

Bayesian molecular clock dating using genome-scale datasets

Mario dos Reis¹ and Ziheng Yang²

1. School of Biological and Chemical Sciences, Queen Mary University of London, London E1 4NS, UK.

2. Department of Genetics, Evolution and Environment, University College London, London WC1E 6BT, UK.

Correspondence to: Mario dos Reis (m.dosreisbarros@qmul.ac.uk) and Ziheng Yang (z.yang@ucl.ac.uk).

Running head: Molecular clock dating

i. Summary

Bayesian methods for molecular-clock dating of species divergences have been greatly developed during the past decade. Advantages of the methods include the use of relaxed-clock models to describe evolutionary rate variation in the branches of a phylogenetic tree, and the use of flexible fossil-calibration densities to describe the uncertainty in node ages. The advent of next-generation sequencing technologies has led to a flood of genome-scale datasets for organisms belonging to all domains in the Tree of Life. Thus, a new era has begun where dating the Tree of Life using genome-scale data is now within reach. In this protocol, we explain how to use the computer program MCMCTree to perform Bayesian inference of divergence times using genome-scale datasets. We use a 10-species primate phylogeny, with a molecular alignment of over 3 million base pairs, as an exemplar on how to carry out the analysis. We pay particular attention to how to set up the analysis, the priors, and how to diagnose the MCMC algorithm used to obtain the posterior estimates of divergence times and evolutionary rates.

ii. Keywords:

Molecular clock, Bayesian analysis, MCMC, fossil, phylogeny, Primates, genome.

1. Introduction

The molecular clock hypothesis, which states that the rate of molecular evolution is approximately constant with time, provides a powerful way to estimate the times of divergence of species in a phylogeny. Since its proposal over 50 years ago (**1**), the

molecular clock hypothesis has been used countless times to calibrate molecular phylogenies to geological time, with the ultimate aim of dating the Tree of Life (2, 3). Several statistical inference methodologies have been developed for molecular clock dating analyses; however, during the past decade, the Bayesian method has emerged as the method of choice (4, 5), and several Bayesian inference software packages now exist to carry out this type of analysis (6–10).

In this protocol, we will explain how to use the computer program MCMCTree to estimate times of species divergences using genome-scale datasets within the Bayesian inference framework. Bayesian inference is well suited for divergence time estimation because it allows the natural integration of information from the fossil record (in the form of prior statistical distributions describing the ages of nodes in a phylogeny) with information from molecular sequences to estimate node ages, or geological times of divergence, of a species phylogeny (6, 11). Another advantage of the Bayesian clock-dating method is that relaxed-clock models, which allow for violations of the molecular clock, can be easily implemented as the prior on the evolutionary rates for the branches in the phylogeny (6). MCMCTree allows analyses to be carried using two popular relaxed-clock models (the autocorrelated and independent log-normally distributed rates models, (12, 13)), as well as under the strict molecular clock. Furthermore, MCMCTree allows the user to build flexible fossil calibrations based on various statistical distributions (such as the uniform, truncated-Cauchy, and skew-t and Skew-normal distributions, (12, 14, 15)). But perhaps the main advantage of MCMCTree is the implementation of an approximate algorithm to calculate the likelihood (6, 16), which allows the computer analysis of genome-scale data sets to be completed in reasonable amounts of time. The disadvantage of the algorithm is that it only works on fixed tree

topologies. Several software packages that perform co-estimation of times and tree topology, but which do not use the approximation, are available **(8, 9, 17, 18)**.

In this protocol, we focus on how to carry out a clock-dating analysis with MCMCTree, paying particular attention to diagnosing the MCMC algorithm (the workhorse algorithm within the Bayesian method). Theoretical details of the Bayesian clock-dating methods implemented in the program MCMCTree are described in **(12–16, 19)**. For general introductions to Bayesian statistics and Bayesian molecular clock dating, the reader may consult **(20, 21)**.

2. Software and data files

To run the protocol you will need the MCMCTree and BASEML programs, which are part of the PAML software package for phylogenetic analysis **(22)**. The source code and compiled versions of the code are freely available from bit.ly/ziheng-paml. All the data files necessary to run the protocol can be obtained from github.com/mariodosreis/divtime. Please create a directory called `divtime` in your computer and download all the data files from the GitHub repository. This protocol was tested with PAML version 4.9e.

You are assumed to have basic knowledge of the command line in Unix or Windows (also known as command prompt, shell or terminal). Simple tutorials for users of Windows, Mac OS, and Linux are posted at bit.ly/ziheng-software. Install MCMCTree and BASEML in your computer system, and make sure you have the `mcmctree` and `baseml` executables in your system's path (see bit.ly/ziheng-paml for details on how to do this). Finally, it is helpful (but not indispensable) to have knowledge of the R

statistical environment (www.r-project.org). R is quite useful to analyse the output of the program, perform convergence diagnostics, and create nice looking plots. File `R/analysis.R` contains some examples for this tutorial.

In this protocol, we will estimate the divergence times of 9 primates and 1 scandentian (an outgroup), using a very long alignment (over 3 million nucleotides long). This data set was chosen because it can be analysed very quickly with MCMCTree, and it is thus suitable to illustrate the method. We also provide a data set of 330 species (276 primates and 4 outgroups) with a shorter alignment, to illustrate time estimation in a taxon-rich dataset (see Exercise 4.5 for details).

2.1 Tree and fossil calibrations

The phylogenetic tree of the 10 species is shown in figure 1. The tree encompasses members of all the main primate lineages. The 10 species were chosen because they have had their complete genomes sequenced. They are a subset of the 36 mammal species analysed in (23). File `data/10s.tree` contains the tree with fossil calibrations in Newick format, which is the format required by MCMCTree. The eight fossil calibrations are shown in Table 1. The calibrations are the same used to estimate Primate divergence times in (24). We discuss fossil calibrations in detail in the “Sampling from the prior” section. The time unit in the analysis is 100 million years (My). Thus, the calibration $B(0.075, 0.10)$ means the node age is constrained to be between 7.5 and 10 million years ago (Ma).

2.3 Molecular sequence data

The molecular data are an alignment of 5,614 protein-coding genes from the 10 species. All ambiguous codon sites were removed, and thus the alignment contains no missing

data. The alignment was separated into two partitions: A partition consisting of all the 1st and 2nd codon positions (2,253,316 nucleotides long) and a partition of 3rd codon positions (1,126,658 nucleotides long). The alignment is a subset of the larger 36 mammal-species alignment in **(23)**. See also **(24)**. File 10s.phys in the data directory contains the alignment. The alignment is compressed into site patterns (a site pattern is a unique combination of character states in an alignment column) to save disk space.

3. Tutorial

We seek to obtain the posterior distribution (i.e. the estimates) of the divergence times (**t**) and the molecular evolutionary rates (**r**, μ , σ^2) for the species in the phylogeny of Fig. 1. Here **t** = (t_{11} , ..., t_{19}) are the 9 species divergence times, **r** = ($r_{1,12}$, ..., $r_{1,19}$, $r_{2,12}$, ..., $r_{2,19}$), are the $2 \times 8 = 16$ molecular rates, one per branch and partition (i.e there are 8 branches in the tree and 2 partitions in the molecular data), and $\mu = (\mu_1, \mu_2)$ and $\sigma^2 = (\sigma_1^2, \sigma_2^2)$ are the mean rates, and the log-variance of the rates, for each partition. The posterior distribution is

$$f(\mathbf{t}, \mathbf{r}, \mu, \sigma^2 | D) \propto f(\mathbf{t})f(\mathbf{r} | \mathbf{t}, \mu, \sigma^2)f(\mu)f(\sigma^2)f(D | \mathbf{r}, \mathbf{t}),$$

where $f(\mathbf{t})$ is the prior on times, $f(\mathbf{r} | \mathbf{t}, \mu, \sigma^2)f(\mu)f(\sigma^2)$ is the prior on the branch rates, mean rates and log-variances of the rates, and $f(D | \mathbf{t}, \mathbf{r})$ is the molecular sequence likelihood. The prior on the times is constructed by combining the birth-death process with the fossil calibration densities (see **(13)** for details). The prior on the rates is constructed assuming a model of rate evolution, in this tutorial, assuming that the branch rates are independent draws from a log-normal distribution with mean μ_i , and log-variance σ_i^2 **(13)**.

Bayesian phylogenetic inference using MCMC is computationally expensive because of the repeated calculation of the likelihood on a sequence alignment. The time it takes to compute the likelihood is proportional to the number of site patterns in the alignment. Thus, longer alignments take longer to compute. For genome-scale alignments the computation time is prohibitive.

MCMCTree implements an approximation to the likelihood that speeds computation time substantially, making analysis of genome-scale data feasible. The approximate likelihood method for clock dating was proposed by Thorne et al. (6) and extended within MCMCTree (16). The method relies on approximating the log-likelihood surface on the branch lengths by its Taylor expansion. Write $\ell(\mathbf{b}_j) = \log f(D | \mathbf{b}_j)$ for the log-likelihood as a function of the branch lengths $\mathbf{b}_j = (b_{j,i} = r_{j,i}t_i)$ for the alignment partition j .

The Taylor approximation is

Note: This should be $b_{\{j,i\}} = r_{\{j,i\}} * \Delta t_i$, where Δt_i is the length timespan of the branch.

$$\ell(\mathbf{b}_j) \approx \ell(\hat{\mathbf{b}}_j) + (\mathbf{b}_j - \hat{\mathbf{b}}_j)^T \mathbf{g}_j + \frac{1}{2} (\mathbf{b}_j - \hat{\mathbf{b}}_j)^T \mathbf{H}_j (\mathbf{b}_j - \hat{\mathbf{b}}_j),$$

where $\hat{\mathbf{b}}_j$ are the maximum likelihood estimates (MLEs) of the branch lengths, and \mathbf{g}_j and \mathbf{H}_j are the gradient (vector of first derivatives) and Hessian (matrix of second derivatives) of the log-likelihood surface evaluated at the MLEs for the partition. The approximation can be improved by applying transformations to the branch lengths (see (16) for details).

To use the approximation, one first fixes the topology of the phylogeny, and then one estimates the branch lengths for each alignment partition on the fixed tree by maximum likelihood. The gradient and Hessian of the log-likelihood are obtained for each partition at the same time as the MLEs of the branch lengths. Note that parameters of

the substitution model –such as the transition/transversion ratio, k , in the HKY model or the α parameter in the discrete-gamma model of rate variation among sites– are estimated at this step. Thus, different substitution models will generate different approximations, because they will have different MLEs for the branch lengths, gradient and Hessian. Note that the time it takes to compute the approximation depends only on the number of species (which determines the size of \mathbf{b} and \mathbf{H}), and not on the alignment length, that is, once \mathbf{g} and \mathbf{H} have been calculated, MCMC sampling on the approximation takes the same time regardless of the length of the original alignment.

3.1 Overview

We will use the approximate likelihood method to speed up the computation of the likelihood on the large genome alignment. The general strategy for the analysis is as follows:

1. *Approximate likelihood calculation:* First, we will calculate the gradient (\mathbf{g}) and Hessian (\mathbf{H}) matrix of the branch lengths on the unrooted tree. For this step, we will need to use the MCMCTree and BASEML programs (BASEML will carry out the actual computation of \mathbf{g} and \mathbf{H}). The substitution model is chosen at this step.
2. *MCMC sampling from the posterior:* Once \mathbf{g} and \mathbf{H} have been calculated and we have decided on our priors, we can use MCMCTree to perform MCMC sampling from the posterior distribution of times and rates. We will then look at the summaries of the posterior (such as posterior mean times and rates, and 95% credibility intervals).

180 3. *Convergence diagnostics*: The MCMC algorithm is a stochastic algorithm that visits
181 regions of the parameter space in proportion to the posterior distribution. Due
182 to its very nature, it is possible that sometimes the MCMC chain is terminated
183 before it has had a chance to explore the parameter space appropriately. The
184 way to guard against this is to run the analysis two or more times, and compare
185 the summary statistics from the two (or more) MCMC chains. If the results from
186 different runs are very similar, then convergence to the posterior distribution can
187 be reasonably assumed.

188
189 4. *MCMC sampling from the prior*: Finally, we will sample directly from the prior of
190 times and rates. This is particularly important in Bayesian molecular clock
191 dating because in most cases the prior on times may look quite different from the
192 fossil calibration densities specified by the user. Thus, sampling from the prior
193 allows the user to check the soundness of the prior actually used.

194
195 Note that in this protocol we assume the user has chosen a suitable sequence alignment
196 and a phylogenetic tree to carry out the analysis. For genome-scale alignments, it is
197 important that the genes chosen among the various species are orthologous, and that
198 the alignment has been checked for accuracy. Several chapters in this volume can guide
199 the user in this purpose.

200 3.2 Calculation of the gradient and Hessian to approximate the likelihood

201 Go into the gH directory and open the `mcmctree-outBV.ct1` file using your favourite
202 text editor. This control file contains the set of parameters necessary for MCMCTree to

203 carry out the calculations of the gradient and Hessian needed for the approximate
 204 likelihood method. Figure 2 shows the contents of the `mcmctree-outBV.ctl` file.
 205
 206 The first two items, `seqfile` and `treefile`, indicate the alignment and tree files to be
 207 used. The third item, `ndata`, indicates the number of partitions in the sequence file, in
 208 this case, two partitions. The fifth item, `usedata`, is very important, as it tells
 209 MCMCTree the type of analysis being carried out. The options are 0: sample from the
 210 prior, 1: to sample from the posterior using exact likelihood, 2: to sample from the
 211 posterior using approximate likelihood, and 3: to prepare the data for calculation of **g**
 212 and **H**. The last is the option we will be using in this step. The next three items, `model`,
 213 `alpha`, and `ncatG`, set up the nucleotide substitution model, in this case the
 214 HKY+Gamma model **(6, 25)**. Finally, the `cleandata` option tells MCMCTree whether to
 215 remove ambiguous data. Our alignment has no ambiguous sites, so this option has no
 216 effect in this case.
 217
 218 Using a terminal, go to the `gH` directory and type
 219
 220 `$ mcmctree mcmctree-outBV.ctl`
 221
 222 (don't type in the `$` as this represents the command prompt!). This will start the
 223 MCMCTree program. MCMCTree will prepare several `tmp????.*` files, and will then
 224 call the BASEML program to estimate **g** and **H**. For this step to work correctly, the
 225 `baseml` executable must be in your system's path. Once BASEML and MCMCTree have
 226 finished, you will notice a file called `out.BV` has been created. Figure 3 shows part of
 227 the contents of this file. The first line indicates the number of species (10), followed by

the tree with branch lengths estimated under maximum likelihood for the first partition (1st and 2nd codon sites). Next, we have the MLEs of the 17 branch lengths (these are the same as in the tree, but printed in a different order). Then we have the gradient, \mathbf{g}_1 , the vector of 17 first derivatives of the likelihood at the branch length MLEs for partition 1. For small datasets, the gradient is usually zero. For large datasets, the likelihood surface is too sharp (i.e. bends downwards sharply and it is very narrow at the MLEs) and the gradient is not zero for numerical issues. But this is fine. Next, we have the 17×17 Hessian matrix, \mathbf{H}_1 , the matrix of second derivatives of the likelihood at the branch length MLEs for partition 1. If you scroll down the file, you will find the second block, with the tree, branch length MLEs, \mathbf{g}_2 and \mathbf{H}_2 for partition 2 (3rd codon positions).

3.3 Calculation of the posterior of times and rates

3.3.1 Control file and priors

Now that we have calculated \mathbf{g} and \mathbf{H} , we can proceed to MCMC sampling of the posterior distribution using the approximate likelihood method. Copy the `gH/out.BV` file into the `mcmc` directory, and rename it as `in.BV`. Now go into the `mcmc` directory. There you will find, `mcmctreectl`, the necessary MCMCTree control file to carry out MCMC sampling from the posterior. Figure 4 shows the contents of the file. The first item, `seed`, is the seed for the random number generator used by the MCMC algorithm. Here it is set to `-1`, which tells MCMCTree to use the system's clock time as the seed. This is useful, as running the program multiple times will generate different outputs.

The `mcmcfile` option tells MCMCTree where to save the parameters sampled (divergence times and rates) during the MCMC iterations. Here we will save them to a file named `mcmc.txt`. Once the MCMC sampling has completed, MCMCTree will read

252 the sample from the `mcmc.txt` file and generate a summary of the MCMC output. This
253 summary will be saved to a file called `out.txt` (`outfile` option).

254

255 The option `usedata` is set to 2 here, which tells MCMCTree to calculate the likelihood
256 approximately by using the **g** and **H** values saved in the `in.BV` file. Option `clock` sets
257 the clock model. Here we use `clock = 2`, which assumes rates are identical, independent
258 realizations from a log-normal distribution **(7, 26)**. Option `RootAge` sets the
259 calibration on the root node of the phylogeny, if none are present in the tree file. In our
260 case, we already have a calibration on the root so this option has no effect. The next
261 three options, `model`, `alpha`, `ncatG`, have no effect as the substitution model was
262 chosen during estimation of **g** and **H**.

263

264 The following options are very important as they determine the prior used in the
265 analysis. `BDparams` sets the prior on node ages for those nodes without fossil
266 calibrations by using the birth-death process **(12)**. Here we use 1 1 0, which means
267 node ages are uniformly distributed between present time and the age of the root.
268 Options `kappa_gamma` and `alpha_gamma` set gamma priors for the κ and α
269 parameters in the substitution model. These have no effect as we are using the
270 likelihood approximation. Options `rgene_gamma` and `sigma2_gamma` set the gamma-
271 Dirichlet prior on the mean substitution rate for partitions and for the rate variance
272 parameter, σ^2 **(19)**. The prior on the mean rate is $\text{Gamma}(2, 40)$, which has mean 0.05
273 substitutions per time 100 My. A symmetric Dirichlet distribution with concentration
274 parameter equal to 1 is used to spread the rate prior across partitions (thus
275 `rgene_gamma = 2 40 1`). See **(19)** for details. The prior on σ^2 is $\text{Gamma}(1, 10)$
276 which has mean 0.1. A Dirichlet is also used to spread the prior across partitions.

277

278 The final block of options, `print`, `burnin`, `sampfreq` and `nsample` control the length
279 and sampling frequency of the MCMC. We will discard the first 20,000 iterations as the
280 burn-in, and then print parameter values to the `mcmc.txt` file every 100 iterations, to a
281 maximum of $20,000 + 1$ samples. Thus, our MCMC chain will run for a total of $20,000 +$
282 $20,000 \times 100 = 2,020,000$ iterations.

283 3.3.2 Running and summarising the MCMC

284 Go into the `mcmc` directory and type

285

286 `$ mcmctree mcmctree.ctl`

287

288 This will start the MCMC sampling. First, MCMCTree will iterate the chain for a set
289 number of iterations, known as the burn-in. During this period, the program will fine-
290 tune the step sizes for proposing parameters in the chain. Once the burn-in is finished,
291 sampling from the posterior will start. Figure 5 shows a screenshot of MCMCTree in
292 action. The left most column indicates the progress of the sampling as a percentage of
293 the total (5%, 10% of total iterations and so on). The next numbers represent the
294 acceptance proportions, which are close to 30% (this is the result of fine-tuning by the
295 program). After the five acceptance proportions the programs prints a few parameters
296 to the screen, and in the last columns the log-likelihood and the time taken.

297

298 The above analysis takes about 2 min and 30 s to complete on a 2.2GHz Intel Core i7
299 Processor. Once the analysis has finished, you will see that MCMCTree has created
300 several new files in the `mcmc` directory. Rename `mcmc.txt` to `mcmc1.txt` and
301 `out.txt` to `out1.txt`. Now, on the command line, type again

302

303 `$ mcmctree mcmctree.ctl`

304

305 This will run the analysis a second time. The results should be slightly different to the

306 previous run due to the stochastic nature of the algorithm. Once the second run has

307 finished, rename `mcmc.txt` to `mcmc2.txt` and `out.txt` to `out2.txt`. If you want to

308 conduct two runs simultaneously, you can create two directories (say `r1/` and `r2/`) and

309 copy the necessary files into them. Then open two terminal windows to start the runs

310 from within each directory.

311

312 Using your favourite text editor, open file `out1.txt`, which contains the summary of

313 the first MCMC run. Scroll to the end of the file (see screenshot, fig. 6). You will see the

314 time used by the program (in my case 2:32), the posterior means of the parameters

315 sampled, and three phylogenetic trees in Newick format. The first tree simply has

316 internal nodes labelled with a number. This is useful to compare the tree with the

317 posterior means of times at the end of the file. The second tree is the tree with branch

318 lengths in absolute time units. The third tree is like the second by includes the 95%

319 credibility intervals (CIs) of the node ages. At the bottom of the file you have a table

320 with all the divergence times (from `t_n11` to `t_n19`), the mean substitution rates for

321 the two partitions (`mu1` and `mu2`), the rate variation coefficients (`sigma2_1` and

322 `sigma2_2`), and finally the log-likelihood (`lnL`). The table gives the posterior means,

323 equal-tail CIs and high-posterior density CIs. For example, the posterior age of the root

324 (node 11, fig. 1) is 116.8 Ma (95% CI: 144.2–92.4 Ma), while for the divergence between

325 human and chimp (node 19, fig. 1) is 8.52 Ma (95% CI: 7.58–9.81 Ma).

326

327 You will also notice that MCMCTree created a file called `FigTree.tre`. This contains
328 the posterior tree in Nexus format, suitable for plotting in the program FigTree
329 (tree.bio.ed.ac.uk/software/figtree/). Figure 7 shows the posterior tree plotted in
330 FigTree, with the time unit set to 1 My.

331 3.4 Convergence diagnostics of the MCMC

332 Diagnosing convergence of the MCMC chains is extremely important. Several software
333 tools have been written for this purpose. For example, the user-friendly Tracer program
334 (beast.bio.ed.ac.uk/tracer) can be used to read in the `mcmc1.txt` and `mcmc2.txt` files
335 and calculate several convergence statistics. Here we will use R to perform basic
336 convergence tests (check out file `R/analysis.R`).

337

338 The first step to assess convergence is to compare the posterior means among the
339 different runs. You can visually inspect the posterior means reported in the `out1.txt`
340 and `out2.txt` files (fig. 8), although this may be cumbersome. Figure 8a shows a plot,
341 made with R, of posterior times for run 1 vs. those from run 2. You can see that the
342 points fall almost perfectly on the $y = x$ line, indicating that both runs have converged to
343 the same distribution (hopefully the posterior!).

344

345 Another useful statistic to be calculated is the effective sample size (ESS). This gives the
346 user an idea about whether an MCMC chain has been run long enough. Tracer calculates
347 ESS automatically for all parameters. Function `coda::effectiveSize` in R will do
348 the same. Figure 9 shows the posterior mean, ESS, posterior variance, and standard
349 error of posterior means calculated with R for run 1 of the MCMC. The longer the ESS
350 the better. As a rule of thumb, one should seek ESS larger than 1,000, although this may

not always be practical in phylogenetic analysis. Note in fig. 9 that some estimates have very low ESSs, while others have substantially higher ESSs. For example, t_{n11} has ESS = 76.1, while t_{n19} has ESS = 1261. Running the analysis again and increasing the total number of iterations (for example, by increasing `samplefreq` or `nsample`) will lead to higher ESS values for all parameters.

Let v be the posterior variance of a parameter. The standard error of the posterior mean of the parameter is $S.E. = \sqrt{(v/ESS)}$. This is why having large ESS is important: Large ESS lead to small S.E. and better estimates of the posterior mean. For example, for t_{n11} , the posterior mean is 116.8 Ma, with standard error 1.53 My (fig. 9). That is, we have estimated the mean accurately to within $2 \times 1.53 \text{ My} = 3.06 \text{ My}$. To reduce the S.E. by half, you need to increase the ESS four times. Note that independent MCMC runs can be combined into a single run. Thus, you may save time by running several MCMC chains in parallel for computationally expensive analyses, although care must be taken to ensure each chain has run long enough to exit the burn-in phase and explore the posterior appropriately.

Trace plots and histograms are useful to spot problems and check convergence. Figure 8b and 8c show trace plots for t_{n19} and t_{n11} respectively. The trace of t_{n19} , which has high ESS, looks like a “hairy caterpillar”. Compare it to the trace of t_{n11} , which has low ESS. Visual inspection of a trace plot usually gives a sense of whether the parameter has an adequate ESS without calculating it. Note that both traces are trendless, that is, the traces oscillate around a mean value (the posterior mean). If you see a persistent trend in the trace (such as an increase or a decrease), that most likely means the MCMC did not converge to the posterior and needs a longer burn-in period.

376

377 Figure 8d shows the smoothed histograms (calculated using `density` in R) for `t_n11`
378 for the two runs. Notice that the two histograms are slightly different. As the ESS
379 becomes larger, histograms for different runs will converge in shape until becoming
380 indistinguishable. If you see large discrepancies between histograms, that may indicate
381 serious problems with the MCMC, such as lack of convergence due to short burn-in, or
382 the MCMC getting stuck in different modes of a multi-modal posterior.

383 3.5 MCMC sampling from the prior

384 Note that fossil calibrations (such as those of table 1) are represented as statistical
385 distributions of node ages. MCMCTree uses these distributions to construct the prior on
386 times. However, the resulting time prior used by the program may be substantially
387 different from the original fossil calibrations, because the program applies a truncation
388 so that daughter nodes are younger than their ancestors (**14, 27**). Thus, it is advisable
389 to calculate the time prior explicitly by running the MCMC with no data so that it can be
390 examined and compared with the fossil calibrations and the posterior.

391

392 Go to the `prior` directory and type

393

394 `$ mcmctree mcmctree-pr.ctl`

395

396 This will start the MCMC sampling from the prior. File `mcmctree-pr.ctl` is identical
397 to `mcmc/mcmctree.ctl` except that option `usedata` has been set to 0. Sampling from
398 the prior is much quicker because the likelihood does not need to be calculated. It takes
399 about 1min on the Intel Core i7 for MCMCTree to complete the analysis. Rename files
400 `mcmc.txt` and `out.txt` to `mcmc1.txt` and `out1.txt` and run the analysis again.

401 Rename the new files as appropriate. Check for convergence by calculating the ESS and
402 plotting the traces and histograms.

403

404 Figure 10 shows the prior densities of node ages obtained by MCMC sampling (shown in
405 grey) vs. the posterior densities (shown in black). Notice that for four nodes t_{n19} ,
406 t_{n18} , t_{n17} , and t_{n16} , the posterior times “agree” with the prior, that is, the
407 posterior density is contained within the prior density. For nodes t_{n15} , t_{n13} , and
408 t_{n11} , there is some conflict between the prior and posterior densities. However, for
409 nodes t_{n14} and t_{n12} , there is substantial conflict between the prior and the
410 posterior. In both cases the molecular data (together with the clock model) suggest the
411 node age is much older than that implied by the calibrations. This highlights the
412 problems in construction of fossil calibrations.

413

414 Each fossil calibration represents the palaeontologist’s best guess about the age of a
415 node. For example, the calibration for the human-chimp ancestor is
416 $B(0.075, 0.10, 0.01, 0.20)$ thus, the calibration is a uniform distribution between 7.5 and
417 10 million years ago (Ma). The bounds of the calibration are soft, that is, there is a set
418 probability that the bound is violated. In this case the probabilities are 1% for the
419 minimum bound and 20% for the maximum bound. The bound probabilities are
420 asymmetrical because they reflect the nature of the fossil information. Minimum
421 bounds are usually set with confidence because they are based on the age of the oldest
422 fossil member of a clade. For example, the minimum of 7.5 Ma is based on the age of
423 †*Sahelanthropus tchadensis*, recognised as the oldest fossil within the human lineage
424 **(28)**. On the other hand, establishing maximum bounds is difficult, as absence of fossils
425 for certain clades cannot be interpreted as evidence that the clade in question did not

exist during a particular geological time **(29)**. Our maximum here of 10 Ma represents the palaeontologist's informed guess about the likely oldest age of the clade; however, a large probability of 20% is given to allow for the fact that the node age could be older. The conflict between the prior and posterior seen in fig. 10 evidence this.

Note that when constructing the time prior, the Bayesian dating software must respect the constraints whereby daughter nodes must be younger than their parents. This means that calibration densities are truncated to accommodate the constraint, with the result that the actual prior used on node ages can be substantially different to the calibration density used (see Exercise 5.4). Detailed analyses of the interactions between fossil calibrations and the time prior and the effect of truncation are given in **(14, 27)**.

4. General recommendations for Bayesian clock dating

Extensive reviews of best practice in Bayesian clock dating are given elsewhere **(4, 20, 21, 30, 31)**. Here we give a few brief recommendations.

4.1 Taxon sampling, data partitioning, and estimation of tree topology

In this tutorial we used a small phylogeny to illustrate Bayesian time estimation using approximate likelihood calculation. In practical data analysis, it may be desirable to analyse much larger phylogenies (see Exercise 5.5). In large phylogenies, there may be uncertainties in the relationships of some groups. The approximate method discussed here can only be applied to a fixed (known) tree topology. If the uncertainties in the tree are few so that just a handful of tree topologies appear reasonable, the approximate method can be used by analysing each topology separately **(23, 32)**. This involves

estimating **g** and **H** for each topology, and then running separate MCMC chains on each topology to estimate the times. Several methods to co-estimate divergence times and tree topology are available (**8, 9, 17, 18**), although they do not implement the approximate likelihood method and are thus unsuitable for the analysis of genome-scale datasets.

We note that partitioning of sites in genomic datasets may have important effects on divergence time estimation. The infinite-sites theory (**13, 33**) studies the asymptotic behaviour of the posterior distribution of times when the amount of molecular data (measured by the number of partitions and the number of sites per partition) increases in a relaxed clock dating analysis. This theory shows that increasing the number of sites per partition will have minimal effects on time estimation when the sequences per partition are moderately long (>1,000 sites, say), but the precision improves when the number of partitions increases, eventually approximating a limit when the number of partitions is infinite. The theory also predicts that very different time estimates may be obtained if the same genomic sequence alignment is analysed as one partition or as multiple partitions (**34**). Furthermore, while more partitions tend to produce more precise time estimates, with narrow CIs, they may not necessarily be more reliable, depending on the correctness of the fossil calibrations and the appropriateness of the partitioning strategies. Unfortunately it is hard to decide on a good partitioning strategy given the genome-scale sequence data, despite efforts to design automatic partitioning strategies for phylogenetic analysis and divergence time estimation (**34–36**).

Commonly used approaches partition sites in the alignment by codon position, or by protein-coding genes of different relative rates (**23**). We recommend the use of the infinite-sites plot (**14**), in which uncertainty in divergence time estimates (measured as

the CI width) is plotted against the posterior mean of times. If the scatter points fall on a straight line, information due to the molecular sequence data has reached saturation and uncertainty in time estimate is predominantly due to uncertainties in fossil calibrations.

4.2 Selection of fossil calibrations

Fossil calibrations are one of the most important pieces of information needed to perform divergence time estimation, and thus should be chosen after careful consideration of the fossil record, although this may involve some subjectivity (29). Parham et al. (30) discuss best practice for construction of fossil calibrations. For example, minimum bounds on node ages are normally set to be the age of the oldest fossil member of the crown group. A small probability (say 2.5%) should be set for the probability that the node age violates the minimum bound (for example, to guard against misidentified or incorrectly dated fossils). Specifying maximum bounds is more difficult, as absence of fossils for a given geological period is not evidence that the clade in question was absent during the period (31). Current practice is to set the maximum bound to a reasonable value according to the expertise of the palaeontologist (see Benton et al. for examples), although a large probability (say 10% or even 20%) may be required to guard against badly specified maximum bounds. Calibration densities based on statistical modelling of species diversification, fossil preservation and discovery are also possible (15). In so-called tip-dating approaches, fossil species are included as taxa in the analysis (which may or may not include morphological information for the fossil and extant taxa) (37–39). Thus, in tip-dating, explicit specification of a fossil calibration density for a node age is not necessary.

4.3 Construction of the time prior

The birth-death process with species sampling was used here to construct the time prior for nodes in the phylogeny for which fossil calibrations are not available. Varying the birth μ , death λ , and sampling ρ , parameters can result in substantially different time priors. For example, using $\mu = \lambda = 1$ and $\rho = 0$, leads to a uniform distribution prior on node ages. This diffuse prior appears appropriate for most analysis. Varying the values of μ , λ , and ρ , is useful to assess whether the time estimates are robust to the time prior. Parameters configurations can be set up to generate time densities that result in young node ages, or in very old node ages (see p. 381 in (20) for examples).

4.4 Selection of the clock model

In analysis of closely related species (such as the apes), the clock assumption appears to be appropriate for time estimation. A likelihood-ratio test can be used to determine whether the strict clock is appropriate for a given dataset (40). If the clock is rejected, then Bayesian molecular-clock dating should proceed using one of the various relaxed-clock models available (7, 13). In this case, Bayesian model selection may be used to choose the most appropriate relaxed-clock model (41), although the method is computationally expensive and thus only applicable to small datasets. Use of different relaxed-clock models (such as the autocorrelated vs. the independent log-normally distributed rates), may result in substantially different time estimates (see (32) for an example). In such cases, repeating the analysis under the different clock models may be desirable.

5. Exercises

5.1 Autocorrelated rates model

Modify file `mcmc/mcmctree.ct1` and set `clock = 3`. This activates the autocorrelated lognormal rates model, also known as the geometric Brownian motion rates model **(6, 13)**. Run the MCMC twice and check for convergence. Compare the posterior times obtained with those obtained under the independent lognormal model (`clock = 2`). Are there any systematic differences in node age estimates between the two analysis? Which clock model produces the most precise (i.e. narrower CIs) divergence time estimates?

5.2 MCMC sampling with exact likelihood calculation

Modify file `mcmc/mcmctree.ct1` and set `clock = 2` (independent rates), `usedata = 1` (exact likelihood), `burnin = 200`, `sampfreq = 2`, and `nsample = 500`. These last three options will lead to a much shorter MCMC chain, with a total of 1,200 iterations. Run the MCMC sampling twice and check for convergence using the ESS, histograms and trace plots. How long does it take for the sampling to complete? Can you estimate how long would it take to run the analysis using 2,020,000 iterations, as long as for the approximate method of section 3.3.2? Did the two chains converge despite the low number of iterations?

5.3 Change of fossil calibrations

There is some controversy over whether †*Sahelanthropus*, used to set the minimum bound for the human-chimp divergence, is indeed part of the human lineage. The next (younger) fossil in the human lineage is †*Orrorin* which dates to around 6 Ma. Modify file `data/10s.tree` and change the calibration in the human-chimp node to B(0.057,

0.10, 0.01, 0.2). Also change the calibration on the root node to $B(0.615, 1.315, 0.01, 0.05)$. Run the MCMC analysis with the approximate method and again sampling from the prior. Are there any substantial differences in the posterior distributions of times under the new fossil calibrations? Which nodes are affected? How bad is the truncation effect among the calibration densities and the prior?

5.4 Comparing calibration densities and prior densities

This is a difficult exercise. Use R to plot the prior densities of times sampled using MCMC (the same as in fig. 10). Now try to work out how to overlay the calibration densities onto the plots. For example, see fig. 3 in **(23)** for an idea. First, write functions that calculate the calibration densities. The `dunif` function in R is useful to plot uniform calibrations. Functions `sn::dsn` and `sn::dst` (in the SN package) are useful to plot the skew-t (ST) and skew normal (SN) distributions. Calibration type S2N (table 1) is a mixture of two skew-normal distributions **(15)**. How do the sampled priors compare to the calibration densities? Are there any substantial truncation effects?

5.5 Time estimation in a super-matrix of 330 species

Good taxon sampling is critical to obtaining robust estimates of divergence times for clades. In the `data/` directory an alignment of the 1st and 2nd codon positions from mitochondrial protein-coding genes from 330 species (326 primate and 4 outgroup species) is provided, `330s.phys`, with corresponding tree topology, `330s.tree`. First, place the fossil calibrations of Table 1 on the appropriate nodes of the species tree. Then obtain the gradient and Hessian matrix for the 330-species alignment using the HKY+G model. Finally, estimate the divergence times on the 330-species phylogeny by using the approximate likelihood method. How does taxon sampling affect node age estimates when comparing the 10-species and 330-species trees? How does uncertainty

in node ages in the large tree, which was estimated on a short alignment, compare with the estimates on the small tree, but with a large alignment?

6. References

1. E. ZUCKERKANDL and L. PAULING (1965) Evolutionary Divergence and Convergence in Proteins, In: *Evolving Genes and Proteins*, pp. 97–166.
2. S. Kumar (2005) Molecular clocks: four decades of evolution., *Nature reviews. Genetics.* 6, 654–662.
3. L. Bromham and D. Penny (2003) The modern molecular clock, *Nature Reviews Genetics.* 4, 216–224.
4. M. dos Reis, P.C.J. Donoghue, and Z. Yang (2015) Bayesian molecular clock dating of species divergences in the genomics era, *Nature Reviews Genetics.* 17, 71–80.
5. P.C.J. Donoghue and Z. Yang (2016) The evolution of methods for establishing evolutionary timescales, *Philosophical Transactions of the Royal Society B: Biological Sciences.* 371, 20160020.
6. J.L. Thorne, H. Kishino, and I.S. Painter (1998) Estimating the rate of evolution of the rate of molecular evolution, *Molecular Biology and Evolution.* 15, 1647–1657.
7. A.J. Drummond, S.Y.W. Ho, M.J. Phillips, et al. (2006) Relaxed phylogenetics and dating with confidence, *PLoS Biology.* 4, 699–710.
8. F. Ronquist, M. Teslenko, P. Van Der Mark, et al. (2012) Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space, *Systematic Biology.* 61, 539–542.
9. N. Lartillot, T. Lepage, and S. Blanquart (2009) PhyloBayes 3: A Bayesian software package for phylogenetic reconstruction and molecular dating, *Bioinformatics.* 25, 2286–2288.
10. T.A. Heath, M.T. Holder, and J.P. Huelsenbeck (2012) A dirichlet process prior for estimating lineage-specific substitution rates, *Molecular Biology and Evolution.* 29, 939–955.
11. H. Kishino, J.L. Thorne, and W.J. Bruno (2001) Performance of a divergence time estimation method under a probabilistic model of rate evolution., *Molecular Biology and Evolution.* 18, 352–61.
12. Z. Yang and B. Rannala (2006) Bayesian estimation of species divergence times under a molecular clock using multiple fossil calibrations with soft bounds, *Molecular Biology and Evolution.* 23, 212–226.
13. B. Rannala and Z. Yang (2007) Inferring speciation times under an episodic molecular clock, *Systematic Biology.* 56, 453–466.
14. J. Inoue, P.C.J. Donoghue, and Z. Yang (2010) The impact of the representation of fossil calibrations on bayesian estimation of species divergence times, *Systematic Biology.* 59, 74–89.
15. R.D. Wilkinson, M.E. Steiper, C. Soligo, et al. (2011) Dating primate divergences through an integrated analysis of palaeontological and molecular data, *Systematic Biology.* 60, 16–31.
16. M. Dos Reis and Z. Yang (2011) Approximate Likelihood Calculation on a Phylogeny for Bayesian Estimation of Divergence Times., *Molecular biology and evolution.*
17. R. Bouckaert, J. Heled, D. Kühnert, et al. (2014) BEAST 2: A Software Platform for

- Bayesian Evolutionary Analysis, PLoS Computational Biology. 10,
18. S. Höhna, M.J. Landis, T.A. Heath, et al. (2016) RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language., *Systematic Biology*. 65, 726–736.
 19. M. Dos Reis, T. Zhu, and Z. Yang (2014) The impact of the rate prior on Bayesian estimation of divergence times with multiple Loci., *Systematic biology*. 63, 555–65.
 20. Z. Yang (2014) *Molecular Evolution: A Statistical Approach*, Oxford University Press, Oxford.
 21. T. a Heath and B.R. Moore (2014) Bayesian inference of species divergence times, In: Chen, M.-H., Kuo, L., and Lewis, P.O. (eds.) *Bayesian Phylogenetics: Methods, Algorithms, and Applications*, pp. 277–318 CRC Press.
 22. Z. Yang (2007) PAML 4: Phylogenetic analysis by maximum likelihood, *Molecular Biology and Evolution*. 24, 1586–1591.
 23. M. dos Reis, J. Inoue, M. Hasegawa, et al. (2012) Phylogenomic datasets provide both precision and accuracy in estimating the timescale of placental mammal phylogeny., *Proceedings. Biological sciences / The Royal Society*. 279, 3491–500.
 24. M. dos Reis, G. Gunnell, J. Barba-Montoya, et al. (2017) Using phylogenomic data to explore the effects of relaxed clocks and calibration strategies on divergence time estimation: Primates as a test case, *bioRxiv*. 201327.
 25. Z. Yang (1996), Among-site rate variation and its impact on phylogenetic analyses.
 26. J.H. Gillespie (1984) The molecular clock may be an episodic clock., *Proceedings of the National Academy of Sciences of the United States of America*. 81, 8009–13.
 27. R.C.M. Warnock, Z. Yang, and P.C.J. Donoghue (2012) Exploring uncertainty in the calibration of the molecular clock, *Biology Letters*. 8, 156–159.
 28. C.P.E. Zollikofer, M.S. Ponce de León, D.E. Lieberman, et al. (2005) Virtual cranial reconstruction of *Sahelanthropus tchadensis*, *Nature*. 434, 755–759.
 29. M.J. Benton and P.C.J. Donoghue (2007), *Paleontological evidence to date the tree of life*.
 30. J.F. Parham, P.C.J. Donoghue, C.J. Bell, et al. (2012), *Best practices for justifying fossil calibrations*.
 31. S.Y.W. Ho and M.J. Phillips (2009) Accounting for calibration uncertainty in phylogenetic estimation of evolutionary divergence times, *Systematic Biology*. 58, 367–380.
 32. M. Dos Reis, Y. Thawornwattana, K. Angelis, et al. (2015) Uncertainty in the Timing of Origin of Animals and the Limits of Precision in Molecular Timescales, *Current Biology*. 25, 2939–2950.
 33. T. Zhu, M. Dos Reis, and Z. Yang (2014) Characterization of the uncertainty of divergence time estimation under relaxed molecular clock models using multiple loci., *Systematic biology*. syu109-.
 34. K. Angelis, S. Alvarez-Carretero, M. dos Reis, et al. (2017) An Evaluation of Different Partitioning Strategies for Bayesian Estimation of Species Divergence Times, *Systematic Biology*.
 35. R. Lanfear, B. Calcott, S.Y.W. Ho, et al. (2012) PartitionFinder: Combined selection of partitioning schemes and substitution models for phylogenetic analyses, *Molecular Biology and Evolution*. 29, 1695–1701.
 36. S. Duchêne, M. Molak, and S.Y.W. Ho (2014) ClockstaR: choosing the number of relaxed-clock models in molecular phylogenetic analysis, *Bioinformatics*. 30, 1017–1019.

- 660 37. T.A. Heath, J.P. Huelsenbeck, and T. Stadler (2014) The fossilized birth-death
661 process for coherent calibration of divergence-time estimates, *Proceedings of the*
662 *National Academy of Sciences*. 111, E2957–E2966.
- 663 38. F. Ronquist, S. Klopstein, L. Vilhelmsen, et al. (2012) A total-evidence approach to
664 dating with fossils, applied to the early radiation of the hymenoptera, *Systematic*
665 *Biology*. 61, 973–999.
- 666 39. J.E. O'Reilly, M. dos Reis, and P.C.J. Donoghue (2015), *Dating Tips for Divergence-*
667 *Time Estimation*.
- 668 40. J. Felsenstein (1981) Evolutionary trees from DNA sequences: a maximum
669 likelihood approach., *Journal of molecular evolution*. 17, 368–76.
- 670 41. T. Lepage, D. Bryant, H. Philippe, et al. (2007) A general comparison of relaxed
671 molecular clock models, *Molecular Biology and Evolution*. 24, 2669–2680.
672
673

Table 1. List of fossil calibrations used in this tutorial.

Node ^a	Crown Group	MCMCTree Calibration ^b
19	Chimp-Human	B(0.075, 0.10, 0.01, 0.20)
18	Gorilla-Human	B(0.10, 0.132, 0.01, 0.20)
17	Hominidae	B(0.112, 0.28, 0.01, 0.10)
16	Catarrhini	B(0.25, 0.29, 0.01, 0.10)
15	Anthropoidea	ST(0.4754, 0.0632, 0.98, 22.85)
13	Strepsirrhini	B(0.38, 0.58, 0.01, 0.10)
12	Primates	S2N(0.698, 0.65, 0.0365, -3400, 0.650, 0.138, 11409)
11	Euarchonta	G(36, 36.9)

a. Node numbers as in figure 1.

b. $B(a, b, p_L, p_U)$ means the calibration is a uniform distribution between a and b , with probabilities p_L and p_U that the true node age is outside the calibration bounds.

ST(location, scale, shape, df) means the calibration is a skew-t distribution. S2N(p , location1, scale1, shape1, location2, scale2, shape2) means the calibration is a $p : 1 - p$ mixture of two skew-normal distributions. $G(\alpha, \beta)$ means the calibration is a gamma distribution with shape α and rate β . See MCMCTree's manual for the full details on fossil calibration formats. The calibrations are from the Primate analysis in (24).

Figure Legends

Figure 1. The tree of 10 species. Nodes with fossil calibrations are indicated with black dots (see Table 1 for calibration densities). Internal nodes are numbered from 11 to 19 according to the nomenclature used by MCMCTree.

Figure 2. The gH/mcmctree-outBV.ct1 file, with appropriate options to set up calculation of the gradient and Hessian matrix for the approximate likelihood method.

Figure 3. The gH/out.BV file produced by BASEML. The first line has the number of species (10), the second line has the tree topology with MLEs of branch lengths, and the MLEs of branch lengths are given again in the third line. The fourth line contains the gradient, **g**, followed by the Hessian, **H**, for partition 1. This file will be renamed in.BV and placed into the mcmc/ directory to carry out MCMC sampling using the approximate likelihood method.

697

698 **Figure 4.** The `mcmc/mcmctree.ct1` file necessary to sample from the posterior
699 distribution using the approximate likelihood method.

700

701 **Figure 5.** Screenshot of MCMCTree's output during MCMC sampling of the posterior.
702 Different runs of the program will give slightly different output values.

703

704 **Figure 6.** The end of the `mcmc/out.txt` file produced by MCMCTree at the end of the
705 MCMC sampling of the posterior.

706

707 **Figure 7.** The dated primate phylogeny with error bars (representing 95% CIs of node
708 ages), drawn with FigTree. The time unit is 1 My.

709

710 **Figure 8.** Convergence diagnostic plots of the MCMC drawn with R (see
711 `R/analysis.R`).

712

713 **Figure 9.** Calculations of posterior mean, ESS, posterior variance, and standard error of
714 the posterior mean in R (see `R/analysis.R`).

715

716 **Figure 10.** Prior (grey) and posterior (black) density plots of node ages plotted with R
717 (see `R/analysis.R`).

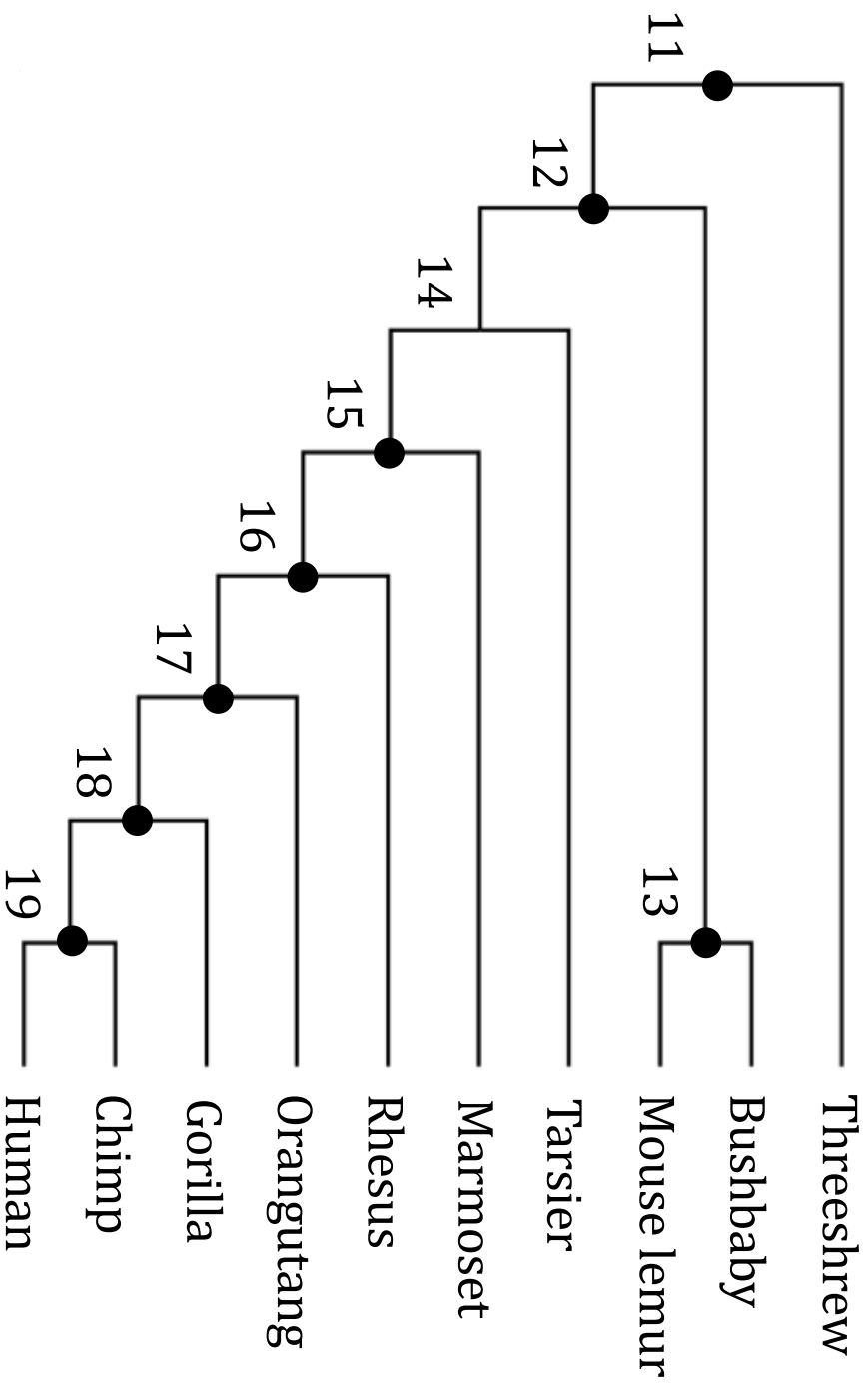


Figure 1

```
seqfile = ../data/10s.phys
treefile = ../data/10s.tree

ndata = 2
seqtype = 0      * 0: nucleotides; 1:codons; 2:AAs
usedata = 3      * 0: no data (prior); 1:exact likelihood;
                  * 2: approximate likelihood; 3:out.BV (in.BV)
clock = 2        * 1: global clock; 2: independent rates; 3: correlated rates

model = 4        * 0:JC69, 1:K80, 2:F81, 3:F84, 4:HKY85
alpha = 0.5      * alpha for gamma rates at sites
ncatG = 5        * No. categories in discrete gamma

cleandata = 0    * remove sites with ambiguity data (1:yes, 0:no)?
```

Figure 2

10

```
((Bushbaby: 0.029523, Mouse_lemur: 0.019653): 0.006547, (Tarsier: 0.030897, Marmoset: 0.0006547 0.029523 0.019653 0.002123 0.030897 0.011754 0.015183 0.003426 0.008716 -2.114230 -2.618861 21.299836 31.765175 20.801006 -3.019251 -14.909946 8.188538 -3.70464
```

Hessian

-2.033e+08	-2.59e+06	-9.717e+06	-4.363e+07	1.799e+06	-5.457e+06	2.055e+06	-1.29e+04
-2.59e+06	-5.71e+07	2.235e+06	1.475e+06	3.315e+06	1.651e+06	3.436e+06	2.134e+06
-9.717e+06	2.235e+06	-8.733e+07	-2.954e+06	2.79e+06	7.275e+05	3.371e+06	1.512e+06
-4.363e+07	1.475e+06	-2.954e+06	-4.622e+08	-5.059e+06	-2.658e+07	3.701e+06	-5.157e+06
1.799e+06	3.315e+06	2.79e+06	-5.059e+06	-5.473e+07	7.951e+05	3.437e+06	2.28e+06
-5.457e+06	1.651e+06	7.275e+05	-2.658e+07	7.951e+05	-1.403e+08	3.724e+06	-1.163e+07
2.055e+06	3.436e+06	3.371e+06	3.701e+06	3.437e+06	3.724e+06	-1.25e+08	-1.69e+07
-1.29e+04	2.134e+06	1.512e+06	-5.157e+06	2.28e+06	-1.163e+07	-1.69e+07	-4.756e+08
3.483e+06	4.548e+06	4.413e+06	-1.406e+05	4.463e+06	2.246e+06	1.979e+06	1.698e+06
8.344e+05	2.861e+06	2.023e+06	1.605e+06	2.021e+06	-5.676e+05	-8.424e+05	-1.722e+07
3.625e+06	4.671e+06	4.894e+06	8.939e+05	4.775e+06	2.595e+06	1.699e+06	5.407e+05
2.701e+06	3.036e+06	2.394e+06	1.777e+06	3.175e+06	6.217e+05	-5.952e+05	-4.592e+06

Figure 3


```

seed = -1
seqfile = ../data/10s.phys
treefile = ../data/10s.tree
mcmcfile = mcmc.txt
outfile = out.txt

ndata = 2
seqtype = 0      * 0: nucleotides; 1:codons; 2:AAs
usedata = 2      * 0: no data (prior); 1:exact likelihood;
                  * 2:approximate likelihood; 3:out.BV (in.BV)
clock = 2        * 1: global clock; 2: independent rates; 3: correlated rates
RootAge = '<1.0'  * safe constraint on root age, used if no fossil for root.

model = 4        * 0:JC69, 1:K80, 2:F81, 3:F84, 4:HKY85
alpha = 0.5      * alpha for gamma rates at sites
ncatG = 5        * No. categories in discrete gamma

cleandata = 0    * remove sites with ambiguity data (1:yes, 0:no)?

BDparas = 1 1 0  * birth, death, sampling
kappa_gamma = 6 2 * gamma prior for kappa
alpha_gamma = 1 1 * gamma prior for alpha

rgene_gamma = 2 40 1 * gammaDir prior for rate for genes
sigma2_gamma = 1 10 1 * gammaDir prior for sigma^2 (for clock=2 or 3)

print = 1      * 0: no mcmc sample; 1: everything except branch rates 2: everything
burnin = 20000
sampfreq = 100
nsample = 20000

```

Figure 4

```

0% 0.26 0.39 0.23 0.39 0.28 1.285 1.243 0.588 1.158 0.541 0.321 - 0.192 0.197 -16.9 0:02
(nsteps = 50)
Current Pjump:      0.26200 0.39475 0.23175 0.38650 0.28000 0.27550 0.39200 0.43750
0.40100 0.29725 0.33725 0.27525 0.32275 0.23475 0.23150 0.29875 0.31600 0.27800
0.25300 0.29975 0.29650 0.32575 0.27500 0.61150 0.29850 0.31225 0.35400 0.23200
0.30800 0.28250 0.33050 0.21325 0.22700 0.25900 0.26725 0.26900 0.33150 0.23725
0.31000 0.20700 0.24225 0.61625 0.30675 0.30150 0.32000 0.21975 0.27650 0.22500
0.36650 0.00000
Current finetune:  0.00365 0.00166 0.00586 0.00182 0.00503 0.00697 0.00486 0.00500
0.00835 0.24230 0.21346 0.71942 0.65595 0.01093 0.01230 0.01256 0.00960 0.01492
0.02008 0.02466 0.03547 0.03942 0.04624 0.17077 0.02425 0.04971 0.01513 0.03626
0.03661 0.04475 0.08082 0.00867 0.00949 0.01146 0.00861 0.01133 0.01263 0.02252
0.02728 0.03996 0.03790 0.14736 0.02025 0.04584 0.01209 0.02975 0.02776 0.03389
0.05173 0.00000
New      finetune:  0.00313 0.00232 0.00438 0.00248 0.00465 0.00632 0.00675 0.00806
0.01194 0.23972 0.24532 0.65158 0.71499 0.00829 0.00918 0.01250 0.01020 0.01367
0.01654 0.02463 0.03499 0.04345 0.04183 0.47928 0.02411 0.05210 0.01846 0.02714
0.03776 0.04175 0.09064 0.00592 0.00694 0.00969 0.00755 0.01000 0.01422 0.01728
0.02835 0.02644 0.02976 0.42023 0.02079 0.04611 0.01305 0.02100 0.02527 0.02454
0.06589 0.00000

5% 0.34 0.30 0.31 0.32 0.28 1.163 0.981 0.622 0.893 0.464 0.295 - 0.129 0.154 -17.0 0:08
10% 0.35 0.30 0.31 0.32 0.27 1.189 0.943 0.607 0.859 0.457 0.293 - 0.128 0.153 -17.0 0:15
15% 0.36 0.30 0.30 0.31 0.27 1.156 0.920 0.604 0.837 0.457 0.290 - 0.133 0.160 -17.0 0:22
20% 0.35 0.30 0.30 0.32 0.26 1.126 0.908 0.600 0.825 0.453 0.290 - 0.137 0.165 -17.0 0:29
25% 0.36 0.30 0.30 0.31 0.26 1.139 0.912 0.605 0.829 0.458 0.293 - 0.138 0.165 -17.0 0:37
30% 0.36 0.30 0.30 0.31 0.26 1.153 0.918 0.609 0.834 0.460 0.293 - 0.136 0.163 -17.0 0:43

```

Figure 5

```

ln Lmax (unconstrained) = -4636133.236961

Time used: 2:26

mean of parameters using all iterations
1.16785 0.91766 0.60797 0.83447 0.46464 0.29132 0.17725 0.10441 0.08519 0.

Species tree for FigTree. Branch lengths = posterior mean times; 95% CIs = labels
(1_Tree_shrew, ((2_Bushbaby, 3_Mouse_lemur) 13 , (4_Tarsier, (5_Marmoset, (6_Rhesus, (7_Orangut
(Tree_shrew: 1.167850, ((Bushbaby: 0.607966, Mouse_lemur: 0.607966): 0.309693, (Tarsier: 0.8344
(Tree_shrew: 1.167850, ((Bushbaby: 0.607966, Mouse_lemur: 0.607966) [x95%={0.50317, 0.735468}]):

Posterior mean (95% Equal-tail CI) (95% HPD CI) HPD-CI-width
t_n11 1.1679 (0.9235, 1.4423) (0.9021, 1.4056) 0.5035 (Jnode 18)
t_n12 0.9176 (0.8015, 1.0484) (0.7965, 1.0423) 0.2458 (Jnode 17)
t_n13 0.6080 (0.5032, 0.7355) (0.5019, 0.7337) 0.2318 (Jnode 16)
t_n14 0.8345 (0.7236, 0.9602) (0.7192, 0.9538) 0.2346 (Jnode 15)
t_n15 0.4646 (0.3966, 0.5340) (0.3964, 0.5335) 0.1371 (Jnode 14)
t_n16 0.2913 (0.2526, 0.3380) (0.2499, 0.3333) 0.0833 (Jnode 13)
t_n17 0.1773 (0.1466, 0.2174) (0.1439, 0.2132) 0.0692 (Jnode 12)
t_n18 0.1044 (0.0995, 0.1164) (0.0988, 0.1139) 0.0152 (Jnode 11)
t_n19 0.0852 (0.0758, 0.0981) (0.0746, 0.0958) 0.0212 (Jnode 10)
mu1 0.0269 (0.0221, 0.0334) (0.0217, 0.0328) 0.0111
mu2 0.1110 (0.0898, 0.1396) (0.0877, 0.1364) 0.0488
sigma2_1 0.1370 (0.0607, 0.2833) (0.0484, 0.2511) 0.2027
sigma2_2 0.1634 (0.0755, 0.3201) (0.0625, 0.2883) 0.2258
lnL -17.0026 (-25.9750, -9.8710) (-24.9110, -9.1170) 15.7940

```

Figure 6

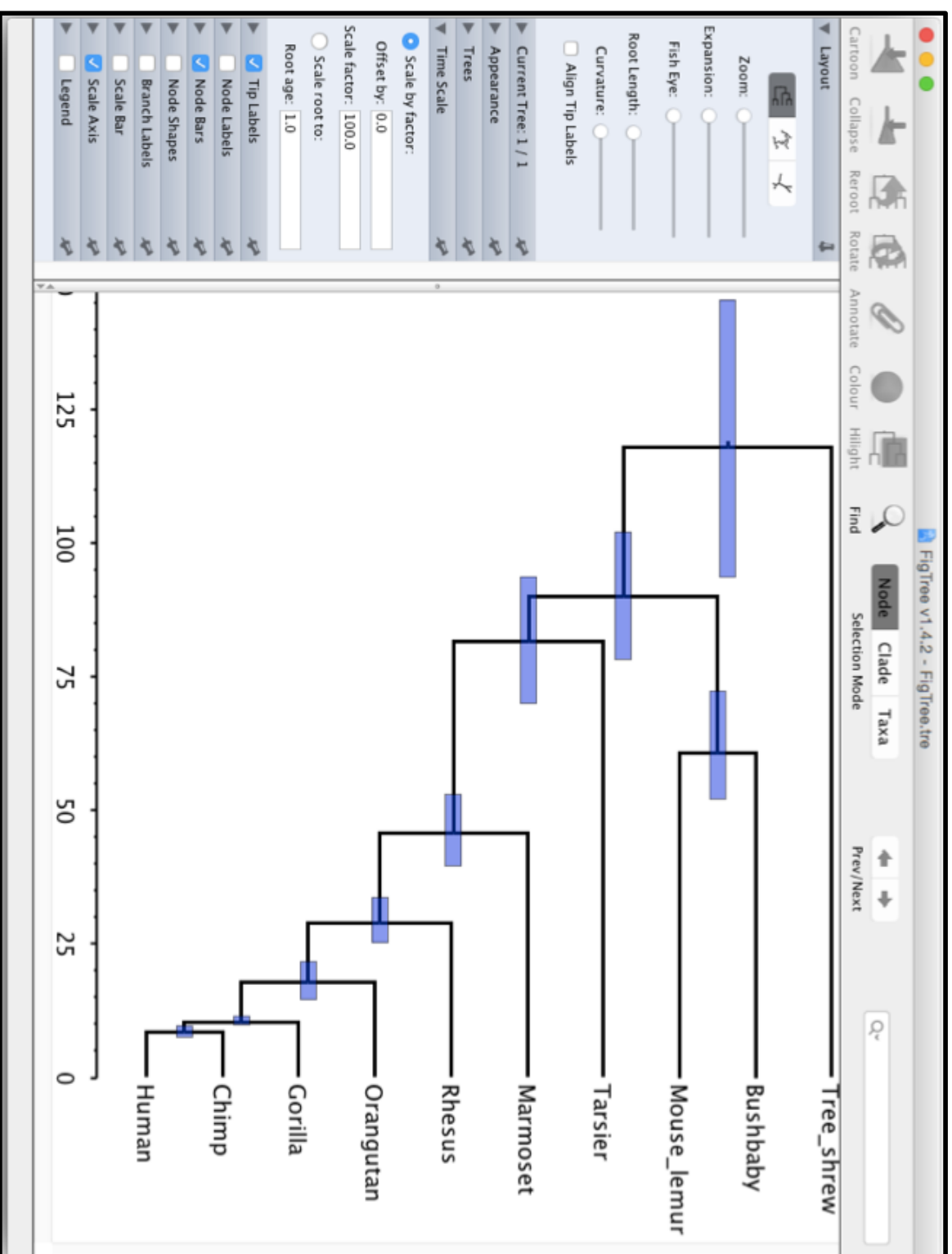


Figure 7

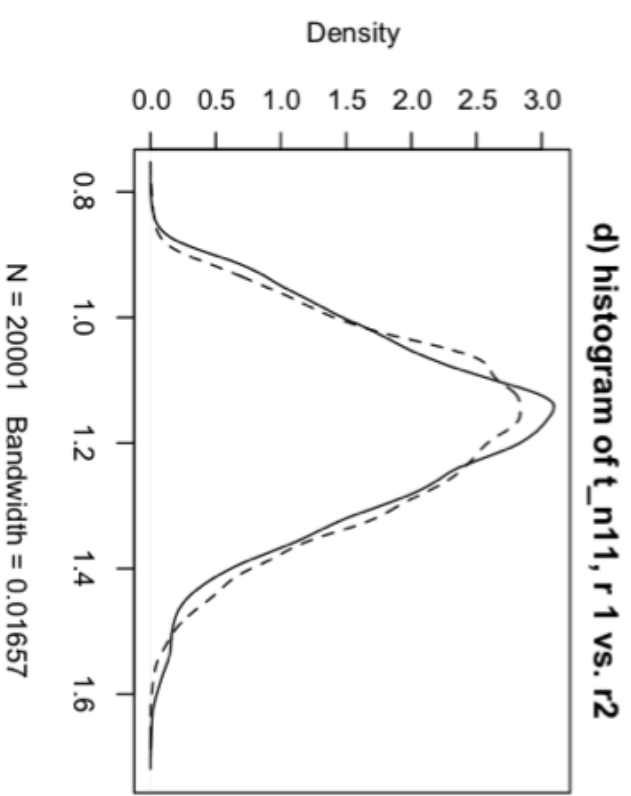
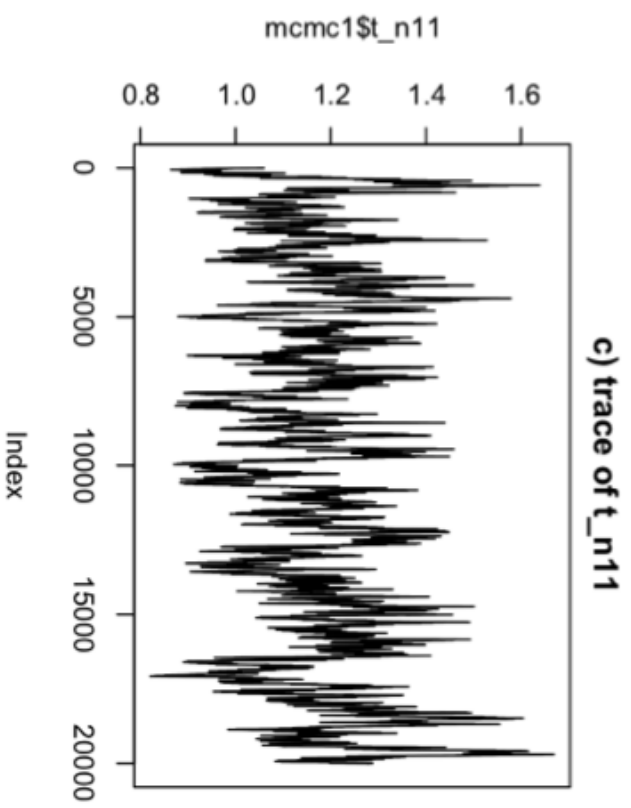
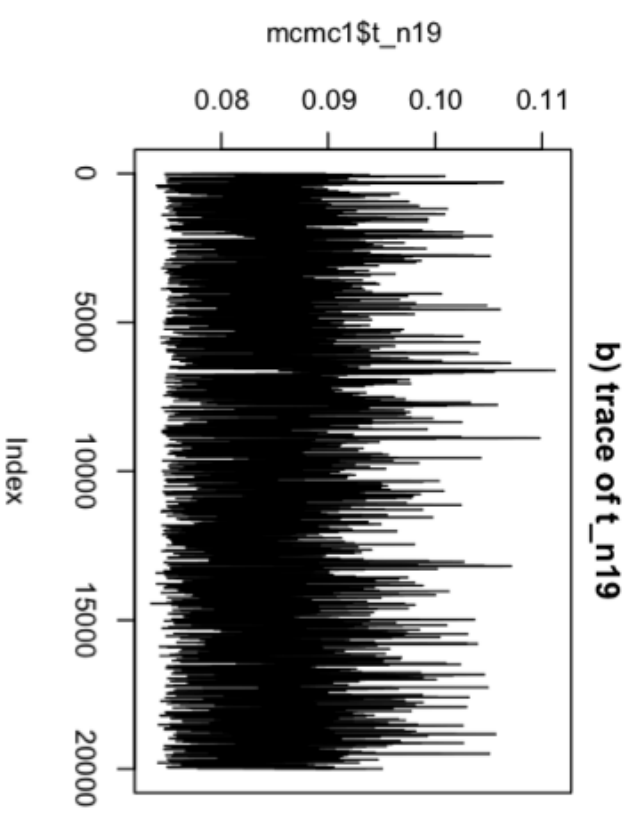
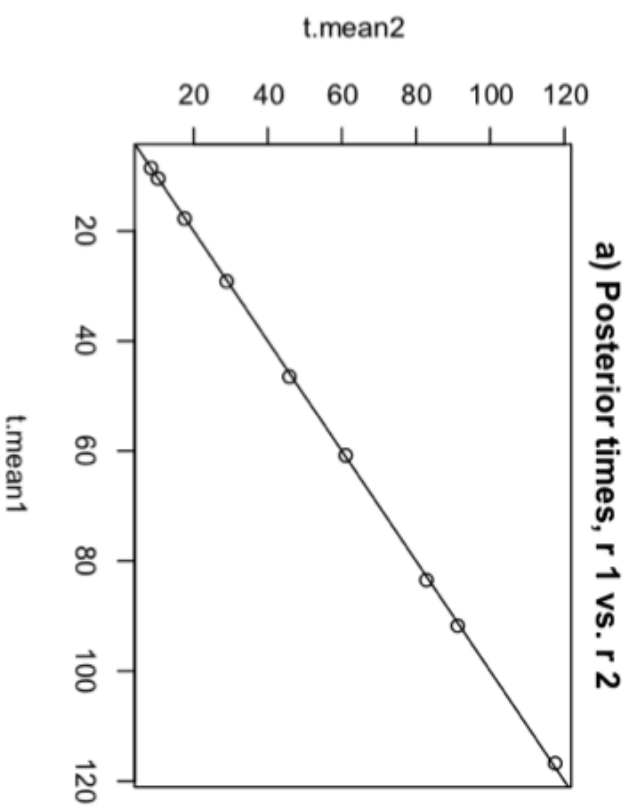


Figure 8

	mean.mcmc	ess.mcmc	var.mcmc	se.mcmc
t_n11	1.16785568	76.14030	1.779905e-02	0.0152894256
t_n12	0.91763459	66.38219	4.085525e-03	0.0078450940
t_n13	0.60801488	151.00623	3.123330e-03	0.0045479066
t_n14	0.83448247	71.93763	3.708967e-03	0.0071803969
t_n15	0.46464686	231.92350	1.211178e-03	0.0022852393
t_n16	0.29131271	353.25425	5.294412e-04	0.0012242361
t_n17	0.17726011	347.03816	3.245757e-04	0.0009670955
t_n18	0.10441651	1035.75332	2.080275e-05	0.0001417203
t_n19	0.08518922	1261.15128	3.363295e-05	0.0001633048
mu1	0.02691074	530.31981	8.464981e-06	0.0001263409
mu2	0.11103179	637.44606	1.577065e-04	0.0004973969
sigma2_1	0.13698819	710.07293	3.298175e-03	0.0021551891
sigma2_2	0.16337732	893.70775	4.046102e-03	0.0021277504
lnL	-17.00256482	20001.00000	1.696800e+01	0.0291265757

Figure 9

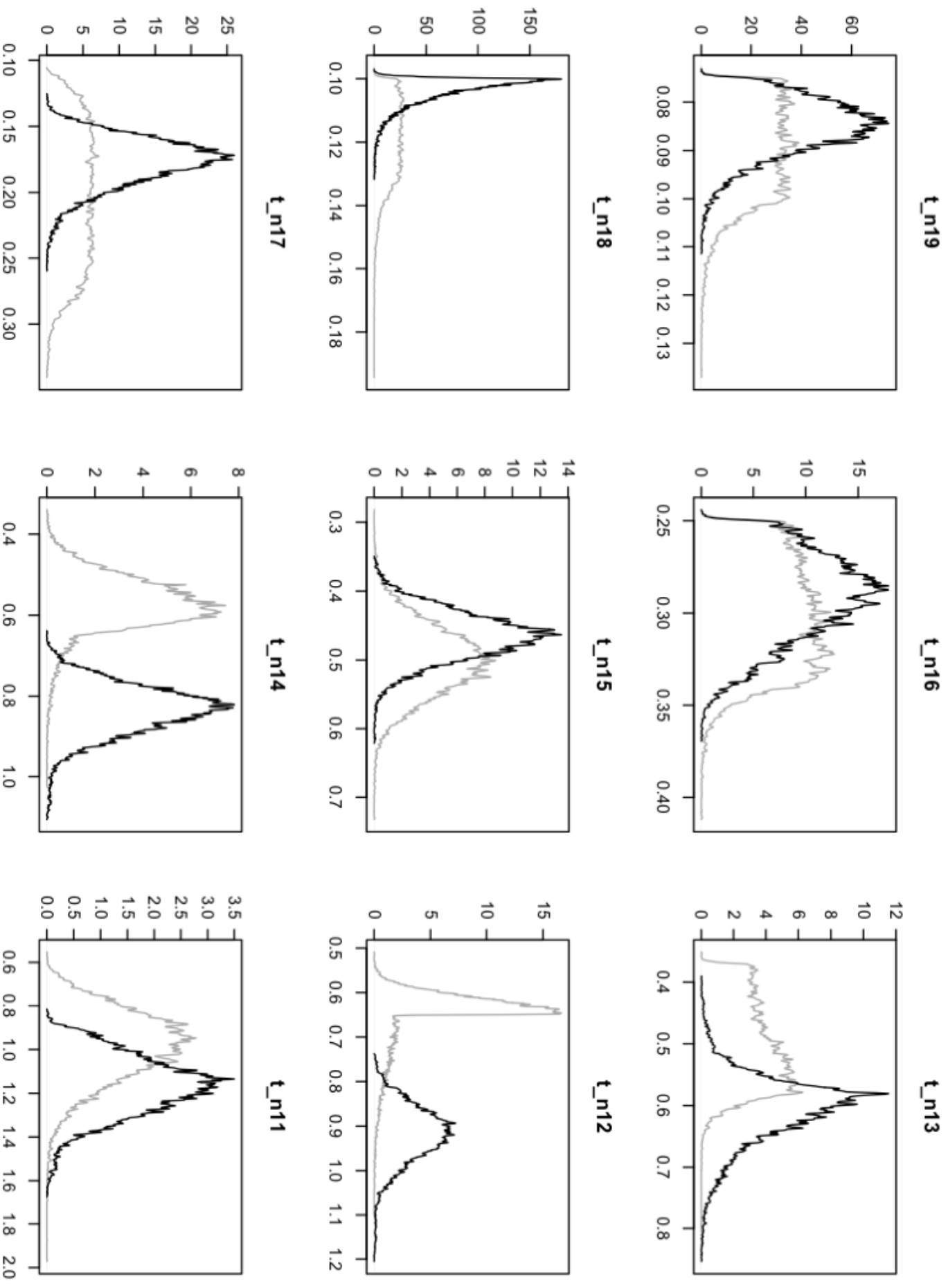


Figure 10