Bioinformatics, 31(15), 2015, 2577–2579
doi: 10.1093/bioinformatics/btv184
Advance Access Publication Date: 29 March 2015
Applications Note

OXFORD

Phylogenetics

ExaML version 3: a tool for phylogenomic analyses on supercomputers

Alexey M. Kozlov[1], Andre J. Aberer[1] and Alexandros Stamatakis[1,2,*]

[1]Scientific Computing Group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany and [2]Department of informatics, Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on January 7, 2015; revised on March 9, 2015; accepted on March 24, 2015

Abstract

Motivation: Phylogenies are increasingly used in all fields of medical and biological research. Because of the next generation sequencing revolution, datasets used for conducting phylogenetic analyses grow at an unprecedented pace. We present ExaML version 3, a dedicated production-level code for inferring phylogenies on whole-transcriptome and whole-genome alignments using supercomputers.

Results: We introduce several improvements and extensions to ExaML: Extensions of substitution models and supported data types, the integration of a novel load balance algorithm as well as a parallel I/O optimization that significantly improve parallel efficiency, and a production-level implementation for Intel MIC-based hardware platforms.

Availability and implementation: The code is available under GNU GPL at https://github.com/stamatak/ExaML.

Contact: Alexandros.Stamatakis@h-its.org

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

1 Introduction

ExaML (Exascale Maximum Likelihood) is a relatively new code for large-scale phylogenetic analyses on supercomputers. It implements the RAxML (Stamatakis, 2014) search algorithm and replaces RAxML-Light (Stamatakis *et al.*, 2012) which was inefficient for large, partitioned phylogenomic datasets that currently represent the typical use case. Hence, in ExaML (v. 1) we implemented a radically different parallelization approach (Stamatakis and Aberer, 2013) that improved parallel efficiency by up to a factor of 3 and also increased scalability. For two large alignments that were recently published in *Science* [51 taxa, unpartitioned, $3.22 \cdot 10^8$ DNA sites & 48 taxa, four partitions, $3.7 \cdot 10^7$ DNA sites in Jarvis *et al.* (2014); 144 taxa, 100 partitions, 1, 240, 377 DNA sites & 144 taxa, 770 partitions, 576 840 AA sites in Misof *et al.* (2014)], we identified and resolved further performance bottlenecks. Note that, ExaML can also be used for analyzing datasets with 10-20 genes and up to 55 000 taxa, but scalability will be limited to at most 100 cores. For ExaML (v. 2), we developed and integrated algorithms for improved load balance (Zhang and Stamatakis, 2012; Kobert *et al.*, 2014) and also optimized the parallel I/O for reading multiple sequence alignments. We also started exploring if and how ExaML can be ported to the Intel MIC (Many Integrated Core) hardware architecture (Kozlov *et al.*, 2014) in a proof-of-concept setting. Here, we present ExaML (v. 3) which offers—apart from new models and data types—a production-level implementation for the Intel MIC architecture that required a substantial amount of re-engineering. We also present a novel parallel alignment I/O method. Finally, we completely re-wrote the user manual.

ExaML is a stable and well-documented code for large-scale phylogenetic inference on x86 Linux/MAC clusters (compiles with `gcc,icc,clang`). It addresses and provides generally applicable solutions for several performance bottlenecks in parallel phylogentic likelihood calculations on partitioned alignments.

2 New features

2.1 Models and data types

Apart from DNA and protein data, ExaML now also supports binary (two-state) characters. This data type can be used, for instance, to analyze genome-wide indel patterns.

The number of available protein substitution models now also includes the LG4M and LG4X models (Le *et al.*, 2012) as well as the recently published stmtREV (Liu *et al.*, 2014) model. In addition, ExaML can automatically determine the best-scoring protein substitution model for each partition via a newly implemented standard test procedure that uses either (i) the likelihood score, (ii) the AIC (Akaike Information Criterion), (iii) the cAIC (corrected AIC) or (iv) the BIC (Bayesian Information Criterion).

Finally, a new option for conducting a maximum likelihood estimate of the base frequencies has become available.

2.2 Parallel I/O optimization

The parallel I/O to read in the input alignment is optimized in two ways. Initially, a plain-text PHYLIP file is analyzed and transformed into a binary file format via a dedicated parser component. The parser can be executed on a standard server and does therefore not consume valuable supercomputer time for issues such as checking the format, detecting duplicate taxon names, compressing site patterns, etc. The binary alignment file contains global data information (alignment length, data types, model options, partition boundaries) as well as the raw sequences stored in the order of the partitions. That is, the sequences for all taxa of partition one are stored first, then, the sequences for all taxa of partition two, etc. This, yet unpublished, binary format allows each ExaML process to concurrently read only those parts of the alignment on which it will be computing likelihoods. This optimization yielded an acceleration of more than one order of magnitude for the start-up phase of ExaML during which the alignment is read (see on-line Supplementary for more details).

2.3 New load balance algorithm

The typical use case for modern likelihood-based inferences are so-called partitioned analyses, where subsets of the alignment sites form partitions that evolve under a distinct set of evolutionary model parameters (e.g. α shape parameter, GTR rates, base frequencies, etc.). ExaML parallelizes likelihood calculations over alignment sites using MPI (Message Passing Interface). The time for likelihood calculations on a partition i consist of a constant part (irrespective of the partition length), mainly the calculation of the P_i matrix via exponentiation of the Q_i matrix given the branch length t, that is, $P_i(t) = e^{Q_i t}$ (Felsenstein, 2004). Once the P_i matrix has been computed one can then calculate the per-site likelihoods for each site in partition i in parallel. We observed that, because of additional synchronization and communication overhead, it is not advantageous to first parallelize all P matrix calculation and subsequently (in a second parallel region) calculate all per-site likelihood calculations. Thus, because P_i needs to be computed redundantly by every process, even if it has just one alignment site belonging to partition i, we need to minimize the number of partitions for which a process calculates the likelihood. At the same time, we need to evenly distribute the sites among processors for optimal load balance (i.e. we need to split up some partitions among processes). To this end, we formulated a bi-criterion problem to define the optimal data distribution of partitions and sites to processors (Kobert *et al.*, 2014). We showed that: (i) the optimization problem is $\mathcal{NP}$-hard, (ii) an approximation algorithm with a guaranteed bound exists, (iii) the algorithm misses the optimum by at most one partition (one additional P_i calculation is required at one or more processes than in the optimal solution). We also showed that this new data distribution algorithm outperforms previous approaches (cyclic distribution of sites, monolithic distribution of partitions) in almost all cases with only minor deviations in cases where the performance was worse. In addition, we showed that ExaML runs up to three times faster [see Fig. 4b in Kobert *et al.* (2014)] than with the previous data distribution schemes.

2.4 Checkpointing

As RAxML-Light, ExaML also allows for checkpointing and subsequently re-starting tree searches from light-weight binary checkpoints. This is particularly useful when using typical supercomputer configurations with 24 or 48 hr run-time limits. Apart from the tree search, ExaML also offers an option to estimate model parameters and branch lengths on a given set of fixed trees. This option is also checkpointable.

2.5 ExaML MIC version

An increasing number of supercomputing centers now operate systems with heterogeneous CPU/MIC compute nodes. To this end, we have transformed our initial proof-of-concept MIC code into production-level hybrid software that can leverage the capacity of all resources in such a system. A detailed description of this novel and non-trivial parallelization approach that requires exploiting parallelism and handling load balance at essentially two levels (MPI among MIC cards and 'normal' CPUs and OpenMP within cards) is provided in the on-line Supplementary. As we show in the supplement, the hybrid code allows to make better use of currently available hardware resources.

3 User support and future work

User support is provided via the RAxML Google group at: https://groups.google.com/forum/?hl=en#!forum/raxml. The ExaML source code contains a comprehensive manual.

Future work includes the continued maintenance and support of ExaML and the implementation of additional models [e.g. models with ascertainment bias correction (Lewis, 2001)], data types and search algorithms (Nguyen *et al.*, 2015).

Acknowledgements

The authors wish to thank lab member Diego Darriba for help with the implementation of the AIC, cAIC and BIC tests.

Funding

All three authors are funded by institutional funds from the Heidelberg Institute for Theoretical Studies for theoretical studies.

Conflict of Interest: none declared.

References

Felsenstein,J. (2004) *Inferring Phylogenies*. Sinauer Associates Sunderland, Sunderland, MA.

Jarvis,E.D. *et al.* (2014) Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, **346**, 1320–1331.

Kobert,K. *et al.* (2014) The divisible load balance problem and its application to phylogenetic inference. In: Brown,D. and Morgenstern,B. (eds.) *Algorithms in Bioinformatics,* Vol. 8701 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 204–216.

Kozlov,A.M. *et al.* (2014) Efficient computation of the phylogenetic likelihood function on the Intel MIC architecture. In: *Parallel Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International,* IEEE Computer Society Washington, DC, Phoenix, Arizona, pp. 518–527.

Le,S.Q. *et al.* (2012) Modeling protein evolution with several amino acid replacement matrices depending on site rates. *Mol. Biol. Evol.*, **29**, 2921–2936.

Lewis,P.O. (2001) A likelihood approach to estimating phylogeny from discrete morphological character data. *Syst. Biol.*, **50**, 913–925.

Liu,Y. *et al.* (2014) Mitochondrial phylogenomics of early land plants: mitigating the effects of saturation, compositional heterogeneity, and codon-usage bias. *Syst. Biol.*, **63**, 862–878.

Misof,B. *et al.* (2014) Phylogenomics resolves the timing and pattern of insect evolution. *Science*, **346**, 763–767.

Nguyen,L.-T. *et al.* (2015) IQ-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.*, **32**, 268–274.

Stamatakis,A. (2014) Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, **30**, 1312–1313.

Stamatakis,A. and Aberer,A. (2013) Novel parallelization schemes for large-scale likelihood-based phylogenetic inference. In: *Parallel Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on,* IEEE Computer Society Washington, DC, USA, Boston, MA, pp. 1195–1204.

Stamatakis,A. *et al.* (2012) Raxml-light: a tool for computing terabyte phylogenies. *Bioinformatics*, **28**, 2064–2066.

Zhang,J. and Stamatakis,A. (2012) The multi-processor scheduling problem in phylogenetics. In: *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International,* IEEE Computer Society Washington, DC, USA, Shanghai, China, pp. 691–698.