Chapter 2

Species Tree Inference with SNP Data

Michael Matschiner

Abstract

While the inference of species trees from molecular sequences has become a common type of analysis in studies of species diversification, few programs so far allow for the use of single-nucleotide polymorphisms (SNPs) for the same purpose. In this book chapter, I discuss the use of the Bayesian program SNAPP, which infers the species tree by mathematically integrating over all possible genealogies at each SNP. In particular, I focus on a molecular clock model developed for SNAPP, allowing the inference of divergence times together with the species tree topology and the population size, directly from SNP datasets in variant call format. With the growing availability of SNP datasets for multiple closely related species, this approach is becoming increasingly relevant for the reconstruction of the temporal framework of recent species diversification.

Key words Genomics, Phylogeny, Species tree, SNPs, Divergence times, SNAPP, BEAST

1 Introduction

Genetic data have long been used to infer relationships among individuals, populations, and species. Traditionally, DNA fragments were sequenced for certain markers and aligned to each other, and the resulting multiple sequence alignment was used to deduce relationships based on pairwise distances, parsimony, or the likelihood of the alignment under certain models of sequence evolution. When the same set of taxa have been sequenced for multiple markers, a common approach has been to join—concatenate—the multiple sequence alignments for these markers into a single alignment before the inference. However, investigations motivated by the growing number of multimarker datasets have identified important issues with this approach. Based on simulations, Kubatko and Degnan [1] demonstrated that under certain conditions, concatenation of alignments can lead to the inference of incorrect relationships among species that even receive greater support with increasing size of the dataset. Their conclusion has since been corroborated by several studies [2–4], including a mathematical

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_2,

proof of the statistical inconsistency of concatenation [5]. Moreover, while the inconsistency affects the estimated topology of the species tree only under certain conditions, the estimates for the lengths of the tree's branches are almost certainly affected by concatenation [6, 7]. This is particularly problematic when the species tree is time calibrated and branch lengths are used as a measure of the amount of time that passed between speciation events.

The underlying cause for the inconsistency resulting from concatenation is the fact that the true genealogies of markers may differ from each other and also from the true species tree, due to recombination. To account for this variation among marker genealogies in the estimation of the species tree, the multispecies coalescent (MSC) model has been developed [8] and implemented in a growing number of inference tools [9–14]. These tools fall into two categories where some estimate the marker genealogies jointly with the species tree and others rely on separately estimated marker genealogies as input. Under the assumptions of the MSC model, which include random mating within species, the absence of gene flow after speciation, and the absence of recombination within markers, inference of species trees with these tools is statistically consistent and therefore reliable [14]. Naturally, all of the assumptions of the MSC model may be violated by empirical systems, but as concatenation has been argued to represent nothing else than a particularly unrealistic special case of the MSC model [15], the use of this model may nevertheless improve the accuracy of species tree estimates. Of these assumptions of the MSC model, particularly the last one—the absence of within-marker recombination—has been criticized, and Springer and Gatesy [16] argued that, depending on population sizes, time between speciation events, and recombination rates, within-marker recombination can change the true genealogy as often as every few base pairs. In such cases, inference with the MSC model may be expected to suffer from the same problems as concatenation [16].

One alternative application of the MSC model that is immune to within-marker recombination is the estimation of species trees directly from single-nucleotide polymorphisms (SNPs) instead of marker sequences. With a length of a single base pair, recombination within a SNP is of course impossible. And while individual SNPs do not carry enough information for the estimation of the genealogy at the position of the SNP, this problem can be circumvented in two ways: In the quartet inference approach implemented in the program SVDquartets [17], the taxon set is decomposed into a large number of quartets (combinations of four species), the support for alternative quartet topologies is assessed, and quartet topologies are finally reassembled into the estimated species tree topology. The second possibility to avoid the uninformative genealogies of individual SNPs is implemented in SNAPP [18], which

integrates over all possible genealogies at each SNP mathematically rather than inferring them. This approach is conceptually elegant, but unfortunately comes at the cost of high computational demand, meaning that SNAPP can usually be applied only to comparatively small datasets of tens of species and thousands of SNPs. In contrast, SVDquartets runs quickly enough to be applied to hundreds of species and millions of SNPs. Besides these two methods based on the MSC, tools that model mutation and allelic drift instead of coalescent variation can also be applied to infer species trees from SNP data; these tools include POMO [19] and the recently developed Snapper [20].

Despite its limitation to smaller datasets, SNAPP is highly useful for the inference of species trees from recently diverged groups. As a Bayesian inference tool, SNAPP produces probabilistic node support values that can be interpreted intuitively, and it allows model comparisons by Bayes factors, which enables its use for species delimitation [21]. And as simulations have shown, the inferred species tree can be accurate and precise even when only hundreds of SNPs are used [7]. Finally, with the molecular clock model developed for SNAPP by Stange et al. [7], the program also estimates population sizes and divergence times, allowing the reconstruction of the temporal framework of species diversification.

In the rest of this chapter, I am going to focus on species tree estimation with SNAPP based on the model of Stange et al. [7], assuming that the reader is not only interested in the topology of the species tree but also in the timeline of diversification. I do not cover species tree inference with SVDquartets or species delimitation with SNAPP but would like to point the readers interested in these analysis types to the excellent tutorials that can be found online at www.phylosolutions.com (by Dave Swofford and Laura Kubatko) and www.evomics.org (by Adam Leaché), respectively.

2 Materials

SNAPP is available as an add-on package for BEAST 2 [22, 23]; therefore, both the BEAST 2 suite of programs and this add-on package are required for species tree inference with SNAPP. To use the model of Stange et al. [7] in SNAPP, the ***snapp_prep.rb*** script, written in the Ruby programming language, is additionally required. To apply this script, several input files, including a genotype data matrix, a file assigning individuals to species, a file with age constraints, and possibly a starting tree are needed. Finally, two more programs, Tracer [24] and FigTree are useful for post-processing the output of SNAPP.

2.1 BEAST 2

The BEAST 2 suite of programs includes BEAST itself, BEAUti, LogCombiner, and TreeAnnotator. BEAST employs Markov-chain Monte Carlo (MCMC) to infer the Bayesian posterior distribution of phylogenetic trees and parameter estimates, under models that are specified in input files in XML format [25]. To facilitate the writing of XML files, BEAST is distributed together with BEAUti, a graphical user interface program through which the model settings can be selected and exported in XML format (however, the SNAPP model of Stange et al. cannot be specified through BEAUti; see below). The user interface of BEAUti also represents an easy way to access the BEAST 2 Package Manager, through which add-on packages like SNAPP can be installed. The program LogCombiner can be used to merge posterior distributions from multiple replicate BEAST analyses into a single file, and TreeAnnotator serves to generate summary trees from the posterior tree distribution. The BEAST 2 suite of programs for Mac OS X, Linux, or Windows can be obtained freely from https://www.beast2.org. All BEAST 2 programs are written in the Java programming language and thus Java is required to run them. Therefore, either the Java Development Kit (version 8 or higher) should be installed (e.g., from https://adoptopenjdk.net) or one of the BEAST 2 versions bundled with Java should be selected from the BEAST 2 website (https://www.beast2.org). The version of Java installed can be identified on the command line with java -version; version number 1.8 or higher corresponds to Java Developer Kit version 8 or higher.

2.2 SNAPP

Owing to SNAPP's integration into BEAST 2, its model settings can be defined with BEAUti, its analyses use the MCMC machinery of BEAST, and postprocessing can be performed with the LogCombiner and TreeAnnotator tools distributed with BEAST 2. The model applied in SNAPP, however, is rather different from those of other BEAST analyses, due to its use of SNP markers instead of sequence alignments and the mathematical integration over all possible genealogies at each SNP. The SNAPP add-on package can be installed with the BEAST 2 Package Manager, accessed through BEAUti as shown in Fig. 1.

2.3 *snapp_prep.rb* and *add_theta_to_log.rb*

While the settings for most SNAPP analyses can be defined with BEAUti's graphical user interface, this is not the case for analyses with the molecular clock model of Stange et al. [7]. To implement this model, the XML file for SNAPP needs to be written differently, and one convenient way in which this can be done is the ***snapp_prep.rb*** Ruby script. The script can be obtained from GitHub at https://github.com/mmatschiner/snapp_prep. A second Ruby script, named ***add_theta_to_log.rb***, is useful for postprocessing of SNAPP results and available from the same repository. To run both scripts, the Ruby programming language (version 2 or higher) is

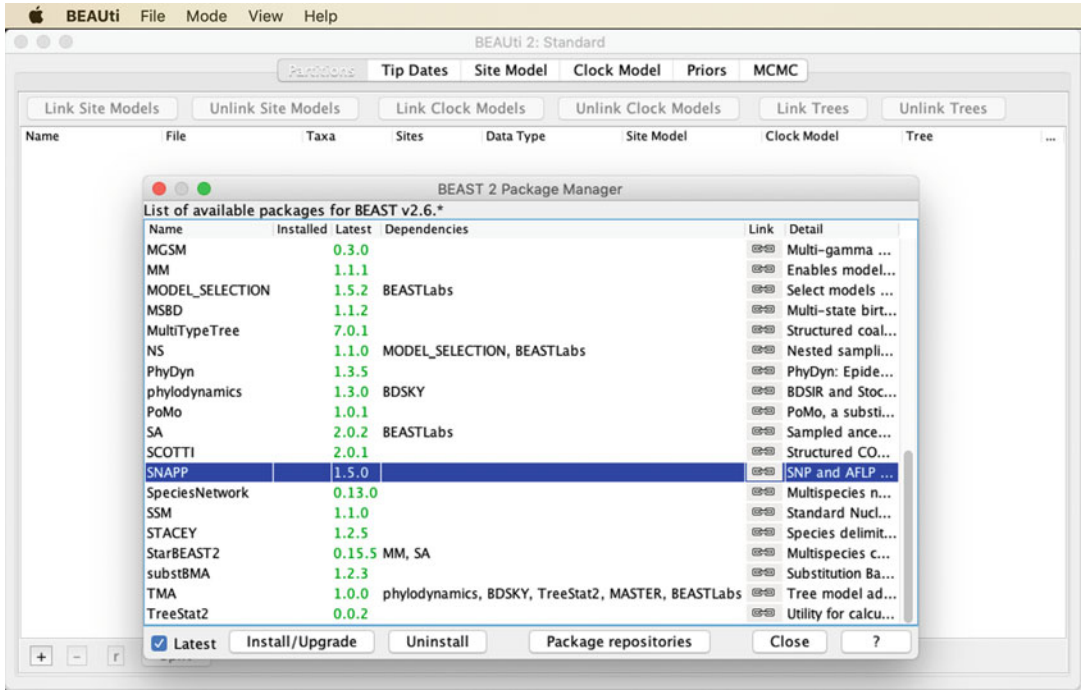

Fig. 1 Screenshot of the BEAUti graphical user interface with the BEAST 2 Package Manager. The Package Manager can be opened by clicking "Manage Packages" in BEAUti's "File" menu. SNAPP can then be installed by selecting the package as in the screenshot and clicking "Install/Upgrade" at the bottom of the Package Manager window

required. The language is included with Mac OS X and Linux operating systems but may first need to be installed on Windows systems. All installation options are described on https://www.ruby-lang.org/en/documentation/installation/. The installed version of Ruby can be identified on the command line with ruby --version.

2.4 Genotype Data Matrix

One of the inputs required by **snapp_prep.rb** to write an XML file for SNAPP is a matrix containing diploid genotype data. This matrix can be provided in Phylip format [26] or in uncompressed variant call format (VCF), the latter of which is probably more convenient for most users as genotyping data is most commonly stored in VCF files. As SNAPP can only handle biallelic SNPs, all indels, multiallelic SNPs, and monomorphic sites should first be removed from the matrix. To further comply with SNAPP's expectation of SNPs that are unlinked [18], it may be advisable to thin the matrix so that no two SNPs are within a short distance of each other on the same chromosome (this can also be done with ***snapp_prep.rb***; see below). Which minimum distance should be chosen may depend on the genome size and the target number of SNPs for the analysis, but minimum distances of at least thousands of base

pairs (bp) may be sensible (that said, the effect of linkage between some SNPs is likely negligible when the matrix includes thousands of SNPs). SNPs with missing data can be used by SNAPP as long as genotypes are known for at least one individual per species; SNPs for which this is not the case will be recognized by the ***snapp_prep.rb*** script and excluded from the produced XML file. Importantly, the genotype matrix should not be filtered by minor allele count or frequency as this filter would introduce bias to the estimated lengths of terminal branches of the species tree. Instead of filters on minor allele count or frequency, filtering for high genotype quality is recommended. The data matrix should not include genotypes for large numbers of individuals per species as these would primarily extend SNAPP's run times without adding much information to the analysis. As a rule of thumb, a total number of 20–30 (diploid) individuals across all species and between 1000 and 10,000 SNPs constitute a suitable dataset size, but if SNAPP's results should turn out to be too uninformative or if the run times are too long, these numbers should be adjusted. Note that SNAPP can provide good species tree estimates even when only a single (diploid) individual is used per species [7].

2.5 Species Table

SNAPP requires information assigning individuals to species, and to write this information into SNAPP's XML file, the ***snapp_prep.rb*** script expects an input file with a two-column table. The file should be in plain text format, the first column should list species IDs, and the second column should list the corresponding IDs of individuals. These individual IDs should exactly match those used in the genotype data matrix. The two columns can be either tab- or space-delimited. The table may include a header row; if it does, the row content should be "Species" in the first column and "Specimen," "Specimens," "Sample," or "Samples" in the second column; these keywords are case-insensitive. An example of a species table, taken from a study by Barth et al. [27], is shown in Table 1.

2.6 Age Constraints

In sequence-based analyses of divergence times, phylogenies are usually time calibrated either by specifying an estimate of the mutation rate or by placing age constraints on one or more divergence events in the tree. In SNP-based species tree inference with SNAPP, however, mutation rates applying to the dataset can usually not be estimated a priori because the SNP data are subject to ascertainment bias as only variable sites are included [7]. Thus, the better approach for time calibration of SNP-based species trees is to specify age constraints for divergences within the tree. The information for these constraints may come from the fossil record or from previous phylogenetic studies, but either way, some age information must be available for at least one divergence, otherwise the molecular clock model of Stange et al. [7] cannot be used. If the user should not be aware of published age estimates for the group

Table 1
Example of a species table assigning individuals to species, taken from Barth et al. [25]

Species	Specimen
mar	BOU15023
mar	BOU15010
mar	SAW16055
mar	BOU15014
mar	SAW16054
meg	VAG12056
meg	SAW17B10
meg	VAG12041
meg	BOU15027
meg	BOU15030
obs	VAG12061
obs	SAW16038
obs	SAW16042
obs	SAW16041
obs	SAW16032
bic	JAV11007
bic	JAV11015
bic	JAV11016
bic	JAV11022
mos	REU03026

of study, it may be worth checking whether some of the recently published large-scale time-calibrated trees [28–32] contain taxa from the study group, which could allow the transfer of age information. If there really is no published age estimate for any divergence event within the study group, a possible solution could be to extend the dataset by adding a closely related species for which a divergence time estimate is available. If this is also not feasible, perhaps because samples of outgroups are not available or too distantly related to map to the same reference, a last option could be to also generate mitochondrial sequence data for some of the species in the dataset and estimate their divergence times based on an assumed mitochondrial substitution rate. This would need to be done a priori in a separate phylogenetic analysis, for example with BEAST 2, and the uncertainty in the assumed substitution rate should be accounted for.

Once a divergence event is identified for which external age information is available, this age information needs to be expressed in the form of a prior probability distribution (simply called "prior" hereafter). The molecular clock model for SNAPP allows the same types of priors that are used by BEAST 2 more generally, including uniform, normal, lognormal, exponential, and gamma distributions. Each of these distributions are defined by a set of parameters, such as the lower and upper boundaries in the case of the uniform distribution or the mean and the standard deviation in the case of the normal distribution. In addition, "offsets" can be used to shift the entire distribution without modifying its shape. A good introduction to the various priors available in BEAST 2 and SNAPP is given in Drummond and Bouckaert [23]. For age constraints based on previous studies, the most suitable prior types are usually normal or lognormal distributions. If, for example, a previous study had found that the group for which SNP data are analyzed began to diverge around 10 million years ago (Ma) with a 95% confidence interval spanning from 9 to 11 Ma, a normal distribution with a mean of 10 and a standard deviation adjusted so that 95% of the probability mass lie between 9 and 11 would be a suitable prior for the age of the root of the SNP-based species tree. If, however, the previously reported confidence interval would be skewed with respect to the mean estimate, which is often the case for age estimates, a lognormal distribution could provide a better fit. For example, when the 95% confidence interval ranges from 9 to 13 Ma and the mean age estimate is 10 Ma, a normal distribution would not be able to accommodate the asymmetry of the estimate, but a lognormal distribution (e.g., with an offset of 8.5, a mean of 1.5, and a standard deviation of 0.55) could approximate it. Identifying the distribution parameter combination that best fits published age estimates may require some trial-and-error testing, aiming for a distribution that approximates both the mean and the confidence interval of the published estimate well. BEAUti's prior preview panel (in the "Priors" tab) may be of help for this testing, but some example data must first be loaded into BEAUti to be able to set an age constraint in this panel (Fig. 2).

With the divergence event identified and the type of prior and the distribution parameters selected, an age constraints input file for ***snapp_prep.rb*** can be written. Based on this file, the script can then translate the constraint to XML format and include it in the input file for SNAPP. The format of the age constraints file for ***snapp_prep.rb*** is relatively simple: For each constraint, a single line with three tab- or space-delimited elements is required (*see* **Note 1**). The first of these three elements specifies the type of the prior (normal, lognormal, uniform, or "CladeAge"; the latter type is described in Matschiner et al. [33]), followed by comma-separated parameter values in parentheses. For normal and a lognormal distributions, the parameters offset, mean (in real space in case of

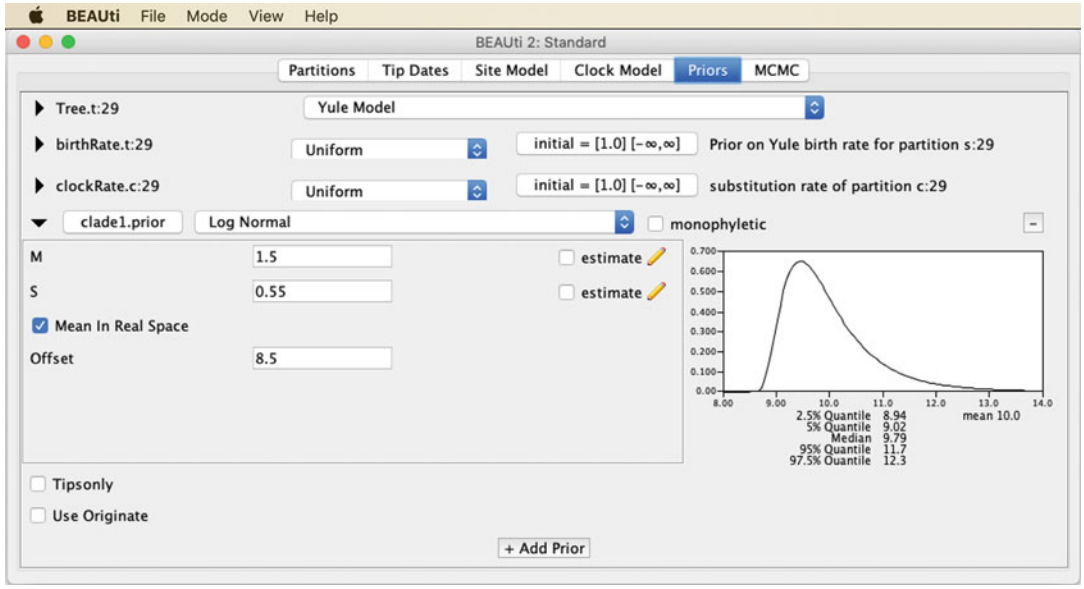

Fig. 2 Screenshot showing BEAUti's prior preview panel. With the chosen parameters, the lognormal prior has a mean of 10 (the sum of the offset, 8.5, and the distribution mean, 1.5) and 95% of the prior probability fall within the range from 8.94 to 12.3, the 2.5% and 97.5% quantiles

lognormal distributions), and standard deviation are expected, while for uniform distributions, only the lower and upper boundaries need to be specified (*see* **Notes 2** and **3**). The second element of the line should be either "crown" or "stem," depending on whether the age constraint should apply to the most recent common ancestor of the selected group ("crown") or the divergence of the group from its sister lineage ("stem"). The species IDs for members of the group should be specified separated by commas as the third element of the line; these should correspond to species IDs used in the species table. The following examples all show valid age constraints:

normal(0,10,0.5) crown speciesA,speciesB,speciesC

(a normally distributed constraint on the age of the most recent common ancestor of three species with a mean of 10 and a standard deviation of 0.5),

lognormal(8.5,1.5,0.55) stem speciesA,speciesB,speciesC

(a lognormally distributed constraint on the divergence time of three species from their sister lineage with an offset of 8.5, a mean of 1.5, and a standard deviation of 0.55),

uniform(10,15) crown speciesA,speciesB,speciesC

(a uniform constraint on the age of the most recent common ancestor of three species with a lower boundary of 10 and an upper boundary of 15).

In Barth et al. [27], a lognormal distribution was used to constrain the age of the most recent common ancestor of five species with an offset of 0, a mean of 13.76, and a standard deviation of 0.1, according to an earlier study based on mitochondrial sequences [34]:

lognormal(0,13.76,0.1) crown mar,meg,obs,bic,mos

Further information on constraint specification can be found in file example.con.txt, which is part of the snapp_prep GitHub repository (https://github.com/mmatschiner/snapp_prep).

2.7 Starting Tree

To initiate the MCMC chain, BEAST requires a starting tree. Usually, BEAST attempts to generate this starting tree itself; however, particularly when multiple age constraints are used, BEAST may not be able to produce a starting tree that is compatible with all constraints. If this is the case, BEAST immediately stops with an error message that includes the line "Fatal exception: Could not find a proper state to initialize" and also the line "P(prior) =--Infinity (was -Infinity)". When this problem occurs, it can be fixed by providing a starting tree in Newick format (http://evolution.genetics.washington.edu/phylip/newicktree.html) that is compatible with all specified constraints. To be readable by ***snapp_prep.rb***, the starting tree should be written to a file that contains only a single line and only the tree in Newick format on this line (*see* **Notes 4** and **5**). Note that besides the requirement that the tree should be compatible with all constraints, the topology and branch lengths chosen for the starting tree should not have any effect on the outcome of the SNAPP analysis and can therefore be chosen arbitrarily. For the dataset used by Barth et al. [27], a suitable starting tree would be the following.

((((mar:3,meg:3):3,obs:6):3,bic:9):3,mos:12);

To verify whether the starting tree is written as intended, the program FigTree (see below) can be used to visualize it.

2.8 Tracer

Tracer [24] is a very convenient and easy-to-use graphical user interface program for the assessment of MCMC stationarity and convergence. The program is available for Mac OS X, Linux, or Windows operating systems from GitHub at https://github.com/beast-dev/tracer/releases. While the instructions in this book chapter assume that Tracer is used to assess stationarity and convergence, it is worth pointing out the coda R package [35] as a useful alternative that also implements many of the functions available in Tracer.

2.9 FigTree

FigTree is a versatile graphical user interface program for the visualization of phylogenetic trees in Newick format. The program is available for Mac OS X, Linux, or Windows systems from GitHub at https://github.com/rambaut/figtree/releases.

3 Methods

3.1 Model

To reduce the computational demand of SNAPP, the method for divergence time estimation developed by Stange et al. [7] implements a model that is even more simplistic than the one used in standard SNAPP analyses. As in other SNAPP analyses, the Yule model [36] of lineage diversification is used, meaning that speciation events are assumed to occur with a constant rate per lineage and extinction is assumed to be absent. Also in common with other SNAPP analyses is the model assumption of a constant mutation rate that is identical in all lineages. While both assumptions are clearly violated by most or all empirical systems, it may be argued that at least within a system undergoing rapid diversification, the effects of extinction and rate variation may be small enough to be ignored. Going beyond the simplicity of standard SNAPP models is the assumption made in the model of Stange et al. [7] that all species have exactly the same population size. Even for recently diverged lineages, this assumption is rather unrealistic [12, 37], but as ancestral population sizes are inherently difficult to estimate and SNAPP analyses would otherwise hardly be possible for datasets of more than ten species, the assumption may nevertheless often be justified. Further reduction of model complexity is achieved by linking the forward and reverse mutation rates, which is not the case in standard SNAPP analyses.

Besides these model simplifications, the method of Stange et al. differs from standard SNAPP analyses also in the choice of priors. To both of the two parameters speciation rate (λ) and clock rate (μ), a scale-independent one-over-x prior is applied. The advantage of this is that the prior works equally well with young or old groups of species and no group-specific adjustments from the user are required. This is also the case for the prior on the population size parameter Θ, for which a very wide and therefore essentially uninformative uniform distribution is used. The model developed by Stange et al., including the above-described priors, is automatically selected when the XML file for SNAPP is written with the ***snapp_prep.rb*** script. For most users of divergence time estimation with SNAPP, no further modifications to the XML file will be necessary.

3.2 Generating the XML File with snapp_prep.rb

The minimum input required by ***snapp_prep.rb*** are three files: the one with the genotype data matrix, the file with the species assignment table, and the file with age constraints. If these are named matrix.vcf, species.txt, and constraints.txt, an XML file can be generated with ***snapp_prep.rb*** using the command:

```
ruby snapp_prep.rb -v matrix.vcf -t species.txt -c con-
straints.txt
```

This command would use all biallelic SNPs with sufficiently complete data, it would specify the default run length of 500,000 MCMC iterations, it would write an XML file with the default name snapp.xml, and it would set the output files of the SNAPP analysis to be named snapp.log and snapp.trees. A different number of MCMC iterations could be specified with the -l option (e.g., -l 100000), and smaller numbers of iterations might be advisable in initial analyses to explore the run time per iteration and how fast the MCMC chain approaches stationarity. The name of the XML file could be changed with the -x option, and different names for SNAPP's output files could be set with the -o option. A file with a starting tree could additionally be provided with the -s option, which may be helpful when BEAST is unable to generate a suitable starting tree itself. An overview of all available options can be displayed with the command:

```
ruby snapp_prep.rb -h
```

Some of the further options may be useful:

The relative weight of topology operators, and with it the frequency at which SNAPP attempts to change the tree topology during MCMC, can be changed with the -w option. The default for this option is 1; with values smaller or larger than 1, SNAPP will attempt to change the topology less frequently or more frequently, respectively, than other parameters. This option may be particularly useful when the user would like to fully fix the tree topology to the topology of a starting tree, which can be done by setting the relative weight to zero with -w 0.

To gain better control of the computational demand of the SNAPP analysis, a maximum number of SNPs can be specified with the -m option. When this option is used, the specified number of SNPs will be randomly selected from all those that are suitable for SNAPP. Similarly, a minimum distance between SNPs can be set with the -q option to reduce the potential effect of linkage among sites.

The effects of these two options are identical to those achieved by reducing and thinning the input VCF file a priori, but it may be more convenient to apply these filters with ***snapp_prep.rb*** because other tools cannot easily discriminate between SNPs suitable for SNAPP (e.g., those that have data for at least one individual per species) and those that will need to be excluded anyway.

While rates of mutations are well known to vary depending on the types of nucleotides that are exchanged [38], SNAPP does not model rate variation. One practical way to account at least partially for varying rates among nucleotide pairs is to reduce the genotype matrix to only transitions or only transversions, given that most rate variation is usually partitioned between these classes rather than within them [39]. This reduction can be done with the -i option to

include only transitions or with the -r option to include only transversions. When unsure which of the two classes of mutations to use, two separate XML files could be produced and SNAPP analyses could be performed separately with both files, allowing an assessment of the robustness of the results to these data subsets.

3.3 MCMC with BEAST

To perform SNAPP analyses with the XML file written with *snapp_prep.rb*, this file needs to be provided as input to BEAST. This can be done either using the command-line version of BEAST or its graphical user interface. If the XML file is named snapp.xml and BEAST is located in /Applications/BEAST/, the command-line version can be used to start MCMC with the command:

```
/Applications/BEAST/bin/beast snapp.xml
```

As SNAPP analyses can be parallelized very efficiently if multiple processors are available, the use of threading is recommended. On the command line, the use of multiple threads can be specified with the -threads option. For example, four threads can be used with the command:

```
/Applications/BEAST/bin/beast -threads 4 snapp.xml
```

The graphical user interface of BEAST can be launched by double-clicking on the program icon, which should open two windows as shown in Fig. 3. The input file can then be loaded by clicking on "Choose File ...", the number of threads can be selected from the drop-down menu next to "Thread pool size," and the MCMC chain can be started by clicking "Run."

During MCMC, BEAST's screen output shows values in eight columns that represent the current MCMC iteration, the posterior probability for this iteration, the cumulative effective sample size (ESS; see below) for the posterior probability, and the likelihood, the prior probability, the tree height (the age of the root of the tree), and the clock rate for this iteration. The last column at first only shows "--" but this is replaced after a certain number of iterations with an estimate of the required run time for one million iterations, as shown in Fig. 4.

It is worth following the screen output for some time. If the ESS value for the posterior increases above 200, the MCMC chain may have reached stationarity and a further extension to the chain may not be required. To verify stationarity, the output file with the ".log" filename extension should be inspected as described in the next section. If, on the other hand, the ESS value remains very low for a long time, stationarity may be difficult to reach and a restart of the analysis with a smaller dataset, or the use of a larger number of threads, should be considered. It is not uncommon for SNAPP analyses to require hours or days to finish, and in some cases the

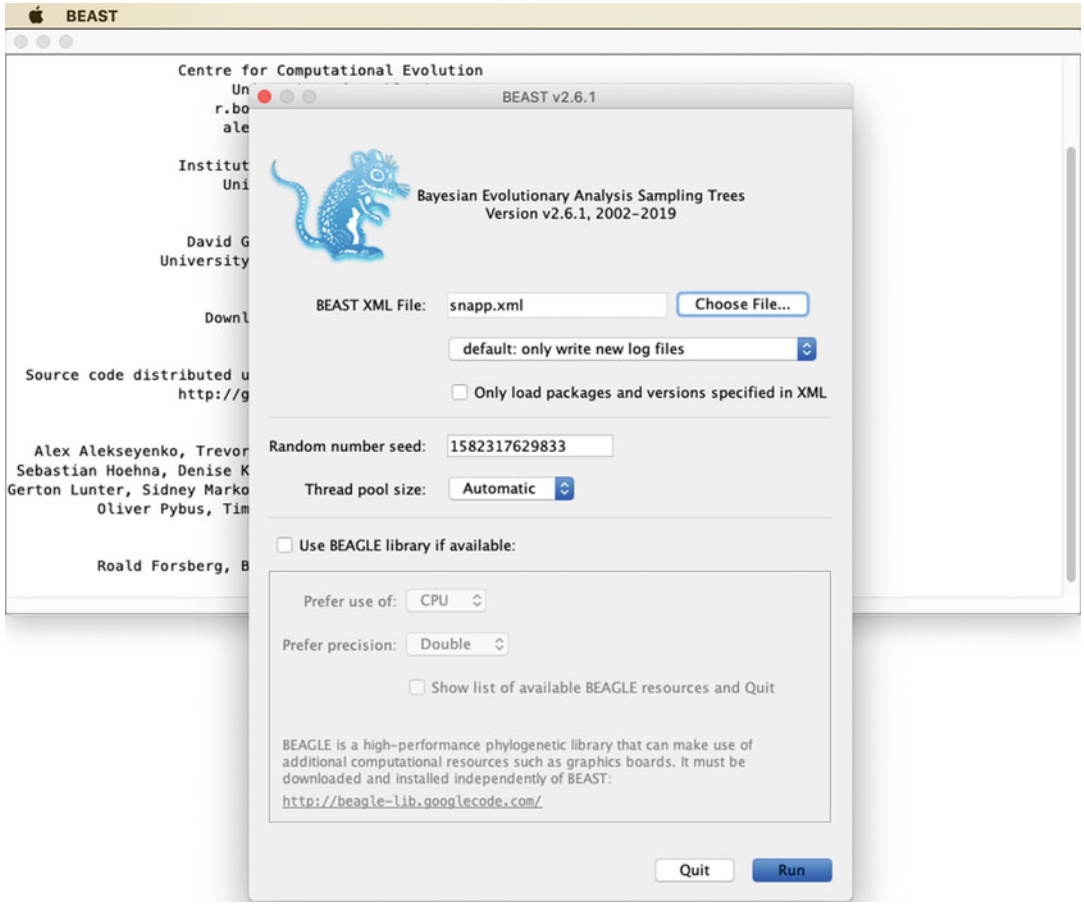

Fig. 3 Screenshot showing the file opening dialog of BEAST's graphical user interface

analysis may take weeks. Ultimately, whether or not the completion of a SNAPP analysis with a given dataset is feasible may depend on the patience of the user and the long-term access to computational resources with multiple processors.

3.4 Assessing Stationarity and Convergence with Tracer

During MCMC, BEAST writes two output files with the ".log" and ".trees" filename extensions. At the end of MCMC, these files should each contain output describing the state of the MCMC chain for 2000 iterations sampled at regular intervals. The output is divided so that all model parameters except the tree and the population size parameter Θ are written to the file with the ".log" extension while the tree, including branch lengths, and Θ are written in annotated Newick format to the file with the ".trees" extension. To assess MCMC chain stationarity, and thus whether or not the analysis should be extended, the file with the ".log" extension should be inspected with the program Tracer.

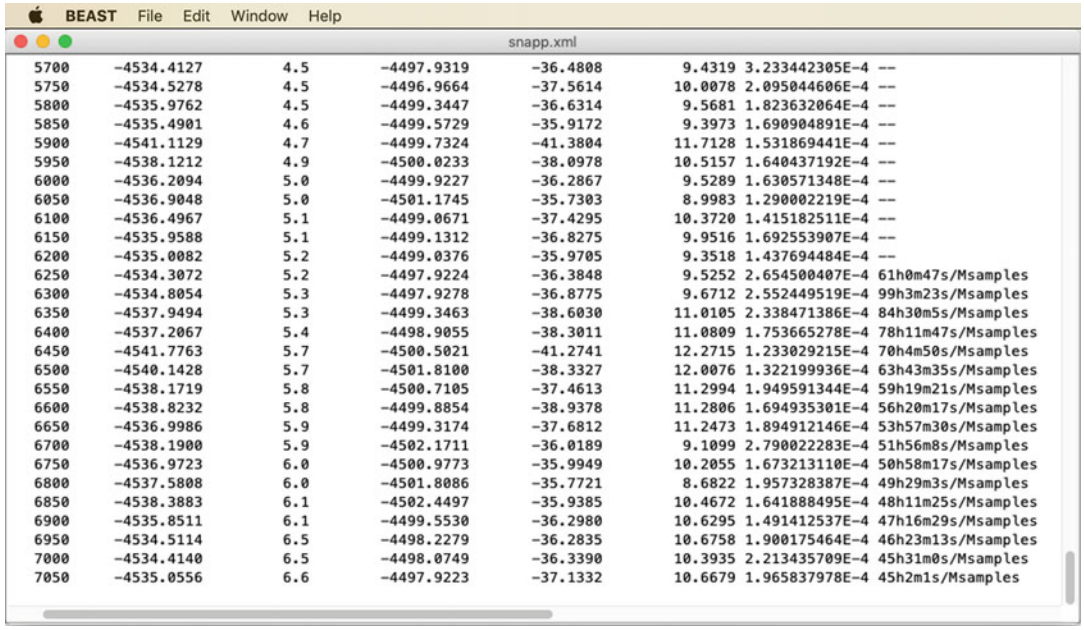

Fig. 4 Screenshot showing the SNAPP screen output in BEAST's graphical user interface

The many ways in which Tracer can be used to analyze MCMC results are described well in its publication [24] and in Drummond and Bouckaert [23]. In brief, Tracer is used to assess whether or not the MCMC chain has run long enough to allow conclusions, to adjust the length of the part of the MCMC chain that is considered as burn-in, and to extract parameter estimates and their confidence intervals. To determine that the chain was sufficiently long, stationarity and convergence must have been reached, indicating that the MCMC chain has sampled from the true posterior distribution. The first of these two criteria—stationarity—can be assumed when trends are no longer recognizable in trace plots of the sampled posterior probability, the likelihood, the prior probability, and all parameter values. One such trace plot, showing samples of the posterior probability, is illustrated in Fig. 5. Perhaps the most important measures of MCMC stationarity are the ESS values that are listed for posterior and prior probabilities, the likelihood, and parameter estimates in the bottom left panel of the Tracer window. These quantify the number of effectively independent samples drawn from the posterior distribution and thus account for autocorrelation in estimates sampled throughout the MCMC chain. As a rule of thumb, all ESS values should be greater than 200 before the MCMC chain can be considered stationary, but even larger values are preferable as they allow better estimates of confidence intervals [23]. To point out problematic estimates, Tracer marks ESS values smaller than 200 in red (ESS < 100) or yellow ($100 \leq \text{ESS} < 200$).

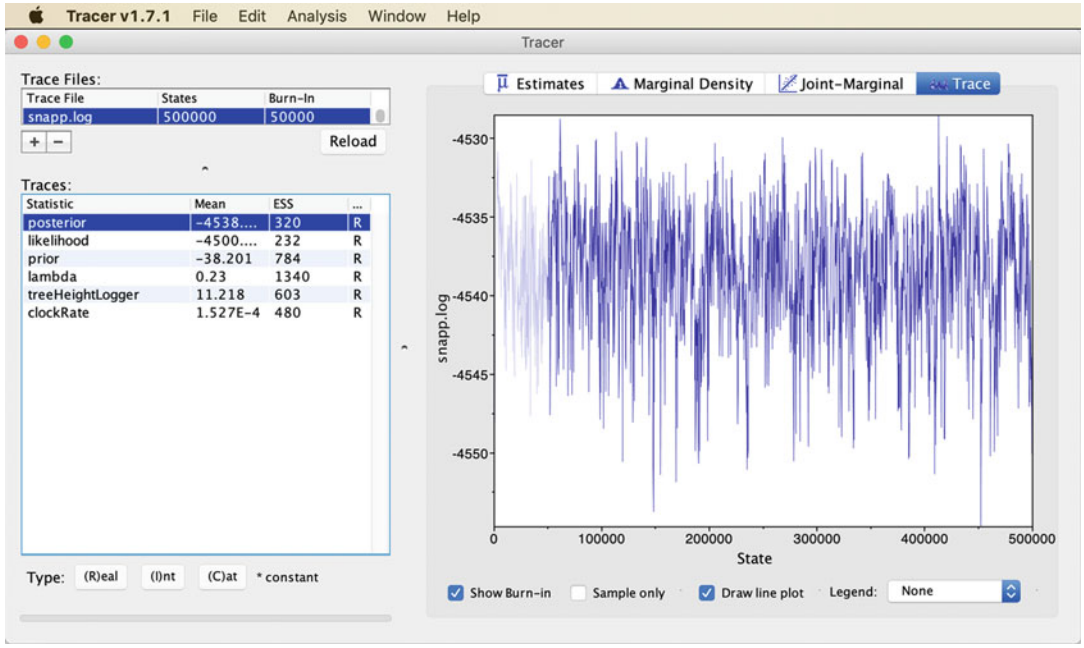

Fig. 5 Screenshot showing the Tracer window with the trace plot of the posterior probability. Trace plots can be displayed by selecting a statistic in the lower left panel and clicking the "Trace" button at the top right of the window

Even when the MCMC chain appears stationary based on visual inspection of trace plots and the ESS values, it may nevertheless not sample from the true posterior distribution. This is possible when the posterior probability surface has multiple peaks and the MCMC chain only explored a peak that is not the highest peak overall. While the probability surface may rarely be complex enough for this to happen with the simple models used in SNAPP, it is important to exclude this possibility. A good way to do so is to run multiple replicate analyses with the same XML file and verify that the MCMC chains in these analyses converge, meaning that they all arrive at roughly the same estimates even though they had different starting points (which is the case unless the same random number seed is reused).

To verify MCMC chain convergence, the files with ".log" extensions resulting from the multiple replicate analyses can be loaded jointly into Tracer. Below the names of these files in the top left panel of the Tracer window, an entry named "Combined" should then appear. When this entry is selected, trace plots will display the combined MCMC chain from the multiple files (excluding the burn-in parts of the individual chains), and all ESS values will be recalculated for the combined chain. If one or more of the run replicates did not converge, this will be obvious from marked steps in the trace plots and substantial decreases of ESS values.

If MCMC chains have reached stationarity and convergence, it may be worth optimizing the percentage of the chain that is considered as burn-in and thus excluded from the calculation of parameter estimates. With the default settings, the number of burn-in samples is specified as 50,000 in the column titled "Burn-in" in the top left panel of the Tracer window (Fig. 5), corresponding to 10% of the default chain length of 500,000 iterations. The length of the burn-in could be increased to, for example, 20%, by clicking on "50000" and writing "100000" instead. Adjusting the burn-in length is advisable if larger burn-in percentages improve the ESS values and the visual appearance of stationarity in trace plots.

If multiple MCMC replicates were performed (and all have converged), downstream analyses can be simplified by combining the result files from these replicates. This can be done separately for the files with the ".log" extension and for the files with the ".trees" extension, using the LogCombiner tool from the BEAST 2 suite of programs. The graphical user interface of LogCombiner can be used intuitively to load multiple input files, specify burn-in percentages for each of these, and set the name of the combined output file.

3.5 Obtaining Parameter Estimates with Tracer

Besides the posterior probability, the likelihood, and the prior probability, Tracer shows only three parameters in the lower left panel (if the XML was prepared with ***snapp_prep.rb***): "lambda," "treeHeightLogger," and "clockRate." Of these, "lambda" refers to the speciation rate (λ), "treeHeightLogger" refers to the age of the most recent common ancestor in the tree (thus, it is the sum of multiple branch lengths rather than a parameter itself), and "clockRate" refers to the rate of the molecular clock (μ). As discussed in Stange et al., the clock rate is subject to ascertainment bias when the dataset includes only SNPs and should not be directly interpreted as the mutation rate. However, as also shown in Stange et al., the clock rate estimate, together with the estimate for the population size parameter Θ, can serve to accurately estimate the effective population size N_e, given that $\Theta = 4N_e\mu g$ (with g being the generation time).

To add an estimate of N_e to a new file with ".log" extension that can be read by Tracer, the Ruby script ***add_theta_to_log.rb***, from the same GitHub repository as ***snapp_prep.rb***, can be used. This script reads the sampled clock rates from the result file with the ".log" ending and the sampled Θ values from the file with the ".trees" ending, calculates N_e from these values and a user-specified generation time, and writes a new file with the ".log" extension that is identical to the first except that it also contains samples for Θ and N_e. For example, with the result files snapp.log and snapp.trees and a generation time of 3 years, the script could be run with the command:

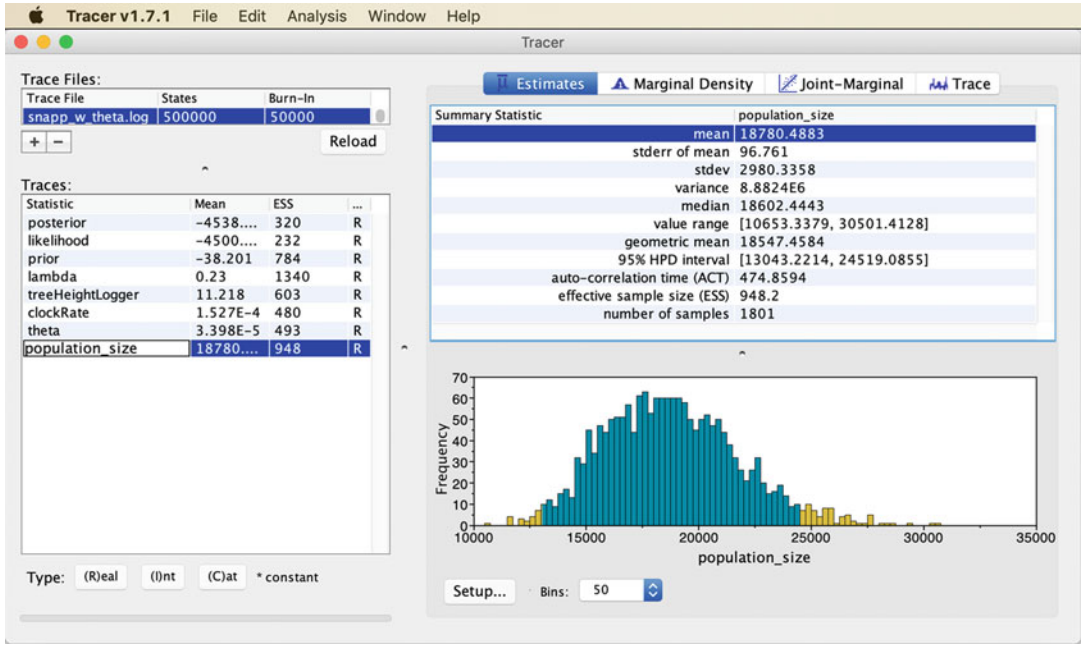

Fig. 6 Screenshot showing the Tracer window with summary statistics for the estimates of the effective population size (N_e)

```
ruby add_theta_to_log.rb -l snapp.log -t snapp.trees -g 3
```

This command would write an output file with the default name snapp_w_theta.log; other names could be specified with the -o option. Opening the output file of ***add_theta_to_log.rb*** in Tracer should show that two entries named "theta" (Θ) and "population_size" (N_e) have been added to the list of parameters in the lower left panel of the window (Fig. 6). Both of these parameters should also be checked for stationarity, but as the estimates for Θ are subject to the same ascertainment bias as those for the clock rate (μ), only the estimates for the population size should be interpreted. Selecting "population_size" in the parameter list in Tracer and clicking the "Estimates" button at the top center of the window should show summary statistics for this parameter, including the mean estimate, the standard deviation, and the 95% highest posterior density (HPD) interval, which in Bayesian analyses serves as the confidence interval (Fig. 6).

3.6 Generating a Summary Tree with TreeAnnotator

Opening the output file of SNAPP with the ".trees" extension in FigTree allows the user to view all the trees sampled during MCMC one by one. Alternatively, all sampled trees could be displayed simultaneously with DensiTree [40], another tool that is included in the BEAST 2 suite of programs. Often, however, a single tree summarizing the information from all sampled trees is required.

Such summary trees can be generated with TreeAnnotator, which identifies the most credible tree topology and representative clade ages based on criteria selected by the user [41]. For the tree topology, either "maximum clade credibility tree" or "maximum sum of clade credibilities" can be chosen; these two options select the tree topology for which either the product or the sum, respectively, of all node support values is highest. The clade ages, on the other hand, can be set either to the mean or the median of each clade's age in all posterior trees that contain this clade. Alternatively, "Common Ancestor heights" can be chosen, which calculates clade ages from all posterior trees, not just those that contain the clade [41]. For most species trees generated with SNAPP, these options should have rather little effect, given that SNAPP trees are usually well-supported and not overly species-rich. Perhaps the most commonly used setting is to produce a maximum clade credibility tree with mean node heights, which should work well for all SNAPP trees. Besides these options, the burn-in percentage should be specified (unless the burn-in part of the MCMC has already been removed, e.g., with LogCombiner), and input and output file names must be given. It is convenient to name the output file exactly like the input tree file, except that the ".trees" file extension is replaced with ".tre".

3.7 Visualizing the Summary Tree in FigTree

After opening the summary tree in FigTree, the program has various options to customize the tree's visualization. These options are accessible from the menu on the left of the FigTree window, within several panels that can be opened by clicking on the triangles and activated by checking the boxes next to these. Generally useful are the following options.

- Uncheck "Scale Bar" but check "Scale Axis," open the panel for "Scale Axis," uncheck "Show grid," and check "Reverse Axis." This adds a time scale in units of millions of years before present.
- Check and open the "Node Labels" panel, then set the drop-down menu next to "Display" to "posterior." This shows the support values for each node in the form of Bayesian posterior probabilities (BPP).
- Check and open the "Node Bars" panel, then set the drop-down menu next to "Display" to "height_95%_HPD". This adds blue bars to each node indicating the confidence interval for its age.

After the tree visualization has been adjusted as described above, a publication-ready figure of the species tree can be exported in PDF format via FigTree's "File" menu.

4 Notes

Issues encountered by users of ***snapp_prep.rb*** are often related to the preparation of the age constraints or species table input files. The following points should be considered if any issues arise in the preparation of these files:

1. If an error message or the resulting estimate of the species tree indicate that the constraints file may not have been read properly by ***snapp_prep.rb***, it may be worth checking that no spaces are included at the very beginning of the line defining the constraint. The same issue can result when the old Mac OS 9 format for line endings is accidentally used in the constraints file.
2. When using normal prior distributions for age constraints, the offset is redundant with the mean; thus, one of the two distribution parameters can always be set to zero.
3. Normal prior distributions may in some cases not work as well as lognormal distributions, because their tails are not bounded in either direction and therefore they do assign a certain prior probability to an age of zero (and even to negative ages). Through the interaction with the priors on the speciation rate and the population size, this may cause the MCMC chain to move toward a tree age of zero. When this issue is encountered, it can be fixed by replacing the normal prior distribution with a similarly shaped lognormal distribution.
4. The individual IDs used in the species table should match those used in the genotype data matrix exactly. No individual IDs should be used only in one of the two files but not the other. Similarly, the species IDs used in the species table should exactly match those in the starting tree if a starting tree is provided. Finally, the IDs used in the definition of age constraints should be species IDs, not individual IDs.
5. If a starting tree is provided, this tree should not contain nodes with only one descendant, and no branch should be included above the root of the tree. One way to test for unintended nodes and branches is to open the tree in FigTree and activate the "Node Shapes" panel, which marks all nodes with circles or other symbols.

If any other issues should arise, I recommend that questions related to SNAPP are posted on the BEAST user group (https://groups.google.com/forum/#!forum/beast-users) while questions related to ***snapp_prep.rb*** should be directed to me by email.

Acknowledgments

I thank Julie Lee-Yaw, Amanda Haponski, Livia Loureiro, Sue Sherman-Broyles, Bohao Fang, Yayan Kusuma, Daniel Poveda-Martínez, Xiaoxi Yang, Cecilia Fiorini, Kristen Finch, Armel Donkpegan, Marta Liber, Jie Gao, and Julia Canitz for testing the ***snapp_prep.rb*** script. Funding was provided by the Research Council of Norway (FRIPRO 275869).

References

1. Kubatko LS, Degnan JH (2007) Inconsistency of phylogenetic estimates from concatenated data under coalescence. Syst Biol 56:17–24
2. Leaché AD, Rannala B (2011) The accuracy of species tree estimation under simulation: a comparison of methods. Syst Biol 60:126–137
3. Liu L, Edwards SV (2009) Phylogenetic analysis in the anomaly zone. Syst Biol 58:452–460
4. Degnan JH, DeGiorgio M, Bryant D, Rosenberg NA (2009) Properties of consensus methods for inferring species trees from gene trees. Syst Biol 58:35–54
5. Roch S, Steel M (2014) Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. Theor Popul Biol 100:56–62
6. Ogilvie HA, Bouckaert RR, Drummond AJ (2017) StarBEAST2 brings faster species tree inference and accurate estimates of substitution rates. Mol Biol Evol 34:2101–2114
7. Stange M, Sánchez-Villagra MR, Salzburger W, Matschiner M (2018) Bayesian divergence-time estimation with genome-wide SNP data of sea catfishes (Ariidae) supports Miocene closure of the Panamanian Isthmus. Syst Biol 67:681–699
8. Maddison WP (1997) Gene trees in species trees. Syst Biol 46:523–536
9. Liu L (2008) BEST: Bayesian estimation of species trees under the coalescent model. Bioinformatics 24:2542–2543
10. Edwards SV (2009) Is a new and general theory of molecular systematics emerging? Evolution 63:1–19
11. Kubatko LS, Carstens BC, Knowles LL (2009) STEM: species tree estimation using maximum likelihood for gene trees under coalescence. Bioinformatics 25:971–973
12. Heled J, Drummond AJ (2010) Bayesian inference of species trees from multilocus data. Mol Biol Evol 27:570–580
13. Yang Z (2015) The BPP program for species tree estimation and species delimitation. Curr Zool 61:854–865
14. Zhang C, Rabiee M, Sayyari E, Mirarab S (2018) ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. BMC Bioinformatics 19:153
15. Edwards SV, Xi Z, Janke A et al (2016) Implementing and testing the multispecies coalescent model: a valuable paradigm for phylogenomics. Mol Phylogenet Evol 94: 447–462
16. Springer MS, Gatesy J (2016) The gene tree delusion. Mol Phylogenet Evol 94:1–33
17. Chifman J, Kubatko LS (2014) Quartet inference from SNP data under the coalescent model. Bioinformatics 30:3317–3324
18. Bryant D, Bouckaert RR, Felsenstein J, Rosenberg NA, RoyChoudhury A (2012) Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Mol Biol Evol 29:1917–1932
19. De Maio N, Schrempf D, Kosiol C (2015) PoMo: an allele frequency-based approach for species tree estimation. Syst Biol 64:1018–1031
20. Stoltz M, Bauemer B, Bouckaert R et al (2021) Bayesian inference of species trees using diffusion models. Syst Biol 70:145–161
21. Leaché AD, Fujita MK, Minin VN, Bouckaert RR (2014) Species delimitation using genome-wide SNP data. Syst Biol 63:534–542
22. Bouckaert RR, Vaughan TG, Barido-Sottani J et al (2019) BEAST 2.5: an advanced software platform for Bayesian evolutionary analysis. PLoS Comput Biol 15:e1006650
23. Drummond AJ, Bouckaert RR (2015) Bayesian evolutionary analysis with BEAST 2. Cambridge University Press, Cambridge
24. Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA (2018) Posterior summarization

in Bayesian phylogenetics using Tracer 1.7. Syst Biol 67:901–904

25. Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W (2002) Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. Genetics 161:1307–1320
26. Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17:368–376
27. Barth JMI, Gubili C, Matschiner M et al (2020) Stable species boundaries despite ten million years of hybridization in tropical eels. Nat Commun 11:1433
28. Kumar S, Stecher G, Suleski M, Hedges SB (2017) TimeTree: a resource for timelines, timetrees, and divergence times. Mol Biol Evol 34:1812–1819
29. Fernández R, Kallal RJ, Dimitrov D et al (2018) Phylogenomics, diversification dynamics, and comparative transcriptomics across the spider Tree of Life. Curr Biol 28:1489–1497
30. Rabosky DL, Chang J, Title PO et al (2018) An inverse latitudinal gradient in speciation rate for marine fishes. Nature 559:392–395
31. Upham NS, Esselstyn JA, Jetz W (2019) Inferring the mammal tree: species-level sets of phylogenies for questions in ecology, evolution, and conservation. PLoS Biol 17:e3000494
32. Janssens S, Couvreur TLP, Mertens A et al (2020) A large-scale species level dated angiosperm phylogeny for evolutionary and ecological analyses. Biodiv Data J 8:e39677
33. Matschiner M, Musilova Z, Barth JMI et al (2017) Bayesian phylogenetic estimation of clade ages supports trans-Atlantic dispersal of cichlid fishes. Syst Biol 66:3–22
34. Jacobsen MW, Pujolar JM, Gilbert MTP et al (2014) Speciation and demographic history of Atlantic eels (*Anguilla anguilla* and *A. rostrata*) revealed by mitogenome sequencing. Heredity 113:432–442
35. Plummer M, Best N, Cowles K, Vines K (2006) CODA: convergence diagnosis and output analysis for MCMC. R News 6:7–11
36. Yule GU (1925) A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. Phil Trans R Soc Lond B 213:21–87
37. Genner MJ, Turner GF (2014) Timing of population expansions within the Lake Malawi haplochromine cichlid fish radiation. Hydrobiologia 748:121–132
38. Yang Z (1994) Estimating the pattern of nucleotide substitution. J Mol Evol 39:105–111
39. Brown WM, Prager EM, Wang A, Wilson AC (1982) Mitochondrial DNA sequences of primates: tempo and mode of evolution. J Mol Evol 18:225–239
40. Bouckaert RR (2010) DensiTree: making sense of sets of phylogenetic trees. Bioinformatics 26: 1372–1373
41. Heled J, Bouckaert RR (2013) Looking for trees in the forest: summary tree from posterior samples. BMC Evol Biol 13:211