Methods in Ecology and Evolution

British Ecological Society

Methods in Ecology and Evolution 2013, **4,** 692–698 doi: 10.1111/2041-210X.12055

APPLICATION

phyloGenerator: an automated phylogeny generation tool for ecologists

William D. Pearse[1,2]* and Andy Purvis[1]

[1]*Department of Life Sciences, Imperial College London, Silwood Park Campus, Ascot, SL5 7PY, UK; and* [2]*NERC Centre for Ecology and Hydrology, Maclean Building, Benson Lane, Crowmarsh Gifford, Wallingford, Oxfordshire, OX10 8BB, UK*

Summary

1. Ecologists increasingly wish to use phylogenies, but are hampered by the technical challenge of phylogeny estimation.

2. We present phyloGenerator, an open-source, stand-alone Python program, that makes use of pre-existing sequence data and taxonomic information to largely automate the estimation of phylogenies.

3. phyloGenerator allows nonspecialists to quickly and easily produce robust, repeatable, and defensible phylogenies without requiring an extensive knowledge of phylogenetics. Experienced phylogeneticists may also find it useful as a tool to conduct exploratory analyses.

4. phyloGenerator performs a number of 'sanity checks' on users' output, but users should still check their outputs carefully; we give some advice on how to do so.

5. By linking a number of tools in a common framework, phyloGenerator is a step towards an open, reproducible phylogenetic workflow.

6. Bundled downloads for Windows and Mac OSX, along with the source code and an install script for Linux, can be found at http://willpearse.github.io/phyloGenerator (note the capital 'G').

Key-words: community phylogenetics, comparative analysis, phylogenetics, phylogeny construction, sequence alignment

Introduction

Ecologists have long recognised the importance of incorporating phylogenetic data in their work. Entire areas of study, such as community phylogenetics (Webb, Ackerly & Kembel 2002; Cavender-Bares *et al.* 2009; Vamosi *et al.* 2009) and comparative analysis (Felsenstein 1985; Harvey & Pagel 1991; Paradis 2012), require detailed phylogenetic, as well as ecological, information. Despite vast amounts of sequence data, progress in these fields has been slowed by the level of expertise required to create reliable phylogenies. Although there has been a recent explosion in the creation of extremely large phylogenies with many species (Smith, Beaulieu & Donoghue 2009; Izquierdo-Carrasco, Smith & Stamatakis 2011), there is often a mismatch between the species sequenced to build such trees and the species in which ecologists are interested. Moreover, while projects such as the '*Open Tree of Life*' (http://opentreeoflife.org/) aim to create a phylogeny of all life on Earth, as yet, no such tree exists for the nonspecialist to use.

Ecologists capable of conducting phylogenetic analyses are rewarded with estimates of phylogenetic uncertainty and the ability to work with novel sequence data. Ecologists without these skills must rely on programs such as Phylomatic (Webb & Donoghue 2005), which allows anyone to generate a phylogeny by adding missing species into a reference phylogeny on the basis of taxonomy and cannot generate a result that conflicts with the user's reference phylogeny or taxonomy. Phylomatic has been used almost exclusively for plant studies largely because the software has always been bundled with an excellent family-level phylogeny (Davies *et al.* 2004), although the latest online version (3 at the time of writing) includes the Bininda-Emonds *et al.* (2007) mammal supertree. Phylomatic is extremely robust and powerful, but when faced with taxa not in its reference phylogeny, its output may contain many polytomies, which can affect measures of phylogenetic diversity (Ricotta *et al.* 2012).

The rapid uptake of Phylomatic suggests there is a need for a method that combines Phylomatic's ease of use with the flexibility and accuracy of *de novo* tree construction. phyloGenerator takes a list of species, candidate genes and (optionally) taxonomic information and from them creates a novel phylogeny using established phylogenetic methods. In contrast with other automated methods, phyloGenerator is intended to allow the nonspecialist to produce a defensible phylogeny with minimal effort.

A nontechnical overview of phyloGenerator

It is beyond our scope to review the entirety of phylogenetics, and in the brief overview below, we assume basic familiarity with the concepts of DNA sequences, phylogenies (or 'trees')

*Correspondence author. E-mail: will.pearse@gmail.com

and Bayesian inference, all of which are covered in depth by Felsenstein (2004) and Roquet, Thuiller & Lavergne (2013). phyloGenerator attempts to find the phylogeny that is most likely given a particular DNA *alignment*. An alignment is intended to represent the same locus in the genome of all species under study, highlighting the differences and similarities in the DNA sequence that provide the basis for inference of the species' phylogeny. We strongly encourage any user to manually inspect their alignment and output phylogeny despite the checks phyloGenerator performs, as many common problems are apparent even to a novice phylogeneticist. The identification and resolution of some common issues is described in Table 1.

First, phyloGenerator downloads DNA sequences from GenBank (Benson *et al*. 2009) for each species from each genetic locus and then *aligns* the sequences to determine how each species' sequence relates to the others (Fig. 1). The choice of locus is important: if a locus' mutation rate is too slow, there will be insufficient variation for analysis, but if it is too fast, then multiple mutations at the same position could confound analysis. Loci with slower mutation rates may be easier to align, but using particularly slow (or fast) loci can make it harder to find the 'true' phylogeny (Yang 1998). A search program constructs a phylogeny from an alignment by calculating the likelihood of a candidate phylogeny given that alignment and then rearranging that phylogeny in an attempt to improve its likelihood score.

In practice it is infeasible to evaluate all possible phylogenies (there are over two million possible phylogenies containing only 10 species), and so there is no guarantee of finding the best estimate of a phylogeny. Maximum likelihood (ML) search programs (phyloGenerator uses RAxML; Stamatakis 2006) can be run multiple times from different starting trees to increase the chances of finding a good tree, and recording how many times a particular clade is found during these searches can provide an estimate of the credibility of that clade (a *bootstrap* support value). Bayesian approaches (phyloGenerator uses BEAST; Drummond *et al*. 2006; Drummond & Rambaut 2007; Drummond *et al*. 2012) can also make use of multiple starting trees and attempt to estimate a posterior distribution of candidate phylogenies. This posterior distribution can be summarised to produce a single phylogeny with estimates of support for each clade, or analyses can be run on all trees in the posterior distribution (see Bollback 2005, for a review of such *posterior predictive* methods). Most Bayesian methods use Markov chain Monte Carlo methods to estimate this posterior distribution and so require that the Markov chain has *converged* on a distribution of likely phylogenies. There are many ways of assessing convergence, and the user should use BEAST only if they are comfortable judging the convergence of its output (see Lemey, Salemi & Vandamme 2009, for more details).

All of these search strategies can be *constrained*, restricting the phylogeny search to trees that do not conflict with a given constraint phylogeny. Users are encouraged to restrict their search to conform to well-known clades (e.g., taxonomic families that have been shown to be monophyletic) and then estimate the unknown relationships within these clades.

The ML estimates of phylogeny produced by RAxML have branch lengths proportional to the rate of evolution at the loci used, rather than to time. Molecular dating techniques can be used in phyloGenerator to transform these branch lengths to

Table 1. Common problems encountered during a phyloGenerator run and their solution. In the majority of cases, problems with phyloGenerator runs result from ignoring the 'warn' column in the DNA alignment stage

Problem	Indicated by	Solution
Inappropriate or poor-quality DNA sequences	Large range of sequence lengths in DNA download stage (extremes marked with '^^^' and '___')	*Reload* sequences (changing target length); *trim* sequences if from a coding region
No DNA data for target species	Sequences of length '0' in DNA download stage	Use *replace* method to find replacements (i.e., surrogates); manually merge clades if no replacement can be found; search again with more loci
Poor alignment	Warning column in DNA alignment stage; inspection of alignment shows regions with large stretches of gaps	As for poor-quality sequences (above); remove outlier regions with trimAl
Topology conflicts with strong *a priori* expectations	Visual inspection of phylogeny	As for poor-quality alignment (above); consider a constraint tree and examine impact of constraint tree on result within program; repeat analysis with more restarts (RAxML) and check for convergence (BEAST)
Extreme variation in root-to-tip distances	Visual inspection of phylogeny	As for poor-quality of DNA sequences (above)—check carefully species subtending from the long branches; re-check any dated clades in constraint tree
Near zero-length branches, or extremely long branches, after molecular dating	Visual inspection of phylogeny	Examine undated phylogeny for extreme variation in branch length, following advice above
Long branch attraction	Visual inspection of phylogeny; species known to be distantly related to rest of phylogeny appear closely-related	Include more species; as for unexpected topology (above)

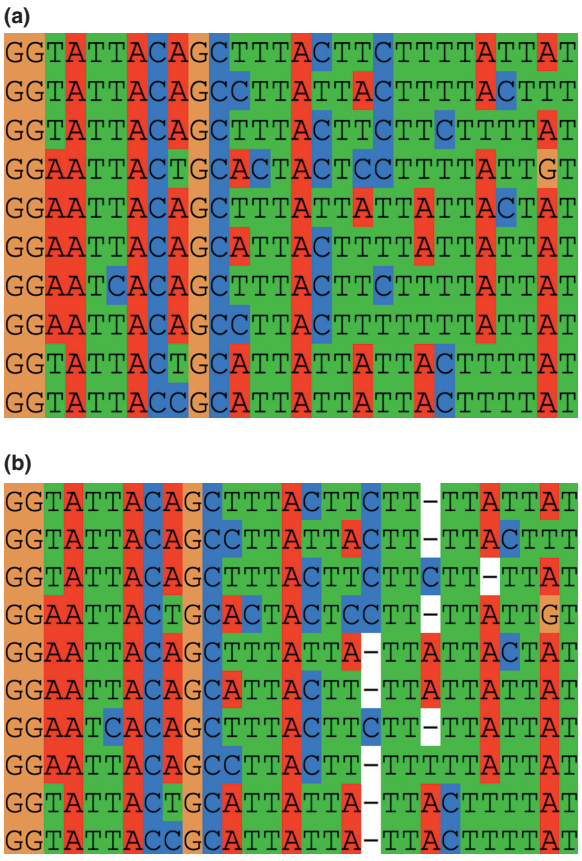

Fig. 1. Description of a DNA alignment. In (a), 16 unaligned DNA sequences are displayed (one per row), while in (b) the same sequences are displayed, but they have now been aligned. Gaps ('—') have been inserted into some sequences and represent where additional base pairs have been inserted (or removed) from the DNA sequences. In the aligned sequence, because each species' nucleotides can be compared with the nucleotides at the same position in every other species, models describing mutation from one nucleotide to another can be fit across the data. These homologous nucleotide positions provide information for phylogenetic analysis.

Fig. 2. Phylogenies of a plant data set made using Phylomatic (black) and phyloGenerator (red). Nodes whose age was constrained in phyloGenerator are marked with circles. phyloGenerator has produced a phylogeny very similar to that of Phylomatic, but without any polytomies. This example is included in the phyloGenerator distribution, and its construction could be described as: "created using phyloGenerator with options '-gene *rbcL*, *matK* -alignment mafft -consTree appendix A-phylogen beast-GTR-GAMMA-chainLength= 4000000'", where the constraint tree and the DNA alignments used were included as appendices. The programs phyloGenerator uses, and phyloGenerator itself, would have to be cited in a publication.

be proportional to divergence time, either by essentially averaging out variation in branch lengths (using PATHd8; see Britton *et al*. 2007, for more details), or a BEAST run where the phylogeny's topology is constrained to that of the most likely phylogeny found by RAxML and so only branch lengths are estimated.

A more technical description of phyloGenerator

phyloGenerator is a command-line application that uses and extends the BioPython framework (Cock *et al*. 2009; Talevich *et al*. 2012). It combines many phylogenetic tools in one distribution, under a single interface; no customisation or set-up, beyond downloading the program, is required for use on Windows or Mac OSX. Users are guided through the process of making a phylogeny by a series of questions, while the advanced user can preselect options from the command line and thus succinctly describe an analysis. Thus, the tool may be used within an automated workflow, providing a step towards an open framework of repeatable phylogenetic methods. The online documentation gives examples of how to succinctly describe an analysis in terms of phyloGenerator commands, and an example is given in the text of Fig. 2.

The program's procedures can easily be customised, and the source code itself has been written to facilitate user modification; phyloGenerator can either be run as a single Python script or imported as a Python module by other scripts. Phylogeneticists can use its features, such as the *replace* method and BEAST analysis templates, within their own pipelines. We wrote the program in Python to allow for this easy integration of phyloGenerator internal functions into advanced users' scripts, while also permitting phyloGenerator to function as stand-alone software without requiring the user to manually configure the programs it uses. Our hope is that user preferences and future methodological advances can be incorporated into its workflow, such that novice phylogeneticists can benefit from the skills of others. We encourage users to submit feature requests online (https://github.com/willpearse/phyloGenerator/issues), which we will endeavour to incorporate into the program. Bundled downloads for Windows and Mac OSX, along with the source code and an installer script for Linux systems, can be found at http://willpearse.github.io/phyloGenerator (note the capital 'G'). An outline of the program's workflow is shown in Fig. 3.

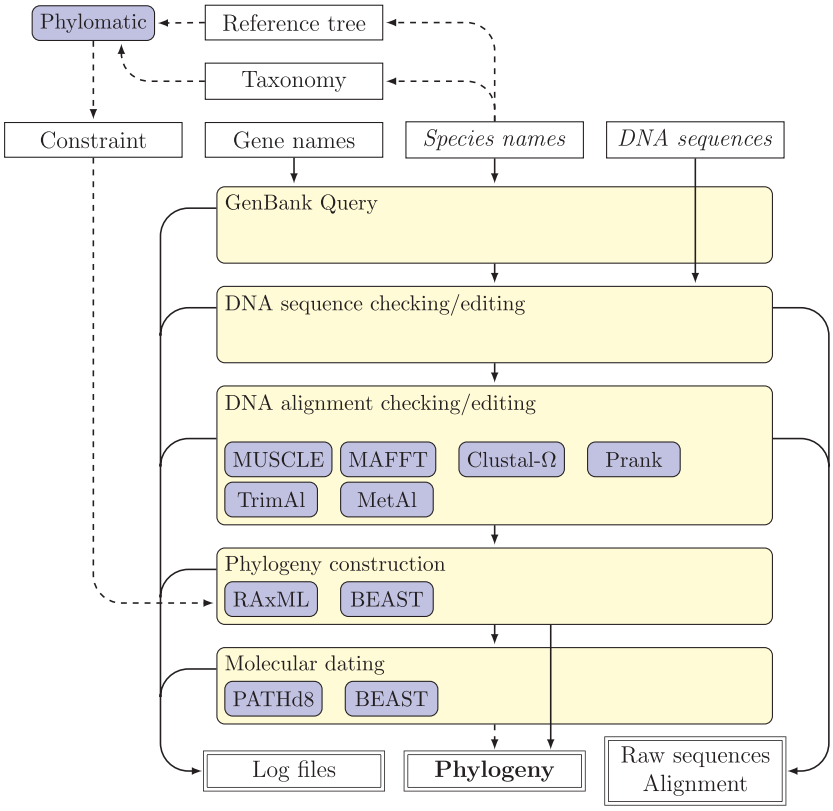

Fig. 3. Outline of phyloGenerator workflow. Stages of the program are coloured yellow, programs used are blue, inputs are white and outputs are white with two lines around them. Optional steps have dashed lines. The user must provide either DNA sequences or species names, but not both (in italics). A constraint tree can either be provided by the user or generated from a reference tree and taxonomic information using Phylomatic.

DNA SEQUENCE DOWNLOAD AND CLEANING

The user provides a list of species and candidate genes, which phyloGenerator downloads from GenBank, choosing between multiple sequences either at random, according to the median, maximum or minimum length of sequences on GenBank, or with reference to a target gene length. phyloGenerator can search for open reading frames in any sequence and extract a gene of interest from annotated sequences. Not all the genes searched for need to be used in the final phylogeny; if the user only wishes to use a certain number of genes, phyloGenerator can select the set of genes that maximises species coverage. If no match is found for a particular species' gene, a relative's gene can be used instead, but only if the NCBI taxonomy (http://www.ncbi.nlm.nih.gov/taxonomy) indicates the species and its replacement would form a monophyletic clade within the phylogeny (the *replace* method). If no such replacement can be found for species, the user can *merge* the missing species with another species that has sequence data; in the final output, the species will form a polytomy dated following the bladj algorithm (Webb *et al.* 2008).

Not all GenBank sequences are labelled in the same way: searches for '*Internal Transcribed Spacer*', '*ITS*', '*ITS1*' and '*ITS2*' will not necessarily yield the same results. phyloGenerator attempts to search both sequence annotations and sequence descriptions for specified genes and allows the user to supply aliases for gene names. Thus, advanced users can use phyloGenerator as an automated, rapid-checking system for exploratory analyses. The user can also select 'preset' sets of candidate genes that are likely to perform well with their taxa (e.g. *COI* for animals).

DNA SEQUENCE ALIGNMENT

DNA data can be aligned using Clustal-Ω (Sievers *et al.* 2011), MAFFT (Katoh *et al.* 2005; Katoh & Toh 2008), MUSCLE (Edgar 2004) and Prank (Löytynoja & Goldman 2005). There is no general consensus on how to identify the most accurate alignment, so several options are offered to help the user choose among candidate alignments generated by different programs within phyloGenerator. Alignments are compared according to their number of gaps, and 'difficult' regions can be removed with trimAl (Capella-Gutiérrez, Silla-Martínez & Gabaldón 2009). Alignments can be directly compared with each other (using the *SSP* metric of MetAl; Blackburne & Whelan 2012) or by their impact on tree searches (the mean Robinson-Foulds distances between RAxML searches with each alignment). Users can reload sequences and align them as many times as they wish and are advised to visually inspect any alignment before proceeding to build a phylogeny.

PHYLOGENY CONSTRUCTION AND MOLECULAR DATING

Using RAxML, a tree can be found and bootstrapped nodal support values calculated for that tree. If desired, molecular dating can be performed using PATHd8 or with a BEAST search where the topology has been constrained to that of the best tree found by RAxML.

BEAST can also be used for the entire search process, in which case, the resulting phylogeny already has branch lengths proportional to time and no molecular dating is required. Nodal support values from the posterior distribution of trees are annotated onto the output phylogeny for the user. We cannot guarantee the convergence of a BEAST run, and so the user is responsible for checking the output of BEAST analyses. AWTY (Nylander *et al.* 2008) and TRACER (Rambaut & Drummond 2012) are excellent tools for checking for convergence, and phyloGenerator outputs BEAST's log file and posterior distribution of trees for use with them.

Table 2. Variation in phyloGenerator execution time. The time taken to download sequences is included in brackets after the total execution time. The 257 British bird species were initially searched for, but sequences for only 233 were used in the final phylogeny, and the runs with 100 species contained a random subset of sequences from the 233 species run. All runs were unconstrained and used the default settings, apart from the RAxML runs which used the integrated Bootstrap = 100 option. No constraint trees were used, and no checking of the sequences or output was performed. All analyses were conducted on a 2·66 GHz Intel Core 2 Duo MacBook Pro laptop with 4 Gb of RAM purchased in 2009. These data sets are included with phyloGenerator

Dataset	No. species	Genes	Method	Execution time (min)
Silwood plants	33	*rbcL*	RAxML	4 (2)
Silwood plants	33	*rbcL*	BEAST	8 (2)
Silwood plants	33	*rbcL*; *matK*	RAxML	12·5 (4)
Silwood plants	33	*rbcL*; *matK*	BEAST	10 (4)
British birds	100	*COI*	RAxML	12
British birds	100	*COI*	BEAST	12
British birds	233	*COI*	RAxML	59 (9)
British birds	233	*COI*	BEAST	84 (9)

Some may be concerned at the idea of a nonspecialist building a phylogeny from sequence data. To mitigate such concerns, the user is encouraged to constrain their tree search using existing strongly supported clades, and Phylomatic can be used to do so. The data's agreement with a constraint can be assessed by comparing tree searches with and without the constraint tree (using the mean Robinson-Foulds distances between RAxML tree searches). If the user provides a constraint tree with named clades, those clades' ages are set as strong priors (a normal distribution with the given age, in Ma, as the mean, and a standard deviation of one) during a BEAST search. By constraining their phylogeny according to strongly supported relationships and dated clades (using Phylomatic if desired), the user can be certain that their phylogeny cannot conflict with established phylogenetic relationships. phyloGenerator attempts to autodetect sequence alignment problems, but the user is strongly advised to inspect their output by eye for misplaced species and unusual branch lengths and to take heed of estimates of clade credibility (Table 1).

Example and comparison with existing methods

Figure 2 shows a phylogeny generated using phylomatic (in black) of plant species in an experiment at Silwood Park (Berkshire, UK). Of the 33 species in the phylogeny, 13 descend from polytomies, suggesting a lack of phylogenetic information for these species. We used this phylogeny as a constraint for phyloGenerator and generated a completely resolved phylogeny (in red on Fig. 2) using the *rbcL* and *matK* genes. By default, phyloGenerator sets strong priors on the ages of all named clades (marked on Fig. 2), dating other clades using DNA data.

Table 2 shows how long phyloGenerator takes to produce phylogenies for two of its example data sets. In general, small phylogenies can be produced fairly rapidly (e.g. 10 min for a two-gene plant phylogeny of 33 species), and while very large phylogenies (e.g. 233 species in Table 2) can be produced with phyloGenerator it is unlikely that BEAST runs of such large phylogenies will reach convergence under default settings. While little user input was required during the execution of

Table 3. Programs with features similar to phyloGenerator. In order from left to right, each column describes whether a program: downloads DNA data from the Internet, aligns DNA data, heuristically searches for an acceptable phylogeny, doesn't require the user to manually customise or run its subcomponents, conducts analyses on the user's computer and attempts to check the user's data or output for obvious sources of error. In each column, ✓ and × indicate whether a program does or does not have a feature, respectively; Phylomatic does not attempt to build a novel phylogeny and so is listed as NA under some columns

Program	DNA download	DNA alignment	Tree search	One-click	Local	Sanity checks
Phylomatic (Webb *et al.* 2008)	NA	NA	NA	✓	✓	×
Peters *et al.* (2011)	✓	✓	✓	×	✓	×
rPlant (Banbury *et al.* 2012)	×	✓	✓	×	×	×
GeneFinder (Lanfear & Bromham 2011)	✓	×	×	×	✓	×
SATé-II (Liu *et al.* 2012)	×	✓	✓	✓	✓	×
phyloGenerator	✓	✓	✓	✓	✓	✓

these runs, we have not attempted to estimate the time required to check the output of phyloGenerator for obvious problems.

A number of other phylogenetic pipelines exist, and Table 3 compares some with phyloGenerator. However, there are few methods available for inexperienced phylogeneticists. For example, Peters *et al.*'s (2011) method requires the user to sequentially run and configure BASH, Perl and Ruby scripts, while rPlant (Banbury *et al.* 2012) is an interface to the online iPlant facilities and requires the user to program their own workflow.

Conclusion

phyloGenerator offers a way for nonspecialists to make phylogenies from existing sequence data, constraining their output according to existing strongly supported systematic information and providing estimates of clades' uncertainty. Users are strongly advised to inspect the quality of both their alignments and their output phylogenies for obvious errors before continuing with any analyses and should be aware that their choice of gene region may affect their output. Experienced phylogeneticists can use phyloGenerator to collect sequence data and conduct exploratory analyses and incorporate phyloGenerator's internal functions into their own pipelines. phyloGenerator is not designed to replace phylogeneticists, but it is intended to facilitate the rapid and broad dissemination of their expertise to those who badly need phylogenies in their work. We hope it is a step towards an open, reproducible way of describing, sharing and implementing phylogenetic methods.

Acknowledgements

We thank A. Humphreys, E. Paradis, A. Papadopulos, D. Quicke, and two anonymous reviewers for helpful comments. D. Orme and D. Roy gave useful feedback on drafts of this manuscript, and we are especially grateful to D. Orme who made the bulk of Fig. 3. WDP was supported by a NERC CASE PhD scholarship. We are grateful to K. Luckett who provided some data for the examples, and I. Fenton, M. Harrison, L. Kirkpatrick, J. Lim, and M. Novosolov who sympathetically (and thoroughly) tested the program.

Conflicts of interest

The authors declare no conflicts of interest.

References

Banbury, B., Michels, K., Beaulieu, J.M. & O'Meara, B. (2012) rPlant: R interface to the iPlant discovery environment, R package version 1.2. Available at: http://CRAN.R-project.org/package=rPlant [accessed 19 July 2012].

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. & Sayers, E.W. (2009) Genbank. *Nucleic Acids Research*, **37**, D26–D31.

Bininda-Emonds, O.R.P., Cardillo, M., Jones, K.E., Macphee, R.D.E., Beck, R.M.D., Grenyer, R., Price, S.A., Vos, R.A., Gittleman, J.L. & Purvis, A. (2007) The delayed rise of present-day mammals. *Nature*, **446**, 507–512.

Blackburne, B.P. & Whelan, S. (2012) Measuring the distance between multiple sequence alignments. *Bioinformatics*, **28**, 495–502.

Bollback, J. (2005) Posterior mapping and posterior predictive distributions. *Statistical Methods in Molecular Evolution*, Chap. 16 (ed. R. Neilsen), pp. 439–462, Springer, New York, NY.

Britton, T., Anderson, C.L., Jacquet, D., Lundqvist, S. & Bremer, K. (2007) Estimating divergence times in large phylogenetic trees. *Systematic Biology*, **56**, 741–752.

Capella-Gutiérrez, S., Silla-Martínez, J. & Gabaldón, T. (2009) trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics*, **25**, 1972–1973.

Cavender-Bares, J., Kozak, K., Fine, P.V.A. & Kembel, S.W. (2009) The merging of community ecology and phylogenetic biology. *Ecology Letters*, **12**, 693–715.

Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. & de Hoon, M.J.L. (2009) Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Davies, T.J., Barraclough, T.G., Chase, M.W., Soltis, P.S., Soltis, D.E. & Savolainen, V. (2004) Darwin's abominable mystery: insights from a supertree of the angiosperms. *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 1904–1909.

Drummond, A. & Rambaut, A. (2007) BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, **7**, 214.

Drummond, A.J., Ho, S.Y.W., Phillips, M.J. & Rambaut, A. (2006) Relaxed phylogenetics and dating with confidence. *PLoS Biology*, **4**, e88.

Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*, **29**, 1969–1973.

Edgar, R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, **32**, 1792–1797.

Felsenstein, J. (1985) Phylogenies and the comparative method. *The American Naturalist*, **125**, 1–15.

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

Harvey, P.H. & Pagel, M. (1991) *The Comparative Method in Evolutionary Biology*. Oxford University Press, Oxford.

Izquierdo-Carrasco, F., Smith, S. & Stamatakis, A. (2011) Algorithms, data structures, and numerics for likelihood-based phylogenetic inference of huge trees. *BMC Bioinformatics*, **12**, 470.

Katoh, K. & Toh, H. (2008) Recent developments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, **9**, 286–298.

Katoh, K., Kuma, K.I., Toh, H. & Miyata, T. (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Research*, **33**, 511–518.

Lanfear, R. & Bromham, L. (2011) Estimating phylogenies for species assemblages: a complete phylogeny for the past and present native birds of New Zealand. *Molecular Phylogenetics and Evolution*, **61**, 958–963.

Lemey, P., Salemi, M. & Vandamme, A.M. (eds). (2009) *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*, 2nd edn. Cambridge University Press, Cambridge.

Liu, K., Warnow, T.J., Holder, M.T., Nelesen, S.M., Yu, J., Stamatakis, A.P. & Linder, C.R. (2012) SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology*, **61**, 90–106.

Löytynoja, A. & Goldman, N. (2005) An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 10557–10562.

Nylander, J.A.A., Wilgenbusch, J.C., Warren, D.L. & Swofford, D.L. (2008) AWTY (are we there yet?): a system for graphical exploration of MCMC convergence in bayesian phylogenetics. *Bioinformatics*, **24**, 581–583.

Paradis, E. (2012) *Analysis of Phylogenetics and Evolution with R*, 2nd edn. Springer, New York.

Peters, R., Meyer, B., Krogmann, L., Borner, J., Meusemann, K., Schutte, K., Niehuis, O. & Misof, B. (2011) The taming of an impossible child: a standardized all-in approach to the phylogeny of hymenoptera using public database sequences. *BMC Biology*, **9**, 55.

Rambaut, A. & Drummond, A. (2009) *Tracer v1.5*. Available at: http://beast.bio.ed.ac.uk/Tracer [Accessed 11 April 2013].

Ricotta, C., Bacaro, G., Marignani, M., Godefroid, S. & Mazzoleni, S. (2012) Computing diversity from dated phylogenies and taxonomic hierarchies: does it make a difference to the conclusions? *Oecologia*, **170**, 501–506.

Roquet, C., Thuiller, W. & Lavergne, S. (2013) Building megaphylogenies for macroecology: taking up the challenge. *Ecography*, **36**, 013–026.

Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soding, J., Thompson, J.D. & Higgins, D.G. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, **7**, 529.

Smith, S.A., Beaulieu, J.M. & Donoghue, M.J. (2009) Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. *BMC Evolutionary Biology*, **9**, 37.

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

Talevich, E., Invergo, B., Cock, P. & Chapman, B. (2012) Bio.phylo: A unified toolkit for processing, analyzing and visualizing phylogenetic trees in biopython. *BMC Bioinformatics*, **13**, 209.

Vamosi, S., Heard, S.B., Vamosi, C. & Webb, C.O. (2009) Emerging patterns in the comparative analysis of phylogenetic community structure. *Molecular Ecology*, **18**, 572–592.

Webb, C.O. & Donoghue, M.J. (2005) Phylomatic: tree assembly for applied phylogenetics. *Molecular Ecology Notes*, **5**, 181–183.

Webb, C.O., Ackerly, D.D., McPeek, M.A. & Donoghue, M.J. (2002) Phylogenies and community ecology. *Annual Review of Ecology and Systematics*, **33**, 475–505.

Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008) Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, **24**, 2098–2100.

Yang, Z. (1998) On the best evolutionary rate for phylogenetic analysis. *Systematic Biology*, **47**, 125–133.

Received 5 February 2013; accepted 27 February 2013
Handling Editor: Emmanuel Paradis