Draft Manual
for

*BayesPhylogenies**

Mark Pagel and Andrew Meade
University of Reading, UK

m.pagel@rdg.ac.uk, a.meade@rdg.ac.uk
http://www.ams.rdg.ac.uk/staff/staffID=41

*code written by A. Meade

BayesPhylogenies

Introduction.
BayesPhylogenies is a general package for inferring phylogenetic trees using Bayesian Markov Chain Monte Carlo (MCMC) or Metropolis-coupled Markov chain Monte Carlo (MCMCMC) methods. The program allows a range of models of gene sequence evolution, models for morphological traits, models for rooted trees, gamma and beta distributed rate-heterogeneity, and implements a 'mixture model' (Pagel and Meade, *Syst Biol*, **55**, 571-581, 2004) that allows the user to fit more than one model of sequence evolution, without partitioning the data.

A number of introductions to the use of MCMC methods and Bayesian inference as applied to phylogenies are available and so we merely describe the program and its commands here. If you use the program in your published work, we would be grateful if you would cite the Pagel and Meade (2004) paper.

Running *BayesPhylogenies*
The program can be run interactively from a console, command line interface or terminal, or its commands can be set in the input file of aligned sequences or other traits. The program allows the user to choose models of evolution, set or modify their parameters, root and unroot trees, partition the data, and so on.

To run the program, first change into the directory that the program was unpacked into. Start the program by typing the binary name followed by the nexus file name.

for OS X and Linux machines type
"**./BayesPhylogenies filename.nex**"

for Windows

"**BayesPhylogenies.exe filename.nex**"

The input file must be in nexus format. The program expects as input a standard nexus file such as generated from PAUP or Clustal. The program will not accept multi-line comments. An example input file of ribosomal RNA data, called rrna.nex, is included and shows the exact input format. Substituting this name into command above will start the program.

Typing "**info**" causes the program to display the current settings to the screen. Using the rrna.nex file, the settings should look like this (seed will differ):

Current settings for BayesPhylogenies

Number of chains:	1
Number of iterations:	Infinite
Print frequency:	1000
Random seed:	1096021629
Output File Base:	rrna.nex

```
Input file:                 rrna.nex
Current partition:          Default
Auto run:                   False
Using an Un-Rooted tree
Partition name:             Default
    No:                     1
    Start:                  1
    End:                    1147
    Model:                  GTR
    Base Frequencies:       Estimated
    No of patterns          1
    No of rates             1
```

The display shows the input file name, describes the default values for the model of evolution, and a number of other settings. Settings are altered from the console by typing the command followed by its parameters. For example, to alter the number of iterations the chain runs, type iterations = [number]. Gamma rate heterogeneity is added to the model using the **Gamma =** command. Typing "**Gamma = 4**", causes a discrete gamma model to be added to whatever model of evolution you have chosen.

A full list of commands can be found below in the "Command List" section. To check that a command entered from the console has been accepted type 'info' again.

The default model of evolution is the GTR (general time reversible model). The **Model** command is used to change to a selection of other models of evolution. The list is available by typing **Model** ?. The program checks current settings of model against **basefreqs** to produce the correct model. For example, selecting HKY85 with a choice of equal base frequencies produces the K2P model and this is what is displayed by the info command.

Once the model of evolution has been chosen and other choices are made, run the program by typing "**run**". This will cause the program to begin a MCMC analysis of the input data. It will print output to the screen showing the iteration number of the chain and the log-likelihood of the tree at that iteration. The first few lines repeat the contents of the info command, and then the log-likelihoods are displayed. The output might look like this (using the rrna.nex file as input and set to 10000 iterations):

```
Current settings for BayesPhylogenies

Number of chains:           1
Number of iterations:       10000
Print frequency:            1000
Random seed:                1096021629
Output File Base:           rrna.nex
Input file:                 rrna.nex
Current partition:          Default
Auto run:                   False
Using an Un-Rooted tree
Partition name:             Default
    No:                     1
```

```
Start:                  0
End:                    474
Model:                  GTR
Base Frequencies:       Estimated
No of patterns          1
No of rates             1
0       -8538.989107
1000    -7440.592042
2000    -7355.585040
3000    -7325.978385
4000    -7316.222354
5000    -7318.259688
6000    -7316.992841
7000    -7315.637355
8000    -7315.806069
9000    -7312.712995
10000   -7311.841898
```

To stop the program at any point hold the *Ctrl* key and press *C*.

Note that the start and end numbers above differ from those obtained from typing info. The numbers on the output represent the file size after compressing the data to identify common site patterns.

After the specified number of iterations has been run the program will return to the console waiting for further input. The acceptable commands are **run**, **quit**, or **iterations** = [Number]. Changing the number of iterations then typing run will cause the program to carry on for that many more iterations from the current state of the chain. Once a Markov chain has been set up and run, the program will not allow you to change its features, as this would violate its equilibrium properties. To set up a new chain with different features quit the program and restart.

Running BayesPhylogenies from the Nexus Input File

For complex sets of parameters or when you wish to run a number of different analyses, possibly making minor changes, it may be more convenient to choose the model of evolution and set the other parameters in the nexus input file. This is done by adding a **BayesPhylogenies** block to the nexus file. At the end of the file type:

> **Begin BayesPhylogenies**;
> Commands….
> **End**

Between the **Begin** and the **End**, any of the commands as shown above or those applicable from the list of commands below can be included. If the **AutoRun** command is used the program starts to run without any further input from the screen.

An example of a valid *BayesPhylogenies* block is below. The file pres-abs-dir.nex contains a *BayesPhylogenies* block that runs the gene presence/absence data in this file with a non-time-reversible (i.e., directional) morphological model. A non-time-reversible

morphological model allows the rates of change form 0 → 1 to differ from the rate of change from 1 → 0. Non-time-reversible models have should only be used with a rooted tree as, unlike a time reversible model, the likelihood for the tree will differ depending upon where the root is placed. To specify the root use the **root** command. In the example using the **root** command below, four taxa are used as the outgroup. This causes the program to root the tree between these four species and the remaining ingroup species. These four are not allowed to move into the ingroup.

```
begin BayesPhylogenies;
        # Set the Model
        Model M2P

        # Root the tree
        root Sulfolobus Methanothermobacter Pyrococcus Thermoplasma

        # Automatically start the run
        AutoRun
end;
```

Setting Up and Running a Mixture Model

BayesPhylogenies allows the user to fit more than one model of evolution to each site in the alignment. Pagel and Meade (2004) provide examples of this, showing how it can detect heterogeneity in the patterns of evolution across sites, and without prior partitioning of the data by the investigator. We find that it can often greatly improve the log-likelihood of the data.

To implement the mixture model, use the **patterns =** command. Typing, "**patterns = 2**" causes two independent rate matrices of the chosen model of evolution to be calculated at each site. This command can be used in conjunction with the gamma command. If the gamma model is invoked the mixture model includes gamma rate heterogeneity (see Pagel and Meade, 2004).

The mixture model uses the same basic model of evolution for each of its rate matrices, but their parameters are allowed to vary independently of one another. It estimates a weighting term for each matrix, and these are printed to the parameters output file. The default setting for a mixture model is to use a single set of base frequencies (the π's) across the rate matrices. To vary both the rate parameters and the base frequencies, use the complete form of the pattern command. This is **patterns** = [no. of matrices] pi=[true/false], where 'true' corresponds to varying the base frequencies. Thus "**patterns** = 2 pi=true" causes two rate matrices to be formed in which both the rate parameters and the base frequencies vary.

We recommend using the GTR model of sequence evolution when fitting a mixture model as the GTR allows the greatest variety of patterns to emerge. Allowing the base frequencies to vary often brings little improvement to a GTR mixture model, but in

alignments with distinct compositional heterogeneity, allowing the base frequencies to vary can be helpful.

Mixture models can return large improvements but may also require a large number of parameters, although often no more than a partitioned dataset. Convergence and mixing of the parameters must be followed carefully (see Pagel and Meade, 2004 for advice), and expect the mixture model to take longer to reach convergence.

As a general point, applicable more widely than just to mixture models, users should be aware that large amounts of data may be required to estimate parameters accurately. We recommend at least a 10-1 rule of data to parameters, and caution that if the true value of a parameter is small, very large amounts of data may be required to estimate it well. In a phylogenetic tree, the topology, branch lengths, and elements of the model of sequence evolution are all parameters to be estimated from the data.

Morphological Data

presence/absence file "pres-abs.nex" contains binary data. To analyse a morphological model, start the program as above but using pres-abs.nex as the input file, and select the one parameter morphological model by typing "**model = m1p**". This implements the simplest morphological model, one in which the rates of gain and loss of the trait are presumed to be equal. The **m2p** model allows the rates of gain and loss to differ and technically must only be used with rooted trees.

Creating and Using Partitions

Although for most applications we advocate using a mixture modelling approach, *BayesPhylogenies* allows the user to partition the data. This is especially useful when different kinds of data are combined. A combined sequence and morphological data set is included in the file combined-data.nex. By default *BayesPhylogenies* applies the same model to all sites. However, any dataset can be partitioned and different model and options can be specified for each partition, using the **NewPart** and **SetPart** commands. Each partition must be given a name, a start site and an end site. Commands affecting the model of evolution are applied to the currently active partition. To set a partition as the active partition type "SetPart *name*".

The example below shows how to create and set partitions in a BayesPhylogenies block. However, these commands can also be entered interactively. Blocks should not overlap.

```
begin BayesPhylogenies;

    # Create a Sequence partition.
    NewPart RNASeq 1 1147

    # Create a Morphological partition.
    NewPart GenePreAbs 1148 3744

    # Set the Sequence partition as active and set up some parameters
    SetPart RNASeq
    Model   GTR
```

```
        Gamma   4

        # Set the Gene presents / absence partition as active
        SetPart GenePreAbs
        Model   M1P

        AutoRun
end;
```

Metropolis-Coupled Markov Chain Monte Carlo (MCMCMC)
To run a MCMCMC simply specify the number of chains using the **chains** command. The output to the screen lists the likelihood of each chain, with the cold chain being the first. The heating schedule is fixed at 1/t where t is the temperature of the chain and t takes the values 1,2,3…. The output also shows for each chain the number of attempted and accepted swaps in the previous interval.

Using the **append** command with MCMCMC causes the program to start running the specified number of chains all of which are started from where the cold chain ended on the previous run.

When using MCMCMC be aware that whenever the cold chain receives a swap, it potentially goes out of equilibrium. This means that sampling from a 'converged' MCMCMC can be difficult.

Output files
Running the program causes two output files to be created. One is a tab-delimited file containing the tree likelihoods and the estimated parameters from each iteration of the Markov chain. The file is called

filename.nex.parameters

Its main use is to follow the chain to convergence by plotting the likelihood against iteration number and to investigate the parameters. It can be read straight into a program such as Excel. The other output file contains the nexus formatted tree files and is called

filename.nex.trees

This file is used as input into tree viewing or manipulating programs such as PAUP or TreeView. The file will have to have "**End;**" added to the end of the file before it can be read into a program such as TreeView or PAUP.

Starting a Run from Other than a Random Starting Tree
The **InputTree**, **InputRates**, and **Append** commands make it possible to start the Markov chain from a specified position rather than form a random starting point. This should be used cautiously but can be useful when the user has confidence in the region of convergence but wishes to sample more trees or parameter values without repeating the iterations that lead to convergence.

LIST OF COMMANDS

COMMANDS FOR MANIPULATING THE MODEL AND ANALYSIS

Command: **Model**
Shortcut: mo
Purpose: Set the model of evolution for the current partition.
Parameters: Valid models for sequence data are GTR, GTNR, HKY85, F81, SYM, JC, K2P. Valid models for binary ('morphological') data are M1P and M2P. For N-State data the KSTATES model will analyse a dataset in which the sites can adopt two or more states, including variables numbers of states across sites. Only the uniform bases frequencies option is allowed for this model.
Examples: model GTR, mo M1P
Notes: Applies to the active partition. The program checks current settings of model against basefreqs to produce the correct model. For example, selecting HKY85 with a choice of equal base frequencies produces the K2P model and this is what is displayed by the info command.

Command: **BaseFreqs**
Shortcut: bf
Purpose: Set the 'base' frequencies of the model of evolution. For models of sequence evolution these are the familiar π values corresponding to the nucleotides. For morphological models they refer to the frequencies of the various discrete character states.
Parameters: options are uniform (uni), estimate (est) or empirical (emp) base frequencies
Example: BaseFreqs uni
bf estimate
Notes: Applies to the active partition.
Only uniform base frequencies are allowed with the KSTATES model.

Command: **Gamma**
Shortcut: ga
Purpose: To apply gamma rate heterogeneity to the model of sequence evolution.
Parameters: the number of discrete rate categories.
Example: gamma 4
ga 6
Notes: To remove rate heterogeneity use **gamma 1**

Command: **Beta**
Shortcut: be
Purpose: To apply beta distributed rate heterogeneity to the model of sequence evolution.
Parameters: A number of discrete categories
Example: Beta 4
be 6

Notes: To remove rate heterogeneity use **beta 1**

Command: **Patterns**
Shortcut: pa
Purpose: To set the number of independent rate matrices in the mixture model and to specify whether both the rate parameters and the base frequencies are allowed to vary.
Parameters: [an integer number of matrices] [pi=true/false]. Typing pi=true allows the base frequencies to vary across matrices. If left unspecified the default is to use a single set of base frequencies across the rate matrices.
Example: Pattern 2
Patttern 2 pi=true
pa 3
Notes: The pattern command can only be used with models that have more than one rate parameter. It cannot be used with JC, or M1P, and is not allowed with KState

Command: **Iterations**
Shortcut: it
Purpose: To set the number of iterations that the Markov chain will be run. To run the program for an infinite number of iterations use -1.
Parameters: an integer
Example: Iterations 1000000
it -1
Notes: None.

Command: **chains**
Shortcut: ch
Purpose: To set the number of simultaneous chains for a Metropolis coupled Markov chain Monte Carlo run.
Parameters: An integer
Example: chains 4
ch 2
Notes: The second, third, fourth and so on chains are heated according to $1/t$ where t is the chain number.

Command: **cooling**
Shortcut: co
Purpose: To set a logarithmic cooling schedule for the current simulation. The cooling schedule runs until the iteration number supplied as a parameter is reached. It has a curvilinear form starting with a heated chain that quickly cools and then slowly converges to an unheated change by the iteration number supplied. The strength of the initial heating is given by the second parameter. Large (in absolute value) negative numbers define stronger initial heating.
Parameters: an integer and a negative floating point value.

Example: Cooling 1000000 -150
Co 50000 -80
Notes: This command can be used as an alternative to MCMCMC. Instead of running a number of independent chains that swap states, this command causes the single chain to explore more of the universe of trees. However, the chain must be allowed to reach convergence after the cooling has finished and before any sampling from the chain can begin. This means that the chain should be run well beyond the value of the first parameter.

Command: **qdev**
Shortcut: qd
Purpose: To set the standard deviation of the distribution from which changes to the rate parameters are drawn.
Parameters: a floating point value
Example: qdev 2.0
qd 0.1
Notes: This parameter normally should not be changed from its default value but can be used if the rate parameters are not mixing satisfactorily.

Command: **Outgroup**
Shortcut: og
Purpose: To specify the out-group of an unrooted tree. If left unspecified the program uses the last species in the alignment file.
Parameters: a single species name, spelled exactly as in the alignment file (see Taxa command).
Example: Root Dog
rt human
Notes: The GTNR and M2P models are directional and require a rooted tree. Otherwise there is no reason to use this command as unrooted trees return the same likelihood for all outgroups.

Command: **Root**
Shortcut: rt
Purpose: To create a rooted tree, with the out-group being defined by the taxa names.
Parameters: one or a list of taxa names, spelled exactly as in the alignment file (see Taxa command)
Example: Root Dog Cat Bat
rt human
Notes: The GTNR and M2P models are directional and require a rooted tree. The other models are time reversible but can still be used with a rooted tree.

Command: **UnRoot**
Shortcut: urt
Purpose: To unroot a tree.
Parameters: none

Example: UnRoot
urt
Notes: None.

Command: **Taxa**
Shortcut: ta
Purpose: causes the program to list the taxon names as they are in the alignment file
Parameters: none
Notes: This command can be used in conjunction with specifying outgroups or roots.

COMMANDS FOR PARTITIONING

Command: **NewPart**
Shortcut: np
Purpose: Creates a new partition in the alignment.
Parameters: A partition name, start site and end site.
Example: NewPart Gene1 1 1250
np Morph 1251 1400
Notes: The default partition encompasses all the sites and is automatically deleted when a valid partition is created.

Command: **SetPart**
Shortcut: sp
Purpose: To make the named partition the current partition.
Parameters: A partition name
Example: SetPart Morph
sp Gene1
Notes: Used to modify some aspect of the named partition, such as to assign a different model

Command: **DelPart**
Shortcut: dp
Purpose: To remove a partition
Parameters: A partition name
Example: DelPart Gene1
dp Morph
Notes: None.

COMMANDS FOR RUNNING AND QUITTING

Command: **autorun**
Shortcut: ar
Purpose: To start the program running automatically from a *BayesPhylogenies* block
Parameters: None
Example: autorun

Ar
Notes: This command can only be used in the *BayesPhylogenies* block in the nexus file.

Command: **Run**
Shortcut: r
Purpose: runs a Markov chain
Parameters: None
Example: Run
r
Notes: none

Command: **Quit**
Shortcut: q
Purpose: To exit the program with out running a Markov chain
Parameters: none
Example: quit
q
Notes: None

INFORMATION AND HELP

Command: # or //
Shortcut: no shortcut
Purpose: To add a comment in the *BayesPhylogenies* block.
Parameters: None
Example: # A comment
// Another comment.
Notes: None.

Command: **Info**
Shortcut: I
Purpose: Displays the current settings.
Parameters: none
Example: Info
i
Notes: None

Command: **Help**
Shortcut: ?
Purpose: Print a list of valid commands
Parameters: none
Example: Help
?
Notes: None

Command: **Valid**

Shortcut: va
Purpose: To check the combinations of parameters, data and partition setting.
Parameters: none
Example: Valid
va
Notes: This command is applied by default once the **run** command is issued.

COMMANDS FOR MANIPULATING THE OUTPUT

Command: **batch**
Shortcut: bt
Purpose: Useful when running a series of analyses from the same input file. Causes the program to create consecutively numbered output files, rather than overwriting the default files.
Parameters: none
Example: batch
bt
Notes: This command cannot be used with the append command. It must be typed in before each new analysis.

Command: **Debug**
Shortcut: de
Purpose: This command causes an extra output file to be written displaying the schedule of changes made to the parameters
Parameters: None
Example: Debug
de
Notes: None

Command: **Ouptut**
Shortcut: op
Purpose: To set the name for the output files if other than he default name is desired.
Parameters: A string
Example: Output DataSet-01
Op SomeData
Notes: If the files can't be created with the name the program will end.

Command: **PrintFreq**
Shortcut: pf
Purpose: Sets the frequency with which the Markov chain is sampled
Parameters: an integer
Example: PrintFreq 1000
pf 5000
Notes: None.

Command: **SiteLh**

Shortcut: sl
Purpose: To output the site-by-site likelihoods for the last sampled tree.
Parameters: None
Example: SiteLh
sl
Notes: The site likelihoods are un-weighted by pattern or rate heterogeneity weight. This command is useful for studying how well different sites are fitted by the model of evolution (see for example Fig. 2 of Pagel and Meade, 2004).

DEFINING THE STARTING POINT OF THE MARKOV CHAIN

Command: **Append**
Shortcut: app
Purpose: To re-start the program from a point where it was stopped in a previous run. This command requires the same parameters setting with which the program was originally run. It takes its information from the two output files.
Parameters: None
Example: Append
app
Notes: None.

Command: **InputTree**
Shortcut: itree
Purpose: To provide the Markov chain with a starting tree other than a random tree.
Parameters: the name of the tree file (from *BayesPhylogenies*) and the iteration number of the tree in that output file. See also **Append.**
Example: InputTree DataSet.trees 1000000
Notes: This command was only designed to be used with trees generated by *BayesPhylogenies*. It is not designed to allow user generated trees to be used. Normally a Markov chain should be given a random starting point but there may be circumstances in which the user wishes to start from a known point.

Command: **InputRates**
Shortcut: irates
Purpose: To load a starting set of rate parameters for the model of evolution. See also **Append.**
Parameters: the name of the parameters file (from *BayesPhylogenies*) and the iteration number of the line of output in that file.
Example: InputRates DataSet.Parameters 25000
Notes: This command was only designed to be used with rates generated by *BayesPhylogenies*. Normally a Markov chain should be given a random starting point but there may be circumstances in which the user wishes to start from a known point.

Command: **Seed**
Shortcut: se
Purpose: To set random number seed. An entire Markov chain can be exactly duplicated by using its seed to begin a new run.
Parameters: An integer
Example: Seed 938343
Se 34243
Notes: None

Some FAQ's

Reading Nexus files.

Error:
"Taxa W (X) has Y sites but was expecting Z"

Problem:
Y sites were found for taxa W, but the nexus block specified Z sites.

Solution:

Check the spelling for taxa W, if the file is interleaved.
Check the number of sites for taxa W against the number specified by "nchar" in the nexus dimensions line.

Error:
"
No Dimensions info found
No Begin Data block found
Error Passing Nexus file
"

Problem:
The nexus file parsing routine could not find a valid

"begin data;"
or a
"dimensions ntax=X nchar=Y;"

lines in the nexus file.

Solution:
Check the spelling of the two lines.
This problem can also be caused if that nexus file has the wrong type of line brakes. E.g. if the text file has Mac OS 9 style line breaks but the program is run using the PC binary.

Error:

"Taxa begin (8) has 10 sites but was expecting X"

Problem:

X taxa were specified in the ntax options. But could not find one or more of them.

Solution:

Check that the number of taxa specified by the ntax option is the same as the number of taxa in the file.

Error:

"A multi-line comment was found ([this is) currently BayesPhylogenies does not support multi-line comments."

Problem:

BayesPhylogenies does not support multi-line comments.

Solution:

Delete the multi-line comment.

Error:

Other problem with loading nexus files.

Problem:

Unknown.

Solution:

Try copy and pasting the data into nexus file that is working.

Acknowledgements and Disclaimer

The work to produce *BayesPhylogenies* has been supported in part by grants from the Biotechnology and Biological Sciences Research Council of the UK. We have used the program for the last few years but cannot ensure that it is free of bugs. Please report any problems to MP or AM.