RAxML Grove: An empirical Phylogenetic Tree Database

Dimitri Höhler[1], Wayne Pfeiffer[3], Vassilios Ioannidis[4], Heinz Stockinger[4], and Alexandros Stamatakis[1,2]

[1]Computational Molecular Evolution group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
[2]Institute for Theoretical Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[3]San Diego Supercomputer Center, University of California, San Diego, La Jolla, United States
[4]SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland

Abstract

The assessment of novel phylogenetic models and inference methods is routinely being conducted via experiments on simulated as well as empirical data. When generating synthetic data it is often unclear how to set simulation parameters for the models and generate trees that appropriately reflect empirical model parameter distributions and tree shapes. As a solution, we present and make available a new database called 'RAxML Grove' currently comprising more than 60,000 inferred trees and respective model parameter estimates from fully anonymized empirical data sets that were analyzed using RAxML (1) and RAxML-NG (2) on two web servers. We also describe and make available two simple applications of RAxML Grove to exemplify its usage and highlight its utility for designing realistic simulation studies and analyzing empirical model parameter and tree shape distributions.

RAxML Grove is freely available at https://github.com/angtft/RAxMLGrove.

Correspondence: ***Alexandros Stamatakis, alexandros.stamatakis@h-its.org***

The field of computational phylogenetics focuses on developing inference methods and models for reconstructing the evolutionary history among distinct species. Inferring the evolutionary history of the species under study commonly involves sequencing and aligning the species' genomes (or parts thereof) to obtain a *multiple sequence alignment* (MSA), which typically serves as input for a *phylogenetic inference* method. There exists a plethora of widely used tools for phylogenetic inference such as, for instance, BEAST (3), MrBayes (4), IQ-TREE (5), and FastTree2 (6). For developing and assessing new phylogenetic algorithms, tools, and models, using empirical data as well as realistic simulated data is mandatory (e.g., for submissions to *Bioinformatics*). To the best of our knowledge, there exist two empirical online databases comprising a sizable amount of phylogenetic data (i.e., thousands of phylogenetic trees): TreeBASE (7) and PhyloFacts (8). TreeBASE offers published peer-reviewed phylogenetic data sets. It also offers programmatic access to all data files via a web API and currently comprises approximately $13,000$ phylogenies. The PhyloFacts database contains over $50,000$ trees with their respective MSAs. However, it appears that the last update of the main database was conducted in September 2011. In addition, it does not offer programmatic data access.

With the RAxML Grove (RG) database we offer a new, freely accessible database with a different focus and data collection model. The main goal of RG is to provide data that allows one to study, summarize, and extract empirical parameter distributions, tree shapes, and other 'interesting' characteristics (e.g., the missing data pattern or the size distribution of MSA data partitions) of phylogenetic inferences on empirical data sets. These data can subsequently be used for informing the design of realistic simulation studies that reflect the properties of empirical data, thereby supporting the development of novel models and methods. In addition, RG is a constantly growing database as it perpetually collects phylogenetic trees and parameter estimates inferred by users on the RAxML/RAxML-NG web servers at the San Diego Supercomputer Center (9) and the SIB Swiss Institute of Bioinformatics (10). In contrast to TreeBASE and PhyloFacts, we do not make available either the MSAs or the original taxon names in the trees to protect unpublished work by the web server users.

Data Collection

RAxML typically requires the user to specify an MSA file and a substitution model to infer a tree. When RAxML terminates, it returns the best tree it was able to find as well as maximum likelihood (ML) model parameter estimates (e.g., the substitution rates, branch lengths, base frequencies etc.). As tree inference under ML is computationally expensive, it can be conducted on the respective RAxML/RAxML-NG web servers ((10), (9)). Anyone can submit jobs to these servers to infer trees with RAxML. The servers report the availability of the result files back to the user once the inference has completed. We use these result files as well as the user supplied MSA and partition files to generate anonymized files comprising numerical information about the MSA and the inferred tree(s) on the respective web servers. During the anonymization, we replace *all* taxon and partition names by generic names and recover only specific subsets of the data available in the RAxML/RAxML-NG log files. The data we collect are the inferred best trees (and per-partition trees, if available) along with branch lengths and the following quantities for every partition: The inferred base frequencies, the substitution model used, the number of alignment sites and alignment patterns, the percentage of

gaps, the proportion of invariant sites, the α shape parameter of the Γ model of rate heterogeneity, and the substitution rates. For partitioned data sets we also compute binary presence/absence matrices using a specific version of IQ-TREE (https://github.com/iqtree/iqtree2/tree/terragen). A binary presence absence matrix B contains columns of 0s and 1s for each partition of a partitioned MSA. $B[i][p]$ is 0 if sequence i in partition p only contains missing data; otherwise it is 1. These presence/absence matrices will help us study the phenomena of terraces in tree space (11).

Afterwards, we upload these files to our GitHub repository at https://github.com/angtft/RAxMLGrove using GitPython (https://github.com/gitpython-developers/GitPython). We chose GitHub because it is easy to use by both developers and users. The tree data are stored in separate directories, one directory per computed web server job. RG is automatically updated with new trees on a monthly basis.

Figure 1 provides an overview of the RG data collection and post-processing procedures which use RAxMLGroveScripts (RGS). RGS is a GitHub repository aimed to help to post-process the RG data and to increase the overall usability of RG. It is available at https://github.com/angtft/RAxMLGroveScripts.

Our implementations of the data collection and post-processing scripts re-use components from the following libraries and tools: Biopython (12, 13), Dawg (14), Genesis (15), GitPython, IQ-TREE (5), Matplotlib (16), NumPy (17), Pandas (18), and SeqGen (19).

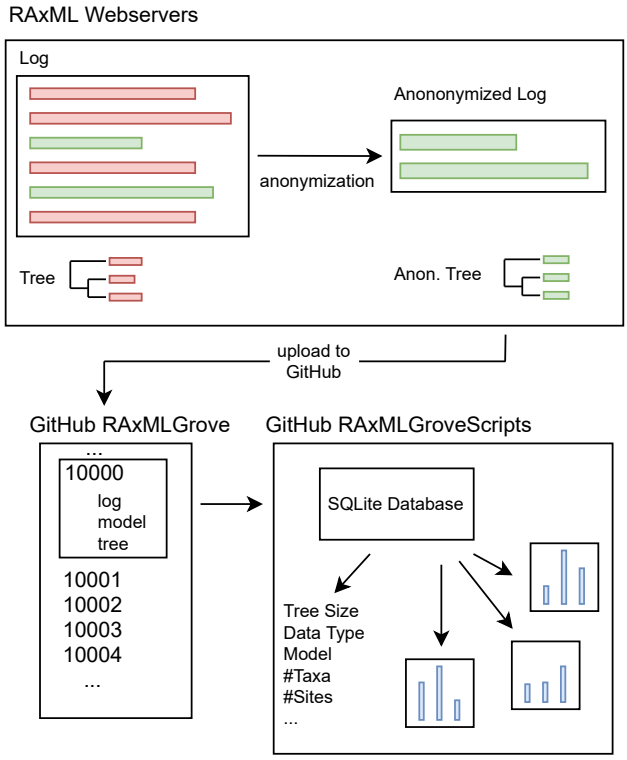

Fig. 1. Data collection and post-processing for RAxML Grove.

Applications

We present two possible usage scenarios for RG. Implemented solutions for the presented scenarios are available in the RGS repository.

Tree Download and Sequence Generation Script. The typical approach to simulate MSAs and respective trees is to generate a true tree using tools such as Zombi (20) or SimPhy (21) and subsequently simulate sequence data along that tree with Dawg (14) or SeqGen (19), for instance. One recurrent challenge in this procedure is to supply the 'correct' parameters to the simulations tools such that the simulated data are comparable to empirical data. In addition, it is difficult to provide rationales for the chosen parameter settings. However, by using RG it is straightforward to generate simulated data that resemble empirical data (e.g., by drawing simulation parameters from the histograms) and to justify the simulation parameter settings.

For such use, simply downloading any random trees including branch lengths and model parameter estimates from RG might be sufficient. However, if one intends to download trees with specific attributes (e.g., the number of taxa being above a certain threshold) or to filter out trees (e.g., trees inferred on protein data or unpartitioned data), one would need to download the entire database and parse it to appropriately sub-sample the data. To facilitate this task, we created a SQLite database with a corresponding Python script for easy access. Currently, storing the full SQLite database requires

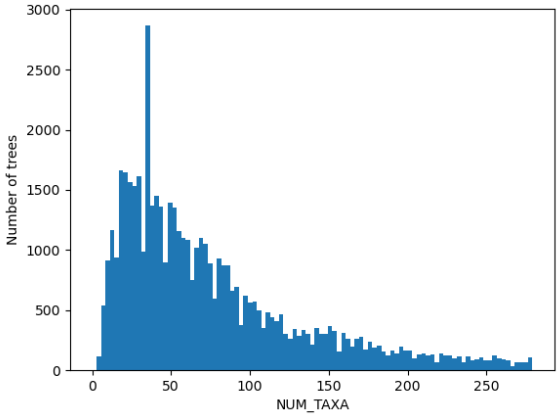

Fig. 2. Histogram of the number of taxa in trees stored in RG. Trees with numbers of taxa outside the range of Tukey's Fences ($k = 1.5$) were filtered.

approximately 124 MB of disk space as opposed to more than 1 GB required for the entire database. It nonetheless contains all data listed in the Data Collection Section as well as other properties, such as the tree diameter or branch length variance. The memory footprint of this SQLite database is small because we only store tree IDs instead of the entire NEWICK-formatted trees. Our script will then only download the trees of interest from the online database for the subsample we want to consider using these IDs. The search for trees with specific attributes can be performed using common SQL syntax. Additionally, the script can filter outliers using Tukey's Fences (22) and also automatically simulate sequences based on the sub-sampled trees and their respective model parameter estimates using Dawg or SeqGen.
Furthermore, the script can simulate sequences with incomplete data for partitioned trees using the presence/absence matrices. For every taxon i and partition p the script simulates a sequence $s_{i,p}$ if the absence matrix B is 1 at $B[i][j]$ and fills the sequence with missing data symbols if $B[i][j] = 0$. Then, the assembled sequence for each taxon i is $a_i = \sum_p s_{i,p}$, where the summation represents a concatenation.

Histograms. One obvious application is to generate empirical statistical distributions for important characteristics of a phylogenetic inference, such as the number of taxa, the evolutionary models used with respective substitution rates and among site rate heterogeneity parameters, or the tree shapes. We used the previously described SQLite database to generate histograms for some of the present columns. Before generating a histogram for a column of interest, we remove the outliers using Tukey's Fences (22) (with $k = 1.5$). Figure 2 shows an example distribution of the number of trees versus the number of taxa in those trees. Additional examples are available in the RGS repository.
These histograms can also be used to set 'good' default starting values for the likelihood model parameters in ML phylogenetic inference tools or serve as empirical prior distributions in Bayesian phylogenetic inference.

Summary

RAxML Grove is a new online database consisting of phylogenetic trees and their respective model parameters as inferred from thousands of RAxML and RAxML-NG runs made via online web servers. To protect unpublished work by users of the servers, taxon names have been anonymized in the trees, and the MSAs are not provided. On the other hand, we have provided presence/absence matrices of partitioned MSAs to capture missing data patterns contained therein.
Two usage scenarios of the RG database have been described. One is to download selected data for use in realistic simulations. The other is to construct histograms corresponding to the distributions of various tree or model parameters of interest.

Acknowledgements

We wish to thank Mark Miller (SDSC) for helpful comments on the initial draft of this manuscript, Hon Wai Wan (SIB) for technical support and Olga Chernomor (CIBIV) for help on extracting binary missing data matrices.

Funding

Part of this work was funded by the Klaus Tschira Foundation.

Bibliography

1. Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 01 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu033.
2. Alexey M Kozlov, Diego Darriba, Tomáš Flouri, Benoit Morel, and Alexandros Stamatakis. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 35(21):4453–4455, 05 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz305.
3. Alexei J Drummond, Marc A Suchard, Dong Xie, and Andrew Rambaut. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular biology and evolution*, 29(8):1969–1973, 2012.
4. John P Huelsenbeck and Fredrik Ronquist. Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.
5. Lam-Tung Nguyen, Heiko A Schmidt, Arndt Von Haeseler, and Bui Quang Minh. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular biology and evolution*, 32(1):268–274, 2015.
6. Morgan N Price, Paramvir S Dehal, and Adam P Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490, 2010.
7. W. H. Piel, L. Chan, M. J. Dominus, J. Ruan, R. A. Vos, and V. Tannen. TreeBASE v. 2: A Database of Phylogenetic Knowledge. *e-BioSphere 2009*, 2009.
8. PhyloFacts. http://phylogenomics.berkeley.edu/. Accessed: 2021-09-07.
9. MA Miller, W Pfeiffer, and T Schwartz. Creating the CIPRES Science Gateway for inference of large phylogenetic trees. Gateway Computing Environments Workshop, 2010, 1–8, 2010.
10. raxml-ng.vital-it.ch. https://raxml-ng.vital-it.ch. Accessed: 2021-09-07.
11. Michael J Sanderson, Michelle M McMahon, and Mike Steel. Terraces in phylogenetic tree space. *Science*, 333(6041):448–450, 2011.
12. Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.
13. Eric Talevich, Brandon M Invergo, Peter JA Cock, and Brad A Chapman. Bio. phylo: a unified toolkit for processing, analyzing and visualizing phylogenetic trees in biopython. *BMC bioinformatics*, 13(1):1–9, 2012.
14. Reed A Cartwright. DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics*, 21(Suppl_3):iii31–iii38, 2005.
15. Lucas Czech, Pierre Barbera, and Alexandros Stamatakis. Genesis and Gappa: processing, analyzing and visualizing phylogenetic (placement) data. *Bioinformatics*, 36(10):3263–3265, 02 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa070.
16. J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.
17. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2.
18. Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Sinhrks, Simon Hawkins, Matthew Roeschke, Adam Klein, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, Marc Garcia, Jeremy Schendel, Andy Hayden, Daniel Saxton, Vytautas Jancauskas, Ali McMaster, Pietro Battiston, Skipper Seabold, chris b1, h vetinari, Kaiqi Dong, Stephan Hoyer, Wouter Overmeire, and Marco Gorelli. pandas-dev/pandas: Pandas 1.1.4, October 2020.
19. Andrew Rambaut and Nicholas C Grass. Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Bioinformatics*, 13(3):235–238, 1997.
20. Adrián A Davín, Théo Tricou, Eric Tannier, Damien M de Vienne, and Gergely J Szöllősi. Zombi: a phylogenetic simulator of trees, genomes and sequences that accounts for dead linages. *Bioinformatics*, 36(4):1286–1288, 2020.
21. Diego Mallo, Leonardo de Oliveira Martins, and David Posada. SimPhy: phylogenomic simulation of gene, locus, and species trees. *Systematic biology*, 65(2):334–344, 2016.
22. John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, Mass., 1977.