Multiple Sequence Alignments with Parallel Computing

Charu Sharma
Research Scholar
Jodhpur National University

Pankaj Agrawal, Ph. D
Professor, IMSEC

Preeti Gupta
Research Scholar
Shubgarti University. Meerut

ABSTRACT

The growth of bioinformatics and computational biology industry, multiple sequence alignment (MSA) applications have become an important emerging workload. In spite of the large amount of recent attention given to the MSA software design, there has been little quantitative understanding of the performance of such applications on modern microprocessors and systems. In this paper we try to analyze performance and characteristics of MSA software from the perspective of multicore machines. We use several popular MSA programs employing a wide variety of alignment approaches. The basic workload characteristics and the efficiencies of various multicore machines features are examined . In order to mapping parallelism in multicore machines we try to explore different parallel programming approaches using threads and MPI

Key words

Multiple Sequence Alignment, Parallelism, multicore machines, parallel strategies.

1. INTRODUCTION

Multiple Sequence Alignment (MSA) is identified as one of the challenging tasks in bioinformatics belongs to a class of hard optimization problems called combinatorial problems [1].On the other hand, with the advent of new breed of fast sequencing techniques it is now possible to generate thousands of sequences very quickly. For rapid sequence analysis, it is therefore desirable to develop fast MSA algorithms that scale well with the increase in the dataset size. The main problem in MSA is its exponential complexity with the considered input data set. These alignments may be used to identify profiles or hidden models that may be used to acquire knowledge for distantly related members of the family sequences, newly discovered sequences, and existing sequence databases. Many of the methods are heuristics ones which attempt to find good alignment that are not necessarily optimal.

Many researcher overviews outlines computational issues related to parallelism, physical machine models, parallel programming approaches and scheduling strategies for a broad range of computer architectures.

Now a day’s processor makers favour multi-core chip designs, and software are written in a multi-threaded or multi-process manner to take full advantage of the hardware. Thus, the only way to catch up is to write applications that leverage parallelism, i.e. engaging multiple CPU cores for handling all tasks rather than a single faster core. Parallel computing or parallelization is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently (“in parallel”). In essence, if a CPU intensive problem can be divided in smaller, independent tasks, then those tasks can be assigned to different processors. Regarding multi-threading and concurrency, many programming languages like Java has had support for Threads since its beginning and back in the old days you could manipulate thread execution using a low-level approach with the interrupt, join, sleep methods.

Today most of the computer systems have four or eight cores with many cores in high end computers. This trend invites researchers to develop parallel MSA algorithms that can effectively exploit the many core designs.

2. BACKGROUND STUDY

2.1 Multisequence Sequence Alignment (MSA)

What is the multiple sequence alignment?

Multiple sequence alignment is a process of determining corresponding residues in homologous sequences. The term homologous means that given sequences share a common ancestor.

Multiple sequence alignment (MSA) can be seen as a generalization of Pairwise Sequence Alignment instead of aligning two sequences, n sequences are aligned simultaneously, where n is > 2.MSA is a fundamental operation performed in computational biology, which helps to provide a wealth of information related to the evolutionary relationships. The main idea behind MSA is to put protein or DNA residues in the same column according to the selected criteria

MSA applies both to nucleotide and amino acid sequences.

Example of MSA

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWWSNG--
```

Figure2: Multisequence alignment

Multiple Sequences Alignment (MSA) of biological sequences is a fundamental problem in computational biology due to its critical significance in wide ranging applications including haplotype reconstruction, sequence homology, phylogenetic analysis, and prediction of evolutionary origins. The MSA problem is considered NP-hard and known heuristics for the problem do not scale well with increasing number of sequences. On the other hand, with the advent of new breed of fast sequencing techniques it is now possible to generate thousands of sequences very quickly. For rapid sequence analysis, it is therefore desirable to develop fast MSA algorithms that scale well with the increase in the dataset size. The high computational costs and poor scalability of existing MSA algorithms make the design of multiprocessor solutions highly desirable.

The purpose of an MSA is to align sequences in such a way as to reflect the biological relationship between the input sequences, but developing a reliable MSA program is a very complex problem. In its most basic form the problem can be stated in the following way: given N sequences and a scoring scheme for determining the best matches of the letters (where each sequence consists of a series of letters), find the optimal pairing of letters between the sequences. Even this simplistic definition requires a consideration of the choice of sequence, choice of comparison model and optimization of the model

MSA can be categories into three approaches:

- Heuristic approach
- Progressive approach
- Iterative approach

The most widely used approach is progressive approach. Some of the available programs of MSA are CLUSTALW [18], T-Coffee [16] are based on progressive approach, SAGA [19] and MUSCLE.

The increasing amount of sequences stored in genomic databases has become unfeasible to the sequential analysis. Then, the parallel and distributing computing brought its power to the Bioinformatics through parallel algorithms to align and analyze the sequences, providing improvements mainly in the running time of these algorithms. In many situations, the parallel strategy contributes to reducing the computational complexity of the big problems. From a computational point of view, there are several ways to address the lack of hard computing power for bioinformatics [14]

1. Developing new, faster heuristic algorithms that reduce computational space for the most time- consuming tasks. [4]
2. Incorporating these algorithms into the ROM of a specialised chip (eg the bio- accelerator at Weizmann Institute5).
3. Parallel computing.

2.2 Parallelism in MSA

With the rapid development of high-throughput sequencing technologies, the number of sequences in public databases is rapidly increasing. There is an increasing demand to align large-scale biological sequence datasets. Due to the computationally intensive operation and vast amount of available sequence data, many parallel MSA algorithms have been developed. In spite of the improvement in speed and accuracy introduced by these parallel programs, most of them are fundamentally modified from a known sequential system.

Nowadays, high-performance processing cannot be understood without new Chip Multiprocessors (CMPs), which are being actively developed. Amongst such chips are the Graphics Processing Units (GPUs) with hundreds of cores [1] and the Sony–IBM–Toshiba Cell Broadband Engine Architecture (CBEA) [2], which allow us to render complex animations and provide as well enough computing power to perform other calculus-intensive tasks [3].A comparison between these different programming approaches and an attempt to automate such processes can be found at [8]. Other companies are working in CMP and multithreaded many-core microprocessors [22].

The multiple sequence alignment is a complex problem and may be very time consuming. Especially, methods that aim to provide high quality results, are very sophisticated and thus require much time to perform the computations. Moreover, the increasing number of sequences is becoming a challenge for current software and hardware solutions. Therefore, a couple of parallel applications addressing this problem have been developed so far, e.g. [45] . These tools, significantly decreasing the computation time, show a great potential of parallelism.

Current laptop computers all have two or four cores, and desktop computers can easily have four or eight cores, with many cores in high-end computers. This trend incites researchers to develop parallel MSA algorithms that can effectively exploit the many-core architecture. Many resercher focus on shared-memory parallel computers, specifically multi-core CPUs, which allow simultaneous execution of multiple instructions on different cores.

A few researches have been done from the aspects of data parallelism. They are still limited in their ability to handle very large amounts of sequences because the system lacks a scalable high-performance computing (HPC) environment with a greatly extended data parallel strategy. The major area of focus for parallel MSA

1. shared-memory parallel computers,
2. distributed memory systems
3. graphical processing units, [8].

This is further categories into Data parallelism and hardware design parallelism.

In order to improve the speed of MSA procedure, there have been numerous attempts to parallelize these sequential MSA systems, such as CLUSTALW-MPI [24,16], Parallel-TCoffee [17], Cloud-Coffee [7], Sample-Align-D [18], and parallelization of the MAFFT [19]. On the other hand, a number of parallel hardware acceleration methods have been developed for biological sequence alignment, which include a new parallelization of the Needleman–Wunsch (NW) [20] algorithm for the 64-core Tile64 microprocessor by Sergio et al. [21], FPGA-based architectures to accelerate DIALIGN [22] by Boukerche et al. [23], network-on-chip hardware accelerators by Sarkar et al. [24], streaming algorithms on commodity Graphics Processing Units (GPUs) by Weiguo et al. [25], and MUMmerGPU [26], a parallel pairwise local sequence alignment program that runs on GPUs in common workstations.

Many researchers have worked on the implementation of the parallel approach in MSA. The parallel approach can be categorized into:

- Parallelism through hardware approach using grid or other parallelism support architecture
- And parallelism through software approach using parallel programming models.

Most of the work currently being done in computational biology involves searching for inter- and intra-sequence homology in massive volumes of genetic and protein sequence data, which are commonly based on a multiple sequence alignments (MSAs) [3]. However, increasing the computational efficiency to solve a variety of real MSA problems is still a challenging task because of the high demand for greater capacity and speed [p5, 4]. Past MSA performance evaluations focused simply on how compute-intense and sensitive the program was with respect to the longest-common-sequence (LCS)-based exact-string matching algorithm (e.g., the Smith-Waterman or Needleman-Wunsch method) [5]. Depending on both the volume of data to be aligned and the accuracy of the comparisons, computation using dynamic programming is extremely time-consuming when large sequence volumes and high accuracy are required simultaneously. Numerous parallel-computation programs, such as parallelized Praline [6], DiAlign P [7], ClustalW-MPI [8], and a commercial SGI parallel Clustal on a shared memory SGI multiprocessor [9].

Geraldo F. [10] has discuss implementation of a parallel score estimating technique for the score matrix calculation stage, which is the first stage of a progressive multiple

sequence alignment. The performance and quality of the parallel score estimating are compared with the results of a dynamic programming approach also implemented in parallel. This comparison shows a significant reduction of running time. In his research he suggests the progressive MSA algorithm is divided into three stages:
first stage is the score matrix calculation,

- The second is the phylogenetic tree construction and
- The last is the multiple alignment.

The score matrix calculation is the most complex. He try to parallelise score matrix calculation stage

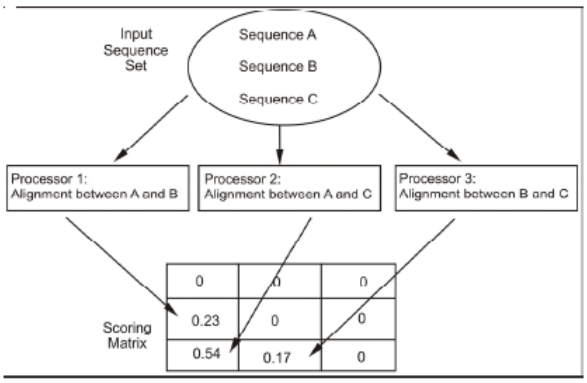
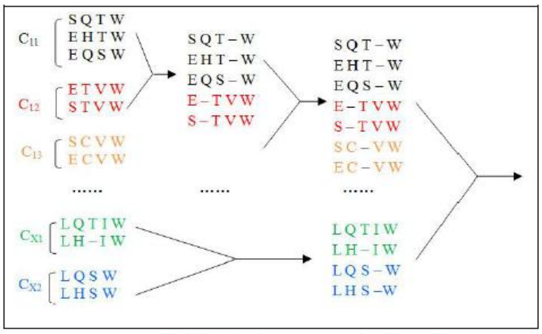


Illustration of parallel pairwise alignment algorithm.

Figure 7: Parallel approach for score matrix calculation

Xiangyuan Zhu[11] address the biological sequence alignment problem, which is a fundamental operation performed in computational biology can be reduce by employing the data parallelism paradigm that is suitable for handling large-scale processing to achieve a high degree of parallelism. He proposed parallel strategy using cluster-distributed –align.
Outline of Cluster – Distribution – Align is

- In parallel,Clustering the sequences on available processors using parallel clustering algorithm
- Redistribute the clusters obtained from step1
- In parallel, align the clusters on each processor using MSA algorithm.



Profile aligning progressively and combining sequence clusters

Figure 8: Profile alignment

This approach shows super linear speed up with comparable quality. Similar approach is used by Fahad saheed[p4] in his Domain Decompose strategy for multi processor.

Agarwal P.[12] highlighted the pipelining approach [p5] in solving sequence alignment problem. In his paper he suggest a two stage pipeline approach for the pair wise alignment. The basic purpose of using the pipelines is to reduce the time-complexity of alignment significantly. He also suggested that his approach can be used for multisequence alignment with parallel approach Assumption of multiple pipelines and functional unit improves the time complexity of the standard algorithms quite considerably from O(n2) to O(n).
Proposed Two stage pipeline model

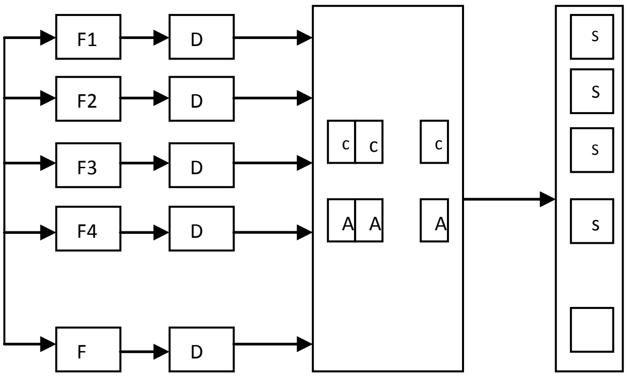


Figure 9: General Architecture for pipeline model

F: fetch Unit, D: Decode Unit ,C:Comparator and S: store unit.

Model with more than one pipeline structure

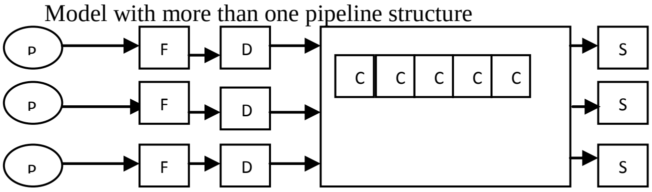


Figure 10: Architecture with multiple function Unit

P: Pipeline F: fetch Unit, D: Decode Unit ,C:Comparator and S: store unit.

Ttrelles O.[14] in his survey report recommend that MIMD machines are more amenable to bioinformatics at hardware level . He also suggested that for parallel programming model we need to consider the two key factors a) granularity and b) Communication. He also consider for parallelism we need to consider task scheduling strategies. Naveed,T.[7] has proposed a fast computation solution through parallel algorithm[7] using Needleman-wunsch algorithm for grid. his approach algorithm he uses the 3 CPU . Two CPU for datamatrix and pointer matrix and third to store the values in global memory. This help to reduce the need of back tracking by storing all values in the global memory of third CPU. This algorithm reduce the calculation time from O(m*n) to O(n +m)[15].
Dohi et al.[25] multi-threaded parallel design and implementation of the Smith-Waterman (SW)
algorithm on graphic processing units (GPUs) with NVIDIA corporation’s Compute Unified Device Architecture (CUDA).This is a divide and conquer approach which divides the computation of a whole pairwise sequence alignment matrix into multiple sub-matrices (or parallelograms) each running efficiently on the available hardware resources of the GPU in hand, with temporary intermediate data stored in global memory. His implementation achieves up to 46% improvement in speed. Zhang[26] accelerate pairwise statistical significance estimation of local sequence alignment using standard substitution matrices. By carefully studying the

algorithm's data access characteristics, he developed a tile-based scheme that can produce a contiguous data access in the GPU global memory and sustain a large number of threads to achieve a high GPU occupancy
Comparative table for different parallel approach of MSA

Table 1: Different approaches of MSA with parallelism

PARALLEL APPORACH	**IMPROVEMENT CRITERIA**
Parallel approach with score matrix	15% improvement in execution time
Data parallel approach(Cluster-Distribtion-Align)	Linear increase in speed of processing
Pineline approach	From O(n2) to O(n)
Parallel approach with dynamic algorithm (Needlemam-Wunsch)	From O(m*n) to O(m+n)

3. METHODOLOGIES CAN BE USED IN PARALLEL MSA

Existing MSA algorithm can be implemented into multi-core machines by modifying them with global shared memory and master slave approach.
Basically we have to focus on two things for parallelism

- Granularity
- Communication

For granularity we need to break them into uniform smallest block. For communication we need to focus overlapping of function to minimize the time consume in communication.
The following strategies we are going to use for parallelism of MSA algorithms

A. Divide and conquer Approach

In this strategy we use two approaches

1. Block division

In this step n input sequences are divided into chunk of blocks. Each of these blocks are execute in parallel for the local pairwise alignments. The result of each block is again align for global alignment.

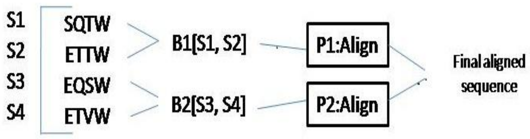


Figure 11: Layout of parallel Block Approach.

2. Block division

In this step we try to merge the locally align block s into one data set using some guided tree approach.

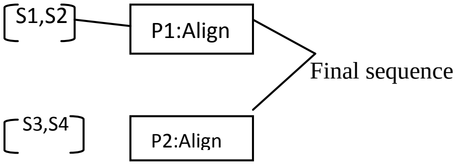


***Figure 13:Merge global alignment* Approach**

B. Master- Slave Cell Computation

In this strategy we apply multithreading concept at Cell computation of Score matrix of pair-wise alignment. Each row level computation is done by individual threads. Master-Slave concept of multithreading is used.

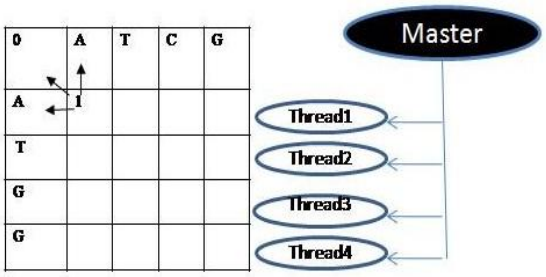


Figure 12: Score matrix Calculation using Master-Slave Multithreading approach. Each slave thread is control by Master Thread.

Local pair-wise align strategies can be used.
In some cases we can use Data decomposition strategies

C. Data Decomposition

N Sequence of MSA can be broken into smaller sequences i.e. N/P where P is processor. Each N/P sequence will execute in parallel independently. Finally two Sequences are combining again for Global alignment process

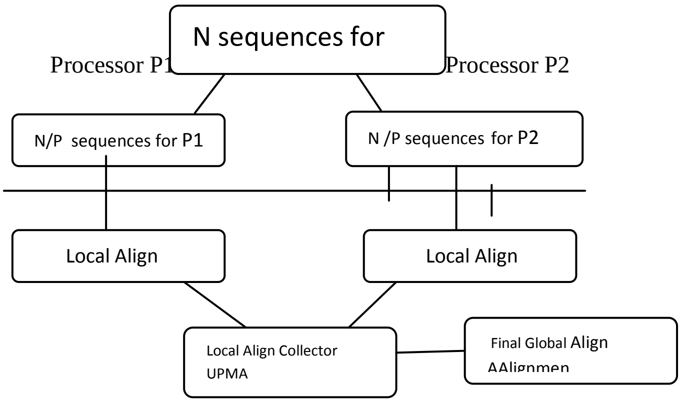


Figure 14: Layout of Data Decomposition Approach

4. IMPLEMENTATION TECHNIQUES

Parallel programming can be implemented in multiprocessors through Multi threading or MPI(Multi Programming Interface).We can use the multithreading techniques of java or .Net c#.

5. CONCLUSION

As demand of use of Multi Sequence Alignment is going to increase and we still need a more reliable and speed up approach for MSA. The parallel programming approach increase the speed and efficiency of our single machine programme. For parallelism we can adopt many approach like Block data, cell computation, merge and data decomposition. In research point of view analysing the working of available MSA programs on the multi-core machines can help us to realise the scalability and computational effect . For analysing them we are going to change only the working environment of these algorithms so that we can implement them in multicore machines. Through this we can explore how existing MSA applications Behaves for multicore machines.

6. REFERENCES

[1] J.D. Thompson, J.e. Thierry, O. Poch. (2003). RASCAL: rapid scanning and correction of multiple sequence alignments, Bioinformatics, Vol. 19, No. 9.

[2] V Amouda et. al. / International Journal of Engineering Science and Technology Vol. 2(11), 2010, 6361-6370.

[3] Thompson JD, Poch O.(2006). Multiple sequence alignment as a workbench for molecular systems biology,Curr Bioinformatics , **1:**95-104.

[4] Boukerche A, Demelo A, Ayalarincon M, Walter M.(2007). Parallel strategies for the local biological sequence alignment in a cluster of workstations, J Parallel Distrib Comput , **67:**170-185.

[5] Essoussi N, Boujenfa K, Limam M.(2008). A comparison of MSA tools**.**Bioinformation , **2:**452-455.

[6] Kleinjung J, Douglas N, Heringa J.(2002). Parallelized multiple alignment,Bioinformatics , **18:**1270-1271.

[7] Schmollinger M, Nieselt K, Kaufmann M, Morgenstern B.(2004).DIALIGN P: Fast pair-wise and multiple sequence alignment using parallel processors**.**BMC Bioinformatics 2004, **5:**128.

[8] Li K-B.(2003). ClustalW-MPI: ClustalW analysis using distributed and parallel computing**,** Bioinformatics , **19:**1585-1586.

[9] Mikhailov D, Cofer H, Gomperts R.(2001). Performance optimization of Clustal W: parallel Clustal W, HT Clustal, and Multiclustals**.** In White papers. Silicon Graphics, Mountain View, CA.

[10] Zafalon,F.D. Geraldo. Et al**.**(2013).“Improvements in the score matrix calculation method using parallel score estimating algorithm”, Journal of Biophysical Chemistry, Vol. 4,No. 2, 47-51.

[11] Zhu, X. Li, K. et al.(2011)**.“**A Data Parallel Strategy for Aligning Multiple Biological Sequences on Homogeneous Multiprocessor Platform”**,** Sixth Annual ChinaGrid Conference.

[12] Agarwal,P. Rizvi,S.A.M.(2009).”Solving sequence Alignment Problems using Pipeline Approach”,BIJIT-BVICAM’s International Journal of Information Technology

[13] Saeed, F. et al.(2009).”A Domain Decomposition Strategy for Alignment of Multiple Biological Sequences on Multiprocessor Platforms”, J.Parallel Distrib. Comput.

[14] Trelles O.(2001). “On the parallelisation of bioinformatics applications**”,** Brief Bioinform, **2:**181-194.

[15] Naveed ,T.Siddiqui, I.S.(2005).”Parallel Needleman-Wunsch Algorithm for Grid”, Biogridpaper.

[16] C. Notredame, D. G. Higgins, and J. Heringa.(2000). T-COFFEE: A novel method for fast and accurate multiple sequence alignment. J. Mol. Biol., 392:205–217.

[17] C. Notredame, L. Holm, and D. G. Higgin.(1998) COFFEE: An objective function for multiple sequence alignment. Bioinformatics, 14(5):407–422.

[18] J. D. Thompson, D. G. Higgins, and T. J. Gibson.(1994). CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucleic Acids Research,22(22):4673–4680.

[19] C. Notredame and D. G. Higgins.(1996). SAGA: Sequence alignment by genetic algorithm, Nucleic Acids Research, 24(8):1515–1524.

[20] Robert C. Edgar.(2004).” MUSCLE: multiple sequence alignment with high accuracy and high throughput**”,** Journal,Nucleic Acid Research, Vol.32(5).

[21] J. Blazewicz et al. (2013) “G-MSA — A GPU-based, fast and accurate algorithm for multiple sequence alignment”, J. Parallel Distrib. Comput. 73: 32–41.

[22]]F.J. Esteban et al.(2013). “Direct approaches to exploit many-core architecture in bioinformatics”, Future Generation Computer Systems Vol.29:15–26.

[23] Rezaei.S et al.(2006).”DIVIDE-AND-CONQUER ALGORITHM FOR CLUSTALW-MPI”,IEEE CCECE/CCGEI, Ottawa, May 2006.

[24] Xiangyuan Zhu. Et al.(2013) “A data parallel strategy for aligning multiple biological sequences on multi-core computers”, Computers in Biology and Medicine 43: 350–361.

[25] Dohi.K, K.Benkrid,C.Ling et al.(2010).“Highly Efficient Mapping of the Smith-Waterman Algorithm on CUDA-compatible GPUs” IEEE,ASAP.

[26] Thompson,J.D., Higgins,D.G. and Gibson,T.J.(1994).“ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting,position-specific gap penalties and weight matrix choice,” Nucleic Acids Research, vol. 22, pp. 4673–4680.

[27] O. Miquel, G. Fernando, N. Cedric, C. Fernando, Exploiting parallelism on progressive alignment methods, J. Supercomput. (2009) 79–87.

[28] B. Azzedine, C. Alba, A.-R. Mauricio, E. Maria, Parallel strategies for the local biological sequence alignment in a cluster of workstations, J. Parallel Distrib. Comput. 67 (2007) 170–185.

[29] K. Taecho, j. Hyun, Clustalxeed: a GUI-based grid computation version for high performance and terabyte size multiple sequence alignment, BMC Bioinformatics 11 (2010) 467. [

[30] H.M. Wong, V. Bharadwaj, Aligning biological sequences on distributed bus networks: a divisible load scheduling approach, IEEE Trans. Inf. Technol. Biomed. 9 (4) (2005) 489–501.

[31] V. Bharadwaj, H.M. Wong, Handling biological sequence alignments on networked computing systems: a

divide-and-conquer approach, J. Parallel Distrib. Comput. 69 (2009) 854–865.

[32] [32] H.P.L. Diana, V. Bharadwaj, A.B. David, On the design of high-performance algorithms for aligning multiple protein sequences on mesh-based multi-processor architectures, J. Parallel Distrib. Comput. 67 (2007) 1007–1017.

[33] D.P. Tommaso, M. Orobitg, F. Guirado, F. Cores, T. Espinosa, C. Notredame, Cloud-coffee: implementation of a parallel consistency-based multiple align- ment algorithm in the t-coffee package and its benchmarking on the amazon elastic-cloud, Bioinformatics 26 (15) (2010) 1903–1904.

[34] H.F.B. Vicente, L.M. David, P. Sylvain, S. Johannes, Parallel geometric algo- rithms for multi-core computers, Comput. Geometry 43 (2010) 663–677.

[35] L. Kuo-Bin, CLUSTALW-MPI: CLUSTALW analysis using distributed and parallel computing, Bioinformatics 19 (12) (2003) 1585–1586.

[36] Z. Jaroslaw, Y. Xiao, R. Adrain, A. Srinivas, Parallel-tcoffee: a parallel multiple sequence aligner, in: Proc. ISCA PDCS, 2007, pp. 248–253.

[37] S. Fahad, K. Ashfaq, A domain decomposition strategy for alignment of multiple biological sequences on multiprocessor platforms, J. Parallel Distrib. Comput. 69 (2009) 666–677.

[38] K. Kazutaka, T. Hiroyuki, Parallelization of the MAFFT multiple sequence alignment program, Bioinformatics 26 (15) (2010) 1899–1900.