# Bayesian molecular dating using PAML/multidivtime

## A step-by-step manual, version 1.5 (July 2005)

**Software and text sources by Jeff Thorne, Hirohisa Kishino, and Ziheng Yang.**

This compilation by Frank Rutschmann.
Please send an e-mail to frank@plant.ch for any error reports, questions, or suggestions.

Many thanks to Jeff Thorne and Ziheng Yang for all their help.

You may cite this manual as follows:
"Rutschmann F. 2005. Bayesian molecular dating using PAML/multidivtime. A step-by-step manual. University of Zurich, Switzerland. Available at http://www.plant.ch". Needless to say that you should cite some of the original papers by Thorne, Kishino and Yang (see "Selected Literature").

## Table of contents



## A. Introduction

**What's described here?**
This step-by-step manual is nothing else than a compilation of the original manuals and readme files by Jeff Thorne and Ziheng Yang. It describes how to perform a Bayesian molecular dating using the software PAML and multidivtime. The manual assumes that you already have one or more DNA sequence matrices and a corresponding phylogenetic tree topology.

The Bayesian dating method implemented in multidivtime (Thorne et al. 1998; Kishino et al. 2001) uses a probabilistic model to describe the change in evolutionary rate over time and uses the Markov chain Monte Carlo (MCMC) procedure to derive the posterior distribution of rates and times. It allows simultaneous use of different substitution models for multiple data partitions as well as multiple calibration windows, and provides direct credibility intervals for estimated divergence times and substitution rates.

**What do you need?**
- a fast PC running Mac OS X, Linux, or Windows
- multidistribute package (Kishino and Thorne, http://statgen.ncsu.edu/thorne/multidivtime.html)
- PAML package (Ziheng Yang, http://abacus.gene.ucl.ac.uk/software/paml.html)
- a DNA sequence matrix and the best topology of a phylogenetic tree you can infer from the sequence data. Your data should include at least one outgroup taxon.

Additional documentation:
- The readme files distributed with the packages mentioned above

**Important:**
The multidivtime package exists in two slightly different versions. One is **the package described here**, composed by Jeff Thorne, available for download at http://statgen.ncsu.edu/thorne/multidivtime.html.

Another distribution has been put together by Ziheng Yang. It's available on ftp://abacus.gene.ucl.ac.uk/pub/T3. Ziheng calls his package the "Thornian Time Traveller" (T3). It contains basically the same, but provides additional example files from Anne Yoder's mouse lemur study (Yang and Yoder 2003), which can be useful to test the program and learn the data formats. Ziheng also added some print commands to the program files to better indicate the status/progress of the programs. His package also provides a small program called TreeTimeJeff.exe which helps to extract important information from the main multidivtime output file.
**Note:** Ziheng's program versions require different data formats then Jeff's originals! Ziheng's package will **not** be discussed here. However, if you prefer to use Ziheng's distribution, it comes with a useful documentation file (T3doc.txt).

**Details useful to know:**
All files have to be saved after editing with UNIX line breaks under Mac OS X or Linux, and with DOS line breaks under Windows. Most text editors allow you to specify this.

The "Gene1" part of the file names mentioned in this manual can be replaced by your own names. Other file names, such as testseq, hmmcntrl.dat or multicntrl.dat are strictly required by the software! An asterisk (*) in a file name represents any number you can give to a partition/gene (only relevant if you want to analyze multiple partitions/genes).

The duration times given in this manual [in squared brackets] have been achieved with a one gene, 42 taxa dataset, containing 1042 characters, analyzed on a Intel Pentium IV 2Ghz machine with 512 MB RAM. The MCMC parameters were: 10000 MC samples, 100 cycles, 100000 burnin cycles. If your dataset contains more taxa (and therefore more branches and nodes), steps 3, 5, and 7 may take much longer! Additionally, the duration of the last step is strongly dependent on the number of MC samples and cycles.

The following software versions have been used to write this step-by-step manual:
- baseml in PAML, version 3.14 (September 2004)
- estbranches, version 8/5/03
- multidivtime, version 9/25/03

# B. Molecular dating steps

## Step 1: Download and compile the program files

*Mac OS X and Linux:*

- Download the multidistribute archive. Extract it to any directory. Delete all files with the file extension .EXE (these are Windows executables). Compile estbranches.c, paml2modelinf.c, and multidivtime.c using a C compiler (see multidistribute readme file how to do that, plus the instructions in the estbranches.c header).
- Download the PAML archive. Extract it to another directory. Delete all files with the file extension .EXE (these are Windows executables). Compile baseml.c using a C compiler (see multidistribute readme file how to do that).
- For starting the programs described below, always add a "./" at the beginning of the commands.
- Sometimes, downloaded program files loose their executable attribute. You can change this by invoking the command `chmod u+x name_of_the_program`.

*Windows:*

- Download the multidistribute archive. Extract it to any directory. The executables estbranches_dna.exe, paml2modelinf.exe, and multidivtime.exe are ready to use.
- Download the paml archive. Extract it to another directory. The program baseml.exe is ready to use.

## Step 2: Prepare DNA sequences and tree topology

**Sequence data:**
You have to transform your sequence data into a special format (see example in Appendix, p. 13). Probably the easiest way for doing this is to open your sequence file in PAUP* and save it from there in the Phylip format (export format=phylip;). Then, open the file in any text editor. Substitute all N, -, gaps (:), or missing data by **?**. Short all taxon names to <= 10 characters and eliminate all spaces within names. Make sure that there are at least 2 spaces between the taxon manes and the beginning of the sequences. Finally, save the file as testseq.Gene1.

*Multiple genes/partitions:*
If you have more than one partition/gene, edit them separately, and save the data in separate files, named testseq.Gene2, testseq.Gene3 etc...

**Tree topology:**
Use the best tree topology you have, e.g. a MP tree topology without branch lenghts, an ML tree, or a MrBayes consensus tree. Root the tree with the outgroup and save it without branch lengths in the Phylip format. This can be best done by using the software TreeEdit (Rambaut and Charleston 2002). Then open your tree file in a text editor. Edit the file as follows: rename taxon names to the same names as in your sequence file (testseq.Gene1). Remove all colons and branch lenghts (if present). Make sure that the outgroup is defined at the very end (right side) of the line. If the outgroup consists of more than one taxon, the outgroup has to be hold together by a separate pair of brackets. Check the number of brackets: the number of left and right brackets should be the same. Leave only the tree definition on one single line with it's brackets and remove all other file content (if present). Save the file as Gene1.tree.

Examples of valid tree definitions:

One outgroup taxon:

```
(Taxon_A,((Taxon_B,Taxon_C),(Taxon_D,Taxon_E)),Outgroup);
```

Two outgroup taxa, hold together by a separate bracket pair:

```
((Taxon_A,((Taxon_B,Taxon_C),(Taxon_D,Taxon_E))),(Outgroup1,Outgroup2));
```

TreeView: check if the tree is readable in TreeView. If something is wrong, TreeView crashes or displays an error message. Once your tree definition is fine, print the tree out.

*Multiple genes/partitions:*
For parameter estimation in baseml (step 3), the number of taxa in a tree file must exactly correspond to the number of taxa in the appropriate sequence file. Therefore, if you have more than one gene/partition, use the best tree toplogy you have for each dataset, and save it as Gene1.tree, Gene2.tree, Gene3.tree etc.

For the steps 4 to 7, use one single (common) tree topology (the tree you want to use in the end for inferring nodal ages) together with the different sequence datasets. Both programs, estbranches (step 5) as well as multidivtime (step 7), are able to account for missing taxa in the sequence files. For example, you can analyze 3 different sequence datasets with 26, 30, and 54 taxa together with a single, common tree topology containing all 54 taxa. What you get in the end is a ultrametric tree with 54 taxa together with inferred nodal ages and substitution rates, based on your three different data partitions and their characteristic models of evolution.

Be sure that taxon names are spelled identically in all sequence datasets and tree files, including upper/lower case letters.

## Step 3: Estimation of model parameters

Obtain estimates for
- unequal nucleotide frequencies
- transition/transversion rate ratio: parameter $\kappa$
- rate heterogeneity among sites: shape parameter $\alpha$ (discrete $\gamma$ model of rates among sites)

Use **baseml** (part of PAML package; Yang 1997):
Input:
- testseq.Gene1 (sequences; ‘seqfile’),
- Gene1.tree (rooted tree topology with outgroup; ‘treefile’)
- Gene1.ctl (options file, specifies substitution model, gap handling etc.)

Output:
- Gene1.out (parameters of the substitution model; ‘outfile’)
- oGene1 (screen output)

First edit the file Gene1.ctl to specify the baseml parameters:
a) adapt the input and output file names (seqfile, treefile, and outfile)
b) select the F84 model (Felsenstein 1984).
c) unequal base frequencies: estimate base frequencies by ML iteration (nhomo = 1)
d) transition/transversion rate ratio: estimate $\kappa$ (fix_kappa = 0)
e) rate heterogeneity among sites: estimate $\alpha$ shape parameter (fix_alpha = 0), 5 categories of rates (ncatG = 5)
See PAML manual and estbranches readme file for more information.

Then open a terminal window, change to the PAML directory, and enter the following command:

```
baseml Gene1.ctl > oGene1
```
[duration: 5 – 10 min]

*Multiple genes/partitions:*
If you have more than one partition/gene, run baseml for each sequence file (testseq.Gene*), together with the corresponding tree topology file (Gene*.tree), with different settings in Gene*.ctl. The number of taxa in a tree file must exactly correspond to the number of taxa in the appropriate sequence file

## Step 4: Transform baseml output files to estbranches input files

Parse the baseml output in file Gene1.out to generate a formatted nucleotide substitution file modelinf.Gene1.

Use **paml2modelinf**:
Input:
• Gene1.out (parameters of the substitution model)

Output:
• modelinf.Gene1 (reformatted file with parameters of the substitution model)

First copy the file Gene1.out from the PAML directory into the multidistribute directory.

Then open a terminal window, change to the multidistribute directory, and enter the following command:
```
paml2modelinf Gene1.out modelinf.Gene1
```
[duration: < 10 s]

*Multiple genes/partitions:*
If you have more than one partition/gene, run paml2modelinf for each Gene*.out file. The result files should be named modelinf.Gene*.

## Step 5: Maximum likelihood estimates

Estimate the maximum likelihood of the
• branch lengths for the rooted tree
• variance-covariance matrix
by approximating the likelihood surface with a multivariate normal distribution centered at the maximum likelihood estimates of branch lengths (Thorne et al. 1998).

Use **estbranches**:
Input:
• modelinf.Gene1 (reformatted file with parameters of the substitution model)
• testseq (copied and renamed sequence file testseq.Gene1)
• Gene1.tree (rooted tree topology with outgroup, this time with a specially added first line, containing any words, see below)
• hmmcntrl.dat (options file, specifies the source file names etc.)

Output:
• oest.Gene1 (tree with estimated branchlenghts and variance-covariance matrix)
• out.oest.Gene1 (screen output)

Important:
Add a first line to the tree file Gene1.tree containing any text, so that the tree definition is then on the second line (use any text editor for that). Estbranches cannot read the treefile without this first "junk line".
Rename the testseq.Gene1 file into testseq. Estbranches does not accept other sequence file names.
Specify the estbranches parameters (source file names etc.) by editing the file hmmcntrl.dat (see estbranches readme file for information how to do this).

Mac OS X and Linux: open a terminal window, change to the multidistribute directory, and enter the following command:

```
./estbranches oest.Gene1 > out.oest.Gene1
```

[duration: 5 – 10 min, the last 2 min without any screen output]

Windows: open a terminal window, change to the multidistribute directory, and enter the following command:

```
estbranches_dna oest.Gene1 > out.oest.Gene1
```

You may run estbranches twice: the first time without output redirection (without the `> out.oest.Gene1` part). The program then writes the results to the screen (and not into the logfile), so you can check if the sequences are read correctly. After the program starts iterating, you can abort it (by pressing Ctrl-C). Then you run the program again as decribed above to generate the output files.

*Multiple genes/partitions:*
Run estbranches for each sequence file separately. Use the appropriate modelinf.Gene* file. Rename the appropriate testseq.Gene* file to the name testseq. Use one single, common tree topology (the tree you want to use in the end for inferring nodal ages) for all partitions/genes. The program estbranches is able to account for missing taxa in the sequence files, unless major parts of the tree (for example the entire outgroup) are missing in a dataset. Don't forget to adapt/edit hmmcntrl.dat to the new file names. The result files should be named oest.Gene* and out.oest.Gene*.

## Step 6: Check the performance of the likelihood optimization

Compare the max. likelihoods obtained from the baseml and estbranches analyses.

Search for the likelihood scores in the two files Gene1.out (PAML directory) and out.oest.Gene1 (multidistribute directory).

Mac OS X or Linux users may use the following `grep` commands to browse the files:

```
grep lnL Gene1.out
```
reports baseml result, e.g. lnL = -12519.13

```
grep 'FINAL LIKE' out.oest.Gene1
```
reports estbranches result, e.g. lnL = -12513.29

The two values should not be very different, otherwise one or both programs failed to optimize the likelihood (usually estbranches).

Then extract the first line of oest.Gene1 with the tree definition into a new file, open it in TreeView, and print the tree out. The tree should now have lost it's outgroup. Label the nodes of the tree with the node numbers assigned by estbranches. You can get them from the file oest.Gene1. You can do this node labeling also later, using the multidivtime output (see Frequently asked questions, p. 9).

*Multiple genes/partitions:*
If you have more than one partition/gene, do the above steps with the corresponding files.

## Step 7: Bayes MCMC analysis

Perform a Bayes MCMC analysis to approximate the posterior distributions of substitution rates and divergence times (Thorne et al. 1998; Kishino et al. 2001; Thorne and Kishino 2002). The likelihood function is approximated using a multivariate normal distribution of estimated branch lengths and is not calculated from the sequence alignment. Thus, the substitution model does not enter the second stage of the analysis.

Use **multidivtime**:
Multidivtime reads not only the branch lengths from oest.Gene1, but also a variance/covariance table. That's the reason why you cannot just take a tree with branch lenghts and analyze it directly with multidivtime, as it can be done in other dating methods, e.g. NPRS and PL (r8s, Sanderson 2003).

Input:
- oest.Gene1 (tree with estimated branchlenghts and variance-covariance matrix; the program estbranches should have removed the tree's outgroup)
- gene1.tree (rooted tree topology still **with** outgroup, the file still contains the first "junk line")
- multicntrl.dat (options file, specifies MCMC parameters and constraints. See multidivtime manual.)
- inseed (contains MCMC initial seed number)

Output:
- out.Gene1 (summarizes posterior means of divergence times and rates, and other information)
- tree.Gene1 (contains the tree definition of the chronogram)
- ratio.Gene1 (contains relative probability ratios for the parameter values sampled by MCMC)
- node.Gene1 (contains detailed information about the MCMC run)
- samp.Gene1 (contains samples from the Markov chain: they can be useful for exploring convergence)

First specify the MCMC parameters and age constraints by editing the file multicntrl.dat (see multidivtime readme file for information how to do this). Check then if all the input files have valid information before you start a long analysis.

For this, open a terminal window, change to the multidistribute directory, and enter the following command:
`multidivtime numbers` [duration: < 10 s]

This program mode just prints the tree topology with the node numbers assigned by estbranches. This is just a check if all the input files have valid information and file names. If not, the program crashes with an unspecified error message. You can use the screen output to label the nodes of your tree with the node numbers assigned by estbranches, if you haven't done it before (see Frequently asked questions, p. 9).

Then, start the analysis by typing in the following command:

```
multidivtime Gene1 > out.Gene1
```
[duration: > 30 min]

You can find the dating results in the files
- out.Gene1 (summarizes the posterior means of divergence times and rates, and other information)
- tree.Gene1 (contains the tree definition of the chronogram, ready to be printed out with TreeView).

**Run the analysis at least twice or more. Compare the results. If the results differ significantly, try different MCMC parameters (number of MC samples and cycles).**

*Multiple genes/partitions:*
If you have more than one partition/gene, adapt/edit multicntrl.dat, so that it uses multiple oest.Gene* files for simultaneous analysis of the different partitions/genes using different substitution models. The topology file you used with estbranches (step 5) serves here again as common tree for all sequence partitions. The program multidivtime is able to account for missing taxa in the sequence files, unless major parts of the tree (for example the entire outgroup) are missing in a dataset.

# C. Frequently asked questions

**1. What can I do if my partitions/sequence files do not have the same number of taxa?**
For parameter estimation in baseml (step 3), the number of taxa in a tree file must exactly correspond to the number of taxa in the appropriate sequence file. Therefore, if you have more than one gene/partition, use the best tree toplogy you have for each dataset, and save it as Gene1.tree, Gene2.tree, Gene3.tree etc. For the steps 4 to 7, use one single (common) tree topology (the tree you want to use in the end for inferring nodal ages) together with the different sequence datasets. Both programs, estbranches (step 5) as well as multidivtime (step 7), are able to account for missing taxa in the sequence files.

**2. How can I perform two multidivtime runs at the same time, with different initial seed conditions?**
Each time a multidivtime analysis ends, the initial seed number in file inseed will be changed automatically to another (odd) random number. If you want to run different analyses at the same time with different initial seed numbers, you have to change the initial seed number by hand (by editing the inseed file) for the second and following runs, in order to have different starting conditions.

**3. Is there an easier way to label the nodes of my tree with the numbers assigned by estbranches/multidivtime?**
A very useful tip from Ziheng Yang: "To figure out the node numbering for specifying the fossil calibration information, note that multidivtime prints out the ingroup master tree at the start of the run. You don't have to run a full multidivtime analysis, just use the command `multidivtime numbers`. The program then only prints the tree topology with the node numbers assigned by estbranches. Jeff Thorne used the symbol : to indicate node numbers. You can copy the tree into a file and then change all the : into another symbol such as # and then read it in TreeView and "Show labels"."

**4. Which multidivtime parameters (in multicntrl.dat) should I choose for my analysis?**
Find the solution to this answer in the multidivtime readme file:

"Markov chain Monte Carlo (MCMC) approaches (such as the one implemented in multidivtime ) approximate distributions of interest. The quality of the approximation improves as the length of the Markov chain increases. In multidivtime, the Markov chain completes **burnin + sampfreq * (numsamps - 1) + 1 cycles**. The first burnin cycles of the Markov chain are not used for approximating the posterior distributions of interest. No samples from the first burnin cycles are taken. Instead, the first sample is taken at cycle burnin+1. Reasonable choices for values of burnin and sampfreq depend on the data set. High values are more likely to lead to good approximations of posterior distributions than low values. On the other hand, the amount of computation time required is proportional to the number of cycles of the Markov chain. The Achilles' heel of MCMC approaches is that they may need too much computational time to get a good posterior distribution approximation. To get a hint of whether the MCMC approach is working well, one technique is to see if separate runs of the Markov chain yield similar approximations.

**rttm** is the mean of the prior distribution for the time separating the ingroup root from the present. rttmsd is the standard deviation of this prior distribution. Choice of the values for rttm and rttmsd depends on what is known by the user regarding the sequences and organisms being studied.

**rtrate** and **rtratesd** are respectively the mean and standard deviation of the prior

distribution for the rate of molecular evolution at the ingroup root node. Choosing a reasonable value of rtrate is difficult. My usual strategy is not statistically rigorous but it seems to work reasonably well. First, I use the estbranches program to estimate amounts of evolution from the ingroup root to the ingroup tips. These estimated amounts of evolution from the ingroup root to the ingroup tips will differ depending on the tips. I usually pick an amount that is close to the median of the amount of evolution for the different tips. I'll refer to this amount as X. Remembering that the amount of evolution is a rate multiplied by a time, I set rtrate to X divided by rttm. For the value of rtratesd, a big standard deviation (e.g., setting rtratesd to equal rtrate) may be reasonable when there is little knowledge about evolutionary rates.

Slightly relevant: you are free to set the time units to be what you want. Of course, you want to then adjust the units for the rates to correspond to the units for the times. For example, let's say that rttm should be about 20 million years and rtrate should be about 0.1 changes per 10 million years. You could make rttm equal to 2.0 where 1.0 time unit is 10 million years. You could then make rtrate equal to 0.1. My preference is to make the time units such that rttm is between 0.1 and 10 time units. This is the range where the MCMC proposal parameters seem to be best for achieving convergence of the Markov chain.

**brownmean** and **brownsd** set the mean and standard deviation of the prior distribution for ``nu" (the variance in logarithm of rate of molecular evolution for an amount of time t is ``nu" multiplied by t). We are still working on what is a good choice for the value of brownmean. My current favorite strategy is to have rttm*brownmean be about 1 or 2. I wish I had a more reasoned approach. My current approach for setting brownsd is to have it equal to brownmean. This generates a pretty flexible prior, I think. If brownmean is set equal to 0.0, then the program analyzes the data with a molecular clock (i.e., evolutionary rates are forced to be constant over time).

**minab** sets the prior for the times of the interior nodes given the time of the root. In the section describing changes subsequent to the Thorne et al. 1998 paper, "a" is used instead of "minab". High values of minab (values greater than 1) specify a prior where internal node times "repel" each other. Low values of minab (values lower than 1) cause internal node times to "attract" one another. I recommend trying a value for minab that is 1.0 or slightly greater than 1. To be honest, I have not done much exploration with values that differ from 1.0.

**newk**, **thek**, **othk** have to do with proposed states for the Metropolis-Hastings algorithm. It's probably a good idea not to change them unless you look at the multidivtime.c code."

Please refer to the multidivtime readme file or Wiegmann et al. (2003) for further information about multidivtime parameters.

### 5. What do I have to know about hardware requirements / memory usage?

Ziheng Yang writes: "Multidivtime keeps the sampled values of the variables (rates and times, for example) in the memory, so the maximum memory required by the program is more or less proportional to **numsamps**. If you have a large tree (so that there are many rates and times) and use a large number for numsamps, the program may well use up all your RAM. We once crashed a workstation with 2GB of RAM. So you should watch out for the memory used by the computer". Mac OS X and Linux users can use the `top` tool for tracking memory and processor usage, Windows users start the "Task Manager" to do the same. Baseml, estbranches, and multidivtime are **not optimized for multiprocessing**. Therefore, it doesn't make much sense to run the software on a multiprocessor machine. As you have to repeat each

multidivtime analysis anyway, you can save time by running the analyses on two machines (or in two sessions on a multiprocessor machine) simultaneously.

### 6. I've generated a huge dataset. Can I analyze it with the software described here?

So far, Bayesian molecular dating with baseml/estbranches/multidivtime has been done with sequences containing more than $1.5*10^4$ characters, and trees with more than 130 taxa. However, if your tree contains more than 200 internal and terminal nodes, you have to increase the **MAXNODE** precompiler definition in the source code (around line 52 in estbranches.c and line 49 in file multidivtime.c) and recompile the program. Otherwise, the program might crash reporting the message "Segmentation fault (core dumped)".

### 7. For branch length estimation in estbranches, can I apply any different model of nucleotide substitution than the F84+gamma model?

- unequal nucleotide frequencies
- transition/transversion rate ratio: parameter $\kappa$
- rate heterogeneity among sites: shape parameter $\alpha$ (discrete $\gamma$ model of rates among sites)

The most common general model that estbranches can handle for nucleotide substitution is F84+gamma, which has the following characteristics:

- unequal base frequencies
- 1 transition/transversion ratio
- $\gamma$ distributed rates among sites

Eventually, careful construction of the modelfile read by estbranches would allow invariant sites to be added to this model. Nevertheless, estbranches is not written to analyze other models such as GTR yet. Because of this, it might not be a good idea to use PAML output from other models to construct the model files for estbranches.

### 8. How can I get a ratogram (a tree where the branch length is proportional to the substitution rate of the branch)?

You can use Torsten Eriksson's Perl program RATO. It reads the multidivtime main output file, generates a chronogram, and writes it in a NEXUS tree file. It's available for download at http://www.plant.ch.

If you used 10 million year units for defining the rttm prior in multicntrl.dat (see FAQ 4), don't forget to divide the rates in the chronogram by 10 to get the usual rate units of substitutions $site^{-1}$ $mys^{-1}$.

### 9. How can I check convergence of the Markov chain?

In principle, it is not possible to say with certainty that a finite sample from an MCMC algorithm is representative of an underlying stationary distribution (Cowles and Carlin 1996). One simple thing you can do is to analyze the multidivtime output file *samp.Gene1* in a spreadsheet application (e.g. OpenOffice.org Calc or Microsoft Excel). This file contains all samples from the Markov chain that

have been generated and then used by multidivtime to calculate the parameters of the posterior distribution as reported in the output file *out.Gene1*. It does not contain any "burnin" samples. In detail, *samp.Gene1* contains first the samples of the divergence times for all nodes (0 to *n*), and then the samples of the rates for all branches (0 to *n*), and finally, the samples of the estimated autocorrelation parameter ν. After importing the data into the spreadheet application, you can highlight any column and then build a histogram for all samples within that column. If the chain has converged, all sample values should be located around a "stable plateau".

*Note:*
The problem of MCMC approaches is that they may need too much computation time to get a good posterior distribution approximation. Reasonable choices for the number of samples (and burnin cycles) depend on the dataset. High values are more likely to lead to good approximations of posterior distributions than low values. On the other hand, the amount of computation time required is proportional to the number of cycles or the Markov chain. Therefore: **Run the analysis at least twice or more and compare the results.** If the results differ significantly, try different MCMC parameters (number of MC samples and cycles). If you are interested in more sophisticated convergence tests, you can find a review on other convergence diagnostic procedures and tools in Cowles and Carlin (1996).

# D. Appendix

**Required format for the sequence file**

The example below shows a sequence data file with 7 taxa and 69 nucleotides. These numbers have to be defined in the first line of the file. Each sequence starts with the name of the sequence (max. 10 characters, without spaces), followed by at least 2 spaces, and the sequence itself.

```
7 69
17Axzeylan  TTTCGAGTAACTCCTCAACCTGGAGTTCCACCTGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
32Axcoriac  TTTCGAGTAACTCCTCAACCTGGAGTTCCACCTGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
19Cpanicu   TTTCGAGTAACTCCTCAACCTGGAGTTCCACCCGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
1Cgriffi    TTTCGAGTAACTCCTCAACCTGGAGTTCCACCCGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
16Cglabri   TTTCGAGTAACTCCTCAACCTGGAGTTCCACCCGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
2Cborneens  TTTCGAGTAACTCCTCAACCTGGAGTTCCACCCGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
3Dactylo    TTTCGAGTAACTCCTCAACCCGGAGTTCCGCCTGAGGAAGCAGGGGCTGCGGTAGCTGCTGAATCTTCA
```

# E. Selected literature


Cowles M.K. and B.P. Carlin. 1996. Markov chain Monte Carlo convergence diagnostics: A comparative review. J. Amer. Stat. Ass. 91, 883-904.

Dayhoff M.O., R.M. Schwartz and B.C. Orcutt. 1978. A model of evolutionary change in proteins. Pp. 345-352 in M.O. Dayhoff, ed. Atlas of protein sequence structure, vol. 5, suppl. 3. National Biomedical Research Foundation, Washington D.C.

Goldman N., J.L. Thorne and D.T. Jones. 1998. Assessing the impact of secondary structure and solvent accessibility on protein evolution. Genetics 149: 445-458.

Jones D.T., W. R. Taylor and J.M. Thornton. 1992. The rapid generation of mutation data matrices from protein sequences. CABIOS 8:275-282

Jukes T.H. and C.R. Cantor (1969) Evolution of protein molecules, pp. 21-132 in Mammalian Protein Metabolism, edited by H. N. Munro. Academic Press, New York.

Kishino, H., J.L. Thorne and W.J. Bruno. 2001. Performance of a divergence time estimation method under a probabilistic model of rate evolution. Mol Biol Evol, 18:352-361.

Korber, B., M. Muldoon, J. Theiler, F. Gao, R. Gupta, A. Lapedes, B.H. Hahn, S. Wolinsky and T. Bhattacharya. 2000. Timing the ancestor of the HIV-1 pandemic strains. Science 288:1789-1796.

Sanderson, M.J. 2003. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. Bioinformatics 19:301-302.

Thorne J.L., H. Kishino and J. Felsenstein. 1992. Inching toward reality: An improved likelihood model of sequence evolution. J. Mol. Evol. 34:3-16.

Thorne J.L., N. Goldman and D.T. Jones. 1996. Combining protein evolution and secondary structure. Mol. Bio. Evol. 13:666-673.

Thorne, J.L., 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.

Thorne, J.L. and H. Kishino. 2002. Divergence time and evolutionary rate estimation with multilocus data. Syst. Biol. 51:689-702 (http://statgen.ncsu.edu/thorne/multidivtime.html).

Wiegmann, B.M., D.K. Yeates, J.L. Thorne and H. Kishino. 2003. Time Flies, a New Molecular Time-Scale for Brachyceran Fly Evolution Without a Clock. Syst. Biol. 52(6): 745-756.

Yang, Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39: 306-314.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. *CABIOS* 13:555-556 (http://abacus.gene.ucl.ac.uk/software/paml.html).

Yang, Z. and A.D. Yoder. 2003. Comparison of likelihood and Bayesian methods for estimating divergence times using multiple gene loci and calibration points, with application to a radiation of cute-looking mouse lemur species. Syst. Biol. 52(5): 705-716.