HAMSA: Highly Accelerated Multiple Sequence Aligner

Naglaa M. Reda
Dept. of Mathematics,
Faculty of Science, Ain Shams University,
Cairo, Egypt

Mohammed Al-Neama
Dept. of Computer Science,
Faculty of Science, Cihan Univeristy,
Sulimanya, Kurdstan Iraq

Fayed F. M. Ghaleb
Dept. of Mathematics,
Faculty of Science, Ain Shams University,
Cairo, Egypt

***Abstract*—For biologists, the existence of an efficient tool for multiple sequence alignment is essential. This work presents a new parallel aligner called *HAMSA*. *HAMSA* is a bioinformatics application designed for highly accelerated alignment of multiple sequences of proteins and DNA/RNA on a multi-core cluster system. The design of *HAMSA* is based on a combination of our new optimized algorithms proposed recently of vectorization, partitioning, and scheduling. It mainly operates on a distance vector instead of a distance matrix. It accomplishes similarity computations and generates the guide tree in a highly accelerated and accurate manner. *HAMSA* outperforms MSAProbs with 21.9-fold speedup, and ClustalW-MPI of 11-fold speedup. It can be considered as an essential tool for structure prediction, protein classification, motive finding and drug design studies.**

***Keywords*—Bioinformatics; Multiple sequence alignment; parallel programming; Clusters; Multi-cores**

I. Introduction

Although diverse methods for aligning multiple sequences have been designed, the accomplishment of alignment's vast computations in a highly accelerated and accurate manner is still a challenge [1]. Most multiple sequence alignment (MSA) tools utilize the progressive method because it is computationally efficient.

First, it calculates a distance matrix illustrating the divergence of each pair of sequences by a similarity score. *Second*, it uses a clustering method to create a guide tree constructed from pairwise sequence distances. *Third*, it builds up the final multiple alignments according to the order given by the guide tree. Some other MSA tools follow the iterative method. It makes an initial alignment of groups of sequences and then revises the alignment to achieve a more reasonable result.

The main contributions of this work are:

1) Designing a highly accelerated parallel tool for aligning multiple sequences on multicore clusters, called *HAMSA*, to align massive sequences rapid,
2) Implementing the proposed HAMSAtool using C++ with MPI and OpenMP on Bibliotheca Alexandrina cluster system, to testits quality,
3) Carrying out comprehensive tests on a variety of actual dataset sizes, to prove that our developed tool outperforms competitive existing tools.

The rest of this paper is organized as follows. *Section 2* summarizes briefly the fundamental tools used for aligning multiple sequences. *Section 3* explains the *HAMSA* proposed tool. *Section 4* presents results and comparisons with diagnosis. Finally, *Section 5* concludes the paper and suggests future work.

II. Related Work

In the last decade, various parallel MSA programs have been developed for reducing time consumption and handling big data. They differ in the parallel platform they use and the way they optimize computations and storage.

The ClustalW is the commonly used tool. Different versions of ClustalW have been developed for shared memory SGI Origin machine, multiprocessors, clusters, and GPUs [2]. Although ClustalW is the most highly cited aligner especially for huge number of sequences, its accuracy is not satisfied enough compered to T-Coffee and MSAProbs. And, it has a problem with long sequences.

The T-Coffee produced two parallel versions for clusters and clouds [3]. The MUSCLE's first parallel attempt was on SMP system then multiscale simulations for HPC cluster and Amazon AWS cloud have been presented [4]. Booth T-Coffee and MUSCLE achieves high accuracy and fast speed, but cannot handle large sized dataset.

The MAFFT have been parallelized using the POSIX Threads library for multi-core PCs [5]. MAFFT is very fast, nevertheless it has poor accuracy.

ParaAT is used to construct multiple protein-coding DNA alignments for a large number of homologs on multi-core machines [6]. It is very good tool for large-scale data analysis, however its speed is unsatisfied.

MSAProbs was optimized for modern multi-core CPUs by employing a multi-threaded design [7]. While MSAProbs is the best tool for demonstrating dramatically accurate alignment, it is very slow.

The parallel version of DIALIGN-TX was implemented using both OpenMP and MPI on a 28-cores heterogeneous cluster [8]. Its speed and accuracy are neutral.

The GPU-REMuSiC [9] was proposed to reduce the computation time of RE-MuSiC; the newest tool with the regular expression constraints. It has good speed but it cannot align long sequences.

III. *HAMSA* PROPOSED TOOL

The main goal of *HAMSA* is to provide biologists with an accelerated multiple sequence aligner with minimal space consumption. It consists of three different stages. The architecture showing the interaction between these three stages is presented in *Fig. (1)*. To illustrate how *HAMSA* works by going through its three stages, the processes of aligning a class of 35 HIV viruses is introduced as a case study.

The distance vector *DV* including the similarity scores between every pair for the input sequences, computed by Stage (1), is introduced in *Fig. (2)*. While, the phylogeny guide tree, constructed by Stage (2), is presented in *Fig. (3)*. And, the final alignment, resulted from Stage (3), is given in *Fig. (4)*. In the following, a detailed discussion of each stage is given.

A. Distance vector computation

Stage (1), the distance vector computation, takes as input a set of N sequences with average length L and produces a distance vector *DV* including all pairwise distances. Each DV cell contains the similarity score for pair of sequences, evaluated by using the *DistVect1* algorithm, proposed in [10].

DistVect1 is a highly efficient parallel vectorized algorithm for computing similarity distances using multicore clusters. It deduces an efficient approach of partitioning and scheduling computations that consumes less space and accelerates computations. *DistVect1* was mainly based on the accelerated vectorization algorithm *DistVect* proposed in [11].

DistVect has solved the problem that real biological applications face when the length of sequences is large and the memory requirement cannot be met.

Instead of seeing the process of distance calculations in a two-dimensional array (matrix), *DistVect* algorithm substitutes the matrix by three vectors only. One vector includes the anti-diagonal of the current computed score, and two other vectors save the two previously calculated anti-diagonals including Northern, Western and North-Western needed values. It reduces the space complexity from $O(L^2)$ to $O(L)$.

DistVect1 also presented a superior performance with very long sequences. For example, for aligning 200 sequences of length 30,488, ClustalW-MPI did not work, SSE2 [12] exhausted 22,949 sec., where *DistVect1* achieved it in 8,017 sec only. This accomplishment is due to its perfect vectorization and hybrid partitioning approaches.

B. Guide tree construction

Stage (2), the guide tree construction, takes the evaluated distance vector *DV* and computes a guide tree. It uses the optimized NJ phylogeny reconstruction algorithm (*NJVect*) presented [13]. *NJVect* is a massively parallel optimized algorithm that compensates matrices used in *NJ* [14] by vectors.

It achieves remarkable reductions in both time and space by eliminating redundant computations and breaking dependences, while preserving the accuracy. It outperforms ClustalW-MPI with 2.5-fold speedup.

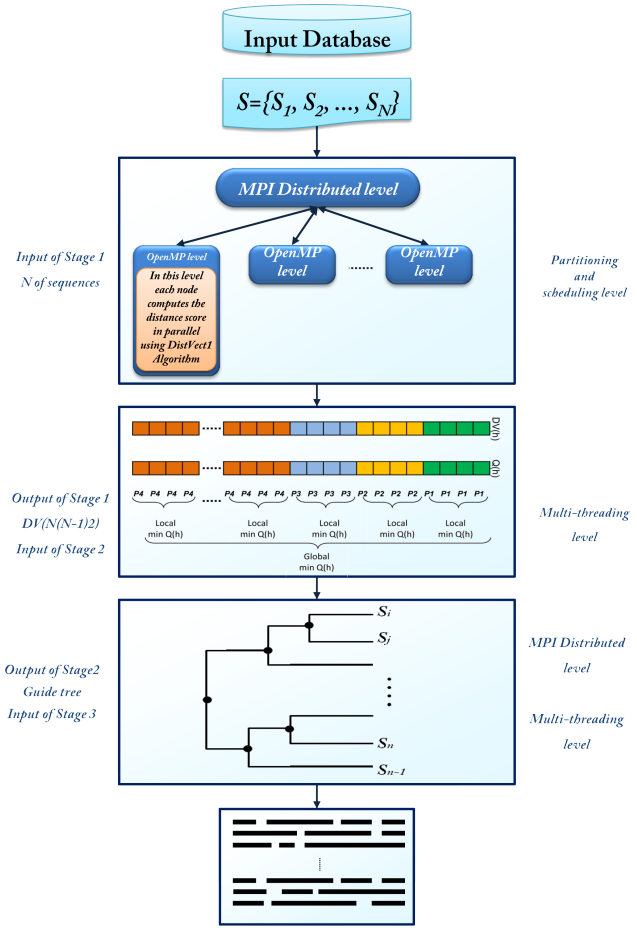

Fig. 1: Architecture of HAMSA.

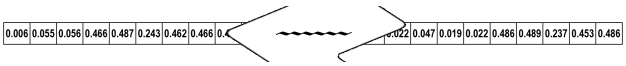

Fig. 2: Distance vector *DistVect1* of pairwise similarity distances.

C. Progressive alignment

Stage (3), the progressive alignment, uses the method provided by ClustalW-MPI for achieving progressive alignments. Its main objective is to distribute all external nodes (n) in the guide tree to be aligned in parallel. The efficiency obviously depends on the topology of the tree. For well-balanced guide tree, the ideal speedup is estimated as n/log n, where n is the number of nodes in the tree.

IV. EXPERIMENTAL RESULTS

HAMSA was implemented in C++, with MPI and OpenMP libraries. It accomplishes the alignment's vast computations in a highly accelerated and accurate manner. The experimental tests were conducted on Sun Microsystems cluster of 32 nodes, provided by LinkSCEEM-2 systems at Bibliotheca Alexandrina, Egypt. Each node contains two Intel Quad core Xeon 2.83 GHz processors (64 bit technology), with 8 GB RAM and 80 GB hard disk, a dual port in fin band (10 Gbps).

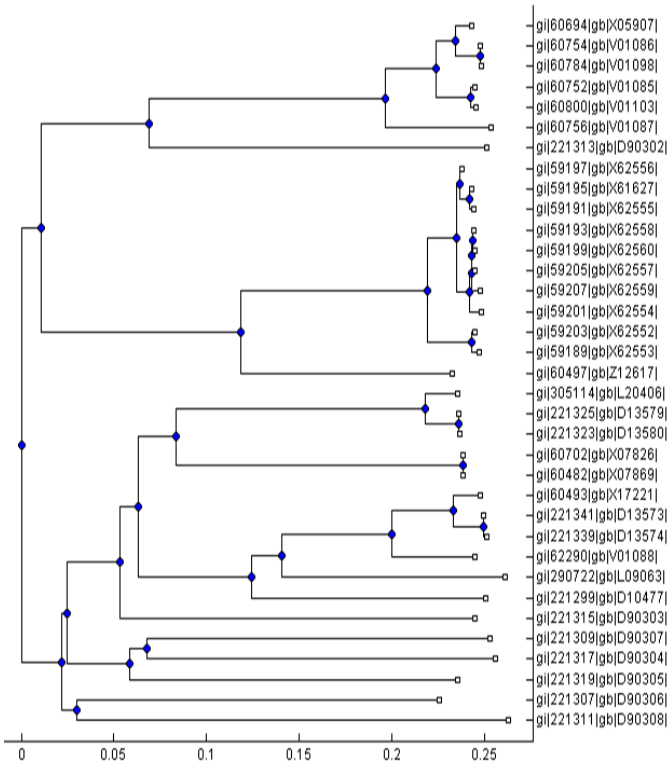

Fig. 3: Resulted phylogeny guide tree.

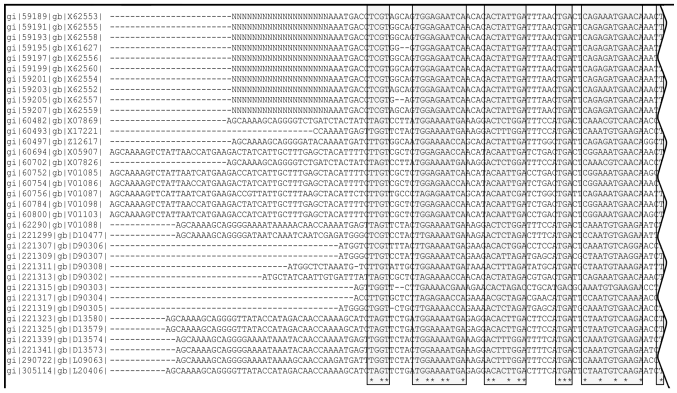

Fig. 4: Aligned sequences for 35 HIV viruses.

The experiments have been conducted using four protein real sequence datasets. These sequences have lengths ranging from 400 to 163,000 DNA residues, which made it possible to study the overall performance of solution against multiple different sizes.

The datasets consist of sequences selected from NCBI [15] and it was comprised of a subset of the Human Immunodeciency Virus (HIV), the Coronaviridae family viruses (COR), the Hemagglutinin (Inuenza B virus (HA)), Herpesviridae (large family of DNA viruses (HRV)), and Plasmid (large family of DNA bacteria Enterobacteriaceae (ENA).

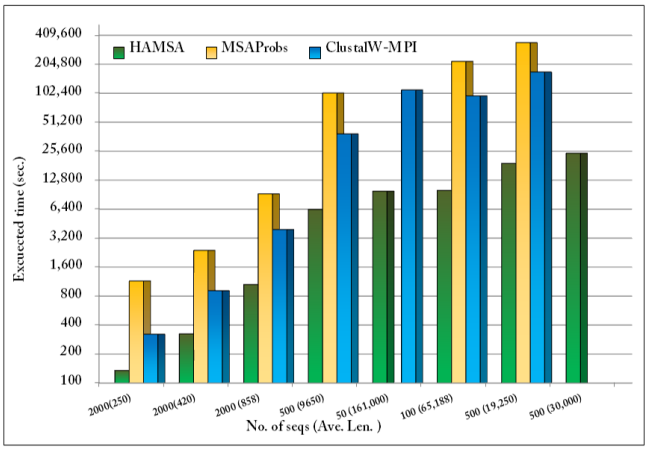

Fig. 5: Execution time measurements.

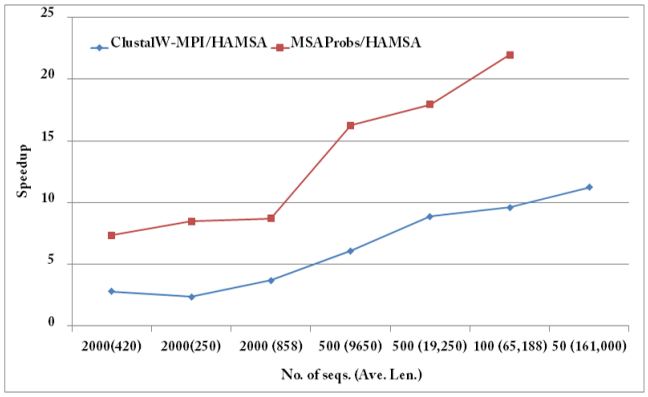

Fig. 6: Speed-up performance comparisons using 32 nodes.

The performance of *HAMSA* has been evaluated by using different metrics, such as: storage, execution time, speedup, efficiency, GCUPS, and occupancy. Its performance has been compared to ClustalW-MPI 0.13 [16], SSE2 [17], and MSAProb 3.0 [18]. *HAMSA* is able to handle the memory perfectly while computing distances between very long sequences; up to 163 k.

Fig. (5) shows that *HAMSA* has exhausted less execution time when aligning variant number of sequences (*N*) with different lengths (*L*) with respect to others. *Fig. (5)* shows that *HAMSA* has ability to achieve the maximum speedup with respect to the ClustalW-MPI and MSAProps for aligning the set 50 (161,000) and the set 100 (65,188), respectively.

Furthermore *HAMSA* can also achieve high GCUPs up to 13.835, while the highest values for ClustalW-MPI and MSAProps were 1.17 and 0.47, respectively, as shown in *Table (I)*. Furthermore, results recorded in *Table (II)* emphasize *HAMSA*'s high efficiency level against others. For CPUs occupancy, *HAMSA* reaches 100%.

TABLE I: *HAMSA* performance comparison in GCUPS

No. of seqs (Ave. Len.)	*HAMSA*	ClustalW-MPI	MSAProbs
500 (30,000)	9.7934	0.870756646	0
2000 (420)	3.2945	1.168593927	0.447014925
2000 (250)	2.83552	1.196421941	0.334377358
2000 (858)	4.1047	1.103413978	0.471208816
500 (9,650)	5.4928	0.901271638	0.338434997
500 (19,250)	7.249043	0.816977685	0.404296877
100 (65,188)	6.34465	0.661107638	0.288997449
50 (161,000)	13.835	0	0

TABLE II: Efficiency of *HAMSA* comparisons using 32 nodes

No. of seqs (Ave. Len.)	ClustalW-MPI	MSAProbs
2000 (420)	0.0881	0.2303125
2000 (250)	0.0740625	0.265
2000 (858)	0.11625	0.27221875
500 (9,650)	0.190453125	0.5071875
500 (19,250)	0.27728125	0.5603125
100 (65,188)	0.29990625	0.6860625
50 (161,000)	0.35146875	0

V. CONCLUSION AND FUTURE WORK

In this paper, *HAMSA* has been proposed for aligning multiple sequences efficiently by using a multi-core cluster system. *HAMSA* applies several optimization methods considering the memory usage and load balancing.

It provides a powerful improved storage handling capabilities with efficient improvement of the overall processing time. The beneficial of *HAMSA* is in relating the molecular structure to the underlying sequences as well as it can operate on local or online databases.

Experimental results show that *HAMSA* is an accelerated competitive MSA tool. *HAMSA* achieves speedup of 21.9 by comparing to MSAProbs and speedup of 11 by comparing to ClustalW-MPI.Its efficiency reaches 0.29, 0.086 and 0.092 over the ClustalW-MPI, SSE2 and MSAProbs, respectively.

Its performance varies from a low of 6.27 GCUPS to a high of 13.835 GCUPS as the lengths of the query sequences increase from 1,750 to 30,500, it also accomplishes 100

ACKNOWLEDGMENT

The authors would like to thank Bibliotheca Alexandria, Egypt, for granting the access to its platform and for the technical support of its Supercomputer laboratory professional team members.

REFERENCES

[1] F. F. M., N. M., and M. W., "An overview of multiple sequence alignment parallel tools," in *Proc. CSCCA '13*, Dubrovnik, Croatia, 2013, pp. 91–96.

[2] C.-L. Hunga, Y.-S. Linb, C.-Y. Linc, Y.-C. Chungb, and Y.-F. Chungc, "Cuda clustalw: An efficient parallel algorithm for progressive multiple sequence alignment on multi-gpus," *Journal of Computational Biology and Chemistry*, vol. 58, pp. 62–68, 2015.

[3] D. T. P., O. M., G. F., C. F., E. T., and N. C., "Cloud-Coffee: implementation of a parallel consistency-based multiple alignment algorithm in the T-Coffee package and its benchmarking on the Amazon Elastic-Cloud," *Bioinformatics*, vol. 26, no. 15, pp. 1903–1904, 2010.

[4] K. Rycerz, K. Poland, N. M., P. P., and C. E., "Comparison of cloud and local HPC approach for MUSCLE-based multiscale simulations," in *e-ScienceW '11, Maui, Hawaii*, 2011, pp. 81–88.

[5] K. K. and T. H., "Parallelization of the MAFFT multiple sequence alignment program," *Bioinformatics*, vol. 26, no. 15, pp. 1899–1900, 2010.

[6] Z. Z., J. X., J. W., H. Z., G. L., X. W., and L. Dai, "Paraat: A parallel tool for constructing multiple protein-coding dna alignments," *Biochemical and Biophysical Research Communications*, vol. 419, no. 4, pp. 779–781, 2012.

[7] L. Y., S. B., and D. L. Maskell, "MSAProbs: multiple sequence alignment based on pair hidden markov models and partition function posterior probabilities," *Bioinformatics*, vol. 26, no. 16, pp. 1958–1967, 2010.

[8] de Araujo Macedo E., M. A. de Melo A.C., P. G.H., and B. A., "Hybrid MPI/OpenMP strategy for biological multiple sequence alignment with DIALIGN-TX in heterogeneous multicore clusters," in *Proc. IPDPSW '11*, Alaska, USA, 2011, pp. 418–425.

[9] L. C.Y. and L. Y.S., "Efficient parallel algorithm for multiple sequence alignments with regular expression constraints on graphics processing units," *Int. J. Comput. Sci. Eng.*, vol. 9, no. 1/2, pp. 11–20, 2014.

[10] M. W. Al-Neama, N. Reda, and F. F. Ghaleb, "An improved distance matrix computation algorithm for multicore clusters," *BioMed Research International*, vol. Article ID 406178, http://dx.doi.org/10.1155/2014/406178, 2014.

[11] ——, "Fast vectorized distance matrix computation for multiple sequence alignment on multi-cores," *International Journal of Biomathematics*, vol. 8, no. 6, p. DOI: 10.1142/S1793524515500849, 2015.

[12] W. A., K. C. K., and S. B., "Multi threaded vectorized distance matrix computation on the Cell/BE and x86/SSE2 architectures," *Bioinformatics Advance*, vol. 26, no. 10, pp. 1368–1369, 2010.

[13] M. W. Al-Neama, N. M. Reda, and F. F. M. Ghaleb, "Accelerated guide trees construction for multiple sequence alignment," *International Journal of Advanced Research*, vol. 2, no. 4, pp. 14–22, 2014.

[14] N. M. Saitou N, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406–425, 1987.

[15] N. C. for Biotechnology Information (NCBI). (2011). [Online]. Available: http://www.ncbi.nlm.nih.gov/

[16] K.-B. Li. (2011) Clustalw-mpi 0.13 clustalw analysis using distributed and parallel computing. [Online]. Available: http://www.mybiosoftware.com/clustalw-mpi-0-13-clustalw-analysis-distributed-parallel-computing.html/

[17] W. A. and S. B. (2013, Jun.) Distance matrix comp on cell be and x86. [Online]. Available: http://www.mybiosoftware.com/clustalw-mpi-0-13-clustalw-analysis-distributed-parallel-computing.html/

[18] D. L. M. Yongchao Liu, Bertil Schmidt. (2010) Msaprobs: Multiple sequence alignment. [Online]. Available: https://sourceforge.net/projects/msaprobs/