Exploring Sequence Alignment Algorithms on FPGA-based Heterogeneous Architectures

Xin Chang[1], Fernando A. Escobar[1], Carlos Valderrama[1], and Vincent Robert[2]

[1] Service d'électronique et de Microélectronique, Université de Mons, Belgium.
[2] CBS-KNAW Fungal Biodiversity Centre, Utrecht, Netherland

Abstract. With the rapid development of DNA sequencer, the rate of data generation is rapidly outpacing the rate at which it can be computationally processed. Traditional sequence alignment based on PC cannot fulfill the increasing demand. Accelerating the algorithm using FPGA provides the better performance compared to the other platforms. This paper will explain and classify the current sequence alignment algorithms. In addition, we analyze the different types of sequence alignment algorithms and present the taxonomy of FPGA-based sequence alignment implementations. This work will conclude the current solutions and provide a reference to further accelerating sequence alignment on a FPGA-based heterogeneous architecture.

Keywords: Sequence Alignment Algorithm, FPGA, Heterogeneous Architectures, Systolic Array, Parallel Computation, Hardware Acceleration

1 Introduction

Bioinformatics is an emerging field which focuses on developing computational methods for collecting, handling and analyzing biological data leading to the discovery and fundamental understanding of the genetic composition in organisms.

Sequence alignment is one of the major research efforts in bioinformatics. Sequence alignment analyzes similarities between DNA, or protein sequences, to assess the genetic relationship between organisms or species. The sequence similarity may also be the consequence of structural or functional relationships. As other bioinformatics applications, sequence similarity analysis is facing a daunting challenge today because the rate of data generation is rapidly outpacing the rate at which it can be computationally processed [1]. Instruments for bioinformatics research, such as next-generation sequencers, are generating vast amounts of data - so much that traditional solutions can't keep up in time or space. As Fig. 1 shows, both the cost and time for DNA sequencing is shrinking spectacularly according to NHGRI Genome Sequencing Program (GSP) [2]. The requirements for the analysis of these data are growing dramatically far beyond what Moore's law can achieve. Sequence alignment shares the same features of the other bioinformatics applications such as data-intensive, data-independent and high parallelism potential. Thus, providing an efficient solution for sequence

alignment can also open the door for improving the other bioinformatics applications.

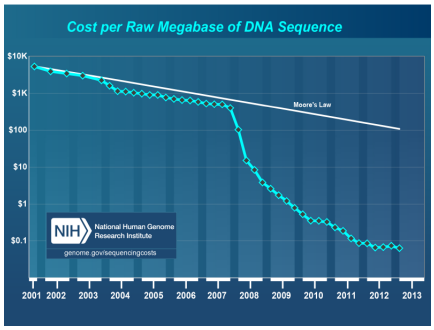

Fig. 1. Cost of per Raw Magebase of DNA Sequence [2]

This paper aims at profiling existing sequence alignment algorithms, providing taxonomy of sequence alignment algorithms focusing on potential candidates for heterogeneous architectures implementation, in particular, FPGA-based (Field Programmable Gate Array) reconfigurable architectures, and concluding with considerations for FPGA-based hardware acceleration of sequences alignment.

The rest of paper is organized as follows. The following section will introduce the basic classification of current sequence alignment algorithms. In Section 3, three distinctive algorithms based on dynamic programming, heuristic programming and hidden Markov model will be described and profiled in detail. After that, the taxonomy will be provided based on the evaluation results on Section 4. Finally, the conclusion will be provided in Section 5.

2 Background

The aims of sequence alignment algorithms are to find these differences between sequences and mark them as soon as possible. Based on the number of sequences can be processed at the same time, sequence alignment algorithms can be classified into pairwise and multiple sequence alignment algorithms . Dynamic programming, heuristic programming, hidden Markov model are the common methodologies for pairwise sequence alignment algorithms. The hybrid method is often used by multiple sequence alignment which is the combination of two or more pairwise sequence alignment algorithm. The typical algorithms of each type are listed in Fig. 2. From the result point of view, all the algorithms based on dynamic programming can be labeled as exact alignment, producing the best result all the time; while the others are labeled as approximated alignment because do not always produce the correct result.

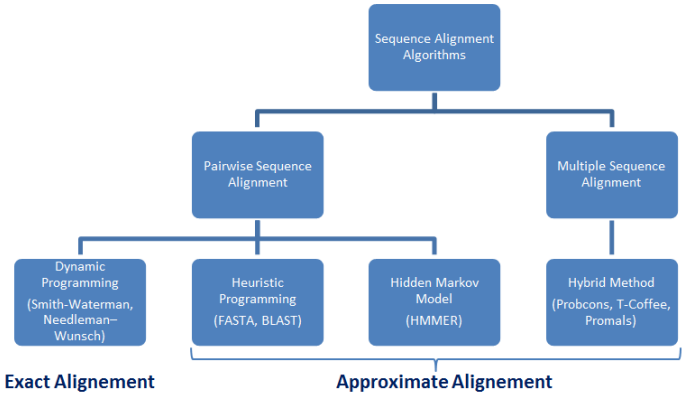

Fig. 2. Classification of Sequence Alignment Algorithms on FPGA

3 Sequence Alignment Algorithms

Currently, multiple sequence alignments are more significant and demand more time to compute than pairwise sequence alignments. There are several popular multiple sequence alignments tools such as Probcons[3], T-Coffee[4], Promals[5] and etc. The algorithms behind the tools are actually the extension and combination of pairwise sequence alignment algorithms. For instance, heuristic algorithms are used in the second stage of Promals, and it takes the overwhelming majority of computational time. So, the study to accelerate the pairwise sequence alignment is essential to improve the performance of multiple sequence alignment. This section will introduce the methodologies pairwise sequence alignment on FPGA in detail.

3.1 Dynamic Programming

Dynamic programming provides the most accurate sequence alignment. Moreover, it is also commonly used in the heuristic programming algorithms to refine the alignment results. However, due to the heavy computational load, dynamic programming is always time-consuming. Therefore, it is critical to find a way to boost their performance.

In pair-wise sequence alignment, two sequences build a matrix based on S-W algorithms. After tracing back from the highest score in the score matrix, the gap will be exposed and the alignment result will come out. The Smith-Waterman algorithm can be illustrated as follow:

$$H_{i,j} = \max \begin{cases} 0 \\ H_{i-1,j-1} + w(\text{match/mismatch}), \\ H_{i,j-1} - d \\ H_{i-1,j} - d \end{cases} \quad (1)$$

$H_{i,j}$ indicates the value in the position (i,j) of the matrix, w stands for the score of match/mismatch, while d means the penalty for gap. The score of each cell in the matrix is calculated by the above equation except for the first column and first row are filled with zeros. Back tracing from the highest score, there might be several paths back to the original point which represent the potential alignments. The alignment with highest sum-up score is the best alignment.

Based on Smith-Waterman algorithm we described above, the implementation on FPGA platform can be divided into 4 parts: Initialization, Matrix Calculation, Backtrack and Result Generation. Initialization refers to prepare the input sequence into the format which can be processed by FPGA. Matrix calculation means to calculate the matrix based on the algorithm in order to find out the maximum score. Backtrack intends to find out the best alignment by tracking back from the cell with maximum score. Result generation aims at output the result in a direct view.

3.2 Heuristic Programming

Algorithms such as BLAST[6] and FASTA[7] are heuristic and give less optimal results compared to dynamic programming. However, the heuristic programming provides much faster sequence alignment than exhaustive dynamic programming algorithms. Indeed, instead of comparing every residue against every other as in dynamic programming, heuristic algorithms partition the sequences into smaller pieces and compare them. Since the basic principle of heuristic algorithms is similar, in this paper we will use the most popular one, the BLAST heuristic algorithm, as example.

In BLAST, input sequences are split into short segments named words (w) to create alignment seeds. BLAST will create a words list from the query sequence with words of specific length that can be defined by the user. Once an alignment is seeded, BLAST extends the seeds in both directions to longer alignments which called High Score Pairs (HSP). Only the HSPs with score higher than the cutoff score S will be taken into further consideration. S is defined by the users. An approximate alignment will be created after the joining the adjacent HSPs together. If there is a mutation or gap between HSPs, dynamic programming can then be used to align the gaps.

There are several versions of BLAST implemented on FPGA, for instance BLASTn, BLASTp, BLASTx, TBLASTn and TBLASTx. They all provide the functionality to compare all possible combinations of query and database se-

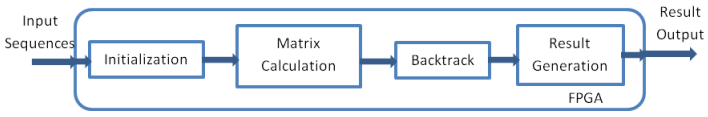

Fig. 3. Work Flow of Smith-Waterman Algorithm on FPGA

quences. However, for each type of search operation, the architectures are almost identically. BLAST can be implemented on FPGA as shown in the Fig. 4.

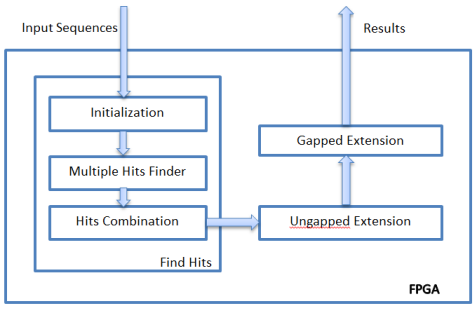

Fig. 4. Block Diagram of BLAST on FPGA

BLAST on FPGA is composed by three main blocks: Find Hits, Ungapped Extension and Gapped Extension. The Find Hits block initializes the input sequences, creates words, detects sub-strings in the database which perfectly match a substring of the query sequence and records the positions of those having an exact match. The Ungapped Extension block extends, in both directions, the exact match to identify a pair of longer sequences between the query and the database. The Gapped Extension block, the last stage of BLAST, uses Smith-Waterman to extend the previous result into a gapped alignment.

3.3 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) was used in speech recognition for a long time. Until 1994, the HMM was first introduced to biology since it well-suited for multiple sequence alignments. A HMM is a probabilistic finite state machine to represent a sequence pattern. Hmmsearch [8] is a tool based on HMM to search a sequence database for matches to an HMM. Fig. 5 shows an example of Hidden Markov Model.

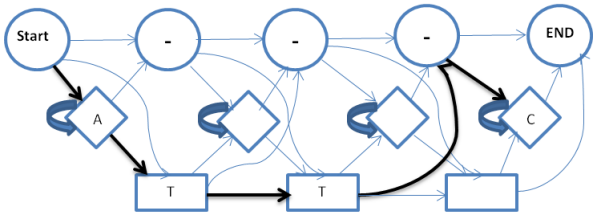

Fig. 5. Probabilistic Finite State Machine of Sequence "ATTC"

In Fig. 5, the sequence "ATTC" found its path in HMM. Any sequence can be represented by a path in the model. The probability of any sequence, given the model, is computed by multiplying the emission and transition probabilities along the path. In order to avoid floating point errors during computations, it is best to transform the probabilities to their logarithms. The resulting score is called the raw score of the sequence, given the HMM. There might be many paths can build an alignment. One way to find the most optimized way is Viterbi algorithm[9]. Viterbi algorithm tries to find out the most probable path leading from a particular amino acid's emission to another at every stage in its journey. It compares the corresponding path of each path and chooses the one with the highest possibility.

The HMM algorithm is complicated; however the implementation on FPGA is not too sophisticated. HMM can be pipeline in different number of stages. FPGA reads a profile HMM and a sequence database as its inputs. It repeatedly reads one sequence from the sequence database and sent it to a process pipeline to process until all the sequences in database are processed. All sequences sent to the pipeline will process by two or three filters.

3.4 Profiling of Sequence Alignment Algorithms

In order to explore the parallelism of sequence alignment algorithms on FPGA, we started by profiling the algorithms running on PCs. The following figures show the percentage of time-consumption by each block of Smith-Waterman, BLAST and HMM. The results are achieved under Intel i5 dual-core working at 2.26 GHz.

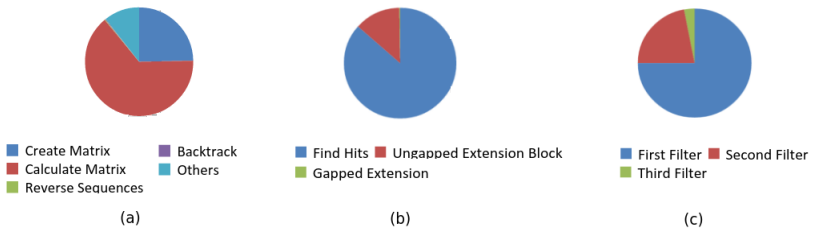

Fig. 6. Time-Consumption by Each Function in S-W algorithm, BLAST and HMM algorithm

Fig. 6. (a) shows the time consumption details of Smith-Waterman algorithm, among four major parts, the most expensive part is the matrix calculation. And as the sequence gets longer, the percentage of matrix calculation will get even larger. For two sequences with 10000 nucleotides, the matrix calculation can take more than 90% of the whole execution time. It is an excellent candidate for accelerating on FPGA since its inherent parallelism. Thus, the major research ef-

fort was in accelerating matrix calculation [10] [11] [12]. The differences between different implementations are also focused on the Matrix Calculation part.

Fig. 6. (b) indicates that the most execution time of BLAST is spent in first two blocks over 99%. Especially in the Find Hits block, it takes over 80% time. Thus, the main research focuses on improving the efficiency of finding hits [13] [14]. In addition, since the Ungapped Extension Block is essentially based on dynamic programming. It also uses the same techniques for dynamic programming to further enhance the performance.

In Fig. 6. (c), the percentage of execution time by each filter in a three-stage HMM is illustrated. About 2% of all sequences will pass first filter and be sent to the next filter, and 0.1% sequences will reach the third filters. And profile result shows that first filter occupies 75% of total execution time and the second filter occupies 22%. Many researches concentrated on offloading the first filter of process pipeline onto FPGA to accelerate the whole program [15][16].

4 Taxonomy of Sequence Alignment Algorithms on FPGAs

This section will compare the sequences alignment algorithms on FPGAs with criteria such as algorithm parallelism, performance speed up, FPGA resource utilization and communication interface.

4.1 Parallelism Exploration

The parallelism of the sequence alignment algorithms can be achieved at two levels. First level is to align one input sequence against different sequences from a database at the same time, by duplicating the algorithm processing unit. The second level depends on the internal parallelism of algorithms itself.

Systolic array is usually applied to achieve this parallelism of algorithms on FPGAs. The systolic array is made of a pipe network arrangement of processing element. Each processing element can compute and store the data of each other. Fig. 7. shows an example of one-dimensional systolic array of processing element.

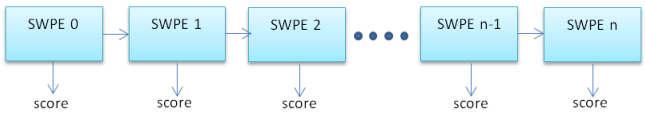

Fig. 7. One Dimensional Systolic Array of PEs

The sequence alignment algorithms can take advantage of systolic array to realize the parallelism. The number of PE contributes to performance increase.

Table 1 shows the several FPGA implementation of sequence alignment algorithms. In the following tables, DP represents the dynamic programming while HR stands for heuristic methods.

Type	Platform	Frequency	PE Number	Speed-up	Reference Design	Ref
DP	XC4VLX160	15MHz	500	172x	AMD Opteron	[17]
DP	XC5VLX330	133MHz	512	500x	Intel Q9400	[18]
DP	XC6VLX760	133MHz	1024	100x	Intel Q9400 CPU	[18]
DP	XC2V6000-4	47.7MHZ	252	241x	1.6GHz Pentium 4	[19]
DP	XC2V6000-4	47.7MHZ	168	160x	1.6GHz Pentium 4	[19]
DP	Spartan-3E	50MHz	92	92x	1.6GHz Pentium 4	[20]
DP	EP2S180-3	66.7MHz	384	185x	AMD Opteron	[21]
HR	XC4VLX160	15MHz	8	22-40x	2.2GHz Intel Centrino	[13]
HR	XC5VLX110T	136HMz	2048	45x	2.8GHz Intel Core i7	[23]
HR	XC2VP70	145MHz	2048	45x	2.8GHz Intel Core i7	[23]
HR	XC4VLX160	-	2048	45x	2.8GHz Intel Core i7	[23]
HR	EP2S130C5	113MHz	3072	-	2.6GHz Pentium 4	[24]
HR	XC4VLX160	189MHz	-	71x	2.6GHz Pentium 4	[24]
HR	XC4VLX160	178MHz	1024	7x	2.6GHz Pentium 4	[25]
HMM	XC6VLX760	100MHz	128	250x	Intel Xeon E5520	[26]
HMM	XC6VLX760	100MHz	24	60x	Intel Xeon E5520	[26]
HMM	XC5VLX110T	130MHz	25	67x	3.2GHz Pentium 4	[27]
HMM	XC2VP100	100MHz	90	-	2.8GHz Pentium 4	[28]
HMM	XC3S1500	70MHz	10	30x	AMD Operon	[29]
HMM	EP2S180	67MHz	85	184x	-	[30]
HMM	EP1S25F102	76.9MHz	44	267x	-	[31]

Table 1. State-of-Art Implementations of Sequence Alignment on FPGA

We can tell from the Table 1 that the algorithm based on dynamic programming and HMM is easier to achieve the better improvement; even the number of PEs is not as much as heuristic and HMM methods. Compare to heuristic methods, HMM also has more potential to be speedup. In addition, the performances of the implementations of the same type are proportional to the higher frequency and bigger number of PEs.

There are two main reasons limit the maximum number of PEs: The first one is the data dependence of the algorithm itself. For instance, in order of keep PE as simplify as possible, the maximum number of PE is constrained by the sum of query and reference sequences' lengths. The other reason is the available resource of FPGA is limited.

4.2 FPGA Resource Utilization

Nowadays, FPGAs are consists of large resources of logic gates and RAM blocks. For the bioinformatics applications such as sequence alignment, they are both

computational and data intensive applications. Thus, not only the logical elements need attention, the Block RAMs on chip also needs to take into consideration. Table 2 shows the researches of accelerating sequences alignment related to the FPGA resources utilization. It is obvious shows that, in general, more

Type	Platform	PE	Slices	BRAM	Speedup	Reference Design	Ref
HMM	XC3S1500	10	34%	-	30x	AMD Opteron	[29]
HMM	XC6VLX760	24	61%	-	60x	Xeon E5520	[26]
HMM	XC5VLX110T	25	90%	-	67x	Pentium 4	[27]
HMM	EP2S180	85	84%	-	184x	-	[30]
HMM	XS2VP100	90	98%	-	-	Pentium 4	[28]
HMM	XC6VLX760	128	97%	-	250x	Xeon E5520	[26]
DP	EP2S180-3	384	78%	57%	185x	AMD Opteron	[21]
DP	XC4VLX160	500	81.3%	-	172x	AMD Opteron	[17]
HR	XC4VLX160	1024	78%	88%	7x	Pentium 4	[25]
HR	XC5VLX110T	2048	55%	20.3%	45x	Intel Core i7	[23]
HR	XC2VP70	2048	87%	42%	45x	Intel Core i7	[23]
HR	XC4VLX160	2048	59.2%	28%	45x	Intel Core i7	[23]
HR	EP2S130C5	3072	87%	11%	-	Pentium 4	[24]
HR	XC4VLX160	-	71%	38%	71x	Pentium 4	[24]

Table 2. Resource Utilization by Different Implementations of Sequence Alignment on FPGA

advance FPGA with more resource on chip can achieve higher performance. Furthermore, besides the resource used by the systolic array, the control units also take a portion of FPGA resources. For instance, in [24], the systolic array takes 70% of used FPGA resource while the control unit takes 26% percentage of used FPGA resources. It means the determination of systolic array size is also related to the control unit and BRAM required.

4.3 Communication Interfaces

Since, the sequence alignment is a data-intensive application, the size of query and database sequences is not negligible, and the configuration command and final score also need to transfer between the host and FPGA. It requires the enough bandwidth to support instead of being the performance bottleneck. Using FPGA as a hardware accelerator, the communication interface between host and the FPGA board is necessary. The current solutions include PCI-bus, USB and Ethernet connections.

The most common solution is the combination of PCI Express and DDR2 memory. PCI Express interface allows efficient data transfer between the host and FPGA, while DDR2 can be utilized to store large mount amounts of data. The maximum data-rate of one PCIe is limited to 250MB/s due to the PCIe connection. By adding extra PCIe connections, the bandwidth can reach 1 GB/s

which fulfill the some implementation's specifications. The disadvantage of the high throughput is a considerable amount of latency depending on the path the data are traveling.

With the development of USB technology, the bandwidth of USB 3.0 can reach to 4G/s [32]. It can also easily fit the requirements of data transferring for sequence alignment. It provides another option to transfer data between FPGA and host. For instance, in [13][22] and etc, USB is applied to support the communication between host and FPGA.

Sometimes, the FPGA-based sequence alignment acceleration is used as the web server to provide the service remotely. In this case, Ethernet turns out to be an option, since it can provide a direct access to the Internet. [33] [34] used the Ethernet to establish the communication with the host. Although Ethernet is easier to get the access to the Internet, the maximum speed of Ethernet is not as fast as USB or PCIe. [34] is implemented as a cluster of FPGA with a hard drive, all the database sequences are stored in the hard drive. FPGA doesn't need to load the database through the Ethernet which greatly relieves the bandwidth bottleneck of Ethernet. Besides that, the data which transfer through the Ethernet can be compressed. This also contributes to lightweight the bandwidth.

5 Conclusion and Future Work

This paper concludes the current sequence alignment implementation on FPGA board in order to represent a future direction for accelerating sequence alignment. After introducing the basic principles of sequence alignment algorithms, classification and profiling, numerous of FPGA implementations are detailed and compared. It reveals the common features of different implementations by exploring the parallelism of the algorithm and FPGA utilization. In addition, three communication interfaces between FPGA and hosts are introduced. This paper as a reference presents the directions and considerations for accelerating the sequence alignment on FPGA platform.

6 References

References

1. S. Sarkar, G. R. Kulkarni, P. P. Pande, and A. Kalyanaraman. Network-on-Chip Hardware Accelerators for Biological Sequence Alignment. IEEE Trans. Comput. 59, 1 (January 2010), 29-41.
2. NCBI, GenBank release note 2013, `http://www.ncbi.nlm.nih.gov`
3. Chuong B Do, Mahathi S P Mahabhashyam, Michael Brudno, and Serafim Batzoglou. ProbCons: Probabilistic consistency-based multiple sequence alignment,Genome Res 15(2):330–40,February 2005
4. J. Rius and F. Cores and F. Solsona and van Hemert, J.I. and J. Koetsier and C. Notredame,A user-friendly web portal for T-Coffee on supercomputers,BMC Bioinformatics, Vol 12, 2012.

5. Jimin Pei, and Nick V. Grishin.PROMALS: towards accurate multiple sequence alignments of distantly related proteins.Bioinformatics 23(7):802-808 (2007)
6. Ian Korf,Mark Yandell and Joseph Bedell, BLAST,O'Reilly & Associates, Inc. Sebastopol, CA, USA 2003,ISBN:0596002998.
7. William R. Pearson,Using the FASTA Program to Search Protein and DNA Sequence Databases,Computer Analysis of Sequence Data: Part I, Methods in Molecular Biology,Vol 24, pp 307-331,1994
8. Rahul P. Maddimsetty, Jeremy Buhler, Roger D. Chamberlain, Mark A. Franklin, and Brandon Harris. Accelerator design for protein sequence HMM search.ICS, page 288-296. ACM, (2006)
9. G. D. Forney,The viterbi algorithm,Proceedings of the IEEE, Vol. 61, No. 3, pp. 268-278, March 1973.
10. Scott Lloyd, Quinn O.Snell. Hardware Accelerated Sequence Alignment with Traceback.International Journal of Reconfigurable Computing, Volume 2009, January 2009 Article No. 9
11. Altera White Paper. Implementation of the Smith-Waterman Algorithm on a Reconfigurable Supercomputing Platform, September 2007.
12. A. Nawazn, M. Nadeem, H. van Someren, K. Bertels. A parallel FPGA design of Smith-Waterman traceback.2010 International Conference on Field-Programmable Technology(FPT), 8-10 Dec.2010.
13. Kasap, S., Benkrid, K., Liu, Y., "Design and Implementation of an FPGA-based Core for Gapped BLAST Sequence Alignment with the Two-Hit Method", Engineering Letters, Vol. 16, Issue: 3, pp. 443-452,2008.
14. Jacob, A.; Lancaster, J.; Buhler, J.; Chamberlain, R.D., "FPGA-accelerated seed generation in Mercury BLASTP," Field-Programmable Custom Computing Machines, 2007. FCCM 2007. 15th Annual IEEE Symposium on , vol., no., pp.95,106, 23-25 April 2007
15. Y. Sun, P. Li, G. Gu, Y. Wen, Y. Liu, and D. Liu. Accelerating HMMer on FPGAs Using Systolic Array Based Architecture, in proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2009.
16. K. Benkrid, P. Velentzas, and S. Kasap, "A High Performance Reconfigurable Core for Motif Searching Using Profile HMM, " presented at Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on, 2008.
17. Ying Liu; Benkrid, K.; Benkrid, A.; Kasap, S., "An FPGA-Based Web Server for High Performance Biological Sequence Alignment," AHS 2009. NASA/ESA Conference on Adaptive Hardware and Systems, 2009, pp.361-368, July 29 2009-Aug. 1 2009
18. Dan Zou, Yong Dou, Fei Xia,Optimization schemes and performance evaluation of SW algorithm on CPU,GPU and FPGA, Concurrency and Computation: Practice and Experience, Vol 24.pp 1625-1644.
19. K. Benkrid, Ying Liu, A. Benkrid, "A Highly Parameterized and Efficient FPGA-Based Skeleton for Pairwise Biological Sequence Alignment," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.17, no.4, pp.561,570, April 2009
20. Chris Fenton, SeqAlign, Internet Source: http://www.chrisfenton.com/seqalign/
21. Peiheng Zhang, Guangming Tan, Guang R. Gao, Implementation of the Smith-Waterman Algorithm on a Reconfigurable Supercomputing Platform in proceeding of HPRCTA '07 Proceedings of the 1st international workshop on High-performance reconfigurable computing technology and applications, pp 39-48
22. Fei Xia, Yong Dou, Jiaqing Xu, Families of FPGA-Based Accelerators for BLAST Algorithm with Multi-seeds Detection and Parallel Extension, p43-57, Springer Berlin Heidelberg, 2008

23. Ouano, M.O.L. ; Jongco, G.F.V.M.G.C.D. ; Escabarte, E.B., FPGA Based AGREP for DNA Microarray Sequence Searching, in proceedings of International Conference on Computer Engineering and Applications, 2009, pp 217-221
24. Xiaoqiang Li, Wenting Han, Gu Liu, Hong An, Mu Xu, Wei Zhou, Qi Li , A Speculative HMMER Search Implementation on GPU, 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, CHINA, 2012
25. Nathaniel McVicar, Walter L. Ruzzo, Scott Hauck, Accelerating ncRNA Homology Search with FPGAs, in Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays, 2013 ,pp 43-52 ,
26. Y. Sun, P. Li, G. Gu, Y. Wen, Y. Liu, and D. Liu. Accelerating HMMer on FPGAs Using Systolic Array Based Architecture, in proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2009.
27. K. Benkrid, P. Velentzas, and S. Kasap, "A High Performance Reconfigurable Core for Motif Searching Using Profile HMM, " presented at Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on, 2008.
28. John Paul Walters, Xiandong Meng, Vipin Chaudhary, Tim Oliver, Leow Yuan Yeow, Bertil Schmidt, Darran Nathan, Joseph Landman, MPI-HMMER-Boost:Distributed FPGA Acceleration, The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, 2007, Volume 48, Number 3, Page 223
29. Rahul P. Maddimsetty , Jeremy Buhler , Roger D. Chamberlain , Mark A. Franklin , Brandon Harris, Accelerator design for protein sequence HMM search, Proceedings of the 20th annual international conference on Supercomputing, June 28-July 01, 2006, Cairns, Queensland, Australia
30. Morales Snchez, Jos Luis, Hardware Design of Algorithm for the Classification of rna sequences, Diss. burgerlijk ingenieur computerwetenschappen, 2006
31. "USB". Implementers Forum. 2013-01-06. Retrieved 2013-01-06 `http://www.usb.org/press/USB-IF_Press_Releases/2008_11_17_USB_IF.pdf`
32. Adam Hall, Short Read DNA sequence Alignment with custom Designed FPGA-based Hardware Dissertation, University of British Columbia, 2011-05
33. Xinyu Guo, Hong Wang, and Vijay Devabhaktuni,A Systolic Array-Based FPGA Parallel Architecture for the BLAST Algorithm, ISRN Bioinformatics, vol. 2012, 11 pages, 2012.