BIOINFORMATICS **APPLICATIONS NOTE**

Vol. 30 no. 9 2014, pages 1312–1313
doi:10.1093/bioinformatics/btu033

Phylogenetics

Advance Access publication January 21, 2014

RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies

Alexandros Stamatakis[1,2]

[1]Scientific Computing Group, Heidelberg Institute for Theoretical Studies, 69118 Heidelberg and [2]Department of Informatics, Institute of Theoretical Informatics, Karlsruhe Institute of Technology, 76128 Karlsruhe, Germany

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Phylogenies are increasingly used in all fields of medical and biological research. Moreover, because of the next-generation sequencing revolution, datasets used for conducting phylogenetic analyses grow at an unprecedented pace. RAxML (Randomized Axelerated Maximum Likelihood) is a popular program for phylogenetic analyses of large datasets under maximum likelihood. Since the last RAxML paper in 2006, it has been continuously maintained and extended to accommodate the increasingly growing input datasets and to serve the needs of the user community.

Results: I present some of the most notable new features and extensions of RAxML, such as a substantial extension of substitution models and supported data types, the introduction of SSE3, AVX and AVX2 vector intrinsics, techniques for reducing the memory requirements of the code and a plethora of operations for conducting post-analyses on sets of trees. In addition, an up-to-date 50-page user manual covering all new RAxML options is available.

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

Contact: alexandros.stamatakis@h-its.org

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

Received on December 22, 2013; revised and accepted on January 14, 2014

1 INTRODUCTION

RAxML (Randomized Axelerated Maximum Likelihood) is a popular program for phylogenetic analysis of large datasets under maximum likelihood. Its major strength is a fast maximum likelihood tree search algorithm that returns trees with good likelihood scores. Since the last RAxML paper (Stamatakis, 2006), it has been continuously maintained and extended to accommodate the increasingly growing input datasets and to serve the needs of the user community. In the following, I will present some of the most notable new features and extensions of RAxML.

2 NEW FEATURES

2.1 Bootstrapping and support values

RAxML offers four different ways to obtain bootstrap support. It implements the standard non-parametric bootstrap and also the so-called rapid bootstrap (Stamatakis *et al*., 2008), which is a standard bootstrap search that relies on algorithmic shortcuts and approximations to speed up the search process.

It also offers an option to calculate the so-called SH-like support values (Guindon *et al*., 2010). I recently implemented a method that allows for computing RELL (Resampling Estimated Log Likelihoods) bootstrap support as described by Minh *et al*. (2013).

Apart from this, RAxML also offers a so-called bootstopping option (Pattengale *et al*., 2010). When this option is used, RAxML will automatically determine how many bootstrap replicates are required to obtain stable support values.

2.2 Models and data types

Apart from DNA and protein data, RAxML now also supports binary, multi-state morphological and RNA secondary structure data. It can correct for ascertainment bias (Lewis, 2001) for all of the above data types. This might be useful not only for morphological data matrices that only contain variable sites but also for alignments of SNPs.

The number of available protein substitution models has been significantly extended and comprises a general time reversible (GTR) model, as well as the computationally more complex LG4M and LG4X models (Le *et al*., 2012). RAxML can also automatically determine the best-scoring protein substitution model.

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

2.3 Parallel versions

RAxML offers a fine-grain parallelization of the likelihood function for multi-core systems via the PThreads-based version and a coarse-grain parallelization of independent tree searches via MPI (Message Passing Interface). It also supports coarse-grain/fine-grain parallelism via the hybrid MPI/PThreads version (Pfeiffer and Stamatakis, 2010).

Note that, for extremely large analyses on supercomputers, using the dedicated sister program ExaML [Exascale Maximum Likelihood (Stamatakis and Aberer, 2013)] is recommended.

2.4 Post-analysis of trees

RAxML offers a plethora of post-analysis functions for sets of trees. Apart from standard statistical significance tests, it offers efficient (and partially parallelized) operations for computing

Robinson–Foulds distances, as well as extended majority rule, majority rule and strict consensus trees (Aberer *et al*., 2010).

Beyond this, it implements a method for identifying the so-called rogue taxa (Pattengale *et al*., 2011), and I recently implemented options for calculating the TC (Tree Certainty) and IC (Internode Certainty) measures as introduced by Salichos and Rokas (2013).

Finally, there is the new plausibility checker option (Dao *et al*., 2013) that allows computing the RF distances between a huge phylogeny with tens of thousands of taxa and several smaller more accurate reference phylogenies that contain a strict subset of the taxa in the huge tree. This option can be used to automatically assess the quality of huge trees that can not be inspected by eye.

2.5 Analyzing next-generation sequencing data

RAxML offers two algorithms for preparing and analyzing next-generation sequencing data. A sliding-window approach (unpublished) is available to assess which regions of a gene (e.g. 16S) exhibit strong and stable phylogenetic signal to support decisions about which regions to amplify. Apart from that, RAxML also implements parsimony and maximum likelihood flavors of the evolutionary placement algorithm [EPA (Berger *et al*., 2011)] that places short reads into a given reference phylogeny obtained from full-length sequences to determine the evolutionary origin of the reads. It also offers placement support statistics for those reads by calculating likelihood weights. This option can also be used to place fossils into a given phylogeny (Berger and Stamatakis, 2010) or to insert different outgroups into the tree *a posteriori*, that is, after the inference of the ingroup phylogeny.

2.6 Vector intrinsics

RAxML uses manually inserted and optimized x86 vector intrinsics to accelerate the parsimony and likelihood calculations. It supports SSE3, AVX and AVX2 (using fused multiply-add instructions) intrinsics. For a small single-gene DNA alignment using the Γ model of rate heterogeneity, the unvectorized version of RAxML requires 111.5 s, the SSE3 version 84.4 s and the AVX version 66.22 s to complete a simple tree search on an Intel i7-2620 M core running at 2.70 GHz under Ubuntu Linux.

The differences between AVX and AVX2 are less pronounced and are typically below 5% run time improvement.

2.7 Saving memory

Because memory shortage is becoming an issue due to the growing dataset sizes, RAxML implements an option for reducing memory footprints and potentially run times on large phylogenomic datasets with missing data. The memory savings are proportional to the amount of missing data in the alignment (Izquierdo-Carrasco *et al*., 2011)

2.8 Miscellaneous new options

RAxML offers options to conduct fast and more superficial tree searches on datasets with tens of thousands of taxa. It can also compute marginal ancestral states and offers an algorithm for rooting trees. Furthermore, it implements a sequential, PThreads-parallelized and MPI-parallelized algorithm for computing all quartets or a subset of quartets for a given alignment.

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 RAxML source code contains a comprehensive manual and there is a step-by-step tutorial with some basic commands available at http://www.exelixis-lab.org/web/software/raxml/hands_on.html. Further resources are available via the RAxML software page at http://www.exelixis-lab.org/web/software/raxml/

Future work includes the continued maintenance of RAxML, the adaptation to novel computer architectures and the implementation of novel models and datatypes, in particular codon models.

ACKNOWLEDGEMENT

The author thank several colleagues for contributing code to RAxML: Andre J. Aberer, Simon Berger, Alexey Kozlov, Nick Pattengale, Wayne Pfeiffer, Akifumi S. Tanabe, David Dao and Charlie Taylor.

Funding: This work was funded by institutional funding provided by the Heidelberg Institute for Theoretical Studies.

Conflict of Interest: none declared.

REFERENCES

Aberer,A.J. *et al.* (2010) Parallelized phylogenetic post-analysis on multi-core architectures. *J. Comput. Sci.*, **1**, 107–114.

Berger,S.A. and Stamatakis,A. (2010) Accuracy of morphology-based phylogenetic fossil placement under maximum likelihood. In: *International Conference on Computer Systems and Applications (AICCSA), 2010 IEEE/ACS*. IEEE, New York, USA, pp. 1–9.

Berger,S.A. *et al.* (2011) Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Syst. Biol.*, **60**, 291–302.

Dao,D. *et al.* (2013) Automated plausibility analysis of large phyolgenies. *Technical report*. Karlsruhe Institute of Technology.

Guindon,S. *et al.* (2010) New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0. *Syst. Biol.*, **59**, 307–321.

Izquierdo-Carrasco,F. *et al.* (2011) Algorithms, data structures, and numerics for likelihood-based phylogenetic inference of huge trees. *BMC Bioinformatics*, **12**, 470.

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.

Minh,B.Q. *et al.* (2013) Ultrafast approximation for phylogenetic bootstrap. *Mol. Biol Evol.*, **30**, 1188–1195.

Pattengale,N.D. *et al.* (2010) How many bootstrap replicates are necessary? *J. Comput. Biol.*, **17**, 337–354.

Pattengale,N.D. *et al.* (2011) Uncovering hidden phylogenetic consensus in large data sets. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, **8**, 902–911.

Pfeiffer,W. and Stamatakis,A. (2010) Hybrid mpi/pthreads parallelization of the raxml phylogenetics code. In *International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE*. IEEE, New York, USA, pp. 1–8.

Salichos,L. and Rokas,A. (2013) Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature*, **497**, 327–331.

Stamatakis,A. (2006) Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, **22**, 2688–2690.

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

Stamatakis,A. *et al.* (2008) A rapid bootstrap algorithm for the raxml web servers. *Syst. Biol.*, **57**, 758–771.