Introductory User's Manual for

MULTISTATE

(copyright M. Pagel)

Mark Pagel
School of Animal and Microbial Sciences
University of Reading
Reading RG6 6AJ
UK
email: m.pagel@rdg.ac.uk
(www.ams.reading.ac.uk/zoology/pagel/)

MULTISTATE

Multistate is a computer program for the analysis of discrete characters on phylogenetic trees. The program implements a continuous time Markov model and was initially described in Pagel (1994). Several papers since then describe other features of my related Discrete model (Pagel, 1997, 1999a,b).

This manual provides a very short introduction to using the **Multistate** method. Some familiarity with my Discrete method will be helpful (a manual is available from for this method: www.ams.reading.ac.uk/zoology/pagel/). The compressed folder contains executable files to run on PC or Mac OSX computers. This version of Multistate allows one to analyze variables with between 2 and 6 states, using an interactive command. A control file and a test data file for a 4-state variable are also included.

Users who may wish to analyze a sample of trees rather than just one can use Multistate in conjunction with our **Bayesian-Multistate** scripts also available on this website. These scripts are intended to be used as a way of accounting for phylogenetic uncertainty in comparative studies (see Lutzoni, Pagel, and Reeb, 2001; Pagel and Lutzoni, 2002). The Bayesian-Multistate scripts along with Multistate make it possible to test comparative hypotheses in each of the trees in a sample of trees and then combine those tests across trees.

The approach that the Bayesian-Multistate scripts adopt documents the *between-tree* variability in comparative parameters, that is the variation that arises from using different phylogenetic trees to test the same hypothesis. For example, if Multistate were used to estimate the ancestral state at a node, the Bayesian-Multistate scripts document how that estimate varies from tree to tree. Another component of variability is that attributable to the inherent uncertainty about the parameters of the model of trait evolution within any given tree. Multistate finds the maximum likelihood (ML) values of these parameters, but it is also possible to ask how the likelihood of the model of trait evolution varies when the parameters are fixed at other than the ML values. The Bayesian-Multistate scripts are not designed to investigate this *within-tree* variability. To do so, it is necessary to vary the parameters of the model of trait evolution by hand from within Multistate.

This manual describes how to use **Multistate** and some of its hypothesis testing capabilities. It is intended to provide enough of an introduction to use the program, although it does not attempt to describe all of the things for which one might use the program.

Some of the main uses of the program are:

- find ancestral states (see especially Pagel, 1999b),
- test for rates of evolution,
- detect directional trait evolution,

- and, investigate the tempo and mode of trait evolution.

INTRODUCTION

Multistate implements a continuous-time Markov model in a maximum likelihood framework. This makes it possible to analyse and test hypotheses about trait evolution without the need ever to reconstruct ancestral states (although ancestral states can be estimated). Instead, the parameters of trait evolution are estimated having summed the likelihood over all possible states at each node of the tree (see Pagel, 1994 for further explanation of the likelihood approach).

An advantage of the likelihood approach over other methods is that uncertainty in the ancestral state reconstructions is automatically taken into account in all likelihood calculations. By comparison, for example, parsimony methods first infer the ancestral states and often treat them in later calculations as if they are known without error. This lends a false degree of certainty to calculations and biases p-values. This bias is most pronounced when traits evolve more than once on a tree, that is when trait evolution is relatively rapid. Under such circumstances parsimony methods are known to underestimate the amount of change, especially in long branches of the tree (Pagel, 1999a gives an example).

Multistate proceeds by estimating the instantaneous rates of transitions among all possible pairs of states. It then uses these to calculate the probabilities of change in a branch of a given length. For a trait that can take only two values (e.g., 0,1), two rates must be estimated, one for transition from $0 \rightarrow 1$, and the other for transitions from $1 \rightarrow 0$. These parameters are sufficient to characterise the evolution of traits in isolation from one another. Six parameters are required for a trait with 3-states, and in general, $n(n-1)$ parameters are required for a character with n states. For three states, one estimates the transitions from 0 to 1, 0 to 2, 1 to 0, 1 to 2, 2 to 0, and 2 to 1 (states are always denoted 0,1,…n-1). These transition rates can be shown to be sufficient to calculate the probability of any kind of change in any branch of the tree, and they can be used to chart the most probable course of evolution from the ancestral state to the contemporary derived state.

Running Multistate

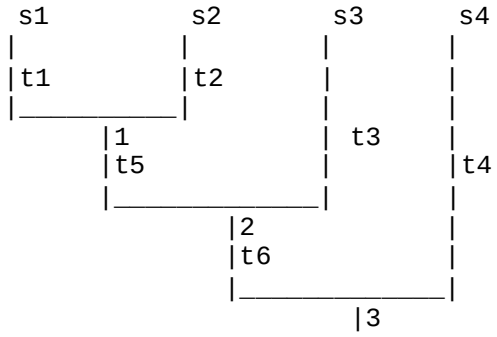
This version of the program runs on PC's under the Windows operating system or on Macintosh OSX systems. Users of pre-OSX Macintosh computers can run Multistate by installing Connectix Virtual PC on their computers.

Data Input Format

Multistate requires a bifurcating and rooted phylogeny and data on species. It uses its own input format ('pag-format') to describe the phylogenetic tree and the data. The attraction of the tree format is its transparency, although it is not as compact as Phylip

(Newick) or Nexus formats. We make available software for converting from Nexus to pag formats (www.ams.reading.ac.uk/zoology/pagel/).

The input format is simple. Consider the tree below of four species, and three internal nodes, 1, 2, and 3 where node 3 is also the root. Trees must be rooted.



The 'pag' input format for this tree is:

4 1
example phylogenetic tree. Comments can precede tree if
preceded by "#" as in this line.

s1, n1, t1, data
s2,n1, t2, data
s3, n2, t3, data
s4,n3, t4,data
n1, n2, t5
n2, n3, t6

The first line in the file gives the number of species and number of traits. The "data" are the comparative data measured across species. Multistate takes one trait (but see Appendix for a special case exception to this). Colons should not be used in the data file.

Read the first line of the input tree as "species 1 goes to node "1" over 'time' or length t1 with data 1; the second line as species 2 goes to node 1 over time or length t2, with data 1", and so on. Data points must be real numbers. Missing data are not allowed. Species with missing data must be removed from the tree.

Beginning with the fifth line of the file, the connections among the internal nodes are described, "node 1 goes to node 2 over time or length 4, and node 2 goes to node 3 over time or length t6". Nodes do not have data, and the branch lengths can be any real number. Branch lengths can be any units but units of time and genetic distance (operational time) are especially useful. If no branch length information is available, one option is to assign them all an arbitrary length of 1.0 (although it should be borne in mind

that doing so implies that more total evolution has taken place between the root and the tips of the tree for those species with more ancestors).

Only tips (species) have data, and the tree must be bifurcating and rooted. If you have so-called polytomies in your tree, resolve them to bifurcations, or if this is not possible, remove species until a bifurcating node remains. If the species that are removed all have the same values on the traits, the analyses will not be affected in any substantial way. Trees must be rooted, although the root itself is not described but inferred by the program from the input format.

The species names can be any alphanumeric character but should start with a letter. They must be one word. Internal nodes can be integers or alphanumeric characters (Note that this input format is more flexible than that for my related program **Continuous** that analyses quantitative comparative data. Users anticipating using both methods may wish to have their input formats conform to that required for **Continuous** – see its manual).

It may often be easiest to number the species from left to right beginning with species s1 to species s*n*. Then label the nodes as n+1, n+2 and so on until you reach the root. Every node in a bifurcating tree must have two and only two descendants. The root does not "go to" any other node as is not described any further. Items are separated by commas.

At the moment, **Multistate** does not give very much useful information when it tries to read in a treefile that is wrong in some respect. The most common errors are failing to separate items by commas, or inserting more than one comma, having too few or too many descendants of a node, specifying the wrong descendant, or incorrectly specifying the number of species or variables. The best way to de-bug a treefile that is not working is to print it out and compare it line by line to a picture of the phylogeny.

Running MULTISTATE cont'd

Once you have an input file, to launch Multistate on Windows, double click on the executable and a window will appear asking for the name of the **control** file. On Macintosh OSX, launch Multistate from the Terminal command-line. Operation is then the same for both systems. Type 'control' (or whatever you have named it – an example control file is given below). If you have specified the **input** file name in the control file, it will read it in. Otherwise you will be prompted from the window for the name of the input file. The input file must be a text file and in the same folder as the executable and the control file. Once you've read in the input file name the program will ask for the name of the **results** file. This is the file to which the contents of the active window will be written, including all of the output from **Multistate** (the output also is printed on the screen. The output frequently scrolls up off the screen so you may wish to set up scrolling windows within Windows).

The program will then await your next command. Typing in 'test' fits the model to the data, and returns a likelihood.

Restricting parameters
Parameter values of the transition rates can be restricted using the 'restrict' command within the window (alternatively, restrictions can be placed in the control file and thus can be saved from one session to the next). For a four state model, the parameter values that can be restricted are (states are numbered 0-3):

q01	*q02*	*q03*
q10	*q12*	*q13*
q20	*q21*	*q23*
q30	*q31*	*q32*

The *qii* (e.g., *q00, q11, q22, q33*) are functions of the others in their row and cannot be changed. In any given row, at least one of the transitions must be allowed to vary. Parameters can be restricted to a constant or to each other.

'restrict q10 = 0.0' forces that parameter to be zero in the model;
'restrict q10 = q12' forces these two parameters to be equal to each other in the model

By comparing the likelihoods of restricted and unrestricted models one can test whether a restriction has an effect (see below and also Discrete manual or Pagel, 1994).

Be sure to unrestrict parameters when making subsequent tests, by using the 'clear' command.

Common sets of restrictions
A 3-state model estimates 6 parameters, a 4-state model estimates 12 (shown above), 5-states requires 20 parameters, and 6 states requires 30 (in general $n^2 - n$). A good rule of thumb is to limit the number of parameters so that there are ten data points per parameter. Thus, one might want at least 300 species to estimate a 6-parameter model.

However, by using restrictions one can retain the flexibility of many states but reduce the number of parameters. A very useful model to compare to the 'full model' (the one that estimates all $n^2 - n$ parameters) is a simple 'forward/backward' model. This model is created by restricting all of the forward parameters to be equal to each other, and the same for all the backward ones. For example, for the 4-state model the following restrictions set all of the forward transitions to each other:

'restrict q01 = q02'
'restrict q01 = q03'
'restrict q01 = q12'
'restrict q01 = q13'
'restrict q01 = q23'

The same can be done for the backward transitions by setting all of them to say q10.

You may wish to make up special control files that contain a set of restrictions that you will commonly use. This saves typing them all in again at the beginning of each new session. An alternative to the forward/backward model is one in which you restrict the parameters in each row to be equal to each other. This would be equivalent to saying that each state has its own rate of moving either forward or backward. For a four-state model this set of restrictions would create a 4-parameter model.

Commands available in the window

quit : terminates interactive session
help : displays this message

status : print information on current restrictions
scaling [0|1] : set/unset branch length scaling
clear [<parameter>] : removes restrictions on parameters
restrict <parameter> = <value | parameter> : restrict a parameter

test : calculate likelihood
fossil [<state> <node>]: sets fossil at given node to given state

Scaling
The 'scaling' command sets or unsets the branch length scaling (see Pagel, 1994 or Discrete manual). It is often good to fit this parameter at least once as it is often zero or very near to zero for discrete data. Then in subsequent runs, the scaling parameter can be restricted to its maximum likelihood value. To restrict it, type in 'restrict scale = x.xxx, where x.xx is the real number value to which you want to restrict it. This often greatly improves the fit of the data to the model (compare the likelihoods with scaling turned on and off).

Reconstructing Ancestral States: the Fossils command
The 'fossil' command is used to reconstruct ancestral states. It fixes the interior nodes of the tree to given values. The synatx is fossil state node, where 'state' is only those values that are legitimate in the model (so, 0,1,2,3 in a 4-state model), and node is the node name corresponding to the node in the pag file. Having fixed a node, run the 'test' command to get the likelihood given the fixed node.

To determine the support for various states at a node, successively fix the node to its n possible states and compare the likelihoods. The proportional support for a given value at a node can be calculated from the following. Define 'numerator' as Exp[log-likelihood for state i], where Exp is simply the base of the natural logarithm, *e*. That is, one raises *e* to the power of the log-likelihood for state i. Most hand held calculators will do this.

Then define 'denominator' as the sum of Exp[log-likelihood for state i] for all *n* states. That is, raise *e* to the power of the log-likelihood separately for each of the states and add up the *n* results.

Finally, divide numerator by denominator to get the proportion of the likelihood associated with state i. In general the support for one state over another at a given node is treated as significant if the difference between their log-likelihoods is more than the value 2 (see Pagel, 1999).

Remember to 'un-fix' the fossil state at a node before going on to other calculations (although it is acceptable to fix more than one node at a time). To do this type in "fossil 127 node". For example, "fossil 127 N12" would un-fossil node N12.

As an alternative, if you wish to reconstruct a binary trait, see the 'testall' command described in the Appendix (below). This will 'fossil' nodes and calculate probabilities automatically using "global" and "local" approaches (Pagel, 1999b), for all nodes.

Testing Likelihoods

To compare two log-likelihoods simply subtract the smaller from the larger. The absolute value of their difference is distributed as a chi-squared variate with degrees of freedom equal to the difference in the number of parameters in the two likelihoods. Thus, to test the full 12 parameter likelihood of the 4-state model to the likelihood obtained from the forward/backward model, compare the difference in their log-likelihoods to a chi-squared distribution with 10 degrees of freedom. If the forward/backward model is not significantly worse that the full model this provides an argument for using it as a simpler representation of the data. Use a liberal alpha value for this test, of perhaps 0.15 to 0.25, because if the two models differ you want to have power to detect that.

To test whether the scaling parameter improves the likelihood, compare the likelihoods with and without it in the model, and test against a chi-squared with 1 df.

Other tests follow this same logic, except for the tests of ancestral state reconstructions (fossils). Here, the model is the same for all states at the node, and one uses the rule of thumb that if the log-likelihoods differ by two or more, then one state is 'significantly' better supported at that site than the other.

****

Example "control" file. It is best to leave untouched the 'interactive' and 'partial' commands. Commands with a # in front are turned off, removing the # swithces it on. A set of restrict commands are shown, each one setting the value of a parameter to zero.

```
#datafile = test4

interactive = 2

#restrict q02= 0.0
#restrict q03= 0.0
#restrict q20= 0.0
#restrict q21= 0.0
#restrict q12=0.0
#restrict q13=0.0
```

```
#restrict q23= 0.0
#restrict q30= 0.0
#restrict q31= 0.0
#restrict q32= 0.0
#converge =
#number =
#step =
#rate =
partial= 1

**********************************
```

References.

Please consider citing one or more of these as appropriate when you publish work using **Multistate**.

Pagel, M. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society (B)* **255** 37-45 (1994).

Pagel, M. Inferring evolutionary processes from phylogenies. *Zoologica Scripta* **26**, 331-348 (1997).

Pagel, M. Inferring the historical patterns of biological evolution. *Nature*, **401**, 877-884 (1999a)

Pagel, M. The maximum likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies. *Systematic Biology*, 48, 612-622 (1999b).

Lutzoni, F. ,Pagel, M., and Reeb, V. 2001. Major fungal lineages derived from lichen-symbiotic ancestors. *Nature*, 411, 937-940.

Pagel, M. and Lutzoni, F. 2002. Accounting for phylogenetic uncertainty in comparative studies of evolution and adaptation. Pp 148-161 in Biological Evolution and Statistical Physics (M. Lässig, and A. Valleriani, eds). Berlin: Springer-Verlag.

Notes.

Multistate was adapted from the computer program Discrete, by Richard Forster, with subsequent modifications by Peter Fredericks, Dr. Andrew Meade and Dr. Daniel Barker

Appendix: the *testall* command

Multistate's 'testall' command causes Multistate to reconstruct the ancestral state of every node in the tree. It is currently only implemented for traits that take two states, 0 and 1. Use of 'testall' with the Bayesian-Multistate scripts is not recommended.

The ***testall*** command must be issued from within Multistate (i.e., at the Multistate prompt), and has syntax:

testall *outputfilename*

testall ignores any user-defined fossils but does respect user-defined restrictions on q_{ij} and scale.

Intermediate calculations are written to the screen, and the probabilities of state 0 for each trait and node are written to the file *outputfilename*. If the file already exists, it will be overwritten. Since only two states are allowed, the probability of state 1 is simply 1 – (probability of state 0).

outputfilename is a tab-delimited text file. The first row contains column headings. The first column gives the trait number, counting from 0. Subsequent columns give the probability of state 0 at each node. For each trait, probabilities for the 'global' approach are given, followed by probabilities for the 'local' approach (Pagel, 1999b). For terminal nodes, probabilities are output as 0 or 1 (if the state is 1 or 0, respectively).

In addition to probabilities, *outputfilename* contains a summary of user-defined restrictions in force during the **testall** command. These are given below the block of probabilities. 'none' indicates no user-defined restrictions.

Note the use (or non-use) of scaling is not currently recorded by 'testall', and should be recorded by the user.

One may continue to use **Mulstistate** interactively after a **testall** command, but any user-defined fossils are cleared by **testall**. They will have to be set again by the user. Continued use of the program after **testall** has only been lightly tested.

If there is more than one trait, traits must be supplied as a comma-separated list. An example data file with four species and three traits is:

```
4 3
s1, 1, 0.5, 1, 0, 0
s2, 1, 0.5, 0, 0, 0
s3, 2, 0.5, 1, 1, 1
s4, 3, 1.0, 0, 0, 1
1, 2, 1.0
2, 3, 1.0
```

‘test’, described in the main part of the manual, will only use the first trait, ignoring the rest. ‘testall’ will use all traits.

Data files of incorrect format may have unpredictable effects.