International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34288-34293

Parallelization of Smith Waterman algorithm using Mat Lab, Open MP, MPI and Hybrid Approach and their Performance Analysis

V. Nagaveni, G T Raju
Assistant Professor Professor
Dept of Computer Science and Engineering Dept of Computer Science and Engineering Acharya Institute of Technology, Bangalore RNS Institute of Technology, Bangalore nagaveni@acharya.ac.in gtraju1990@yahoo.com

Abstract- This paper aims to describe the various challenges involved and the approaches made to parallelize one of the dynamic programming algorithms, the Smith Waterman algorithm-a local alignment approach, which is used in finding similar regions between 2 nucleotide DNA sequences. Local alignment of two DNA sequences is done using the *BLOSUM50* scoring matrix with the default values for the *Gap Open* and *Extend Gap* properties, which returns the optimal local alignment score. In this paper, Smith-Waterman algorithm has been implemented in both sequential and parallel processing approaches. Here we explore Open MP, Mat Lab and MPI as well as hybrid approach using Open MP with MPI for parallel implementations.

Experiments have been conducted on NCBI database of DNA sequences. Performance analysis on sequential and parallel implementation has been done. Results show that parallel implementation using Mat Lab on Multi core architectures performs better in terms of speed, memory usage, gap score, etc.

Keywords- *Smith Waterman algorithm; parallel programming; Mat Lab; OpenMP; MPI*

1. Introduction

Search operation on large database of DNA sequences using the algorithm are quite slow on conventional sequential processing systems, so many heuristic alternatives have been developed, such as FASTA and BLAST, these methods have reduced the running time by a factor of up to 40 compared with the Smith Waterman sequential implementation. However, at the expense of sensitivity. a distantly related sequence may not be found in a search using these heuristic algorithms for generation and retrieval of large DNA sequences. The use of parallel computers has brought some hope on improving the performance of the pair wise sequence comparison operation when using the Smith-Waterman algorithm. Parallel computers can be broadly divided according to the memory architecture as multicomputer systems with distributed memory and multiprocessor systems with shared memory. It is predicted from Moore's law that the numbers of cores per processor chip will double every 18–24 months [6]. These advances make it available to construct clusters using low priced multi core commodity computers. In this paper we compare the implementation of various parallel programming models on the smith-waterman algorithm on a cluster of shared memory computers, including MPI, Open Mp and Hybrid MPI/Open MP in terms of the execution time, memory usage and gap score.

The remainder of the paper is organized as follows: section II presents the related work done in this algorithm. In section III, the system architecture of smith-waterman algorithm is discussed. In section IV, working of smith-waterman algorithm is discussed .In section V proposed parallel programming approaches are discussed. In section VI Experimental results and discussions are presented and the paper is ended with conclusion.

2. Related Work

The most commonly used method in molecular biology to check for two given sequences are related or not, is to identify their structure or function and to compare their sequences. ***Sequence*** is a collection of nucleotides or amino acid residues which are connected with each other. A typical DNA/RNA sequence consists of nucleotides while a protein sequence consists of amino acids.

Sequencing **is** the process to determine the nucleotide or amino acid sequence of a DNA fragment[11] or a protein. There are different experimental methods for sequencing, and the obtained sequence is submitted to different databases like NCBI, Gene bank etc.

Sequence Alignment and importance:

Sequence Alignment or sequence comparison lies at heart of the bioinformatics, which describes the way of arrangement of DNA/RNA or protein sequences, in order to identify the regions of similarity among them. It is used to infer structural, functional and evolutionary relationship between the sequences. Alignment finds similarity level between query sequence and different database sequences. The algorithm works by dynamic programming approach which divides the problem into smaller independent sub problems. It finds the alignment more quantitatively by assigning scores. When a new sequence is found, the structure and function can be easily predicted by doing sequence alignment. Since it is believed that, a sequence sharing common ancestor would exhibit similar structure or function. Greater the sequence similarity, greater is the chance that they share similar structure or function.

Methods of Sequence Alignment: There are mainly two methods of Sequence Alignment: These two methods of alignments are defined by different algorithms, which use scoring matrices to align the two different series of characters or patterns (sequences). The two different alignment methods are mostly defined by Dynamic programming approach for aligning two different sequences.

Global Alignment: Closely related sequences which are of same length are very much appropriate for global alignment. Here, the alignment is carried out from beginning till end of the sequence to find out the best possible alignment.

Local Alignment: Sequences which are suspected to have similarity or even dissimilar sequences can be compared with local alignment method. It finds the local regions with high level of similarity.

Dynamic programming: Dynamic programming is used for optimal alignment of two sequences. It finds the alignment in a more quantitative way by giving some scores for matches and mismatches (Scoring matrices), rather than only applying dots. By searching the highest scores in the matrix, alignment can be accurately obtained. The Dynamic Programming solves the original problem by dividing the problem into smaller independent sub problems. These techniques are used in many different aspects of computer science. Needleman-Wunsch and Smith-Waterman algorithms for sequence alignment are defined by dynamic programming approach. Smith Waterman algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981. The algorithm explains the local sequence alignment, it gives conserved regions between the two sequences, and one can align two partially overlapping sequences, also it's possible to align the subsequence of the sequence to itself. These are the main advantages of Local Sequence Alignment. This algorithm mainly differs within two aspects from the Needleman-Wunsch algorithm. First aspect being, local alignment differs by having only a negative score for the mismatch and when the matrix value becomes negative it has to be set to zero (need to take maximum value of the scorings compared with zero). They are also predefined scoring matrices for nucleotide or protein sequences.

Scoring matrices: In optimal alignment procedures, mostly Needleman-Wunsch and Smith-Waterman algorithms use scoring system. For nucleotide sequence alignment, the scoring matrices used are relatively simpler since the frequency of mutation for all the bases are equal. Positive or higher value is assigned for a match and a negative or a lower value is assigned for mismatch. These assumption based scores can be used for scoring the matrices. There are other scoring matrices which are predefined mostly, used in the case of amino acid substitutions.

Mainly used predefined matrices are PAM and BLOSUM. **PAM Matrices:** Margaret Dayhoff was the first one to develop the PAM matrix, PAM stands for Point Accepted Mutations. PAM matrices are calculated by observing the differences in closely related proteins. One PAM unit (PAM1) specifies one accepted point mutation per 100 amino acid residues, i.e. 1% change and 99% remains as such. **BLOSUM:** BLOcks Substitution Matrix, developed by Henikoff and Henikoff in 1992, used conserved regions. These matrices are actual percentage identity values. Simply to say, they depend on similarity. Blosum 62 means there is 62 % similarity.

Gap score or gap penalty: Dynamic programming algorithms use gap penalties to maximize the biological meaning. Gap penalty is subtracted for each gap that has been introduced. There are different gap penalties such as gap open and gap extension. The gap score defines a penalty given to alignment when we have insertion or deletion. During the evolution, there may be a case where we can see continuous gaps all along the sequence, so the linear gap penalty would not be appropriate for the alignment. Thus gap open and gap extension has been introduced when there are continuous gaps (five or more). The open penalty is always applied at the start of the gap, and then the other gap following it is given with a gap extension penalty which will be less compared to the open penalty. Typical values are –12 for gap opening, and –4 for gap extension.

Assumed scoring schemas: If the residues (nucleotide or amino acids) are same in both the sequences the match score is assumed (Si, j) as +5 which is added to the diagonally positioned cell of the current cell (i, j position). If the residues are not same, the mismatch score is assumed as -3. This score should be added to the diagonally positioned cell of the current cell. The gap penalty score is assumed as -4 which is added to left and above positioned cells of the current cell. These scores are not unique, they can be user defined also, but the mismatch and gap penalty should be the negative values.

Alignment can be done completely from start to finish of both the sequences to locate the best conceivable alignment [5]. Firmly related sequences which are of same length are all that must fit for global alignment [4]. In situations where genes are aligned whose whole gene is homologous, global alignment may be viewed as appropriate. This strategy for arrangement does not accept that the two successions have the likeness over the whole length [6]. Arrangements which are suspected to have comparability or even different groupings can be contrasted and nearby arrangement strategy [4]. The mainly challenging and important responsibilities in Bioinformatics is sequence database searching from [22], where the fast growing amount of genetic sequence information accessible represents a regular confront to developers of a software and hardware database search and managing. The range of nucleotide database has been doubling-up every 15 months. In this paper [23], a parallel execution methodology of Smith-Waterman algorithm is obtainable. This approach reduced the time complexity from O(mn) to O(m+n) for a single sequence comparison and from O(mnk) to O(m+nk) for multiple sequence pairs comparison. Taking into account that sequences can have up to 10^9 nucleotides each, the time and memory required to solve this problem in a sequential manner is impracticable. This leads to the parallelization of the algorithm with the help of powerful parallel architectures.

3. SYSTEM ARCHITECTURE

Figure 1 shows the proposed system architecture. It consists of Input module having Reference and User Sequence, Smith Waterman algorithm steps, methodologies having both Sequence Alignment and Parallel Alignment and output module,

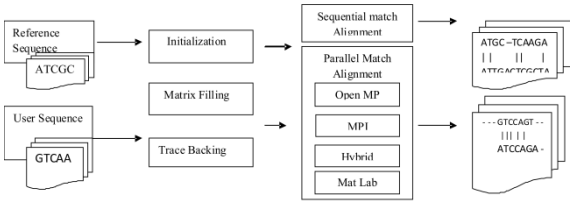


Fig 1: System Architecture

At the input module, reference and user sequence are fed as an input to Smith Waterman module. In Smith Waterman module, the following steps are carried out in an order. They are: Initialization, Matrix Filling and Trace backing. In initialization step, matrix will be initialized with zero in first row and first column. In Matrix filling step, matrix will be filled as per the equation no 2.It is followed by Trace backing, which helps to find the optimal gap score. This result is then given to either or both sequential and all parallel methods one after the other, which gives the optimal local alignment.

4. Smith Waterman Algorithm

The Smith Waterman algorithm is a dynamic programming method for determining similarity between nucleotide or protein sequences. The algorithm was first proposed in 1981 by Smith and Waterman and is identifying homologous regions between sequences by searching for optimal local alignments. To find the optimal local alignment, a scoring system including a set of specified gap penalties is used [Smith and Waterman, 1981].

Homology identified by sequence database searches often implies shared functionality between sequences and further research and development might depend on the accuracy of the search results. The Smith-Waterman algorithm is build on the idea of comparing segments of all possible lengths between two sequences to identify the best local alignments as per equation no (1). This means that the Smith-Waterman search is very sensitive and ensures an optimal alignment of the sequences. Unfortunately, this also has the effect that the method is both time and CPU intensive [15].

A matrix H is built as follows:

$H(i,0)= 0, 0 \le i \le m$

$H(0,j)= 0, 0 \le j \le n$

$$H(i,j)= \max \begin{cases} 0 \\ H(i-1,j-1)+s(ai,,bj) \text{ Match/Mismatch} \\ \max_{k \ge 1} \{H(i-k,j)+ W_k)\} \text{ Deletion}, \\ 1 \le i \le m, 1 \le j \text{ ------equation} \\ \max_{k \ge 1} \{ H(i,j-l)+ W_l) \text{ Insertion} \end{cases} \quad (1)$$

where:

- a,b:Strings over the alphabet ∑
- m=length(a)
- n=length(b)
- s(a,b) is a similarity function on the alphabet
- H(i,j)- is the maximum similarity-score between a suffix of a[1…i] and a suffix of b[1...j]
- W_i is the gap scoring scheme

WORKING OF SMITH-WATERMAN ALGORITHM

The basic steps for algorithm are:

- Initialization of a matrix
- Matrix filling with the appropriate scores.
- Trace back the sequences for a suitable alignment.

i. INITIALIZATION OF MATRIX

Consider the sequences: *CGTGAATTCAG* and *GACTTAC* with the proposed algorithm the table will be filled from the position (1, 1), the first entry in the first row as in figure 2.

	-	C	G	T	G	A	A	T	T	C	A	T
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0											
A	0											
C	0											
T	0											
T	0											
A	0											
C	0											

Fig 2: initialization of matrix

ii. MATRIX FILLING

The second step of the algorithm is filling the entire matrix. To fill the matrix it is important to know the neighbor values to fill each and every cell.

$M_{i,j} = Maximum [M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + W, M_{i-1,j} + W, 0]$ ------*equation* *(2)*

The entire matrix is filled using the assumed scoring schema and initial values.The first residue in both sequences is 'C' and 'G', the matching score or the mismatching score is going to be added the neighboring value which is diagonally located. The upper and left values added to the gap penalty score from the matrix.

The scoring schema equation is:

$$M_{1,1} = Maximum[M_{0,0} + S_{1,1}, M_{1,0} + W, M_{0,1} + W, 0]$$

$$M_{1,1} = Maximum[0(-3), 0 + (-4), 0 + (-4), 0]$$

$$M_{1,1} = Maximum[(-3), +(-4), +(-4)]$$

$$M_{1,1} = 0$$

From the above calculations the maximum value obtained is 0. Finding the maximum value for Mi, j position, there is no negative values in the matrix, since 0 is taken as lowest value. After filling the matrix as in figure 3, keep the pointer back to the cell from where the maximum score has been determined. Each cell is back pointed by one or more pointers from where the maximum score has been obtained.

	-	C	G	T	G	A	A	T	T	C	A	T
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	5	1	5	1	0	0	0	0	0	0
A	0	0	1	2	1	10	6	2	0	0	5	1
C	0	5	1	0	0	6	7	3	0	5	1	2
T	0	1	2	6	2	2	3	12	8	4	2	6
T	0	0	0	7	3	0	0	8	17	13	9	7
A	0	0	0	3	4	8	5	4	13	14	18	14
C	0	5	1	0	0	4	5	2	9	18	14	15

Figure 3: Backtracking of scores

iii. TRACE BACKING THE SEQUENCE

The final step for the appropriate alignment is trace backing; first find the maximum score obtained in the entire matrix for the local alignment of the sequences. The maximum score in

the matrix is 18as in figure 4. So trace back begins from the position which has the highest value, pointing back with the pointers, thus find out the possible predecessor, then move to next predecessor and continue until it reach 0 score.

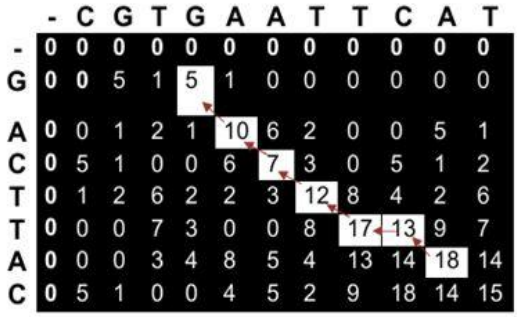


Figure 4:Trace back of possible alignment

The local alignment is obtained as in figure 5.

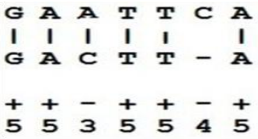


Figure 5: Scoring for best alignment

The alignment can be given with a score, for matching as +5, mismatch as -3 and gap penalty as -4, sum up all the individual scores and the alignment which has maximum score after this can be taken as the best alignment.

5. Parallel programming methodologies

As mentioned in the previous section query sequence is compared separately with user sequence in the database which can be implemented using Parallel programming languages like OpenMP, MPI ,Hybrid approach using both MPI and OPenMP, as well as it can also be implemented using Mat Lab using its constructs . In the following subsections each of these models are discussed.

A. MPI

The Message-Passing Interface (MPI) is a standardization of a message-passing library interface specification. MPI defines the syntax and semantics of library routines for standard communication patterns. Language bindings for C, C++, Fortran-77, and Fortran-95 are supported. Freely available MPI libraries are MPICH, LAM/MPI and OpenMPI [8, 9, 10]. An MPI program consists of a collection of processes that can exchange messages. Normally, each processor of a parallel system executes one MPI process, and the number of MPI processes started should be adapted to the number of processors that are available. Typically, all MPI processes execute the same program in an SPMD style [11].

B. Open MP

API of Open MP supports multi-platform shared-memory parallel programming in C/C++ and Fortran. The Open MP API defines a portable, scalable model with a simple and flexible interface for developing parallel applications on platforms from the desktop to the supercomputer.With OpenMP, the threads number is reduced to one at each machine, messages volume in MPI is reduced, and the time/speed is improved greatly.OpenMP is used to parallelize the matrix/vector operations automatically (just a flag to toggle on/off, based on OpenMP's parallel for). OpenMP's task is used for graph visit.

C. Hybrid MPI/OpenMP

By utilizing a mixed mode programming model we should be able to take advantage of the benefits of both models. For example a mixed mode program may allow us to make use of the explicit control data placement policies of MPI with the finer grain parallelism of OpenMP [7]. In mixed mode model two level of communication pattern are used, inter-node & intra-node communication. intra-node communication is implemented through common access to each node's shared memory and inter-node communication is achieved through message passing between different nodes [14].

D:Mat Lab

MAT LAB has emerged as one of the languages most commonly used by scientists and engineers for technical computing, with approximately one million users worldwide. The primary benefits of MAT LAB are reduced code development time via high levels of abstractions, interpretive, interactive programming, and powerful mathematical graphics. The compute intensive nature of technical computing means that many MAT LAB users have codes that can significantly benefit from the increased performance offered by parallel computing. Some of the functions available in Mat Lab are: ***parfor, parpool, delete, gcp, add Attach files, parfeval, fetch Next, parallel. Pool etc***

6. Experimental results and discussions

The experiments have been conducted on a cluster consisting of ten machines where one machine is the master node responsible for distributing data and nine machines as worker nodes responsible for applying smith waterman algorithm upon receiving data. All machines are of the same configuration, 2.2 GHz Intel core 2 Duo processor and 4 GB of RAM running ubuntu 10.04 32-bit operating system except the master node which is 2.2 GHz Intel quad core processor. The cluster is interconnected using Ethernet.

The MPI implementation used during experiments is MPICH3 which is prominent freely available portable MPI implementation. GNU GCC compiler implementation of Open MP is used [17, 18].

The databases used during the experiments are a set of randomly generated DNA sequences taken from NCBI from [19]. The size of the databases ranges from 1000 kb up to 75 MB. The length of the sequences ranges from 50 to 25000 characters.

The main performance measure in our experiments is the relative speed up. Given two algorithms, where T1 and T2 are the execution times of the serial and parallel algorithms respectively. T2 varies according to the deployed parallel paradigm. MPI, Open MP, and hybrid models are deployed such that the relative speed up S is calculated according to (3).Figure 6 illustrates the execution time

elapsed by the three parallel programming models, Open MP MPI and hybrid model. Hybrid model which combines the MPI and Open MP models gives better performance in terms of the execution time than the pure MPI model and obviously than the Open MP time. In this experiment, Six core 2 Duo CPUs are used as workers.

Speed Up= (T2/T1)*100 --- equation (3)

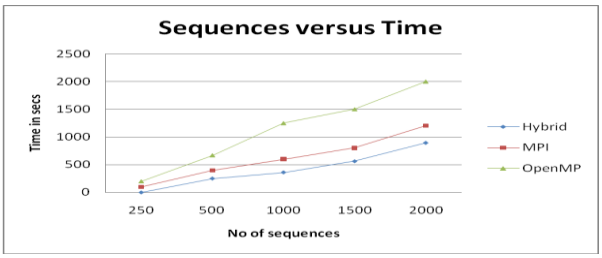


Fig 6: Execution time for three models on different database sizes with 10 CPU.

Figure 7 illustrates the speed up comparison between the three parallel programming approaches, Open MP, MPI and the Hybrid model with different number of CPUs on the same database size of 30000 sequences. When running the experiment using only two worker, on pure MPI model the speed up was 1 which is obviously worse than the serial model. On hybrid model the speed up is raised to 2. It is clear that the speed up increases linearly as the number of CPU and cores increase.

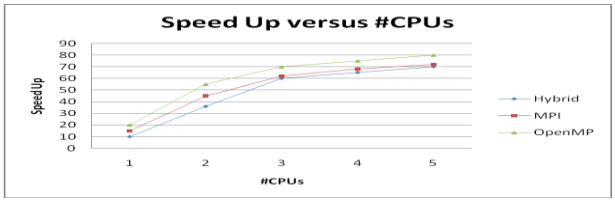
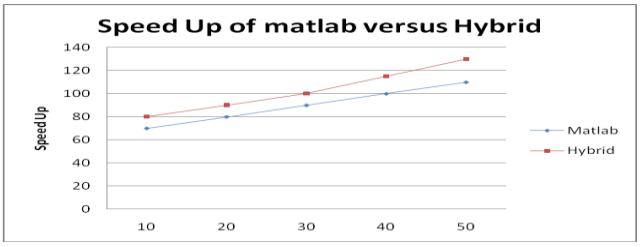


Fig 7: Speed up with respect to CPUs.

On our final experiment we compare the speed up of an MPI implementation with 6 worker nodes and an Mat lab implementation on one Node with 4 cores. Figure 8 illustrates the result. As mentioned previously, the results indicate that the Mat lab can also be used to perform better than the pure MPI. This indicates that Mat lab is more scalable than MPI when applied to various sequence alignment, as it contains many supporting functions, which will increase the performance.



#CPUs
Figure 8: Comparison of Hybrid vs Mat Lab

Comparison study of Sequential with Parallel algorithm using Mat Lab:

Table 1 indicates the advantage of parallel approach using Mat Lab in respect of gaps, score, similarity, memory usage and taken .

Table 1 Sequential versus parallel algorithm

Characteristics	***Sequential Approach***	***Parallel Approach***
Score	380.370000	457.900000
Similarity	582	178
Gaps	611	205
Memory used by reference Sequence	2.33kb	2.89kb
Memory used by User Output Sequence	2.33kb	2.89kb
Time Taken	26.021365	6.601200

8. CONCLUSION

The Smith-Waterman Algorithm presents a standard dynamic programming method for identifying the improved local sequencing alignments of biological gene pairs. The improvements of parallel sequencing makes the efficiency of the system to estimate gaps,similarity,score,memory usage and the time taken for alignment. Because of its highest optimum sensitivity for accomplishing local alignments, this kind of parallel programming approach is a fundamental process in bioinformatics computation, comprising exploration for biological sequence database and sequence alignment. The local alignment increases accuracy of the sequence matching within two sequences, reducing running time as well as a good challenge in bioinformatics computing. The time taken for sequence comparison by parallel alignment of Smith-Waterman Algorithm is less compared to the time taken by sequential alignment.

The proposed methods and the previous approaches have their own advantages and disadvantages. For example Needleman-Wunsch algorithm and discriminative model have their own complexities. The future development is to test the complexity of each model so that a framework can be designed which can enhance added efficiency.. Hence as a further enhancement a new novel algorithm can be expressed to accept any length data set directly, with the help of standard databanks, so that efficiency as a feature can be improvised still more..

References

[1] Kun-Mao Chao and Louxin Zhang, *Sequence Comparison: Theory and Methods*, Springer, 2012 pp. 35.

[2] Michael Farrar, "Striped Smith--Waterman speeds database searches six times over other SIMD implementations," Bioinformatics, 2007, pp. 156-161, doi: 10.1093/bioinformatics/btl582.

[3] Vipin chaudhary, feng liu, vijay matta, and laurence t. yang, "Parallel implementations of local Sequence alignment: hardware and software," Parallel Computing for Bioinformatics and Computational Biology: Models, Enabling Technologies, and Case Studies, Wiley Series on Parallel and Distributed Computing , 2006, pp. 234.

[4] Hsien-Yu, L., Meng-Lai, Y., and Yi, C. "A parallel implementation of the Smith-Waterman algorithm for massive sequences searching," Engineering in Medicine and Biology Society, 2004. IEMBS apos;04.
[5] 6th Annual International Conference of the IEEE, pp. 2817-2820, San Francisco, CA, USA.T. Smith and M. Waterman., "Identification of common molecular subsequences," Journal of Molecular Biology, 1981, pp. 195–197.
[6] Thomas Rauber and Gudula Rünger, *Parallel Programming: for Multicore and Cluster Systems*, springer, 2011, pp. 93
[7] L.A. Smith. "Mixed mode MPI / OpenMP programming," UK High-End Computing Technology Report, 2000.
[8] Faith Han Calglayan, Matthijs Geers, Roelof Willem Heji, "Low-Cost Smith-Waterman Acceleration" DELFT University of technology august 2013.
[9] Yvonne Hermann, "Dynamic Programming and Smith-Waterman Algorithm" classical papers in Bioinformatics may 3rd, 2010.
[10] Ananth Prabhu G, Dr.Ganesh Aithal, "Automatic parallelization for Parallel Architectures using Smith-Waterman Algorithm" volume 3, issue 9, April 2014.
[11] "Global Alignment of two Sequences Needleman-wunsch Algorithm" www.amritavlab.com.
[12] Rajesh Mehra, Sonali Vijan, " Biological Sequence Alignment for Bioinformatics Application using MAT LAB" Electronics Engineering, NITTTR, Chandigarh.
[13] Azzedine Boukerche, Jan M. Correa, Alba Cristina M, A de Melo, Ricardo.P.Jacobi, "A Hardware Accelerator for the Fast Retrival of DIALIGN Biological Sequence Alignments in Linear Space" IEEE Transactions on Computers, vol.59, No-6, pp.808-821, 2010.
[14] Jane B. Reece and Neil A. Campbell. Campbell biology/Jane B Reece.[et al]. Pearson Australia Frenchs Forest, N.S.W, 9th ed. Edition, 2012. ISBN 9781442531765.]Laiq Hasan and Zaid Al-Ars, "An overview of hardware-based Acceleration of Biological Sequence Alignment" Computational Biology and Applied Bioinformatics, pages 187-202, 2011.
[15] S.B.Needleman and C.D. Wunsch, "A general method Applicable to the search for Similarities in the amino acid sequence of two proteins" Journal of Molecular Biology, 48(3):443-453, 1970.
[16] Temple F Smith and Michael S Waterman, "Comparison of Biosequences" Advances in applied Mathematics, 2(4):482-489, 1981.
[17] L.Hasan, Z.Al-Ars and S.Vassiliadis, "Hardware Acceleration of Sequence Alignment Algorithms" In design Technology of Integrated Systems in Nanoscale Era, 2007, DTIS, International Conference on, pages 92-97, 2007. Jacques Cohen. Bioinformatics an Introduction for computer scientists. ACM Comput. Surv., 36(2):122-158, June 2004. ISSN 0360-0300. Doi: 10.1145/1031120.1031122.
[18] Erik Vermij, "Genetic Sequence Alignment on a Supercomputing Platform" Master's thesis, Delft University of Technology, 2011.
[19] K.R.Sharma, "Bioinformatics: Sequence Alignment and Markov Models", McGraw-Hill, 2008. ISBN 9780071593069.
[20] RD page. Gene tree: Comparing gene and Species Phylogenies using Reconciled trees. Bioinformatics, 14(9):819-820, 1998.
[21] P. Zhang, X. Liu, N. Sun and X. Jiang, "A Reconfigurable Accelerator for Smith-Waterman Algorithm" IEEE Trans. Circuits and Systems, vol. 54, no. 12, pp. 1077-1081, Dec (2007).
[22] De Giusti Armado.E, Rucci Enzo, "Parallel Smith-Waterman Algorithm for DNA Sequences Comparison on Different Cluster Architectures" {erucci, degiusti, HUUUfrancoch}@lidi.info.unlp.edu.arUH Universidad nacional de la plata. Argentina.
[23] J.M.Correa, R.P.Jacobi, Boukerche, A.CM.A, de Melo and A.F.Rocha, "Reconfigurable Architecture for Biological Sequence Comparison in Reduced Memory Space" , Proc. IEEE Int'1 parallel and Distributed Processing Symp. (IPDPS), pp.1-8, (2007).
[24] Meng-Lai Yin, Hsien-yu Liao, Cheng.Y, "A parallel implementation of the Smith-Waterman Algorithm for Massive Sequences Searching" Engineering in Medicine and Biology Society, 26th Annual International Conference of the IEEE, Vol.2,pp.2817, 2820, 1-5 Sept (2004).
[25] William R.Pearson, Kun-Mao Chao and Webb Miller, "Aligning two sequences within a specified within a specified diagonal band" , Department of Computer Science, University Park, PA 16802 and Department of Biochemistry, University of Virgina.
[26] Abdullah.R Rashid, N.A.A, Talib, A.Z.H, Ali.Z, "Fast Dynamic Programming Based Sequence Alignment Algorithm", Distributed Frameworks for Multimedia Applications, 2006. The 2nd International Conference on Vol-no-pp.1, 7, May (2006).
[27] Al-Ars.Z, Hasan.L, "An efficient and high performance linear recursive variable expansion implementation of Smith-Waterman algorithm," Engineering in Medicine and Biology Society, Annual International Conference of the IEEE3-6 Sept (2009).
[28] W.Miller and X.Huang, "A time-efficient linear-space local similarity algorithm" Adv.Appl.Math, vol-12.
[29] F.Guinand "Parallelism for computational molecular biology", in ISThmus 2000 Conference on Research and Development for the Information Society, Poznan, Poland, 2000.
[30] A.N Arslan, "Multiple Sequence alignment containing a sequence of regular expressions" computational Intelligence in Bioinformatics and Computational Biology, 2005.
[31] Faith Han Caglayan, Matthijs. Geers, Roelof Willem Heij, "Low-Cost Smith-Waterman Acceleration", DELFT University of Technology August 2013.
[32] Wai Yang, Brain Hang, "A Parallel Implementation of Smith-Waterman Sequence Comparison Algorithm".
[33] S.Aluru and S.Rajko, "Space and Time Optimal Parallel Sequence Alignments", IEEE Transaction. Parallel Distributed Systems
[34] Zaiq Al-Ars, Laiq Hasan, "Performance improvement of Smith-Waterman Algorithm" Delft University of Technology computer Engineering Laboratory