User's Manual for

# *Discrete*

*(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/)

# *Discrete*

**Discrete** is a computer program for the comparative analysis of binary 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 Discrete (Pagel, 1997, 1999a,b). I would be grateful if you would cite these papers, as appropriate, when you use Discrete. A recent application can be found in Lutzoni, Pagel, and Reeb, 2001.

The application program can be used to

- test for correlated evolution between pairs of traits,
- find ancestral states (see especially Pagel, 1999b),
- test for rates of evolution,
- detect directional trait evolution,
- investigate the tempo and mode of trait evolution,
- to detect differential rates of evolution in different branches of the tree using a gamma rate heterogeneity model, and
- to conduct Monte Carlo simulation studies of results.

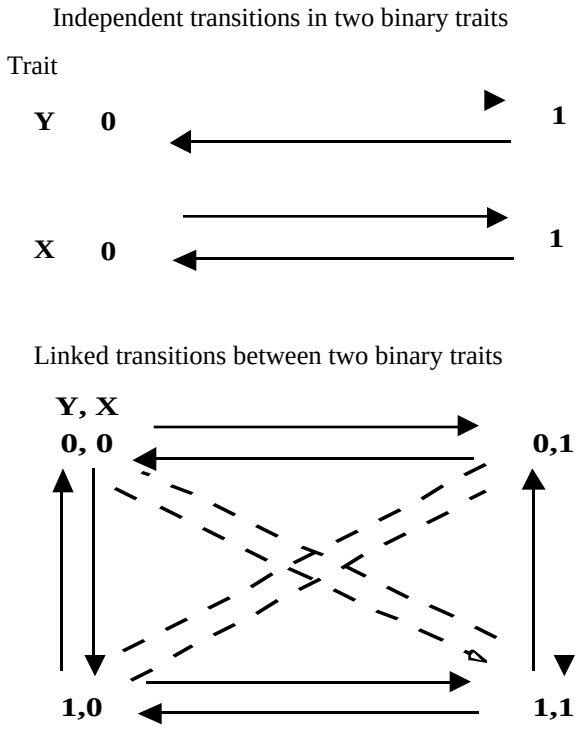
This manual describes how to use **Discrete** and its hypothesis testing capabilities. The manual is intended to provide enough of an introduction to use the program, although it does not attempt to describe all of the things one might use the program for. Some features of Discrete are not yet implemented in the program even though they appear in the menus. The current computer interface was designed and programmed by Mr Peter Fredericks; Drs Heath and Richard Forster worked on earlier versions of the program.

For variables with more than two states my related program called **Multi-state** is available. However, Multi-state is intended only to study trait evolution and not correlated evolution between pairs of traits. With more than two states a very large number of parameters is required to test correlated evolution. But see Characters with more than two states under **Hypothesis Testing**.

**Discrete** implements the 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 this approach over parsimony methods is that uncertainty in the ancestral state reconstructions is automatically taken into account in all likelihood calculations. By comparison, parsimony methods first infer the ancestral states and then 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 (see Pagel, 1999a for an example).

One of the principal uses for Discrete is to test for correlated evolution between two binary discrete characters. This is achieved by comparing the fit (likelihood)of two models to the data. In one the two traits are allowed to evolve independently; in the other they evolve in a correlated fashion. Evidence for a correlation is found if the model of correlated evolution fits the data significantly better than the model of independent evolution (Pagel, 1994).

For a trait that can take only two values (e.g., 0,1), two rates must be estimated, one for transition from "0" to "1", and the other for transitions from "1" to "0". These parameters are sufficient to characterise the evolution of traits in isolation from one another. Four parameters are required for two traits evolving independently (see Figure). The model of correlated or dependent trait evolution considers the four possible states that two binary characters can jointly adopt (0,0; 0,1; 1,0; 1,1). It then allows one of the variables to change state in any branch of the tree, yielding eight possible transitions to be estimated (Figure). These 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.




Independent transitions between two binary states in two traits (upper); Linked or correlated transitions in two binary traits (lower). Dashed lines are not calculated.

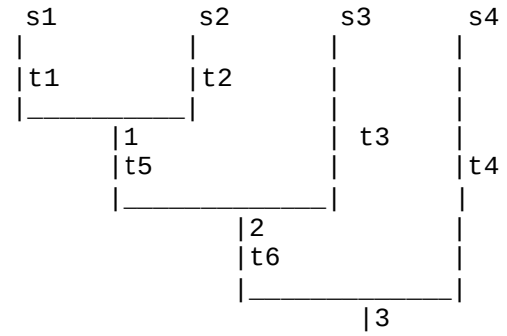
**Running Discrete**

This version of the program runs on PC's under the Windows operating system. Macintosh users can run Discrete by installing Connectix Virtual PC on their computers.

**Data Input Format**

**Discrete** requires a bifurcating 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 hope to make available software for converting among these three formats.

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:

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

s1, 1, t1, data1,data2
s2,1, t2, data1, data2
s3, 2, t3, data1, data2
s4,3, t4,data1, data2
1, 2, t5
2, 3, t6

The "data1, data2" are the comparative data measured across species. Discrete takes two traits. If only one trait is being investigated it can be duplicated to create a 'dummy' second trait.

Read the first line of the input file as "species 1 goes to node "1" over 'time' or length t1 with data 1, 2; the second line as species 2 goes to node 1 over time or length t2, with data 1, 2", 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 shouldbe 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. 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, **Discrete** 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.

# Loading and Viewing the Data: File Menu

When **Discrete** is started up a blank window will appear. Select **Open** from the **File** menu and look for the input file in the box that appears. Input files must be text files in the format as described above. If the input file has been saved by a word processor such as Microsoft Word be sure to choose the text only option. Sometimes word processors can insert invisible characters that can interfere with the input file being read in. If you suspect this, open the file in a word processor and save it as some other file choosing a text only format.

If the treefile loads properly, the message “Data Loaded in Successfully” will appear, and describe the file. To view the contents of the input file select **Display Input** from the **File** menu. The **Save Output** command saves a copy of the analysis window in a text file and is useful when a number of analyses have been run.

Subsequent files can be loaded the same way and will replace the previous file as the one that Discrete will analyse.

**Analysing Data**

**Discrete** can be used to characterise and test hypotheses about the evolution of single traits, to find ancestral states, to test for evidence of correlations among traits, and to conduct computer simulations. All parameters are estimated by maximum likelihood, and found by searching a likelihood surface for the value that maximises the likelihood of observing the data given the value of the parameter, the phylogenetic tree, and the model of evolution. All likelihoods are expressed as log-likelihoods.

The **Independent**, **Dependent**, **Simulation**, and **Graphics** menus provide the features for analysing data. The features in these menus will be described in order below, although this will not normally correspond to how they will be used when analysing data.

# Independent Menu

The **Independent** menu provides a number of ways to investigate the evolution of single traits, do ancestral state estimation, and provides the starting analyses for the test of correlated evolution. The results to complete the test of correlated evolution are obtained from the **Dependent** menu. How to conduct significance tests is described in the **Hypothesis Testing** section. The following sections describe the Independent menu. All calculations in this menu fit the model of independent trait evolution in contrast to the model of dependent evolution described in the next section.

**Run Independent Test** command

This command calculates the log-likelihood of the model of independent evolution for the two traits (see Pagel, 1994, 1997). This model allows the traits to evolve independently on the tree. Each trait is characterised by a forward and backward transition rate, labelled “alpha” and “beta”, respectively. They are the instantaneous transition rates from state 0 to state 1 (alpha) and from state 1 to state 0 (beta). As rates they depend upon the lengths of the branches of the phylogenetic tree. They are not probabilities. Alpha1 abd beta1 correspond to trait 1 and alpha2 and beta2 to trait 2.

When the independent test is run the program prints out the likelihood of the model (which is the sum of the likelihoods for each variable separately), the transition rate parameters, and information on the state of other parameters that can be fixed or estimated.

**Set Independent Variables** command

This option opens a menu box allows the user to estimate ancestral state at the root of the tree (other ancestral state reconstruction described under **Graphics** menu), calculate scaling

parameters, fix the values of parameters to predetermined values, and to test for differential rates of trait evolution.

The **Parameter Restriction** box allows one to choose a parameter to be fixed to a scalar value or to be restricted to be equal to another parameter. The kind of restriction is chosen from the Restriction Type menu. Setting a parameter to a constant fixes it at that value in all likelihood calculations. Fixing a parameter to the value 0f 0.0 can be used to compare the likelihood obtained when the parameter = 0.0 with that obtained when it is allowed to take its maxmimum likelihood value (see Hypothesis testing). Fixing a parameter to be equal to another parameter means that they are restricted to take the same estimated value in the model. This feature makes it possible to test simpler models (for example, rate of forward changes equals rate of backward changes is achieved by fixing alpha1=beta1 or alpha2 = beta2) against unrestricted models.

The choices made in the **Parameter Restriction** box are implemented the next time **Run Independent Test** option is chosen.

The **Set Model** box allows the user to fix the root (**Fix Root** option) or allow it to remain free (if not fixed the likelihood calculations sum over both values). If **Root Reconstruction** is switched on, then the program automatically estimates the likelihood of the two alternative values at the root and prints out the posterior root probabilities based upon what I have called the "local" method (see Pagel, 1999b for an explanation of the calculations). The **Bayesian Weights** option is described in Pagel (1999b) but is not fully tested.

The Independent Scaling option allows the user to estimate the value of the parameter $\kappa$ (kappa). Kappa is described in Pagel (1994). The **kappa** parameter differentially stretches or compresses individual phylogenetic branch lengths and can be used to test for a punctuational versus gradual mode of trait evolution. Kappa > 1.0 stretches long branches more than shorter ones, indicating that longer branches contribute more to trait evolution (as if the rate of evolution accelerates within a long branch). Kappa < 1.0 compresses longer branches more than shorter ones. In the extreme of Kappa = 0.0, trait evolution is independent of the length of the branch. Kappa = 0.0 is consistent with a punctuational mode of evolution.

Kappa is interesting in its own right and can be valuable for smoothing the likelihood surface. If the phylogeny contains a wide range of branch lengths – some very long, others very short – it can be difficult to fit the likelihood model. Kappa will often take a value $<<1.0$ on such trees, making all branches roughly the same length.

The **Advanced Options** box contains parameters that the numerical analysis algorithm uses in its 'hill-climbing' routine. These are best left untouched, save for the Convergence value. This value determines when to stop the likelihood search: if two successive likelihoods from the search procedure differ by less than the Convergence value, the search is stopped. Smaller numbers therefore cause a more stringent stopping rule to be enforced.

The choices made in the **Set Model** box are implemented the next time **Run Independent Test** option is chosen.

**Gamma Settings** command

This menu implements a gamma rate heterogeneity model of trait evolution (using code based upon Yang's discrete gamma model. J. Mol. Evol. 39,306,1994). This model allows the traits to evolve at different rates in different branches of the tree, where the distribution of rates is assumed to follow a gamma distribution with a mean of 1.0. When the gamma parameter is estimated, the likelihood of the basic model of trait evolution (Independent model or Dependent model) is summed over the distribution of possible rates.

If the gamma model improves the fit of the data to the underlying model, the likelihood will be improved and this indicates that rates of evolution are significantly faster or slower in some branches of the tree. The model does not at present identify which branches.

Gamma rate parameters can be estimated separately for the X and Y traits (trait 1 and trait 2), they can be restricted to be equal to each other (via the **Parameter Restriction** window), or they can be restricted to a constant. The gamma distribution is, for purposes of calculation, divided into a number of discrete classes of equal area. Four divisions usually provides sufficient resolution.

Choosing the gamma option greatly slows calculations as the parameter is often difficult to fit and the number of likelihood calculations is increased by a factor equal to the number of divisions chosen. It is recommended that the model is run a number of times when the gamma option is on as the value of the parameter often varies from run to run (usually indicating that it has little affect).

**Discrete** automatically incorporates the maximum likelihood values of the kappa and gamma scaling parameters into its calculations, when these options are switched on.

**Ancestral States** command

This command allows one to estimate the best simultaneous set of ancestral states on the tree. There are $2^n$ possible assignments of ancestral states of a binary character to *n* nodes. The option calculates the likelihood of each of them and identifies the single assignment of ancestral states to the n nodes that has the highest likelihood. For trees of more than about 18 nodes it can take a very long time, especially if the 'local' option is used (Pagel, 1999b). This option re-calculates the Independent model for each of the $2^n$ assignments. The global option simply applies the parameter values from the Independent model to each reconstruction.

The set of ancestral states derived from this option can differ from those obtained by separately calculating the most probable ancestral state at each node (**Graphics** menu), allowing the others to vary.

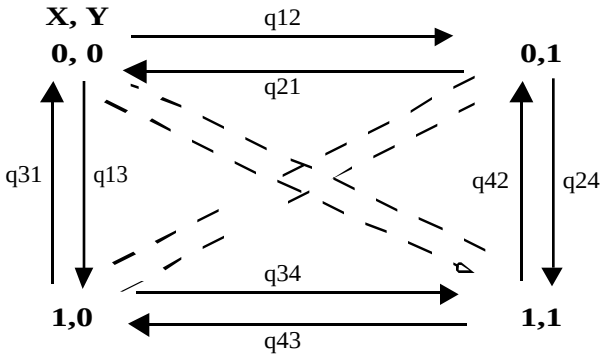
# Dependent Menu

This menu effectively repeats the options of the Independent menu but here implements them for the model of dependent trait evolution.

**Run Dependent Test**
This option calculates the likelihood of the 8 parameter model of dependent trait evolution. The parameters are displayed as qij values and a table is drawn showing their correspondence to the actual states of the traits. Thus, q12 estimates the rate at which the Y or trait2 character changes from 0 to 1 when the X character is in state 1. The q34 parameter measures the same rate, but now against a background of character X in state 1. Careful choice of comparisons of pairs of parameters tests specific hypotheses of trait evolution. A number of these are described in Pagel (1994).

The four 'forward' and four 'backward' transition rate parameters can be used to construct a 'flow diagram'. The flow diagram charts the most probable way that the traits have evolved from some ancestral state to some derived state. For example, if in the diagram below the state "0,0" is thought to be ancestral, one may be interested in how evolution got to the state "1,1". The diagram shows that it could have gone via the intermediate state '1,0' or via '0,1'. By testing each qij for significance it will often be the case that one of the possible pathways is significant but the other is not (see Czeilly, DuBois, and Pagel, Animal Behavior, 59, 1143-1152., 2000 for an example).

**The Flow Diagram**




**Set Dependent Variables** menu

This menu repeats the options of the Set Independent Variables menu but now they are applied to the dependent model. As before it is possible to fix parameters, set them to each other, reconstruct ancestral states at the root, and choose the kappa scaling parameter.

The Root Reconstruction option calculates the most probable joint set of states at the root, and prints out their probabilities. They will often be equal to the product of the corresponding root probabilities from the Independent model, although they need not be.

**Gamma Setting**

This option finds a single gamma value that is optimal for scaling the dependent model. It can be very difficult to fit.

# Simulation menu

The simulation menu allows the user to set up and run a Monte Carlo simulation study of the independent or dependent model. Its principal used is to find the approximate null hypothesis distribution for the test of correlated evolution. This test compares the log-likelihoods of the model of independent evolution with that of the model of dependent evolution, via what is known as the likelihood ratio statistic (see Testing Correlated Evolution under **Hypothesis Testing**).

**Run Simulations**

This command runs the Monte Carlo simulations following the choices made in the **Simulation Setup** menu.

The simulations print results to the screen: IL = independent likelihood of independent model as fitted to simulated data; DL = likelihood of dependent model on same data; LR = likelihood ratio = (DL-IL); (0,0),…(1,1) = the proportion of simulated tip values (species) with these trait-combinations.

At the end of a run of *n* simulations the approximate p-value is printed out for the likelihood ratio that was observed in the real data.

> **Important**: for the p-value result and the simulations to be meaningful the exact forms of the independent model and dependent model that are being tested should be run in succession just before running the simulations.

Sometimes simulations fail owing to unusual combinations of data that cause floating point errors. The simulation results are written out to a file so if a simulation fails, the runs to that point can be retrieved. Simulations can be combined to yield larger data sets.

**Simulation Setup**

The Simulation Type menu allows the user to choose the Independent or the Dependent model as the model that is used to generate the simulation data. The default is the independent model as this is the model used to derive the null hypothesis sampling distribution for the test of correlated evolution.

**Fossil Records**

If ‘fossil’s have been set (that is, nodes fixed to one or the other value of the trait)on the phylogeny (**Graphics** menu) this option allows them to be included or not in the simulations. If they have been used when the independent model was calculated then they can be employed in the simulations.

**Number of Simulation Runs**

A minimum of 100 runs is recommended, although use fewer to inspect the runs to see if the settings are correct and that the run is producing meaningful results. Simulation results (likelihoods and distributions of tip states are printed out to the screeen).

**Simlimit and Variance**

The simlimit command prevents the hill-climbing algorithm from getting stuck (2000 iterations is a useful figure) and the varianced command ignores simulated runs in which the variance of the characters acoss the tips is too low. A value of 5 seems useful. Simulated data sets can, by chance, all come up with the same value at the tips and then there is nothing for the model to analyse.

**Parameter Output**
Selecting these options produces output files of the estimated parameter values for the simulated data.

# Graphics Menu

The Graphics menu allows the user to inspect the phylogeny, reconstruct ancestral states, assign ancestral states to nodes, and to calculate likelihood surfaces for specified parameters.

**Draw Phylogeny**
This option produces a picture of the phylogeny drawn to scale from the branch length information in the input file.

Clicking on a node (this can be tricky) brings up the **Node Information** box for that node. The box gives information about the node, including its ancestral state (default = no state information). By clicking on Fossil1 or Fossil2 it is possible to fix the value of the node at a specified state, or return it to the 'free' or unfixed state.

Re-calculating the likelihood having set the node successively to state 0 and then 1 gives information about which is the more probable state. This procedure is automated in the **Calculate Fossil Likelihood** command. Pressing 'GO' instructs the program to calculate the likelihoods of a '0' and then a '1' at the node and to print out their probabilities under the model. This is a very quick way to do ancestral state reconstruction by maximum likelihood. Results are printed to the text window.

**NOTE potential problem:** When 'fossil' likelihoods are calculated two kinds of calculation are done, called 'local' and 'global' (see Pagel 1999 Systematic Biology for a description of these). There is a mistake in the current version of Discrete (4.0) that means that if a series of ancestral states are calculated in succession, the global estimates will only be correct for the first set. This is because after calculating the global and local fossil likelihoods, the global alpha and beta parameters of the independent model get lost. The consequence is that if one estimates a number of ancestral nodes in a row, the global estimate no longer represents the true global estimate because the initial parameter estimates are no longer the same. The local estimates are not affected.

To get global estimates of ancestral state, first fix the alpha and beta parameters to their ML estimates using the settings in the Set Independent Variables menu. When this is done the local and global estimates using the "Go" button will necessarily be the same. The local estimates can be obtained by unfixing the parameters and re-doing the analyses.

Clicking on the end of a terminal branch reveals information about the species.

**Surface Plot**
This command draws the likelihood surface for a parameter, given the instructions from the **Surface Setup** command.

**Surface Setup**
It is often desirable to see how the likelihood changes for differing values of a parameter – this is a likelihood surface in one dimension. The parameter is successively fixed at a series of values and all other parameters are free to vary when the likelihood is calculated. The option allows the user to choose a parameter to be plotted, specify the accuracy of the curve (number of points in curve), and specify whether 95% confidence intervals should be included.

These plots can be quickly calculated for the standard parameters of the independent model, but may take a long time for scaling parameters, gamma parameters, and parameters of the dependent model.

# Hypothesis Testing

All hypotheses are tested using the likelihood ratio statistic. The likelihood ratio statistic compares the log-likelihood of a null hypothesis model to that of an alternative hypothesis model. **Discrete** automatically calculates the log-likelihood of whatever model is chosen in the **Independent** or **Dependent** menus, and displays this likelihood in the text window.

Once you have run the dependent test a likelihood ratio will be printed out. The value that is printed out to the screen (and in the simulations) is the simple difference between the dependent likelihood based upon the last Dependent model run, and the independent likelihood based upon the last model run under the Independent analysis. Thus, for example, if one wishes to test for correlated evolution, the Independent model should be run followed by the Dependent model. Then, the likelihood ratio printed out will reflect the difference between these two models. Conventionally, this difference is multiplied by 2 to form the likelihood ratio statistic. The likelihood ratio (LR) test compares the goodness of fit of a model to the data with that of a simpler model that lacks one or more of the parameters. The LR statistic is then defined as

$$LR = -2\log_e\left[H_0 / H_1\right],$$

where $H_0$ represents the simpler (null) model and $H_1$ the (alternative) model containing the parameters representing the evolutionary processes one wishes to estimate.

If the simpler model is a special case of the more complicated one, the LR statistic is asymptotically distributed as a chi-squared variate with degrees of freedom equal to the difference in the number of parameters between the two models, i.e., $LR \sim \chi^2(v)$, where $v$ is the number of degrees of freedom. One test is a special case of another if it is possible to collapse the more complicated model to the simpler model by setting some parameters to zero or to other fixed values. For example, the model in which a parameter such as kappa is estimated collapses

to the default null hypothesis model of kappa = 1: the kappa=1 or null model is a special case of the alternative hypothesis model in which kappa is free to take any value. In such circumstances the two models are often referred to as being 'nested', and here they differ by one degree of freedom.

Testing Correlated Evolution. One of the principal uses of Discrete will be to test for correlated evolution. Elsewhere (Pagel, 1994) this is called the omnibus test. This test is performed by comparing the likelihoods of the models of independent and dependent evolution via a likelihood ratio test. If the traits are correlated in the sample the dependent model will fit the data significantly better. In the above equation for LR the log-likelihood of the independent model is $H_0$ and $H_1$ is the log-likelihood of the dependent model.

In their default states, these two models differ by four parameters. Simulation studies (Pagel, 1997) show that the likelihood ratio test ratio in this instance is asymptotically distributed as a chi-squared variate with 4 degrees of freedom. However, for small phylogenies or for traits that show very little change on the tree, the null hypothesis distribution may be less than a chi-squared with 4 degrees of freedom, approximating to a 3 or even 2 degree of freedom distribution.

What this means is that if the result of the test exceeds the chi-squared 4 df criterion for $p<0.05$ (for example), one can safely reject the null hypothesis. If it doesn't, it may still be possible to reject the null if simulations show that the distribution is less than chi-squared with 4 df. This is what the simulations options determine (see Simulation Setup under **Simulations**).

**Other Examples of Likelihood Ratio tests with Discrete** (Pagel, 1994 provides an outline of tests and Pagel, 1999a gives an example of the test of correlated evoltuion with binary traits).

Models of Evolution. Do forward and backward transitions proceed at the same rate? Is the rate of back transitions not different from zero? These and other examples can be tested by simple LR tests with one degree of freedom. Compare the restricted model of independence (alpha=beta; beta=0.0) to the unrestricted model.

The dependent model has 8 parameters. Frequently it is possible to show that some of them do not differ from zero. These parameters can then be set to zero to produce a simpler model of dependent evolution. By implication, this model says something about how the two traits evolved.

Conditional or Contingent trait evolution. Does the rate at which Trait 2 changes from 0 to 1 depend upon the state of Trait 1. This and other conditional tests are performed by restricting the dependent model. Comparing the likelihood of a model in which q12 is restricted to q34 with the likelihood of the unrestricted model tests this hypothesis of conditional evolution. The test has 1 df. An alternative form of this test separately asks whether each differs from 0.0. If one does and the other does not, then it might be argued that they differ from each other. This test can be slightly more powerful than the preceding test.

Punctuational and Gradual Trait Evolution. Perform a LR test of kappa = 0.0 (null) to kappa = ML value. If kappa(ML) is not significantly different from 0.0, then trait evolution is consistent with a punctuational mode of change. Kappa > 0.0 implies some form of gradualism. Test whether kappa < 1.0 to see if default or 'scaled' gradualism is better supported.

More generally, the test of kappa is one of whether the branch lengths are informative about trait evolution. If they are not, kappa will tend to go to zero.

Constant-rate of change: Perform a LR test of the independent model with Gamma turned off versus the same model with Gamma turned on. This test will have one degree of freedom for each value of gamma estimated. Thus, if gamma is estimated only for trait1, the test will follow a chi-squared 1 df distribtuion.

Ancestral States. The conventional cut-off point for preferring one state at a node over the other is if their likelihoods differ by more than 2 log units or by more than 4 in the LR test.

Characters with more than two states. Discrete is not set-up to calculate likelihoods for traits with more than two states. However, any trait with more than two states can be represented as a series of binary traits, each one contrasting a group labelled "1" with all of the others. Careful choice of assignment of 1's and 0's in successive traits can account for the comparisons one may wish to make. Each of the successive binary traits can then be correlated with some other binary trait of interest.

**General Tips**

Finding the maximum likelihood can be difficult for some data sets. Users should repeat analyses of the independent and dependent models several times to get a sense of the stability of the result. Sometimes a "local" optimum exists and the program will occassionally find that result rather than the global optimum. Repeating the analysis it will become obvious which of the two is the preferred result.

Some data sets have very difficult likelihood surfaces that return highly unsatisfactory results. Data sets with a very large ratio of the longest to the shortest branch can sometimes behave badly. These cases can often be dealt with by introducing a scaling parameter substantially less than 1.0. This has the effect of shrinking all branches, but shrinking longer ones more than shorter ones. There is nothing wrong with doing this; in fact the optimal branch length scaling is interesting in its own right (see Pagel 1994, 1997). The scaling reflects the transformed space in which the underlying model of evolution best fits the data.

Sometimes the best fit model returns very large values for some of the rate parameters. They can be so large as to seem unrealistic. Usually this means that the likelihood surface is 'flat' for that parameter and so, effectively, all values of the parameter return the same likelihood. The large value then does not indicate a large effect.

## References

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.

## IX. Disclaimer

Discrete has been tested and gives correct results, to the best of my knowledge. However, no specific claims are made for its accuracy and users are responsible for the interpretation and use of all results derived from it.

## X. Known Problems

Restricting a parameter of the Independent model to 0.0 will sometimes cause a log SING error. The problem seems to be most acute on small phylogenies. If restricting a parameter to zero is important for a hypothesis test, try setting it to a very small value such as 0.000001.

See the section on ancestral state reconstruction about a bug in one aspect of this code (easily dealt with within the program).

Users are invited to report problems to m.pagel@reading.ac.uk

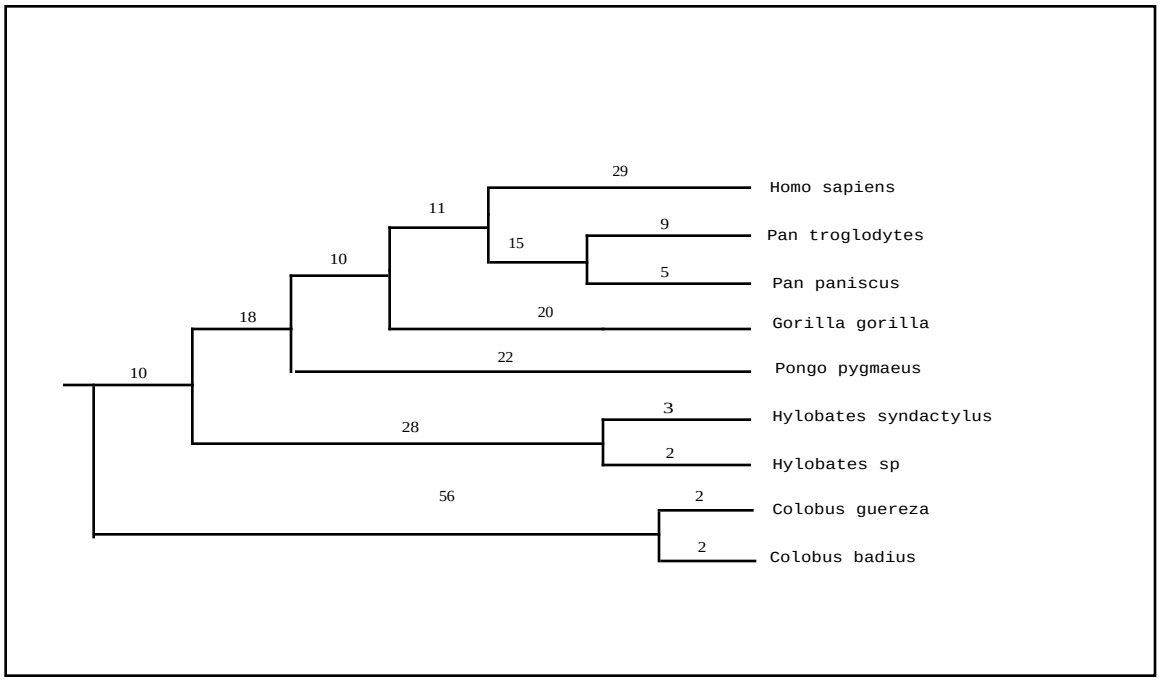
Appendix: Example data set

This data set includes values of mating system (1=multi-male, 0=monogamy or unimale) and presence/absenece of oestrous advertisement (1=present, 0=absent) for nine Old World primates. It is in the correct pag-file input format for Discrete. S. Branch lengths are genetic distances.

#data on primates: trait 1 = advertisement, trait 2 = mating system

```
homo.sapiens,n11,29,0,0
pan.trog,n10,9,1,1
pan.paniscus,n10,5,1,1
gorilla,n12,20,0,0
pongo.pyg,n13,22,1,0
Hylo.syndact.,n14,3,0,0
Hylo.sp,n14,2,0,0
col.guer,n16,2,0,0
col.bad,n16,2,1,1
n10,n11,15
n11,n12,11
n12,n13,10
n13,n15,18
n14,n15,28
n15,n17,10
n16,n17,56
```

The Phylogeny implied by this treefile



Analyses of Example data set

log-likelihood of

Independent model: ≈ -10.52

Dependent Model: ≈ -7.05; LR test ≈ 2 *X* 3.47≈ 6.94

approximate p-value (100 runs of simulation) ≈ 0.02

Models of Evolution

restrict alpha1 ≈ 0.0 (set to 0.00001 setting to 0.0 may cause computational error in this case. See **X.** Known Problems) likelihood ≈ -14.206

restrict alpha2 ≈ 0.0 (set to 0.00001 setting to 0.0 may cause computational error in this case. **X.** Known Problems) likelihood ≈ -15.051

Setting either alpha1 or alpha2 to zero causes a large increase in the likelihood. This is equivalent to saying that these two parameters are statistically different from zero.

Estimated ancestral states at root

Trait 1: approximately equally likely to be 0 or 1

Trait 2: approximately equally likely to be 0 or 1

Scaling Parameter κ

kappa ≈ 0.002 log-likelihood with kappa = -10.42 (no improvement over default independent model)