BIOINFORMATICS APPLICATIONS NOTE Vol. 18 no. 9 2002
Pages 1270–1271

Parallelized multiple alignment

*Jens Kleinjung, Nigel Douglas and Jaap Heringa**

Division of Mathematical Biology, National Institute for Medical Research, Mill Hill, London NW7 1AA, UK

Received on February 15, 2002; revised on March 11, 2002; accepted on March 12, 2002

ABSTRACT

Summary: Multiple sequence alignment is a frequently used technique for analyzing sequence relationships. Compilation of large alignments is computationally expensive, but processing time can be considerably reduced when the computational load is distributed over many processors. Parallel processing functionality in the form of single-instruction multiple-data (SIMD) technology was implemented into the multiple alignment program Praline by using 'message passing interface' (MPI) routines. Over the alignments tested here, the parallelized program performed up to ten times faster on 25 processors compared to the single processor version.

Availability: Example program code for parallelizing pairwise alignment loops is available from http://mathbio.nimr.mrc.ac.uk/~jkleinj/tools/mpicode. The 'message passing interface' package (MPICH) is available from http:/www.unix.mcs.anl.gov/mpi/mpich.

Contact: jhering@nimr.mrc.ac.uk

Supplementary information: Praline is accessible at http://mathbio.nimr.mrc.ac.uk/praline.

RESULTS

A key procedure in bioinformatics is sequence comparison by multiple alignment which can provide a wealth of information about structure–function relationships, such as evolutionary conservation of functional residues or conserved hydrophobicity patterns. Multiple alignment programs like Clustal *W* (Thompson *et al.*, 1994), T-Coffee (Notredame *et al.*, 2000) and Praline (Heringa, 1999, 2002) are based upon the so-called progressive alignment strategy (Feng and Doolittle, 1987) and are all able to produce high-quality alignments as demonstrated in a recent benchmark (Heringa, 2002) over 144 alignments in the BAliBASE repository (Thompson *et al.*, 1999), although their results are not necessarily identical, particularly with more divergent sequence sets. The compilation of a multiple alignment consisting of n sequences, using the progressive alignment strategy, typically involves the generation of all possible $n(n-1)/2$ pairwise sequence alignments (PSA) to generate a so-called guide tree, and then the construction of the final multiple alignment by progressive inclusion of the n sequences through $n-1$ sequence or profile alignments in the order given by the guide tree. In Praline, however, an alternative and more computer intensive progressive protocol is followed (Heringa, 1999, 2002).

The progressive phase here requires a total of $(n-2)(n-3)/2$ pairwise profile alignments (PPA), as a full profile search is conducted after each inclusion of a sequence (Figure 1a). This is done to select the highest scoring alignment (sequence–sequence, sequence–profile or profile–profile) at each step during the progressive phase. Praline thus determines the alignment order on-the-fly during progressive alignment, such that the tree becomes available after completion of the final alignment.

The Praline method also has a pre-processing mode, shown to be effective in enhancing alignment quality, which involves the generation of a profile for each of the initial sequences, constructed using the pairwise alignments resulting from the PSA phase (Heringa, 1999, 2002). Each pre-processed profile contains information from other sequences deemed reliable enough to increase the information content of the profile, and is based on a master–slave alignment, where pairwise alignments containing the master sequence are stacked onto that sequence. Pre-processing can be performed in global or local mode: globally pre-processed profiles contain information from related complete sequences within the initial sequence set, whereas locally pre-processed profiles hold local alignment information. A multiple alignment is then constructed using these pre-processed profiles rather than the individual sequences. The pre-processing option has been found to dramatically increase the multiple alignment quality (Heringa, 2002), but comes at a price of an additional round of $n(n-1)/2$ PPAs. This leads to a total of $n(n-1)/2$ PSAs to construct the pre-processed profiles, plus $n(n-1)/2$ PPAs and $(n-2)(n-3)/2$ PPAs for the Praline multiple alignment strategy as before, but now based on the pre-processed profiles, so that $n(n-1)/2$ PPAs instead of PSAs are carried out here.

*To whom correspondence should be addressed.

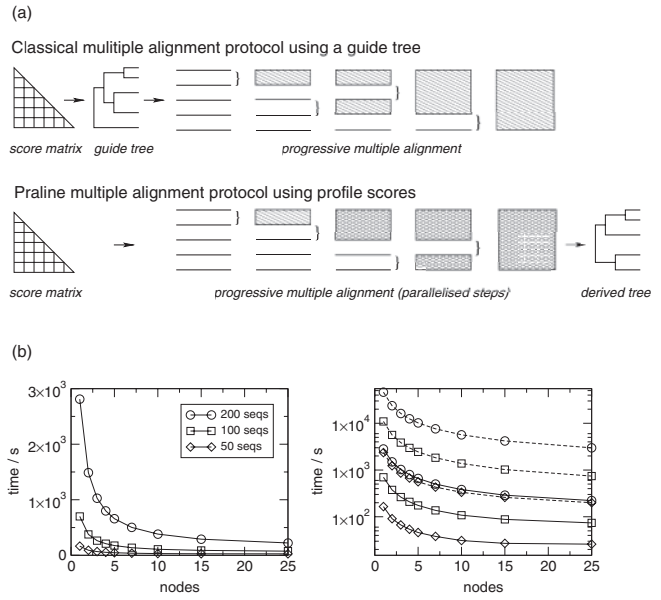

Fig. 1. (a) Two different strategies for progressive multiple alignment. Top: Classical progressive alignment—A matrix of $n(n-1)/2$ pairwise alignment scores is used to construct a guide tree. The multiple alignment is constructed following this guide tree (top scoring alignment indicated by brackets). Bottom: The Praline progressive alignment strategy—Praline does not use a guide tree, but progressively evaluates at each step the alignment score of all single sequences or sequence blocks with the current profile, leading to $(n-2)(n-3)/2$ alignments. The top-scoring alignment (indicated by brackets) will then be selected as a building block for the final alignment. A phylogenetic tree can be derived from the final multiple alignment. Note that the trees and final alignments of the two methods may differ as shown in this example. (b) Computational times of parallelized Praline on different numbers of nodes for three sets of 200, 100 and 50 sequences, each 200 residues long. Left: The performance improvement levels off at about 5 nodes for high computational load and the maximum gain at 25 nodes is a ten-fold reduction in processing time. Right: Logarithmic plot of Praline performance with the pre-processing mode switched on (dotted lines) and off (solid lines). The parallel code shows the same scaling with increasing computational load.

Highly repetitive procedures, such as the PSA phase in progressive alignment and the additional PPA phase(s) in Praline, are favourable targets for parallelized (or distributed) computing designed to split the total computational task into sub-tasks that are being processed on separate CPUs (nodes). If all nodes execute the same operations but on different sub-sets of distributed data, the parallelization technology is called single-instruction multiple-data (SIMD). Parallel code is most efficient at a minimum amount of communication between the nodes and at optimal balancing of the computational load over the CPUs.

We parallelized both the PSA and PPA phases in Praline by implementing parallelization routines provided by the MPICH package (Gropp *et al.*, 1996; Pacheco, 1997) for SIMD technology. In the PSA phase, each pairwise alignment is independent of the others, and only one inter-node communication event for gathering node-specific results is required at the end of the overall process, which also holds for the extra $n(n-1)/2$ PPAs carried out when pre-processing is used. In contrast, the calculation of $(n-2)(n-3)/2$ PPAs during progressive alignment build-up requires data passing after each of the sequence inclusions, implying an additional burden of $n-2$ communication events. The scaling of computational times *versus* the number of employed nodes is plotted in Figure 1b for three differently sized sets of sequences. Parallelized Praline generated a multiple alignment up to ten times faster than the single processor version, when tested on a set of 200 random sequences of 200 residues length. The parallel code is expectedly most efficient at high computational load. The smaller sets of 100 and 50 random sequences of 200 residues length should therefore be compiled on a few nodes only. However, when using the pre-processing mode of Praline, the computational load is high enough to justify parallel execution even on small sets of sequences (Figure 1b).

Parallelized Praline should be useful for analyzing large sets of sequences, as those emerging from the genome databanks. Prerequisites are the presence of a computer network and installation of the MPICH package. Praline is available upon request from the authors.

REFERENCES

Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **21**, 112–125.

Gropp,W., Lusk,E., Doss,N. and Skjellum,A. (1996) A high-performance, portable implementation of the MPI Message-Passing Interface standard. *Parallel Computing*, **22**, 789–828.

Heringa,J. (1999) Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. *Comp. Chem.*, **23**, 341–364.

Heringa,J. (2002) Local weighting schemes of protein multiple sequence alignment. *Comp. Chem.*, **26**, 459–477.

Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.

Pacheco,P.S. (1997) *Parallel programming with MPI*. Morgan Kaufmann, San Francisco.

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

Thompson,J.D., Plewniak,F. and Poch,O. (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple sequence alignment programs. *Bioinformatics*, **15**, 87–88.