ISSN NO. 2320-5407

Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/2607
DOI URL: http://dx.doi.org/10.21474/IJAR01/2607

RESEARCH ARTICLE

PROTEIN SEQUENCE ALIGNMENT ANALYSIS USING GPU ACCELERATED BASED TOOLS FOR HUMAN GENOME.

G. Anitha Mary*[1] and G. Anjan Babu[2].

1. Asst.Prof, Dept of MCA, Loyola Academy Degree & PG College, Old Alwal, Secunderabad-10, India , Research sc6holar(Ph.D - PartTime)Sri Venkateswara University Tirupati.
2. Professor & Head , Dept. of Computer Science, Sri Venkateswara University, Tirupati, Chittoor District , A.P, India.

Manuscript Info

Manuscript History

Received: 28 October 2016
Final Accepted: 27 November 2016
Published: December 2016

Key words:-
Bioinformatics, Dataset, Protein, Sequence analysis, Visual Processing

Abstract

Basic Local Alignment Search Tool (BLAST) is the program that finds regions of similarity between biological sequences (DNA, RNA and Protein). The program compares nucleotide or protein sequences to databases with sequences and calculates the statistical significance. It finds regions of local similarity between sequences. There are many types of BLAST present based on the query sequence input given. The protein-protein BLAST (blastp) is a program in which the given protein query returns the most similar protein sequences from the protein database when a user specifies. Most of the sequence similarity search runs on Central Processing Unit (CPU). A Graphics Processing Unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory. It is occasionally called as Visual Processing Unit (VPU). It accelerates the creation of images in a frame buffer intended for output to a display. GPUs are used in mobile phones, personal computers, workstations, embedded systems, and game consoles.

Modern GPUs are very efficient at manipulating computer graphics and image processing. Their highly parallel structure makes them more efficient than general-purpose CPUs for algorithms, where processing of large blocks of data is done in parallel. A GPU can be present on a video card in a personal computer. It can be embedded on the motherboard or in certain CPUs it is present on the CPU die. In the present work, the review of the GPU bases software analysis for sequence analysis was being discussed. Blastp Algorithm is being implemented for analysis of Human Protein Sequences. The DNA sequences were first analyzed using BLAST +2.2.28 and then analyzed using GPU BLAST and GPU softwares like BarraCUDA, CUSHAW, G-DNA and G-MSA. The interpreted results were obtained and compared from one another and also with CPU based BLAST programs.

Corresponding Author:- G. Anitha Mary.
Address:-Asst.Prof, Dept of MCA, Loyola Academy Degree & PG College, Old Alwal, Secunderabad-10, India , Research scholar(Ph.D - PartTime)Sri Venkateswara University Tirupati.

Introduction:-

BLAST is one of the most widely used programs in bioinformatics for sequence searching (Casey, R. M., 2005). It finds solution for a fundamental problem in bioinformatics research. The algorithm used is heuristic method and is much faster than other approaches. The emphasis of calculating an optimal alignment on speed is vital and to make the algorithm practical on the huge genome databases that are currently available. Although the subsequent algorithms can be even faster BLAST is mostly used. Many attempts have been made in the last decade to design and develop new BLAST software tools for specific hardware (Fei et al., 2008; Jacob et al., 2007; Sotiriades and Dollas, 2007; Zhang et al., 2000) or even for parallel supercomputers (Lin et al., 2008). Unfortunately, most of the researchers do not have access to these hardware platforms. BLAST software tools using multiple CPU cores for increased speed is National Center for Biotechnology Information (NCBI) BLAST that supports multithreading in the preliminary stage of the BLAST algorithm (Camacho et al., 2009). An indexed Mega-BLAST module is being supported by NCBI-BLAST that uses the database index to achieve an approximate speedup of 2~4X (Morgulis, 2008). To achieve better alignment speed, PLAST is a parallel implementation of BLAST (Nguyen and Lavenier, 2009) that applies a new indexing technique together with multithreading and SSE instructions. To compare two sets of DNA sequences, KLAST is an optimized and extended version of PLAST which includes a module KLASTn. It can achieve good speedup with comparable sensitivity when compared with NCBI BLASTN. Recently, Graphics Processing Units (GPUs) have been widely accepted as high-performance computing platforms with low-cost (Owens et al., 2008). AMD, Intel and other manufacturers turn to focus on improving CPU architecture on the single chip integrating more processor core. It makes a CPU to turn into the direction of Multi-Core development. At the same time, the GPU technology is also in constant development. The birth and development of GPU technology can be divided into four eras i.e., Ex-GPU era, fixed function pipeline era, programmable shader pipeline era and unified programmable shaders era.

GPUs have much higher computational horsepower and memory bandwidth compared with traditional multi-core CPUs. The significant difference between CPU and GPU architectures has created many challenges in developing highly efficient GPU software (Nickolls, 2007). Many bioinformatics tools have been accelerated by GPUs in recent years (Dematte and Prandi, 2010; Liu et al., 2012a, b; Lu et al., 2012, 2013; Manavski and Valle, 2008). For protein sequence alignment also some GPU-based software tools have been developed. Ling's GPU-based BLAST software can achieve a speedup of 1.7~2.7X compared with NCBIBLAST (Ling and Benkrid, 2010).

GPU-BLAST that can typically achieve acceleration speedup of 3~4X relative to the sequential NCBI-BLAST is being developed recently by Vouzis and Sahinidis (2011). Design and implementation of G-BLASTN, an open source software tool for nucleotide alignment based on the widely used NCBI-BLAST is being developed by (Kaiyong Zhao and Xiaowen Chu, 2014). The major advantage of GPU-BLAST is that it can produce the same results as NCBI-BLAST.

Methodology:-

BLASTP Algorithm:-

The overview of the BLAST Program is as follows : 1. Removing the low-complexity region or sequence repeats in the given query sequence; 2. Making a k-letter word list of the query sequence; 3. Listing the possible matching words; 4.Organizing the remaining high-scoring words into an efficient search tree; 5.Repeating the steps from 3 to 4 for each k-letter word in the given query sequence; 6.Extending the exact matches to high-scoring segment pair (HSP); 7.Listing all of the HSPs in the database whose score is high enough to be considered; 8.Evaluation of the significance of HSP score; 9.Making two or more HSP regions into a longer alignment; 10.Showong the gapped Smith-Waterman local alignments of the query and each of the matched database sequences; 11.Report every match whose expect score is lower than a threshold parameter E (Fig:1).

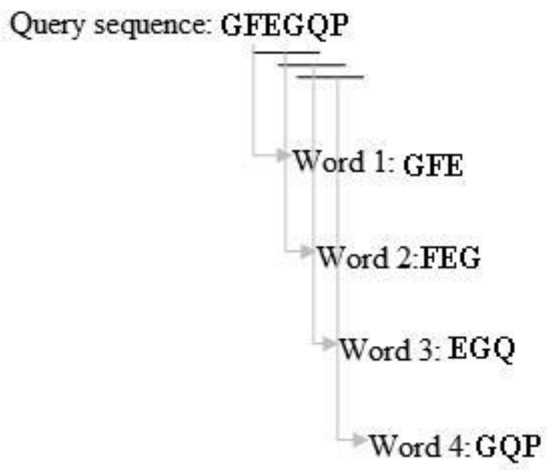


Fig 1:- BLAST Algorithm.

Protein-protein BLAST (BLASTP):-
In this program when given a protein query, it returns the most similar protein sequences from the protein database when the user specifies. For protein database searches, the BLASTP algorithm first makes a list of three-letter words in the query sequence and then scores these words for matches with themselves and with all other possible words using the BLOSUM62 scoring matrix. The 50 highest scoring matches are kept. Database sequences are then scanned for matches to these high-scoring words, and if such are found, then a local alignment is made with the query sequence by dynamic programming.

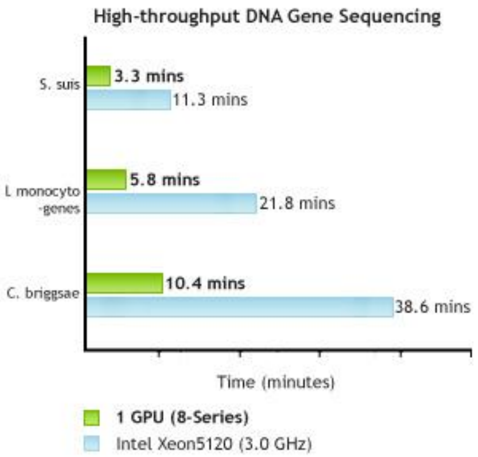
Suppose that the three-letter word GFE is in the query sequence, what is the log odds score of a match of GFE with itself? Scan through the table and find the highest scoring match with G (say amino acid Z, where Z is not equal to G). What would be the score for GFE in our query sequence matching ZFE in the database sequence? Scan again and find the worst match (es) with G (say amino acid W). What is the score for a match of GFE with ZFW? Repeat the last two questions for the second and third letters in GFE. How many possible matches are there with GFE? (BLASTP uses approximately the best 50.) How many words will be used in a search starting with a query sequence that is 400 amino acids long?

BLAST+ 2.2.28:-
BLAST+ 2.2.28 can be accessed from the URL ftp://ftp.ncbi.nih.gov/blast/ executables/blast+/ 2.2.28/ and download on to the system to run the software. The characteristics of this type BLAST consists of composition based statistics support in rpsblast, support for query coverage, subject sequence title, and taxonomy data in custom tabular output format, blastdbcmd support for batch subsequence retrieval and Adaptive BATCH_SIZE. It performs the incremental XML output by formatting of asterix character in XML output. The segmentation fault on out-of-memory and Prevention of extension of alignment into Ns, segmentation fault in DeltaBLAST when used with -remote and -out_ascii_pssm. It has Replace tabs with spaces in FASTA deflines, blastdbcmd displaying internal sequence ID for databases built without -parse_seqid, blastdbcmd not fetching sequence data for complete sequence ID and -target_only, blastn missing a hit for small word sizes, Crash in blastn when it fetches sequence data from Genbank,DeltaBLAST returning no hits when used with -remote option and searching more than one query, Initialization problems for indexed megablast,psiblast problem using -import_search_strategy,blast_formatter displaying empty query for DeltaBLAST RID,makeblastdb problem with ASN.1 input, dustmasker errors with acclist and maskinfo_xml output formats, blastx reporting of HSPs dependent on -max_target_seqs, psiblast's display of number of queries in tabular output format, blastx error when -ungapped and -comp_based_stats F are used.

GPU accelerated Softwares:-
GPU-accelerated computing is the use of a graphics processing unit (GPU) together with a CPU to accelerate deep learning, analytics, and engineering applications. Pioneered in 2007 by NVIDIA, GPU accelerators now power energy-efficient data centers in government labs, universities, enterprises, and small-and-medium businesses around the world. They play a huge role in accelerating applications in platforms ranging from artificial intelligence to cars,

drones, and robots. GPU-accelerated computing offloads compute-intensive portions of the application to the GPU, while the remainder of the code still runs on the CPU. From a user's perspective, applications simply run much faster.

A simple way to understand the difference between a GPU and a CPU is to compare how they process tasks. A CPU consists of a few cores optimized for sequential serial processing while a GPU has a massively parallel architecture consisting of thousands of smaller, more efficient cores designed for handling multiple tasks simultaneously. GPUs have thousands of cores to process parallel workloads efficiently. Sequencing and protein docking are very compute-intensive tasks that see a large performance benefit by using a CUDA-enabled GPU. There is quite a bit of ongoing work on using GPUs for a range of Bioinformatics and life sciences codes. (Fig 2 &3)



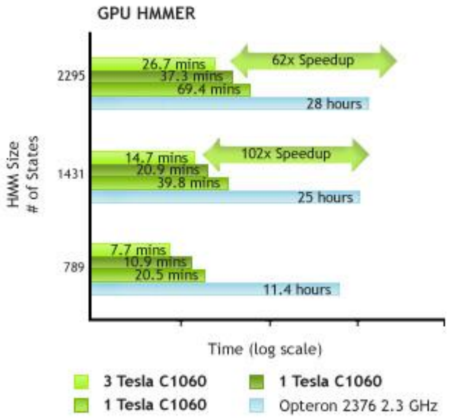


Fig 2:-Accelerating HMMER using.**Fig 3:-**MUMmerGPU: High-through DNA sequence GPUsScalable Informaticsalignment using GPUs (Schatz, et al) (Ref: nvidia)

Modern GPUs are tiny computing centers with thousands of cores specialised to carry out mathematical routines. Figure 1 shows the computing time of GPU versus CPU depending on the input length. The advantage of GPU to accelerate computations increases with the input length. Figure 2 illustrates the acceleration achieved exclusively by hardware optimization by porting an algorithm for matrix multiplication to the GPU.(Fig 4 & 5)

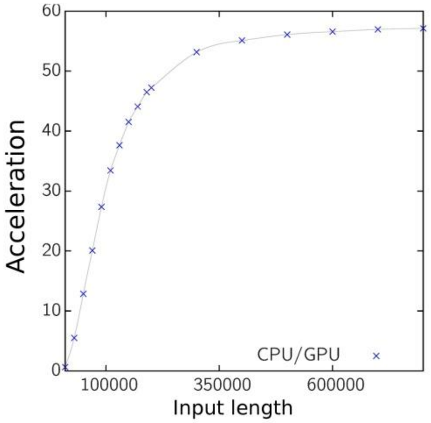


Fig 4:- Comparison of the computing timedepending on the input length. (Ref: oak-labs)

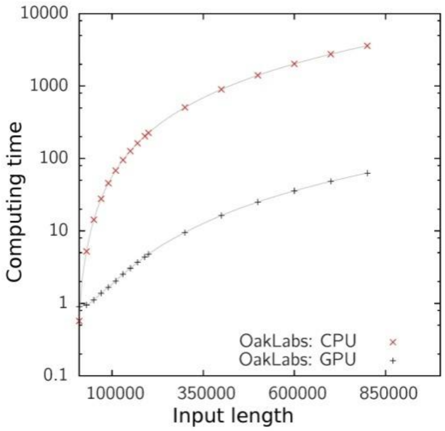


Fig 5:- Acceleration achieved in computations on GPU.

BarraCUDA:-

BarraCUDA utilises NVIDIA's GPGPU implementation called Compute Unified Device Architecture (CUDA) to parallelize the alignment of sequence reads. It is a next generation sequencing alignment software to perform mapping of sequencing reads to reference genomes using NVIDIA graphics cards also. The program loads the complete Burrows-Wheeler transform (BWT)-encoded reference sequence and sequence reads from disk to GPU memory. Then this is followed by launching a GPU alignment kernel. Here the alignment task of each of the sequence reads are distributed to hundreds of processors within the GPU and computations are performed in parallel. Once the kernel finishes the alignment results, then these are transferred from GPU back to disk.

BarraCUDA is based on BWA and can perform gapped alignment with gap extensions supporting mappings for single- and paired-end reads with comparable alignment accuracies. It also generates alignments in the SAM format for compatibility with downstream data analysis applications. In Bioinformatics using GPGPU, this software lays an important milestone in low-cost and energy efficient computing.

CUSHAW:-

CUSHAW is the first release of CUSHAW software package for next-generation sequencing read alignment. It is a CUDA compatible short read alignment algorithm for multiple GPUs sharing a single host. This aligner is designed based on the BWT and programmed using CUDA C++ parallel programming language. It is on Performance evaluation and uses the simulated and real short read datasets. It reveals that the aligner achieves significant speedups in terms of execution time. It yields comparable or even better alignment quality for paired-end alignments, compared to three popular BWT-based aligners like Bowtie, BWA and SOAP2. This aligner only provides support for ungapped alignment and has been incorporated in to NVIDIA Tesla Bio Workbench.

G-DNA:-

G-DNA is referred as GPU-based DNA aligner, is the first highly parallel solution that has been optimized to process nucleotide reads (DNA/RNA) from modern sequencing machines. The results show that the software is very efficient on both multi-GPU machines and MPI+GPU clusters. It computes scores and shifts for a given set of sequence pairs and might be used as an efficient aligning tool in de-novo assembly. G-DNA is an extremely efficient software tool that performs pairwise sequence alignment for selected pairs from a given set of nucleotide reads. The software is freely available and may be run on commodity hardware which makes it a perfect tool for the everyday scientific use. It is therefore perfectly suited to be used in the DNA assembly problem or Protein analysis.

DNA/RNA sequencing has recently become a primary way researchers generate biological data for further analysis ie. Protein sequence analysis. However, some of them require pair wise alignment to be applied to a great deal of reads as assembling algorithms are an integral part of this process.

Although several efficient alignment tools have been released over the past few years including those taking advantage of GPUs. But none of them directly targets high-throughput sequencing data. As a result there is a need to create software that could handle such data as effectively as possible. G-DNA, means GPU-based DNA aligner is the first highly parallel solution that has been optimized to process nucleotide reads ie., DNA/RNA from modern sequencing machines. Results show that the software reaches up to 89 GCUPS (Giga Cell Updates per Second) on a single GPU. As a result it is the fastest tool in its class. It scales up well on multiple GPUs systems, including MPI-based computational clusters where its performance is counted in TCUPS (Tera CUPS).

G-MSA:-

G-MSA is a valuable tool for the Multiple Sequence Alignment (MSA) problem which is efficient and can be run on a common personal computer equipped with NVIDIA GPU (G80, GT200 or Fermi). When extensive tests conducted they show its great speedup in comparison to the T-Coffee method on which it was based. The quality of the results remained very high and multi-GPU support influences the execution time considerably. Multiple sequence alignment methods are essential in biological analysis as several MSA algorithms have been proposed in recent years. The quality of the results produced by those methods is reasonable. But there is no single method that consistently outperforms others.

Besides, the increasing number of sequences in the biological databases is perceived as one of the upcoming challenges for alignment methods. The lack of performance concerns not only the alignment problems but also may be observed in many areas of biologically related research. To overcome this problem in the field of pair wise alignment several GPU computing approaches have been proposed which show a great potential of GPU platform. Performed tests show that the method G-MSA is highly efficient achieving up to 193-fold speedup on a single GPU while the quality of its results remains very good.

Conclusion:-

Traditional CPU parallel levels only allow instruction-level parallelism which can only improve the performance of the CPU by improving the working efficiency of processor. Under the traditional architecture, due to the limitations of storage mechanism, memory and bandwidth the processors that above 16 nuclear can't bring performance improvements for super computer. It might even lead to efficiency dropped substantially. Compared with the CPU multi-core computing, GPU advantage is that its high performance price ratio, direct visualization, low power dissipation and good portability. GPU-based general-purpose computing research has played a linking role both in academia and industry; it is of great deal of enthusiasm to the GPU-based parallel high performance algorithm.

The comparative study using GPU based analysis tools for human protein sequences is useful for increasing input length, sequence analysis of nucleotides (DNA, RNA) & proteins and comparisons for a broad range of applications. Further the study can be implemented for study on development and implementation of a Hybrid Monte Carlo algorithm for bimolecular and parallelization of algorithms for genomic selection also.

References:-

1. Casey, R. M. (2005). "BLAST Sequences Aid in Genomics and Proteomics". Business Intelligence Network.
2. Fei,X. et al. (2008) FPGA-based accelerators for BLAST families with multi-seeds detection and parallel extension. In: Proceedings of the 2nd International Conference in Bioinformatics and Biomedical Engineering. IEEE, Shanghai, China, pp. 58–62.
3. Jacob,A. et al. (2007) FPGA-accelerated seed generation in Mercury BLASTP. In: Proceedings of 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. IEEE, California, USA, pp. 95–106.
4. Sotiriades,E. and Dollas,A. (2007) A general reconfigurable architecture for the BLAST algorithm. J. VLSI Signal Process. 48, 189–200.
5. Zhang,Z. et al. (2000) A greedy algorithm for aligning DNA sequences. J. Comput. Biol., 7, 203–214.
6. Lin,H. et al. (2008) Massively parallel genomic sequence search on the Blue Gene/P architecture. In: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing. ACM/IEEE, Austin, USA, pp. 1–11.
7. Camacho,C. et al. (2009) BLASTþ: architecture and applications. BMC Bioinform., 10, 421.
8. Morgulis,A. et al. (2008) Database indexing for production MegaBLAST searches. Bioinformatics, 24(16), 1757–1764.
9. Nguyen,V.H. and Lavenier,D. (2009) PLAST: parallel local alignment search tool for database comparison. BMC Bioinform., 10, 329.
10. Owens,J.D. et al. (2008) GPU Computing. IEEE Proc., 96, 879–899.
11. Nickolls,J. (2007) Nvidia GPU parallel computing architecture. In: Proceedings of the IEEE Hot Chips 19. IEEE, Stanford, CA, USA.

12. Dematte,L. and Prandi,D. (2010) GPU computing for systems biology. Brief. Bioinform., 11, 323–333.
13. Liu,C.M. et al. (2012a) SOAP3: ultra-fast GPU-based parallel alignment tool for short reads. Bioinformatics, 28, 878–879.
14. Liu,Y. et al. (2012b) CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows–Wheeler transform. Bioinformatics, 28, 1830–1837.
15. Lu,M. et al. (2013) GPU-accelerated bidirected De Bruijn graph construction for genome assembly. Web Tech. Appl. Lect. Notes Comput. Sci., 7808, 51–62.
16. Lu,M. et al. (2012) High-performance short sequence alignment with GPU acceleration. Distrib. Parallel Dat., 30, 385–399.
17. Manavski,S. and Valle,G. (2008) CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. BMC Bioinform., 9 (Suppl. 2), S10.
18. Ling,C. and Benkrid,K. (2010) Design and implementation of a CUDA-compatible GPU-based core for gapped BLAST algorithm. Proc. Comput. Sci. USA, 1, 495–504.