An Overview of Multiple Sequence Alignment Parallel Tools

FAYED F. M. GHALEB
Department of Mathematics
Faculty of Science
Ain Shams University
Cairo, Egypt
fmghaleb@yahoo.com

NAGLAA M. REDA
Department of Mathematics
Faculty of Science
Ain Shams University
Cairo, Egypt
r_naglaa2012@yahoo.com

MOHAMMED W. AL-NEAMA
Department of Mathematics
Faculty of Science
Al-Azhar University
Cairo, Egypt
mwneama@uomosul.edu.iq

Abstract: - Multiple sequence alignment is a key problem to most bioinformatics applications. The last ten years have witnessed a big improvement to existing multiple alignment tools and the development of new ones. Various parallel architectures have been experimented for reaching the highest level of accuracy and speed. This paper surveys most popular tools to clarify how parallelism accelerates the processing of large biological data set and improves alignment accuracy. It aims at guiding biologists/scientists to the appropriate software.

Key-Words: - Bioinformatics; Multiple sequence alignment; Distributed and parallel computing; Clouds; Grids; Clusters; Supercomputers; Multi-cores

1 Introduction

Bioinformatics [1] is the interface between biological and computational sciences. Its focus is on developing and applying computationally intensive techniques to increase the understanding of biological processes. One of its most important and basic challenging problem is the sequence alignment. It is a fundamental tool in molecular function prediction, intermolecular interactions, residue selection, and phylogenetic analysis. It is used to extract functional and evolutionary information of genes and proteins.

Sequence alignment is the problem of comparing biological sequences by searching for a series of nucleotides or amino acids that appear in the same order in the input sequences, possibly introducing gaps into them. When the number of sequences is two then it is referred to pairwise sequence alignment, otherwise it’s multiple sequence alignment (MSA). Global alignment is to find the best match between the entire sequences. While local alignment must find the best match between certain regions of the sequences.

Most MSA methods are based on one of the two pairwise alignment algorithms: the optimal algorithm proposed by Needleman and Wunsh (NW) for global alignment, and the improvement to the NW algorithm proposed by Smith and Waterman (SW) to obtain the local alignment. Both algorithms are composed of three phases: initialization, distance matrix computation and trace back. Nevertheless, they differ in their applied techniques at each phase.

The majority of MSA tools utilize the progressive method. For aligning N sequences, it first generates all possible N(N-1)/2 pairwise sequence alignment in order to calculate a distance matrix giving the divergence of each pair of sequences. Second it creates a guide tree constructed from pairwise sequence distances using a clustering method such as Neighbor-Joining. Third it builds up the final multiple alignments by progressive inclusion of the N sequences profile alignment according to the order given by the guide tree. This method is computationally efficient, but it doesn’t guarantee a global optimal alignment. Other MSA programs follow the iterative method. It was proposed in order to circumvent the inherent errors in progressive alignment method. It refines the alignment by making an initial alignment of groups of sequences and then revising the alignment to achieve a more reasonable result.

Although high quality MSA is an essential task, it becomes a big dilemma nowadays due to the gigantic explosion in the amount of molecular data. So, MSA programs were enforced to face the problem of the huge number of residue-to-residue comparisons when searching for similarities. Therefore, programmers were directed to the use of high performance computing systems that provide high computational capability and uniform high-speed memory access [2].

This paper surveys the most efficient and popular parallel MSA programs, emphasizing their requirements, capabilities, and limitations.
2 MSA Parallel Tools Evolution

Last decade has witnessed magnificent developments in MSA tools because of the parallel/distributed systems technology evolution. Various parallel versions have been developed that concentrate on the most time consuming tasks. They differ in the way they handle parallelism and the system they use as explained below.

2.1. Clustal series

The Clustal series [3] are the most widely used tools for global multiple sequence alignment. The first Clustal program combined the progressive alignment strategy with dynamic programming using a guided tree. Then ClustalW incorporated a number of improvements to the alignment algorithm, including sequence weighting, position-specific gap penalties and the automatic choice of a suitable residue comparison matrix.

SGI parallel Clustal [4] was the first attempt to accelerate ClustalW by parallelizing all three stages on a shared memory SGI Origin machine using OpenMP. It shows speedup of up to 10 folds when running ClustalW on 16 CPUs. So as HT ClustalW the other parallel programming approach that adopts high-throughput automation.

ClustalW-MPI [5] was targeted for workstation clusters with distributed memory architecture. It overcomes the problem of allocating time-independent tasks to parallel processors during the parallelization of the distance matrix calculation by using the fixed-size chunking scheduling strategy. It also uses MPI with dynamic scheduling to parallelize the searching of sequences having the highest divergence from all other sequences. It achieved an overall speedup of 4.3 using 16 processors and pairwise distances calculations scale up to 15.8. Also a multithreaded algorithm was proposed in [6] and was incorporated in ClustalW-MPI to enable a better use of CPU cycles and a better memory usage. It eliminates the synchronization delays bottlenecks by using threads and different scheduling approach.

pCLUSTAL [7] is another parallel version of ClustalW using MPI. It distributes the pairwise alignment computation on available processors to be computed in parallel. It can be run on a range of distributed and shared memory parallel machines, from high-end parallel multiprocessors to PC clusters, to simple networks of workstations. It achieved speedup up to 10 on a 64 node PC cluster. Another acceleration of ClustalW have been developed in [8] on multiprocessor SMP cluster using a hybrid MPI/OpenMP method. A speedup of 80 and 9.2 was obtained for the first and third stages respectively, and an overall speedup of 35 using 40 nodes and 80 processors.

MT-ClustalW [9] presented a multithreading optimized version of ClustalW which utilize the machine resources and achieve higher throughput on multicore computers. It achieves over 2 times faster than the sequential ClustalW with 8 threads.

MSA-CUDA [10] parallelizes all three stages of the ClustalW processing pipeline by the GPU using CUDA. It demonstrates average speedups of 36.91 for long protein sequences on a GeForce GTX 280 GPU.

ClustalXeed [11] has incremental improvements over previous versions. It can compute a large volume of biological sequence data sets, which were not tractable before. It uses both physical RAM and a distributed file-allocation system for distance matrix construction and pair-align computation to solve the conventional memory-dependency problem. It markedly improves the computation efficiency of disk-storage system by implementing INSTA load-balancing algorithm. Tests show that the average speed-up for 50 nodes was about 19.6 on 100 CPU cluster system.

In [12] a new method was presented, in which computing resources such as existing PCs and small cluster systems can be utilized as private cloud computing infrastructure for an inter-query style of bioinformatics tools. It presents a metadata repository schema and 6 query routing algorithms on it. Proposed algorithms were applied to ClustalW. Experimental results show remarkable benefits on a proposed private cloud system in terms of performance and user requirements.

2.2. T-Coffee series

T-Coffee [13] was the first MSA software that uses a consistency-based objective function optimized using progressive alignment. It tries to maximize the score between the final multiple alignment and a library of pairwise residue-by-residue scores derived from a mixture of local and global pairwise alignments.

Parallel T-Coffee (PTC) [14] was the first parallel implementation of T-Coffee. It is based on MPI and RMA mechanisms. It realized a speedup of about 3 with 80 processors on cluster consisting
of dual Intel Xeon 3GHz nodes. Most of the speedup comes from parallelizing and distributing pairwise alignment tasks dynamic scheduling for a near linear speedup during library generation.

Cloud-Coffee [15] is another parallel implementation of T-Coffee but in a different approach. It is based on shared-memory architectures, like multi-core or multi-processors. It relies entirely on the UNIX fork () function, with child and parent processes communicating via temporary files. It was benchmarked on the Amazon Elastic Cloud (EC2) and showed that the parallelization procedure is reasonably effective.

2.3. Muscle series

MUSCLE [16] is a widely used program. It has achieved a higher rank in accuracy and a faster speed compared to ClustalW and T-Coffee. It includes fast distance estimation using kmer counting; progressive alignment using a new profile function called the log-expectation score; and refinement using tree-dependent restricted partitioning.

MUSCLE-SMP [17] was the first parallel attempt of MUSCLE on shared memory system. It achieves an overall speedup of 15.2 on a 16 processors SMP system using OperMP. It was incorporated with the multithreaded algorithm in [6]. It used the bag-of-tasks model. Tests on 16 node cluster showed interesting improvement for progressive alignments and efficiency scales with the growth in the problem size.

MUSCLE-based multiscale simulations [18] have been presented in the two types of infrastructures: local HPC cluster and Amazon AWS cloud solutions. It has been integrated with grid space virtual laboratory that enables users to develop and execute virtual experiments on the underlying computational and storage resources through its website based interface.

2.4. MAFFT series

MAFFT [19] is another popular MSA program. It included two novel techniques that reduce the CPU time. It identified homologous regions by the fast Fourier transform. It was updated in [20] with two new techniques. The PartTree algorithm improves the scalability of progressive alignment and the Four-way consistency objective function improves the accuracy of ncRNA alignment.

All stages of MAFFT have been parallelized in [21] using the POSIX Threads library with the best-first and simple hill-climbing parallelization strategies. This approach was tested on a 16 core PC and achieved a peak speedup of 10.

2.5. DIALIGN series

DIALIGN was proposed in [22]. It compares every pair of sequences, generating a set of ungapped fragments with high score. These fragments are used to iteratively, generate the final alignment. DIALIGN-TX [23] used progressive and greedy approaches for segment-based MSA. It incorporated anchors optimizations for accurate alignments.

DIALIGN-TX-MPI [24] is the parallel version of DIALIGN-TX. It used an iterative heuristic method for MSA and generates alignments by concatenating ungapped regions with high similarity. It was implemented using both OpenMP and MPI on a 28-cores heterogeneous cluster. The best obtained speedup was 3.13.

2.6. Others

Sample-Align-D [25] is another parallel MSA program. It was based on partitioning the set of sequences into smaller subset using k-mer count based similarity index (k-mer rank). Then each subset is independently aligned in parallel. It has been implemented on a cluster of workstation on 16 node using MPI library, and shows a remarkable speedup. It was able to align 2000 randomly selected sequences in less than 10 minutes, compared to over 23 hours on sequential MUSCLE.

MSAProbs [26] is a new and practical multiple protein sequence alignment algorithm designed by combining a pair-HMM and a partition function to calculate posterior probabilities. It also investigates two critical bioinformatics techniques, namely weighted probabilistic consistency transformation and weighted profile-profile alignment, to achieve high alignment accuracy. It statistically demonstrates dramatic accuracy improvements over several top performing aligners: ClustalW 2.0.12, MAFFT 6.7.17, MUSCLE 3.8.31, ProbCons 1.12 [27], and Probalign 1.3 [28].
MSACompro [29] is a new efficient and reliable multiple protein sequence alignment program. It incorporates predicted secondary structure, relative solvent accessibility, and residue-residue contact information into the currently most accurate posterior probability-based MSA methods. It used a multiple-threading implementation on a 32 CPU cores machine. Benchmarks clearly show improvements in accuracy over the leading tools including MSAProbs.

ParaAT [30] is a recent parallel program that is capable of constructing multiple protein-coding DNA alignments for a large number of homologs. It is well suited for large-scale data analysis in the high-throughput era. It assigns each homolog to one of the slave threads; enable user to customize one of multiple sequence aligners (including ClustalW, Mafft, Muscle, T-Coffee); consolidates the results from all slave threads; then parallely back-translates multiple protein sequence alignments into the corresponding DNA alignments. Tests performed on a 24 cores machine provide good scalability and exhibits high efficiency.

3. Discussion

Despite the frequent use of the MSA tools by biologists and scientists, the decision about which tool to use is a difficult problem. Lately some researches [31], [32], [33] have been produced for surveying, comparing and evaluating sequential tools to address this critical issue and highlight a number of their strengths and weaknesses. But with parallel tools the situation is much difficult. The assessment and the choice of the most convenient tool are subjected to variant metrics. Most important metrics that affect the usability and popularity of the aligner were selected and summarized below. A comprehensive comparison of the most recent and efficient parallel MSA software tools against these metrics emphasizing their strength is presented in table 1. The given results are for BAliBASE, PREFAB and HomFam benchmark tests.

3.1. Availability

In fact, it is still very difficult for researchers and biologists who are not specialized in information technology to fully utilize high-performance computing technologies. Thus, the ease to obtain and use the software (availability) is a fatal requirement for them. The program needs to be publicly available and user-friendly.

3.2. Portability

Portability refers to the ability to run the program on variant type of platforms with different operating systems (OSs). This is very significance as most scientists need to run it on their available/ accessible platform such as supercomputers, clusters, grids, clouds, and multi-cores.

3.3. Performance

In time-critical applications, alignment speed is an essential factor to consider for assessing its performance. It is always measured by the execution time (T); that is the elapsed time from the start moment of the first processor to the end moment of the last processor execution. It clarifies the program’s high computational capability and high-speed memory access. It is highly affected by the sequences dataset size, i.e. the number of sequences (SqN) multiplied by the average sequence length (Sql).

3.4. Accuracy

The term accuracy (Acc) refers to the quality of producing good alignment. It is measured using different scores. The most important score for biologists is the column score (CS). It tests the ability of the programs to align all of the sequences correctly. It depends on the alignment method and the sequences’ number and length.

4. Conclusion

The aim of this paper is to direct both biologists and scientists to choosing the most appropriate MSA tool for their specific needs, thus enabling more efficient research. The general and central considerations of the MSA methods have been reviewed, to clarify their main differences. The most popular and widely used tools have been surveyed with a study of their evolution from the first appearance till now aiming at discovering their strengths and bottlenecks. A detailed discussion of existing parallel tools has been introduced supported by a comparative study.
according to variant metrics for emphasizing their main characteristics.

Discussion shows that Clustal is the most highly cited aligner especially for huge number of sequences. MAFEET is the fastest at expense of accuracy. While for small number of sequences, MUSCLE and T-Coffee achieved both high accuracy and fast speed. MSAProbs is the most accurate but with long run times. Also Sample-Align-D is a recent promising algorithm that is quit suitable to practical use and further improvement.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Availability</th>
<th>SqN</th>
<th>SqL</th>
<th>Platform</th>
<th>OS</th>
<th>T (Sec.)</th>
<th>Acc (CS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClustalXeed 1.0</td>
<td><a href="http://clustalxeed.blogspot.com/">http://clustalxeed.blogspot.com/</a></td>
<td>52750</td>
<td>440</td>
<td>100 × CPU AMD Opteron 244 cluster</td>
<td>Fedora Linux</td>
<td>200</td>
<td>High (0.5)</td>
</tr>
<tr>
<td>MAFEET 6.857</td>
<td><a href="http://mafft.cbrc.jp/alignment/software/">http://mafft.cbrc.jp/alignment/software/</a></td>
<td>50157</td>
<td>100</td>
<td>4 × Quad-Core AMD Opteron Processor 8378</td>
<td>Unix, Linux, Windows, MacOSX</td>
<td>6119</td>
<td>Low (0.253)</td>
</tr>
<tr>
<td>Parallel T-Coffee PBS</td>
<td><a href="http://gcd.udl.cat/ptc">http://gcd.udl.cat/ptc</a></td>
<td>1048</td>
<td>523</td>
<td>HECToR supercomputer with 180-core</td>
<td>Unix, Linux, cygwin and MacOSX</td>
<td>9264</td>
<td>High (0.615)</td>
</tr>
<tr>
<td>MUSCLE 3.8.31</td>
<td><a href="http://www.drive5.com/muscle/">http://www.drive5.com/muscle/</a></td>
<td>4000</td>
<td>1000</td>
<td>Beowulf cluster of 16 Intel Pentium 4 processors</td>
<td>Unix, Linux, Windows, MacOSX</td>
<td>140</td>
<td>High (0.550)</td>
</tr>
<tr>
<td>MSAProbs 0.9.4</td>
<td><a href="http://sourceforge.net/projects/msaprobs/">http://sourceforge.net/projects/msaprobs/</a></td>
<td>1682</td>
<td>100</td>
<td>Intel i7 quad-core</td>
<td>Linux</td>
<td>51286</td>
<td>High (0.737)</td>
</tr>
</tbody>
</table>

Table 1: A comparison of parallel MSA tools

References:


E. de Araujo Macedo et al., Hybrid MPI/OpenMP Strategy for Biological Multiple Sequence Alignment with DIALIGN-TX in Heterogeneous Multicore Clusters, IPDPSW ’11, 2011, pp. 418-425.