BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 10 2014, pages 1476–1477
doi:10.1093/bioinformatics/btu053

Phylogenetics Advance Access publication January 28, 2014

PUmPER: phylogenies updated perpetually

Fernando Izquierdo-Carrasco[1,*], John Cazes[2], Stephen A. Smith[3] and Alexandros Stamatakis[1,4]

[1]Scientific Computing Group, Heidelberg Institute for Theoretical Studies (HITS gGmbH), Schloss-Wolfsbrunnenweg 35, D-69118 Heidelberg, Baden-Württemberg, Germany, [2]Texas Advanced Computing Center, University of Texas, Austin, TX, 78758, USA, [3]University of Michigan, Ann Arbor Department of Ecology and Evolutionary Biology, 48109, MI, USA and [4]Karlsruhe Institute of Technology, Institute for Theoretical Informatics, Postfach 6980, 76128 Karlsruhe, Baden-Württemberg, Germany

Associate Editor: David Posada

ABSTRACT

Summary: New sequence data useful for phylogenetic and evolutionary analyses continues to be added to public databases. The construction of multiple sequence alignments and inference of huge phylogenies comprising large taxonomic groups are expensive tasks, both in terms of man hours and computational resources. Therefore, maintaining comprehensive phylogenies, based on representative and up-to-date molecular sequences, is challenging. PUmPER is a framework that can perpetually construct multi-gene alignments (with PHLAWD) and phylogenetic trees (with ExaML or RAxML-Light) for a given NCBI taxonomic group. When sufficient numbers of new gene sequences for the selected taxonomic group have accumulated in GenBank, PUmPER automatically extends the alignment and infers extended phylogenetic trees by using previously inferred smaller trees as starting topologies. Using our framework, large phylogenetic trees can be perpetually updated without human intervention. Importantly, resulting phylogenies are not statistically significantly worse than trees inferred from scratch.

Availability and implementation: PUmPER can run in stand-alone mode on a single server, or offload the computationally expensive phylogenetic searches to a parallel computing cluster. Source code, documentation, and tutorials are available at https://github.com/fizquierdo/perpetually-updated-trees.

Contact: Fernando.Izquierdo@h-its.org

Supplementary information: Supplementary Material is available at *Bioinformatics* online.

Received on August 14, 2013; revised on November 22, 2013; accepted on January 23, 2014

1 INTRODUCTION

Sequence data continues to accumulate in public databases at an increasing pace. With the addition of new data for species and individuals or genes, existing phylogenies of taxonomic groups need to be updated. Reinitiating phylogenetic inferences from scratch every time new data are added to public databases represents wasted effort in the form of man-hours and energy consumption (computations). Nonetheless, adding new taxa or genes to a phylogenetic tree may also unravel new evolutionary relationships that were not supported by previous smaller datasets.

We present the PUmPER framework and make available a phylogenetic analysis pipeline that can automatically update existing huge reference phylogenies *and* alignments by new sequence data, without the need to recompute everything from scratch. We call this procedure a perpetual tree update.

Other pipelines, such as STAP (Wu *et al*., 2008), have used publicly available databases and packages like BLASTN, ClustalW and PhyML to automate the process of alignment construction and phylogenetic inference. The *mor* (Hibbett *et al*., 2005) pipeline implements the automated extension of phylogenetic trees with new taxa, albeit with the specific focus of producing automated taxonomies of *homobasidiomycetes*.

2 FRAMEWORK OVERVIEW

In the following, we outline our framework for perpetually updating phylogenetic trees that can handle > 20 000 taxa. PUmPER comprises (i) an extension of PHLAWD (Smith *et al*., 2009) that retrieves GenBank sequences and subsequently builds or extends multiple sequence alignments (MSAs), and (ii) the phylogenetic tree inference component, based on ExaML (Stamatakis and Aberer, 2013) and RAxML-Light (Stamatakis *et al*., 2012), that infers/extends the trees via maximum likelihood (ML) tree searches. ExaML and RAxML-Light are dedicated HPC versions of RAxML that runs on clusters using the Message Passing Interface (MPI). It can infer new trees from scratch or extend given trees by additional taxa. Additionally, we developed an iterative procedure that perpetually updates trees. Each iteration consists of the execution of two stages: the generation of an MSA, and the subsequent inference of a set of trees.

The *initial iteration* is special because it builds the initial MSA and ML tree set from scratch. Setting up an initial iteration requires editing a PHLAWD configuration file. This file contains the NCBI taxonomic rank (clade name) and the gene(s) for which a MSA shall be assembled. PHLAWD then queries GenBank to construct the MSA. Based on this initial MSA, our framework conducts a given number of independent ML tree searches (on distinct randomized stepwise addition order parsimony starting trees) and executes them in parallel to generate an initial set of ML trees. The number of tree searches to conduct, and the size of the tree set to keep, is specified by the user in a separate configuration file.

[*]To whom correspondence should be addressed.

We call all subsequent iterations *update iterations* because they extend the MSA and trees of the preceding iteration. An update iteration carries out the following four steps: (i) MSA extension with PHLAWD according to the initial configuration file; (ii) generation of distinct randomized stepwise addition order parsimony starting trees with Parsimonator (unpublished; available under GNU GPL at https://github.com/stamatak/Parsimonator-1.0.2), extending the set of trees from the previous iteration by the newly added taxa; (iii) ML optimization of the comprehensive parsimony starting trees that now include all new taxa with ExaML or RAxML-Light; and (iv) selection of a subset of these ML trees (based on their likelihood scores) that will be used as starting points for the next iteration.

Update iterations are either initiated manually via the command-line interface or triggered automatically. A tree update iteration is initiated if (i) the alignment from the previous iteration has been extended *and* (ii) the phylogenetic analyses of the previous iteration have been completed.

Our framework also supports generating multigene alignments with PHLAWD. For each gene region of interest, we execute an independent PHLAWD instance to generate a single-gene MSA. Each PHLAWD instance has its own configuration and sequence-seed file. Thereafter, we concatenate all single-gene MSAs into a multi-partitioned dataset and store the gene boundaries in a RAxML-formatted partition file.

3 IMPLEMENTATION

All components of the framework are open source. PUmPER is based on Ruby modules that can be seamlessly used in Ruby scripts. Each Ruby module encapsulates an independent function or wrapper, that is, the user does not need to be aware of the specific usage of the underlying basic tools. Configuration files specify clade- and gene-specific settings. Although our main use case is the automated update of phylogenetic trees, the framework can easily be adapted to build custom phylogenetic pipelines. For example, if alignments are already available, the PHLAWD component is not required. The online repository includes a detailed installation guide, as well as some basic usage configuration examples.

Under the default configuration, PUmPER operates in stand-alone mode on a single server. PHLAWD and ExaML are executed locally on this server. The individual ExaML tree searches are conducted one after the other. Although PUmPER deploys ExaML by default, it can be configured to use the PThreads version of RAxML-Light on multi-core servers. Thus, this stand-alone version already allows for updating large trees with thousands of taxa on a medium-sized lab server.

For huge trees with tens of thousands of taxa, the computational resources of a single server may be insufficient because of memory and/or time constraints. Thus, PUmPER can offload the computationally intense ML calculations to a cluster system. Thereby, the trees are updated in a timely manner while the process is still being orchestrated by a local server. This requires the perpetual tree framework to interface with remote systems using standard communication tools (scp and ssh), batch schedulers (we have successfully used the framework with SGE and SLURM) and to also use optimized executables for the remote target system (Parsimonator, ExaML, RAxML-Light and RAxML). Although this adds some complexity, it is required to infer trees whose size requires a large amount of computational resources.

Table 1. Original run and two update iterations for rbcL alignments of the Embryophyta clade

Iteration	Taxa	Sites	Avg LH (30)	Runtime (h)
2008	12 072	1437	−848794.80	46.55
2010	16 962	1427	−1005824.25	68.36
2010 (scratch)	16 962	1427	−1005931.37	70.89
2012	21 791	1424	−1108161.66	93.40
2012 (scratch)	21 791	1424	−1108243.29	97.42

Note: The Iteration numbers reflect the amount of data available in GenBank in past years (2008–2012). The complete data are available in Supplementary Table S1.

4 RESULTS AND DISCUSSION

According to our first results (see Table 1 and Supplementary Material for details), the iterative MSA and tree extension approach does not yield statistically significantly better (or worse) trees than the standard (inference from scratch) approach with respect to the likelihood scores. The topological accuracy in our simulations is comparable in both approaches. The runtimes of the perpetual inferences are slightly, but consistently, lower than for inferences from scratch. We view the main contribution, however, in saving man-hours; alignment construction, job setup, filtering and post-analyzing results are tedious tasks that consume a significant, and hard to quantify, amount of time.

We are currently operating an instance of our framework as part of the iPlant collaborative (http://www.iplantcollaborative.org/) to maintain and make available perpetually updated trees for the *Viridiplantae* clade (using the *rbc*L, *matK* and *atpB* genes). The details of our setup can be found in the Supplementary Material.

ACKNOWLEDGEMENT

The authors acknowledge Texas Advanced Computing Center (TACC) and Extreme Science and Engineering Discovery Environment (XSEDE).

Funding: F.I.-C. was funded via DFG grants (STA 860/2 and STA 860/3). S.A.S. was funded by NSF (1207915 and 0735191). iPlant Collaborative (NSF Grant #DBI-0735191).

Conflict of Interest: none declared.

REFERENCES

Hibbett,D.S. *et al.* (2005) Automated phylogenetic taxonomy: an example in the homobasidiomycetes (mushroom-forming fungi). *Syst. Biol.*, **54**, 660–668.

Smith,S.A. *et al.* (2009) Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. *BMC Evol. Biol.*, **9**, 37.

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.* pp. 1195–1204.

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

Wu,D. *et al.* (2008) An automated phylogenetic tree-based small subunit rRNA taxonomy and alignment pipeline (STAP). *PLoS One*, **3**, e2566.