Performance Evaluation of SW Algorithm on NVIDIA GeForce GTX TITAN X Graphic Processing Unit (GPU)

Ahmad Hasif Azman[1], Syed Abdul Mutalib Al Junid[2], Abdul Hadi Abdul Razak[3], Mohd Faizul Md Idros[4], Abdul Karimi Halim[5], Fairul Nazmie Osman[6]
[1,3,4,5,6]Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[1,2,3,4,5,6]Electronic Architecture and Application Research Group (EArA), Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Article Info

Article history:

Received Jun 2, 2018
Revised Aug 1, 2018
Accepted Aug 11, 2018

Keywords:

SW algorithm
Graphic Processing Unit (GPU)
DNA sequence alignment**.**

ABSTRACT

Nowadays, the requirement for high performance and sensitive alignment tools have increased after the advantage of the Deoxyribonucleic Acid (DNA) and molecular biology has been figured out through Bioinformatics study. Therefore, this paper reports the performance evaluation of parallel Smith-Waterman Algorithm implementation on the new NVIDIA GeForce GTX Titan X Graphic Processing Unit (GPU) compared to the Central Processing Unit (CPU) running on Intel® CoreTM i5-4440S CPU 2.80GHz. Both of the design were developed using C-programming language and targeted to the respective platform. The code for GPU was developed and compiled using NVIDIA Compute Unified Device Architecture (CUDA). It clearly recorded that, the performance of GPU based computational is better compared to the CPU based. These results indicate that the GPU based DNA sequence alignment has a better speed in accelerating the computational process of DNA sequence alignment.

Corresponding Author:

Syed Abdul Mutalib Al Junid
Faculty of Electrical Engineering,
Universiti Teknologi MARA,
40450 Shah Alam, Selangor, Malaysia
Email: samaljunid@salam.uitm.edu.my

1. INTRODUCTION

The demand for advanced and high-performance computational method for comparing and searching biological sequences have increased, according to the exponential growth rate of biological sequences database [1-4]. Besides this demand, the requirement for high performance and sensitive comparison, and alignment tools have also increased after the advantage of the system for defining the solution is related to the deoxyribonucleic acid (DNA), human genomes and molecular biology has been figured out through bioinformatics study [5-6]. In DNA sequence alignment, the performance of comparison and alignment affect a lot of application processes such as vaccine design, drugs design, genetics detection, disease identification and curing method. Hence, with the high performance and high sensitivity DNA sequences alignment or comparison; the vaccines, drug, disease detection and disease curing method can be designed and defined in a faster way.

In consumer cases, the effect of a chemical reaction due to the bad chemical contains in cosmetic and health product can be proven using DNA [7-8]. The demand for DNA sequence alignment for court evidence and reference is high in the past decade, proving a lack of cosmetic and pharmaceutical manufacturer, criminal and forensic cases [9-10]. To satisfy this need, high performance and sensitive biological comparison tools are very important for research and application of molecular biology today.

The demand for high sensitivity and high-performance DNA sequence alignment is higher and important since it can extract useful information from DNA sequences. The sensitive algorithm for DNA sequence alignment has been proposed and highlighted in [10] but the performance has degraded due to the high sensitivity and accuracy. The increase in performance and sensitivity in DNA sequence alignment contributes to the faster identification of information.

Nowadays, the molecular biology research is focusing intensively on the genetic study in order to gain information regarding the genetic diseases and other diseases from cell level. The study of DNA repairs can be done in preventing the disease from occurring in the future through vaccines designed. In this study, the disease can be detected from the cells level. The mobility of the cell in protecting it from the disease can be analyzed in order to find the suitable protecting method or vaccines. All of these can be performed faster when having a sensitive and high-performance DNA sequence alignment tool for analyzing the DNA sequence from the cell. The database for disease due to social-demography and boundaries can be developed from the DNA sequences data and the control mechanism can develop faster for preventing the disease from spreading out.

In bioinformatics study, there are two basic approaches currently being adopted in order to identify similarity scores of DNA between two sequences or referred to as pairwise alignment. This technique, known as a heuristic and dynamic programming. There are a number of important differences between heuristic and dynamic programming which is speed and accuracy, in term of speed the heuristic is faster to compute the score between pairwise alignment than dynamic programming, but it's less sensitive in accuracy compared to dynamic programming. This paper aims to show the computational of the DNA similarity score using a Smith-Waterman algorithm based on dynamic programming. The smith waterman is one of the algorithm that has been derived from Needleman-Wunsch another dynamic programming which use the technique of divide and conquer in order to obtain optimal score alignment in biological sequence [11-13] due to the large complexity of this algorithm $0(mn)$ [14] and the rapid growth the DNA data size in database is exponential [1] it has make the process become slow when involve large sequence search. As a solution, the technique of parallel programming has been implement to reduce the computational time of SW algorithm.

To improve this problem, method of parallel programming has been implanted reported in [15-19] as a way to improve the performance of SW algorithm by using FPGA as a medium to accelerate the time of algorithm, but FPGA can become excessive in term of price and development of it has become slow. To overcome this issue, the GPU (graphics processing unit) has been used to accelerate the SW algorithm, the GPU is a device that suitable for parallel programming it takes advantage of a large number of threads to execute work thus lead to minimizing the duration of the program. To take advantage of this system, in 2007 NVIDIA has released a platform that works with C / C++ language known as CUDA (Compute Unified Device Architecture). This paper shows the evaluation performance of the SW algorithm on the NVIDIA GeForce GTX TITAN X GPU and Intel® CoreTM i5-4440S CPU 2.80GHz.

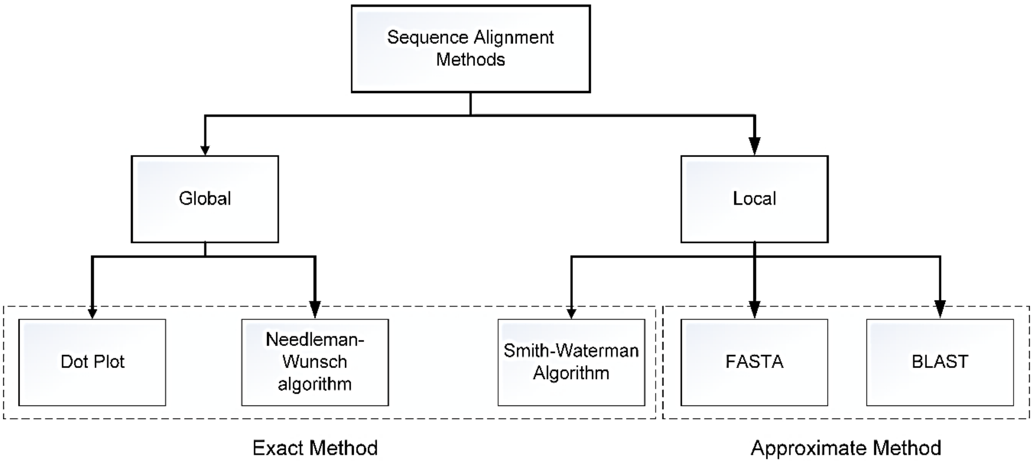

Figure 1. Sequence alignment methods [20]

The Smith-Waterman algorithm has been introduced by Temple F. Smith and Michael S. Waterman in 1981. This algorithm is one of the popular and widely been used as a method to find the similarity of pairwise alignment and the process of the SW algorithm can be divided into 3 stages, initialization '0' in the matrix, fill up the matrix and traceback. The process of initialization begins by initialize Hi,j = 0, where H0,j = 0 and Hi,0 = 0. Below in Figure 2 shows the data dependency in S algorithm, the comparison value of H(0,0), H(0,1), and (1,0) will be used as a score value for H(1,1).

H(0,0)	H(0,1)	H(0,2)	H(0,3)	H(0,4)	H(0,5)
H(1,0)	H(1,1)	H(1,2)	H(1,3)	H(1,4)	H(1,5)
H(2,0)	H(2,1)	H(2,2)	H(2,3)	H(2,4)	H(2,5)
H(3,0)	H(3,1)	H(3,2)	H(3,3)	H(3,4)	H(3,5)
H(4,0)	H(4,1)	H(4,2)	H(4,3)	H(4,4)	H(4,5)
H(5,0)	H(5,1)	H(5,2)	H(5,3)	H(5,4)	H(5,5)

Figure 2. Computational cells of DNA sequence alignment

Next, the matrix cell will be computed based on the sequences of S1 and S2, the computation of each cell calculated by the following equations:

$$H(i,j)\,\text{max} = \begin{cases} 0 \\ Hi-1,j-1+Si,\dfrac{j}{D} \\ Hi-1,j-d \\ Hi,j-1-d \end{cases}$$

In this stage, the computation of the matrix cell consists of the 4th cycle where each of this cycle plays the role to gain the final optimal score alignment in a matrix cell. In the 1st cycle $H(i-1,j-1)+S(i,j)$ equation used by comparing the sequence where Si,j represent similarity and D dissimilarity score based on the character between S1 and S2 sequence. Then on 2nd cycle, the $H(i-1,j)$ used upper cell value and d is a gap value and in 3th cycle $H(i,j-1)$ used horizontal value to compute the cell value.

The last cycle '0' in the equation represent restart, if the score value of computation reaches negative value it will be set as 0 which make it differ scoring method in the Needleman-Wunsch method. Once the matrix cell has been filled up with the score, the final stage of traceback will begin. On this stage, the process starts by mapping the path of the score, the starting point to trace the alignment starts at the highest score of the matrix cell inclined toward the upper cell, from this the optimal of pairwise alignment will be produced.

		P	U	N	C	H	E	S
	0	0	0	0	0	0	0	0
P	0							
I	0							
N	0							
C	0							
H	0							
E	0							
S	0							

Figure 3. Example of DNA sequence cell initialization
S1: --NCHES
S2: --NCHES

2. RESEARCH METHOD

As shown in Equation 1 the SW algorithm involved high level computational, where its complexity $0(mn)$ will increase proportional to the length of the DNA sequences. By implementing SW algorithm using NVIDIA CUDA an extension of the C programming language, make the code run on GPU [21] that act as the accelerator for this code. As a result, the duration computational of the dynamic programming will be shorten. The issue of complexity of workloads can be reduced because GPU design consist of hundreds of core and thousands of threads, and relatively the GPU threads are tremendously lightweight which suitable for parallel programming. The parallelism means that GPU system are able to operate under a large number of data concurrently, where all algorithm equation executed independently at the same time.

a) Graphic Processing Unit

As stated above, theoretically the S-W algorithm performance can be improved by implementing CUDA C language. The NVIDIA CUDA concept can be described as heterogeneous computing that consists of two distinct parts; host (CPU) and device (GPU) [22].

Figure 4, shows the implementation of the SW algorithm using NVIDIA CUDA C and from the illustration represent of two blocks; Intel(R) Core(TM) i5-4440S CPU @ 2.80GHz as the host and GTX TITAN X with computes capability 6.1 as the device. The memory architecture of the GPU consists of registers, shared memory, constant memory, texture memory, local and global memory as shown in Figure 5. In this study, the parameter for parallel computation of quadratic equation of the SW algorithm will undergo inside of global memory and the transition of data between host and device done by using PCI Express (Peripheral Component Interconnect Express). In general, the purpose of using the GPU is to show the transformation of computation in term of performance compared to the CPU.

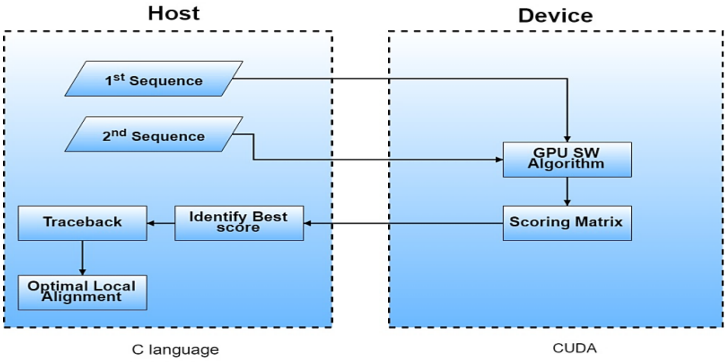

Figure 4. Implementation of NVIDIA CUDA

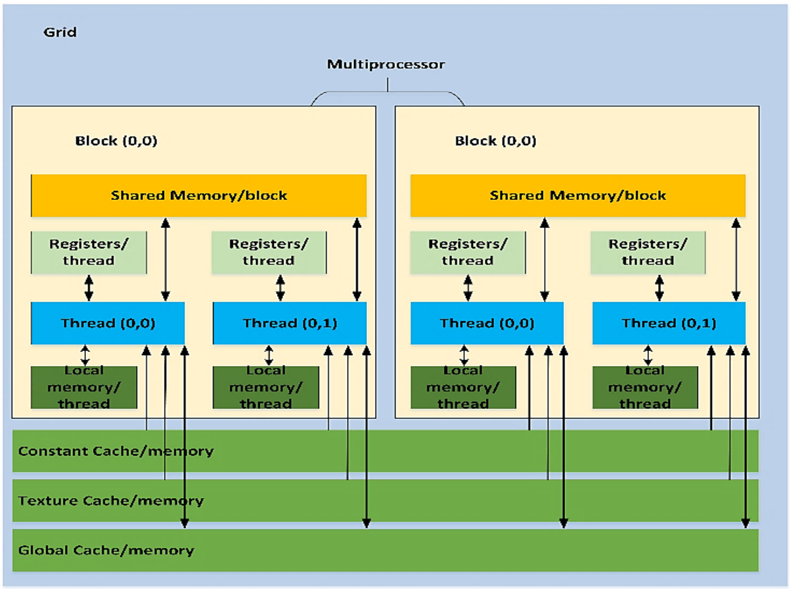

Figure 5. GPU Memory Architecture [22-23]

The SW algorithm represents a matrix of m × n where this matrix having of the 2D array using two threads of indexes for it accessing the data in a 2D way. In order to maximize its performance, the kernel launch; a number of block and threads need to be configured appropriately depending on the size of m × n to achieve high occupancy.

b) Central Processing Unit (CPU)

The implementation of the C language in this study as shown in Figure 6, the illustration is shown the flow of a running process of the SW algorithm where the computation of the equation, trace back and at the end of the final process making optimal alignment occurs in the host environment. The workload has been focused on the CPU, which lead to making the performance of the algorithm differ than the heterogeneous concept which distributes workload between GPU and CPU.

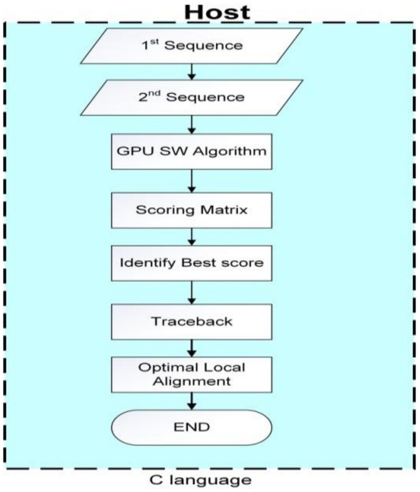

Figure 6. Implementation in CPU

3. RESULTS AND ANALYSIS

This section shows the computational performance analysis of SW algorithm between two platforms; a Central Processing Unit (CPU) and Graphics Processing Unit (GPU), whereas the length of DNA sequences acts as a benchmark in this comparison.

3.1. Data Collection From Base Pair Sequences

In this study, the data have collected from GPU using base pair method to show the benchmark between the similar size of data from 1st sequence and 2nd sequence. The collection of data involved random of DNA sequences with the range from 1 to 100 of the sequence.

Table 1. Smith-Waterman GPU model

Base pair Sequences	Time (ms)
1	0.015
2	0.018
3	0.025
4	0.039
5	0.045
6	0.063
7	0.083
8	0.107
9	0.133
10	0.167
20	0.827
30	2.32
40	5.583
50	9.142
60	15.692
70	23.883
80	34.535
90	47.958
100	64.531

Table 2. Smith-Waterman CPU model

Base pair Sequences	Time (ms)
1	0.938
2	1.114
3	3.202
4	6.654
5	7.619
6	8.177
7	8.875
8	9.718
9	12.184
10	21.251
20	33.438
30	56.863
40	79.169
50	106.796
60	216.025
70	260.426
80	301.393
90	365.466
100	392.137

Based on the implementation of both model the result depicted in Table 1 and Table 2, where the comparative value of runtime in milliseconds and GCUPS for all base pair sequences of the DNA. From the obtained data, has shown significant result between GPU and CPU can be perceived where the GPU data Table 1 lead to a dramatic reduction of runtime compared to CPU Table 2.

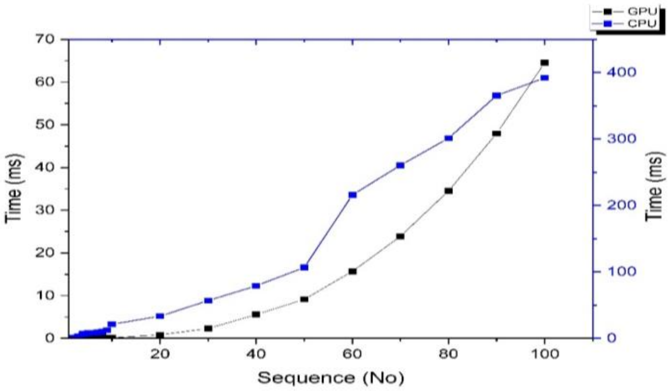

Figure 7. Computational Time Comparison between GPU and CPU

The effect from both approaches, methods can be analyzed in Figure.7, from the illustration the SW algorithm model that has been utilized its usage by the GPU has shown an acceleration in performance in computational time and with the increment of each number of sequences the computations required less of time compared to the CPU model. From the data that has collected between both platforms by using parallelization, the algorithm is capable to produce a better result.

Table 3. Speed-Up and Efficiency CUDA GPU and CPU

Query Sequence	Query Sequence Length	GPU Computational Time (ms)	(GPU) Computational Time (ms)	Speed Up
3CW1_w	138	164.23	427.735	2.60
HZ785699	178	334.96	695.472	2.08
BV097121	201	461.79	866.190	1.87
G29020	241	751.21	955.277	1.27
BV097130	260	921.74	1365.767	1.48
U18389	286	1.197s	1378.209	1.15
		Average		1.74

Based on the implementation of the SW algorithm which divides into two; sequential based on CPU and parallel methods based on the GPU. The result produces from Table 3 shows, the difference computational time achieved between these two platforms. The performance of the parallel technique empowers by the GPU are able to gain high performance, which reduces the computational time of the SW algorithm in the same number of DNA length compared to sequential. To implement comparison test, its involve several query sequences such as 3CW1_w, HZ785699, BV097121, G29020, BV097130 and U18389 with a sequence length of 138, 178, 201, 241, 260 and 286, respectively.

Another important finding that can be related to the obtained results, the differences in how sequential and parallel work. The computational problem of the SW algorithm needs to be solved by dividing and conquering the task. As normally, to solve this problem, it's easy to break it down into small parts and compute it in a series manner. By computing it in sequential the runtime takes to solve the problem increased compared to the parallel which takes that small piece of the problem and computes it simultaneously.

4. CONCLUSION

In this paper, we discuss the comparison of computational time taken of DNA sequence using the Smith-Waterman algorithm between Central Processing Unit (CPU) and Graphic Processing Unit (GPU) method. Base on the result the SW algorithm using NVIDIA CUDA has successfully achieved the best result compared to the CPU, which show, the advantage of the parallelism of the GPU can bring out the high performance of SW algorithm. Other than use for sequence alignments such as DNA and protein, the SW algorithm also can be applied as a spam filter or plagiarism detection and another area which required the need for data comparison. Moreover, with the current growth, development of Graphic Processing Unit the performance of the algorithm can be improved in the future

ACKNOWLEDGMENT

The authors would like to acknowledge the NVIDIA Corporation US for granting the NVIDIA GeForce GTX TITAN X Graphic Processing Unit (GPU) via NVIDIA academic research donation; Faculty of Electrical Engineering for laboratory facilities during the design and testing; Ministry of Higher Education Malaysia for providing financial support under the Fundamental Research Grant Scheme (FRGS) 600-RMI/FRGS 5/3 (134/2015).

REFERENCES

[1] Kim KB, Yu DH, Hwang S. "Analysis of DNA Sequence Alignment Using Fuzzy Membership Degree The Proposed Algorithm for DNA Sequence Analysis". 2012;352-354.

[2] Junid SAM Al, Tahir NM, Majid ZA, Osman FN, Shariff KKM. "Comparative study for DNA data minimization technique on FPGA". *ICIAS 2012 - 2012 4th Int Conf Intell Adv Syst A Conf World Eng Sci Technol Congr - Conf Proc*. 2012;2:765-767.

[3] Junid S a. M Al, Haron M a., Majid ZA, Halim a. K, Osman FN, Hashim H. "Development of Novel Data Compression Technique for Accelerate DNA Sequence Alignment Based on Smith–Waterman Algorithm". *2009 Third UKSim Eur Symp Comput Model Simul*. 2009;181-186.

[4] Al Junid SAM, Haron MA, Abd Majid Z, et al. "Optimization of DNA sequences data for accelerate DNA sequences alignment on FPGA". *AMS2010 Asia Model Symp 2010 - 4th Int Conf Math Model Comput Simul*. 2010;231-236.

[5] Shu JJ, Yong KY, Chan WK. "An improved scoring matrix for multiple sequence alignment". *Math Probl Eng*. 2012;2012.

[6] Khaled H, Gohary R El, Badr NL, Faheem HM. "Accelerating pairwise DNA Sequence Alignment using the CUDA compatible GPU". *Int J Comput Appl*. 2013;84(1):25-31.

[7] Neves N, Sebastiao N, Patricio A, et al. "BioBlaze: Multi-core SIMD ASIP for DNA sequence alignment". *Proc Int Conf Appl Syst Archit Process*. 2013;241-244.

[8] Chakrabarti T, Bengal W. "DNA Multiple Sequence Alignment by a Hidden Markov Model and Fuzzy Levenshtein Distance based Genetic Algorithm". 2013;73(16):26-30.

[9] Zhang J, Yang C. "DNA sequence recognition based on the Markov model". *2013 6th Int Conf Biomed Eng Informatics*. 2013;(Bmei):536-540.

[10] Halim AK, Harun MH, Mohamed S, Majid ZA, Mansor MA, Junid SAMA. "Low power study on trace back and reconstruction modules for DNA sequences alignment accelerator". *Proc - 2012 14th Int Conf Model Simulation, UKSim 2012*. 2012;(1):117-125.

[11] Abdul S, Al M, Majid ZA, Halim AK. "Development of Dna Sequencing Accelerator Based on Smith Waterman Algorithm With Heuristic Divide and Conquer Technique for Fpga Implementation". 2008:994-996.

[12] Al Junid SAM, Md Tahir N, Abd Majid Z, Othman Z, Mohd Shariff KK. "Reducing memory complexity using data minimization technique on FPGA". *2012 Int Conf Comput Inf Sci*. 2012:431-434.

[13] Razip MIM, Junid SAM Al, Halim AK, Majid ZA. "Sequence alignment using systolic array for an accelerator". *2014 IEEE 8th Int Power Eng Optim Conf*. 2014;(March):663-665.

[14] Junid SAM Al, Tahir NM, Majid ZA, Halim AK, Shariff KKM. "Improved data minimization technique in reducing memory space complexity for DNA local alignment accelerator application". *ISCAIE 2012 - 2012 IEEE Symp Comput Appl Ind Electron*. 2012;(Iscaie):153-156.

[15] Rucci E, Garcia C, Botella G, Giusti A De, Naiouf M, Prieto-Matias M. "Smith-Waterman Protein Search with OpenCL on an FPGA". *Proc - 14th IEEE Int Conf Trust Secur Priv Comput Commun Trust 2015*. 2015;3:208-213.

[16] Casale-brunet S, Bezati E, Mattavelli M. "Design space exploration of dataflow-based Smith-Waterman FPGA implementations".

[17] Zahid SK, Hasan L, Khan AA, Ullah S. "A novel structure of the Smith-Waterman Algorithm for efficient sequence alignment*". 2015 3rd Int Conf Digit Information, Networking, Wirel Commun DINWC 2015*. 2015:6-9.

[18] Causapruno G, Urgese G, Vacca M, Graziano M. "Protein Alignment Systolic Array Throughput Optimization". *Tvlsi*. 2015;23(1):68-77.

[19] Junid SAM Al, Idros MFM, Razak AHA, Osman FN, Tahir NM. "Parallel processing cell score design of linear gap penalty smith-waterman algorithm". *Proc - 2017 IEEE 13th Int Colloq Signal Process its Appl CSPA 2017*. 2017;(March):299-302.

[20] Hasan L, Al-Ars Z. "An Overview of Hardware-Based Acceleration of Biological Sequence Alignment". *Comput Biol Appl*. 2011. http://cdn.intechopen.com/pdfs/18997/InTechAn_overview_of_hardware_based_acceleration_of_biological_sequence_alignment.pdf.

[21] Bustamam A, Ardaneswari G, Lestari D. "Implementation of CUDA GPU-Based Parallel Computing on Smith-Waterman Algorithm to Sequence Database Searches". 2013;978-979.

[22] Cheng J, Grossman M, Mckercher T. P*ROFESSIONAL CUDA C Programming*. 2014.

[23] Hasan L, Kentie M, Al-Ars Z. "DOPA: GPU-based protein alignment using database and memory access optimizations. BMC Res Notes". 2011;4(1):261.