reasons the parallel version of FDASA is proposed. In the parallel version when the data partitioned, using Open MP the available threads apply to partitions and align then in parallel. TABLE V, TABLE VI and TABLE VII list the execution time in parallel and sequential FDASA for data sets BALIBASE, OXBENCH and SMART respectively.

TABLE V. ELLAPSED TIME SEQUENTIAL AND PARALLEL FDASA FOR BALIBASE DATA SET

Data Set	Time millisecond	
	Sequential FDASA	***Parallel FDASA***
RV11_BB11025	15999	15997
RV11_BBS11002	8001	7984
RV11_BBS11008	16001	8011
RV11_BBS11025	8001	7982
RV12_BBS12039	16001	7982

TABLE VI. ELLAPSED TIME SEQUENTIAL AND PARALLEL FDASA FOR OXBENCH DATA SET

Data Set	Time millisecond	
	Sequential FDASA	***Parallel FDASA***
4t2	11006	10990
22s33	34903	14007
44t42	16029	13201
139s4	14643	9008
512	27019	13010

TABLE VII. ELLAPSED TIME SEQUENTIAL AND PARALLEL FDASA FOR SMART DATA SET

Data Set	Time millisecond	
	Sequential FDASA	***Parallel FDASA***
AXH	10007	2001
LCCL	12012	8008
POU	10026	6040
SAND	15005	12006
VWC_out	9989	6988
WH1	18012	12038
WIF	27022	14994

In the graphs Fig.8, Fig.9 and Fig.10 the execution time for the sequential and parallel version of FDASA are done. The graphs show that in the parallel version the execution time will be decreased when compared with sequential algorithm. This is the second problem in the previous algorithms. the parallel version solves it perfectly.

Fig. 8. Execution time in Sequential and Parallel FDASA in BALIBASE

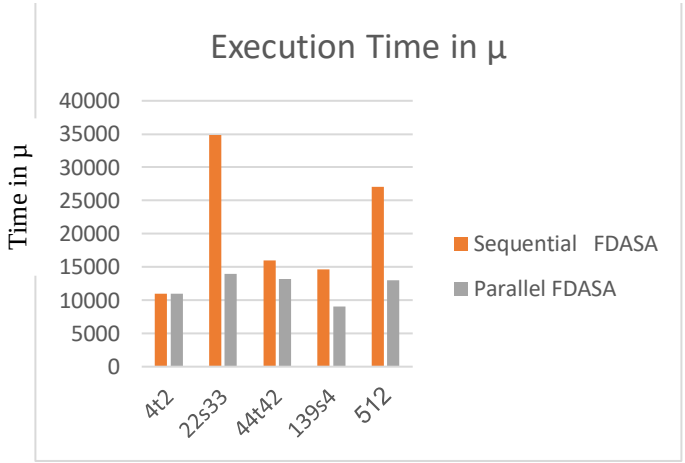

Fig. 9. Execution time in Sequential and Parallel FDASA in OXBENCH

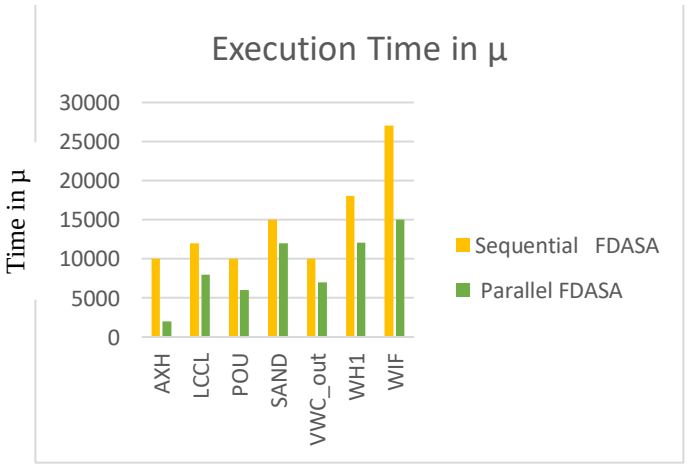

Fig. 10. Execution time in Sequential and Parallel FDASA in SMART

V. CONCLOUSION

In this paper the parallel version of Fast Dynamic Algorithm for Sequence Alignment is proposed. It is aims at solving two important problems in pairwise alignment. The first problem is when using large data set the SP score isn't the optimal and the sensitivity is low. The second problem is execution time. the proposed parallel version of FDASA solve these problems by using two schemes. The first scheme is partitioning the input data, the results indicate that SP in the partitioning is larger than non-partitioning. The second scheme is parallel version of FDASA with Open MP and assign each partition to available thread in parallel. The results prove that the parallel version of FDASA is minimize the execution time when compared with sequential one. So that the parallel version of FDASA is considered one of the most useful algorithms in aligning large data.

REFERENCES

[1] Mount DM,"Bioinformatics: Sequence and Genome Analysis (2nd ed.)," Cold Spring Harbor Laboratory Press: Cold Spring Harbor,NY. ISBN 978-0-87969-608-5, 2004.

[2] Needleman S, Wunsch.,"A general method applicable to the search for similarities in the amino acid sequences of two proteins," J Mol Biol, 48:443-453, 1970.

[3] Smith, Temple F. & Waterman, Michael S,"Identification of Common Molecular Subsequences," Journal of Molecular Biology, 147 (1): 195–197, 1981.

[4] Sara A. Shehab,Waiel Fathi, Arabi Keshk, Hany Mahgoub, " Fast Dynamic Algorithm for Sequence Alignment Based On Bioinformatics," International Journal of Computer Applications37(7): 54-61, January 2012 .

[5] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. S¨oding, et al,"Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega", Molecular systems biology, 7(1):539,2011.

[6] K. Katoh and D. M. Standley," Mafft multiple sequence alignment software version 7: improvements in performance and usability," Molecular biology and evolution,30(4):772–780, 2013.

[7] R. C. Edgar.,"Muscle: multiple sequence alignment with high accuracy and high throughput," Nucleic acids research, 32(5):1792–1797, 2004.

[8] Tahir Naveed, Imitaz Saeed Siddiqui, Shaftab Ahmed,"Parallel Needleman-Wunsch Algorithm for Grid," Proceedings of the PAK-US International Symposium on High Capacity Optical Networks and Enabling Technologies , Islamabad, Pakistan,Dec 19 -21, 2005.

[9] SérgioAnibal de Carvalho Junior ,"Sequence Alignment Algorithms," thesis, 2002/2003.

[10] Rong X,.(Jan 2003)." Pairwise Alignment -CS262 Lecture 1 Notes(online)". Stanford University. Available: http://ai.stanford.edu/~serafim/cs262/Spring2003/Notes/1.pdf.

[11] Bin Wang.(2002)." Implementation of a dynamic programming algorithm for DNA Sequence alignment on the Cell Matrix Architecture (online)", Utah State University, Logan, Utah.Available:http://www.cellmatrix.com/entryway/products/pub/wang2002.pdf.

[12] Chand T. John.(April 2004). "CS273: Algorithms for Structure and Motion in Biology". Stanford University.Available:http://www.stanford.edu/class/cs273/scribing/8.pdf .

[13] Suzuki,H.andKasahara,M.,"Introducing difference recurrence relations for faster semi-global alignment of long sequences," BMC bioinformatics,19(1), 45,2018.

[14] Suzuki, H. and Kasahara, M. , " Acceleration of nucleotide semi-global alignment with adaptive banded dynamic programming," BioRxiv, page 130633,2017.

[15] Rahn, R., Budach, S., Costanza, P., Ehrhardt, M., Hancox, J., and Reinert, K. ," Generic accelerated sequence alignment in seqan using vectorization and multi-threading," Bioinformatics,34(20), 3437–3445,2018.

[16] Santiago Marco-Sola, Juan Carlos Moure, Miquel Moreto and Antonio Espinosa, "Fast gap-affine pairwise alignment using the wavefront algorithm," Published by Oxford University Press,2020.