BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 8 2011, pages 1164–1165
doi:10.1093/bioinformatics/btr088

Phylogenetics

Advance Access publication February 17, 2011

ProtTest 3: fast selection of best-fit models of protein evolution

Diego Darriba[1,2], Guillermo L. Taboada[2], Ramón Doallo[2] and David Posada[1,*]

[1]Department of Biochemistry, Genetics and Immunology, University of Vigo, 36310 Vigo and [2]Department of Electronics and Systems, Computer Architecture Group, University of A Coruña, 15071 A Coruña, Spain

Associate Editor: Martin Bishop

ABSTRACT

Summary: We have implemented a high-performance computing (HPC) version of ProtTest that can be executed in parallel in multicore desktops and clusters. This version, called ProtTest 3, includes new features and extended capabilities.

Availability: ProtTest 3 source code and binaries are freely available under GNU license for download from http://darwin.uvigo.es/software/prottest3, linked to a Mercurial repository at Bitbucket (https://bitbucket.org/).

Contact: dposada@uvigo.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on December 29, 2010; revised on February 10, 2011; accepted on February 11, 2011

1 INTRODUCTION

Recent advances in modern sequencing technologies have resulted in an increasing capability for gathering large datasets. Long sequence alignments with hundred or thousands of sequences are not rare these days, but their analysis imply access to large computing infrastructures and/or the use of simpler and faster methods. In this regard, high-performance computing (HPC) becomes essential for the feasibility of more sophisticated—and often more accurate—analyses. Indeed, during the last years HPC facilities have become part of the general services provided by many universities and research centers. Besides, multicore desktops are now standard.

The program ProtTest (Abascal *et al.*, 2007) is one of the most popular tools for selecting models of amino acid replacement, a routinary step in phylogenetic analysis. ProtTest is written in Java and uses the program PhyML (Guindon and Gascuel, 2003) for the maximum likelihood (ML) estimation of model parameters and phylogenetic trees and the PAL library (Drummond and Strimmer, 2001) to handle alignments and trees. Statistical model selection can be a very intensive task when the alignments are large and include divergent sequences, highlighting the need for new bionformatic tools capable of exploiting the available computational resources.

Here we describe a new version of ProtTest, ProtTest 3, that has been completely redesigned to take advantage of HPC environments and desktop multicore processors, significantly reducing the execution time for model selection in large protein alignments.

2 PROTTEST 3

The general structure and the Java code of ProtTest has been completely redesigned from a computer engineering point of view. We implemented several parallel strategies as distinct execution modes in order to make an efficient use of the different computer architectures that a user might encounter:

(1) A Java thread-based concurrence for shared memory architectures (e.g. a multicore desktop computer or a multicore cluster node). This version also includes a new and richer graphical user interface (GUI) to facilitate its use.

(2) An MPJ (Shafi *et al.*, 2009) parallelism for distributed memory architectures (e.g. HPC clusters).

(3) A hybrid implementation MPJ - OpenMP (Dagum and Menon, 1998) to obtain maximum scalability in architectures with both shared and distributed memory (e.g. multicore HPC clusters).

Moreover, ProtTest 3 includes a number of new and more comprehensive features: (i) more flexible support for different input alignment formats through the use of the ALTER library (Glez-Peña *et al.*, 2010): ALN, FASTA, GDE, MSF, NEXUS, PHYLIP and PIR; (ii) up to 120 candidate models of protein evolution; (iii) four strategies for the calculation of likelihood scores: fixed BIONJ, BIONJ, ML or user defined; (iv) four information criteria: AIC, BIC, AICc and DT (see Sullivan and Joyce 2005); (v) reconstruction of model-averaged phylogenetic trees (Posada and Buckley, 2004); (vi) fault tolerance with checkpointing; and (vii) automatic logging of the user activity.

3 PERFORMANCE EVALUATION

In order to benchmark the performance of ProtTest 3, we computed the running times for the estimation of the likelihood scores of all 120 candidate models from several real and simulated protein alignments (Table 1). When these data were executed in a system with shared memory, e.g. a multicore desktop, the scalability was almost linear as far as there was enough memory to satisfy the requirements. For example, in a shared memory execution in a 24-core node the speedup was almost linear with up to 8 cores, also scaling well with datasets with medium complexity, like HIVML or COXML (Fig. 1). In a system with distributed memory like an cluster, the application scaled well up to 56 processors (Fig. 2). With more processors, a theoretical scalability limit exists due to the hetereogeneous nature of the optimization times, from a few seconds for the simplest models to up to several hours for the models that include rate variation among sites (+G). This problem was solved with the hybrid memory approach. In this case, the scalability went beyond the previous limit, reaching up to 150 in the most complex cases with 8-core nodes (Fig. 3).

*To whom correspondence should be addressed.

Table 1. Real and simulated alignments analyzed

Dataset Abbreviation	Protein	Size $N \times L$	Base tree	Sequence execution time
RIB	Ribosomal protein	21 × 113	Fixed BIONJ	5.5 min
RIBML	”	”	ML tree	28 min
COX	Cytochrome C oxidase II	28 × 113	Fixed BIONJ	9.5 min
COXML	”	”	ML tree	55 min
HIV	HIV polimerase	36 × 1,034	Fixed BIONJ	44 min
HIVML	”	”	ML tree	160 min
10K	Simulated aln	50 × 10K	Fixed BIONJ	9.2 h
20K	”	50 × 20K	”	24.5 h
100K	”	50 × 100K	”	80 h

N indicates the number of sequences and *L* the length of the alignment. *Base tree* is the tree used likelihood optimization and *Seq. exec. time* is the time required to calculate the likelihood scores using the sequential version (i.e. a single thread).

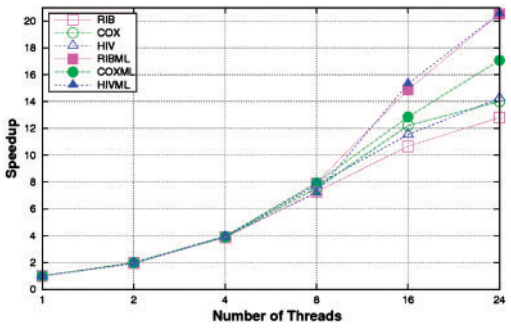

Fig. 1. Speed-ups obtained with the shared memory version of ProtTest 3 according to the numbers of threads used in a 24-core shared memory node (4 hexa-core Intel Xeon E7450 processors) with 12 GB memory.

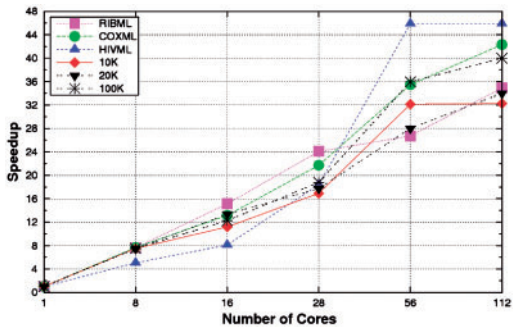

Fig. 2. Speed-ups obtained with the distributed memory version of ProtTest 3 according to the numbers of cores used in a 32-node cluster with 2 quad-core Intel Harpertown processors and 8 GB memory per node. Up to 4 processes were executed per node because of the memory requirements of the largest datasets (10K, 20K, 100K).

4 CONCLUSIONS

ProtTest 3 can be executed in parallel in HPC environments as: (i) a GUI-based desktop version that uses multicore processors; (ii) a cluster-based version that distributes the computational load among nodes; and (iii) as a hybrid multicore cluster version that achieves

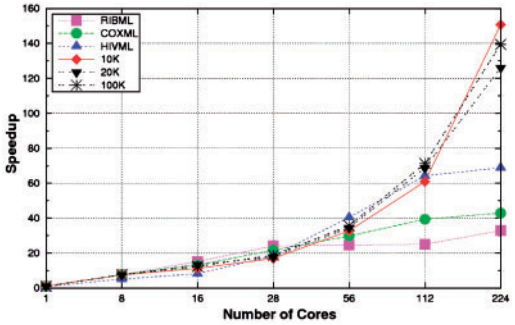

Fig. 3. Speed-ups obtained with the hybrid memory version of ProtTest 3 according to the numbers of cores used in the same 32-node cluster as Fig. 2. Up to 4 MPJ Express processes per node and at least 2 OpenMP threads for each ML optimization were executed.

speed through the distribution of tasks among nodes while taking advantage of multicore processors within nodes. The new version has been completely redesigned and includes new capabilities like checkpointing, additional amino acid replacement matrices, new model selection criteria and the possibility of computing model-averaged phylogenetic trees. The use of ProtTest 3 results in significant performance gains, with observed speedups of up to 150 on a high performance cluster. In this way, statistical model selection for large protein alignments becomes feasible, not only for cluster users but also for the owners of standard multicore desktop computers. Moreover, the flexible design of ProtTest-HPC will allow developers to extend future functionalities, whereas third-party projects will be able to easily adapt its capabilities to their requirements.

ACKNOWLEDGEMENTS

Special thanks to Stephane Guindon and to Federico Abascal for their help.

Funding: This work was financially supported by the European Research Council (ERC-2007-Stg 203161-PHYGENOM to D.P.); the Spanish Ministry of Science and Education (BFU2009-08611 to D.P.); Xunta de Galicia (Galician Thematic Networks RGB 2010/90 to D.P. and GHPC2 2010/53 to R.D.).

Conflict of Interest: none declared.

REFERENCES

Abascal,F. *et al*. (2007) ProtTest: selection of best-fit models of protein evolution. *Bioinformatics*, **24**, 1104–1105.

Dagum,L. and Menon,R. (1998) OpenMP: an industry-standard API for shared-memory programming. *IEEE Comput. Sci. Eng.*, **5**, 46–55.

Drummond,A. and Strimmer,K. (2001) Pal: an object-oriented programming library for molecular evolution and phylogenetics. *Bioinformatics*, **17**, 662–663.

Glez-Peña,D. *et al*. (2010) ALTER: program-oriented conversion of DNA and protein alignments. *Nucleic Acids Res.*, **38** (Suppl. 2), W14–W18.

Guindon,S. and Gascuel,O. (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.*, **52**, 696–704.

Posada,D. and Buckley,T.R. (2004) Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Syst. Biol.*, **53**, 793–808.

Shafi,A. *et al*. (2009) Nested parallelism for multi-core HPC systems using Java. *J. Parallel Distr. Com.*, **69**, 532–545.

Sullivan,J. and Joyce,P. (2005) Model selection in phylogenetics. *Annu. Rev. Ecol. Evol. S*, **36**, 445–466.