phylogenetic comparative methods

fiona jordan
university of bristol
fiona.jordan@bristol.ac.uk

QMSS 2016 @ MPI Science of Human History, Jena
05.2016

INTRODUCTION

These practical exercises are designed to give you a feel for the sorts of different questions you can ask with phylogenetic and comparative methods. We will be using a few different data sets and programs. These examples have been written with the anthropological application of PCM in mind, so there are biological, linguistic, and cultural data throughout.

In this booklet you'll find exercises on:

Viewing comparative data on phylogenies
Ancestral state reconstruction
Correlated evolution
Models of character evolution

There won't be time to cover everything, but I'll direct you to the exercises for our session.

TYPES OF DATA

While biologists frequently use molecular data for the construction of phylogenetic trees, more often than not phylogenetic comparative hypotheses use character data. This is good news if you have linguistic or cultural data, as the computer programs we wish to use will be able to understand our data as well. Cultural data are often coded in discrete form either as **binary** presence or absence (e.g. cognate set members, particular marriage practices, material culture motifs) or as **multistate** (e.g. bigman, chiefdom, state; word-order patterns). Multistate data can often be easily turned into binary variables, although if there are actual or potential "orderings" of the states you may want to retain the multistate categorisation. Data may also be **continuous** (e.g. measurements on artifacts, population density, brain size). Comparative methods differ in their handling of continuous, discrete and multi-state data, so we will address each separately.

Notes preceded by **>>** will indicate useful tips for your further use of the program.

NOTE: To give you a feel for the messiness of real data, the datasets used in these practical have been kindly provided by people who are using them in active research. *I ask that you do not use these data for analyses of your own beyond the masterclasses without EXPRESS PERMISSION from the authors.* Unpublished data will have aspects that are fabricated or slightly altered.

In this manual there are exercises to follow.

- Menu commands that you will click or type are shaded bold in **`courier font`**.
- Programs, file and directory names are in **bold type.**
- Notes in italics preceded by **>>** *will indicate useful tips for your further use of the program*.
- *Q: Denotes a question to be answered as a result of the analysis.*

CHECK: For the exercises in this handbook you should have installed the following programs on your laptop by downloading from the websites.

Mesquite (version 3.04) For visualizing and analyzing comparative data on phylogenies.
@ http://mesquiteproject.org by Maddison & Maddison for description
@ https://github.com/MesquiteProject/MesquiteCore/releases for download
You'll also need Java 7 or 8.

FigTree (version 1.4.2) For visualizing phylogenies and producing figures.
@ http://tree.bio.ed.ac.uk/software/figtree by Rambaut

BayesTraits (version 2.0) For ML and Bayesian comparative analysis.
@ http://www.evolution.reading.ac.uk/SoftwareMain.html by Pagel & Meade

SplitsTree (version 4.0) For inferring various types of networks.
@ http://splitstree.org by Huson & Bryant

CHECK: The folder of data files for the exercises is /**exercises**

NOTE on R: Much (if not most) of what these separate programs do can be implemented in the statistical environment of R, and there is a thriving community of phylogeneticists who are migrating many "old-school" aspects of phylogenetic inference and comparative inference to R. Because R requires its own steep learning curve I often teach short courses using these stand-alone programs, but you will be using some of the phylogenetic R packages in the sessions on continuous character evolution.

If things are incorrect or unclear in this manual, please let me know!
Feedback is most welcome, in person or email fiona.jordan@bristol.ac.uk

SECTION A. WORKING WITH TREES

1. Viewing trees in FigTree

We'll start with some precooked trees to get you familiar with file formats and viewing trees. As well, you should always get to know your tree(s) before embarking on any sort of comparative analysis. Here we'll use **FigTree**, useful for viewing trees and manipulating them for further analyses and/or publication.

Open **FigTree** by double-clicking on the icon.

`File > Open`. Navigate to the exercises/adzes folder and open "**adzes.trees**". These are 1000 trees found by Bayesian analysis of a data set coding Polynesian adze morphology.

You can get a feel for the different topologies of trees here. Scroll through the trees by using the arrow icons (Prev/Next). Experiment with the menus at the left to see what you can change about the trees' appearance (nothing is permanent until you save, so click around and if you get really lost just go **`> File > RevertToSaved`**).

Experiment with selecting clades/branches/nodes (buttons up top). Collapsing clades to triangles is useful for large trees. Try colouring branches and nodes, and note the different topologies you can present by rotating nodes/clades.

Q: *Try to alter Tree 11 to look the same as Figure 1 below.*

The most important aspect of FigTree is the ability to produce good graphics.
Go **`File > ExportGraphic`**. No matter what sort of graphics program you use, there will be some sort of file type there that you can export your tree in. Click on **Options** in the dialogue for further tweaking of size etc.

Don't bother to save any changes. When you're finished experimenting, quit FigTree.

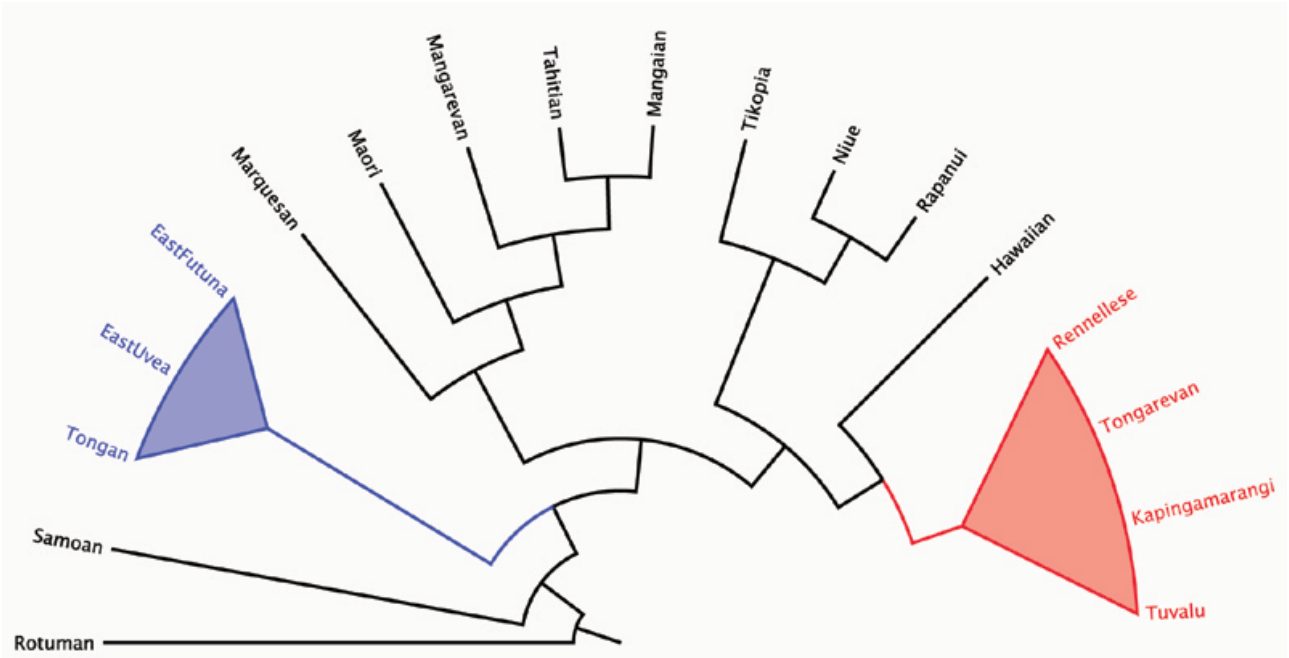


Figure 1. Phylogeny of Polynesian societies derived from adze morphology data.

2. The Nexus file format

One of the most widely used file formats for phylogenetic data is the Nexus file (Maddison, Swofford & Maddison 1997). You can read in detail about the format and the different types of "blocks" at the following websites, so I won't repeat the information here.

http://molecularevolution.org/resources/fileformats
http://www.eeb.uconn.edu/eebedia/index.php/Phylogenetics:_NEXUS_Format
http://wiki.christophchamp.com/index.php/NEXUS_file_format

As with most software, the most common problem with running an analysis is having errors in your file format, so you should become familiar with the conventions of the format.

There are two main types of Nexus files you'll encounter: *.nex (containing data and possibly trees) and *.tree(s) (trees only). We'll take a look at a combined *.nex file and talk through some of the key features.

The data here come from the important paper:

Smith, AS (2005) Are Jaffa Cakes really biscuits? Using Cladistics to classify biscuits. *Journal of Unlikely Science* 1(7): http://www.plesiosauria.com/dinobiscuits/biscuit.htm

In a text editor: **`> File > Open`** "**biscuits/biscuits.nex**". You should see the following.

```
#NEXUS
[written Tue Aug 12 14:27:15 BST 2008]

BEGIN TAXA;
	TITLE Biscuits;
	DIMENSIONS NTAX=22;
	TAXLABELS
		Simplespongecake Bourbon Digestive Chocolatedigestive Richtea Penguin
Custardcream Partyring Hobnob Jaffacake Chocchipcookie Nice Jammiedodger Figroll Minigems
Pinkwafer Shortbread Garibaldi Chocolatefinger Macaroon Rusk Gingernut
	;

END;

BEGIN CHARACTERS;
	TITLE  Biscuits;
	DIMENSIONS  NCHAR=20;
	FORMAT DATATYPE = STANDARD GAP = - MISSING = . SYMBOLS = "  0 1";
	MATRIX
	Simplespongecake    00000000000010100000
	Bourbon             11101000001100010100
	Digestive           00000100001100000001
	Chocolatedigestive  00010100001000000001
	Richtea             00000000001100000001
	Penguin             11111000010000010100
	Custardcream        11001000000100000100
	Partyring           00000001100000000010
	Hobnob              01000100000001000001
	Jaffacake           00011010000010101000
	Chocchipcookie      00100100000011000010
	Nice                10000000000100000000
	Jammiedodger        01001011000000001000
	Figroll             11001010000011101000
	Minigems            00000000110000000010
	Pinkwafer           11001000100000000010
	Shortbread          10000000001010000000
	Garibaldi           10000010000000000000
	Chocolatefinger     10010000000000000000
	Macaroon            00000110000011000000
```

```
	Rusk		00000100000011000000
	Gingernut		00100100000001000010

;

END;
BEGIN TREES;
	Title 'Trees from "biscuits.nex"';
	TRANSLATE
		1 Simplespongecake,
		2 Bourbon,
		3 Digestive,
		4 Chocolatedigestive,
		5 Richtea,
		6 Penguin,
		7 Custardcream,
		8 Partyring,
		9 Hobnob,
		10 Jaffacake,
		11 Chocchipcookie,
		12 Nice,
		13 Jammiedodger,
		14 Figroll,
		15 Minigems,
		16 Pinkwafer,
		17 Shortbread,
		18 Garibaldi,
		19 Chocolatefinger,
		20 Macaroon,
		21 Rusk,
		22 Gingernut;
	TREE shortest =
(1,((21,(20,((22,11),(9,(4,(3,5)))))),(17,((18,(12,(19,(15,8)))),((7,(6,2)),(16,(13,(14,1
0)))))));

END;
```

2. You'll notice that as well as the data, there is a single tree in this file. Open up the "**`adzes.trees`**" file in your text editor and compare. What differences do you see?

Some features of the Nexus file format:

- Structured in blocks (Data, Trees, Characters, Assumptions etc) all of which start with a "Begin" statement and end with an END; statement
- Case-insensitive:
  BEGIN
  Begin
  begin
  ... are all equivalent
- Commands end in a semi-colon ;
- Indents and whitespace don't matter.
- Comments go in square brackets [This is a comment].
- Tree notation uses brackets for nesting and colons for branch lengths
  e.g. tree1 = (A:3.2,(B:4.7, C:6.1):5);

3. Networks

With any data set our first step should be to examine what the data look like. Is there any phylogenetic signal in the data at all? Are we warranted in using phylogenetic approaches?

Here we will use **SplitsTree** to examine what the Polynesian adze data look like, and to help us decide whether we want to infer trees from these data.

SplitsTree is a program for inferring phylogenetic networks (and trees) of many different kinds. It takes a Nexus file and can use molecular, binary or distance data. The most popular algorithm for networks is NeighborNet (Bryant & Moulton 2004).

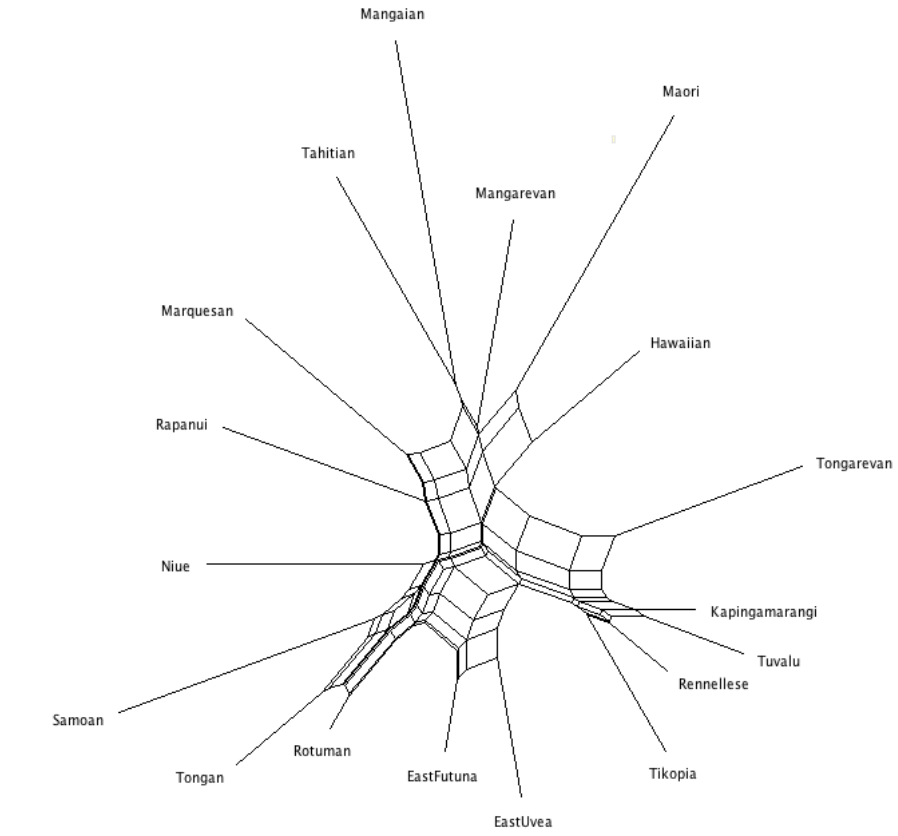
1. Open **SplitsTree**. Enter your program key if you have not done so already.

2. Navigate to the exercises/adzes folder.

```
> File
> Open "a18.adze.nex"
```

You may be prompted to choose a Character or Trees block. Choose Characters.

A network will appear looking something like the following. **NeighborNet** is the default network method: you can see this by clicking on the **Network** menu, where there is a tick next to NeighborNet.



Q: Given what you know about how NNet represents conflicting signal, what do you think about the "treeness" of these data?

You can click on the NNet itself to highlight splits and to move edges around. Click on the **Data** bar, and then the drop-down arrow next to Distances. Here you can see the distance matrix that NNet uses to compute "weakly compatible splits". The Splits drop-down arrow will give you a description of the splits themselves.

One of the useful things about NNet is that it allows you to see the effects of individual characters or taxa on your data.

> Data
> Filter Taxa

Choose two taxa to exclude and then click
> Hide
> Apply

Experiment with adding and removing taxa to see what effect they have on the NNet. If the NNet "simplifies" greatly, losing boxy structure and becoming more treelike, we can infer that those taxa have something to do with the conflicting signal. You can reset the taxa by:

> Data
> Restore all taxa

Q: Which taxa seem to have the greatest effect in terms of reticulation?

You can also add and remove characters in the same fashion using the **Data** menu.

SplitsTree allows you to also view your data as a tree.

> Trees
> UPGMA
> Draw > Phylogram

Q: Based on what you've seen of this data, would you proceed with a phylogenetic analysis? Why or why not?

There are a large number of options to compute networks and trees of different sorts in **SplitsTree**: I encourage you to experiment. Try also opening the biscuit data set and viewing it as a split network and computing a distance tree.

4. Inferring Phylogenies

You'll be learning how to use **BEAST** for phylogeny inference at the QMSS (see Simon Greenhill's session.

BEAST, MrBayes/RevBayes and **PAUP*** are probably the main software programs for inferring phylogenies. **PAUP*** has been around for many years and is excellent for parsimony analyses, and can do many things that other programs need a workaround for. It is however a commercial package, and some systems only run a command-line version. It's also been in "beta" for something like a decade. **MrBayes** and **BEAST** have to a great extent overtaken **PAUP*** as the phylogenetic inference programs of choice for biologists using sequence data, although there are a large number of other likelihood / Bayesian contenders out there (e.g. **RaxML**, **Phylip**). For those of us with binary data, **MrBayes**, **BayesPhylogenies** and **BEAST** are the best options for tree inference.

SECTION B. COMPARATIVE METHODS

EXAMING THE EVOLUTION OF DISCRETE TRAITS

We'll use a combination of programs (Mesquite and BayesTraits) to look at ancestral state inference, models of change, and correlated evolution. Discrete and continuous characters each require distinct approaches, and, as most cultural and linguistic data tends to be discrete, we'll concentrate on those models.

Program used: MESQUITE v. 3.04
http://mesquiteproject.org/Mesquite_Folder/docs/mesquite/manual.html

Mesquite is a program for examining the evolution of characters on phylogenies, rather than building trees (although it can do that in a limited fashion).

"Mesquite is a general system for phylogenetic computing to which different programmers could contribute modules. Bringing different analytical tools into a common system increases possible analyses more than additively. A second goal of Mesquite is to provide a graphical user interface that will operate, more or less without modification, under different operating systems (being written in Java)."

These properties make it a useful program to have in your repertoire, and the friendly GUI takes away some of the learning-curve-nightmares that other programs induce! It is freely available for download at the website indicated.

>> Menu commands that you will follow are shaded bold in **`courier font`**.
>> Programs, file and directory names are in **bold type**.

The first thing is always to see how our characters look when plotted on a tree. We'll do this in Mesquite, and use a likelihood implementation of the Discrete test to warm-up.

1. Getting started with Mesquite 

Open **Mesquite** by double-clicking on the Mesquite icon. Two windows will flash up as the program loads all of the modules - you will see some of the names of these. The Log window will remain open at the "Log" tab and will tell you what you are doing and what you have done. If you get lost at any time it can be useful to refer to this window. The "Search" tab is a useful way to find help or features. Click on the "Projects and Files" tab. You should see "No Projects Open".

Go **`File > OpenFile`**.

Navigate to the "**exercises/ase**" folder and click on the project "**res_fish.mesquite.nex**". A Project window will open. Your Project window should allow you to see options to select aspects of the characters and trees. You may need to right click on the "Traits" / "Trees" or the little blue arrows (depending on Mesquite version) so that they drop down and reveal options. Contextual menu bars, and new tabs, will then open.

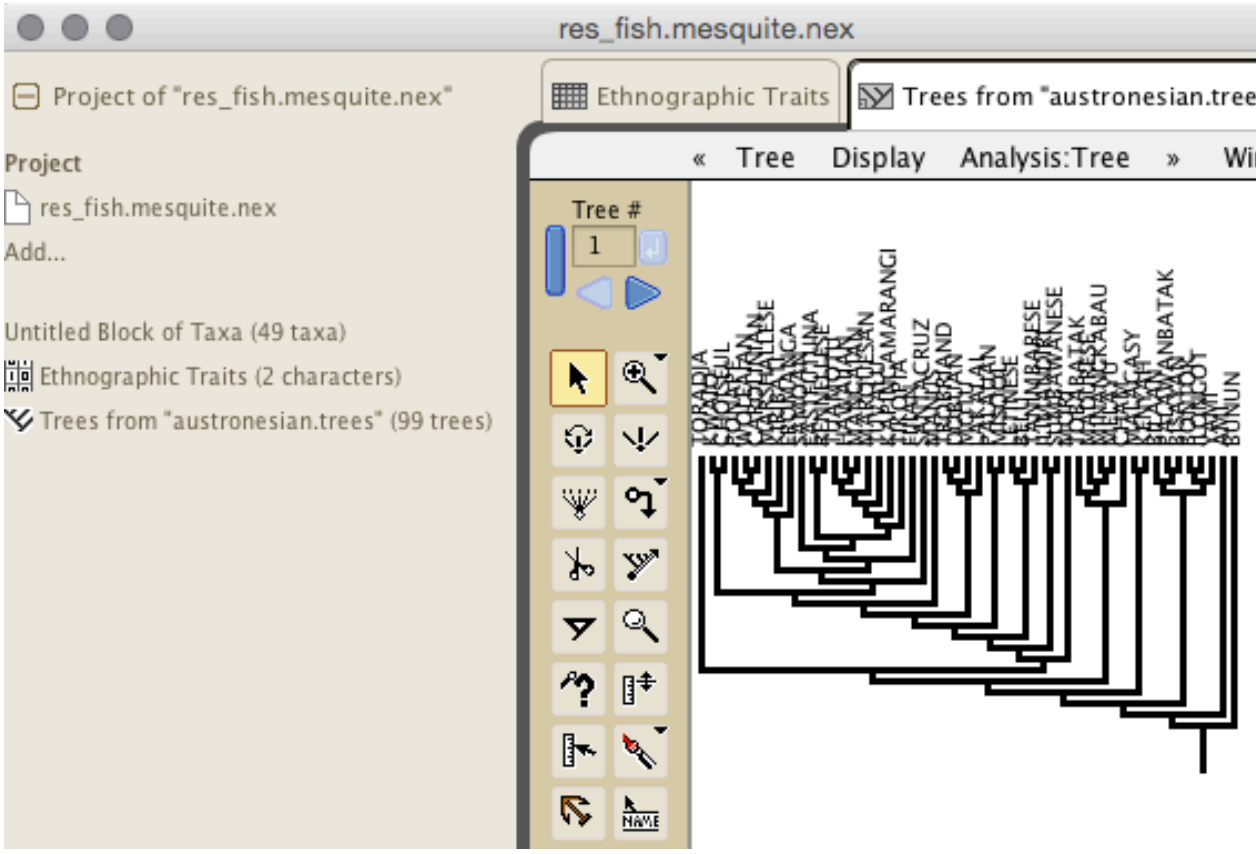


FIGURE 3

About the data:
This file contains data on 49 Austronesian-speaking societies. There are 99 trees in this file, derived from a Bayesian analysis of language data from the Austronesian Comparative Dictionary, kindly provided by Simon Greenhill and Russell Gray. You can see the data for over 1000 Austronesian languages at their website: http://language.psy.auckland.ac.nz/.

2. Characters

Click on the **`Character Matrix`** (here, "Ethnographic Traits") for **`Show Matrix`** to see the two characters. They are coded for two variables, postmarital residence and the degree to which that society depends on fishing for subsistence. Both of these variables have been taken from the Ethnographic Atlas and coded as binary.

RESIDENCE	0 = matrilocal, where the husband lives with the wife's kin 1 = patrilocal, where the wife lives with the husband's kin
FISHING	0 = low dependence 1 = high dependence

We have these characters because eventually we'll test the hypothesis that societies where men are absent for long periods of time are more likely to sustain matrilocality (female-biased residence), and we're using fishing as a proxy for male absence. But first we'll do some more basic things.

3. Viewing the trees and characters

Click on **`Trees from "Austronesian…"`** and select **`View Trees`** to open the Tree Window tab.

You should now see a new window that displays Tree 1 of these societies (like in Figure 4).

>> In Mesquite, many of the menu options are tab-specific and the menu will change for you. For instance, you need to be in a Character Tab to see all of the relevant options for character manipulation and analysis.

Click through the trees using the arrows to see the variation in the Bayesian tree sample.

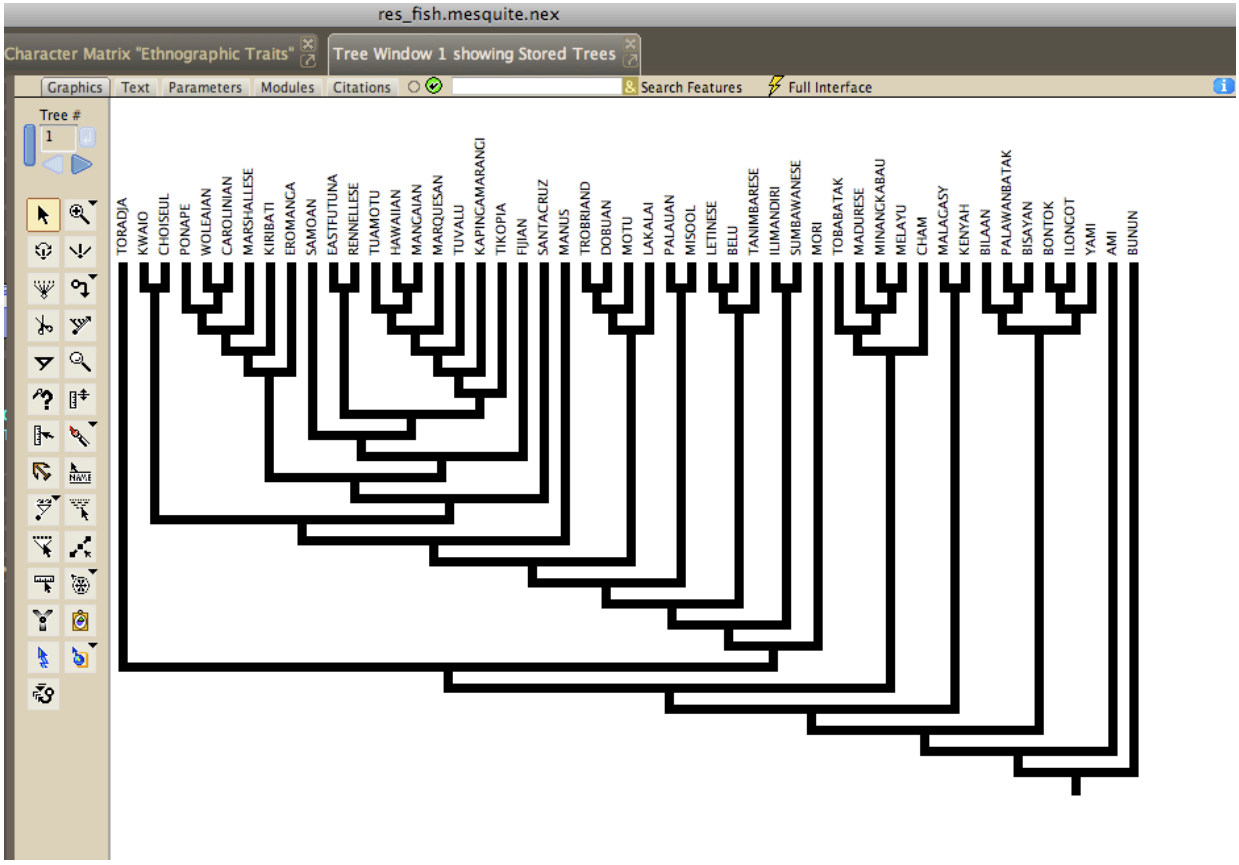


FIGURE 4

4. Viewing the characters on the trees

Return to tree #1. Ladderize the tree view using the  button. Click this icon, then at the root of the tree.

To view the characters plotted on the tree, we will use **Trace Character History**.

> **Analysis:Tree**
> **Trace Character History**
> **Stored Characters** (if prompted for source of characters)
> **Parsimony Ancestral States**

Your branches in the tree window should now appear as shaded bars that correspond with the character states. A **Trace Character** palette has appeared, with a key to the character states, and the number of steps required for this character's evolution under a parsimony model.

Q: How many steps do you observe for RESIDENCE? What about for FISHING?

Note that we now have a new item in the Menu Bar **Trace**. To close this, go **Trace > CloseTrace**. This is a general feature of Mesquite: contextual menus appear whenever you select a procedure that has many more options (indicated by "...").

5. Inferring ancestral states

To visualize these data in a different way:

> **Display**
> **Tree Form**
> **Balls&Sticks**

Then repeat the **Trace Character History** steps:

Try also

> **Analysis:Tree**
> **Trace Character History**
> **Stored Characters** (if prompted for source of characters)
> **Likelihood Ancestral States**
> **Current probability models**

The current probability model for Likelihood should be a one-parameter model. You can change this under the **Trace > Probability Model** options.

Hover the mouse over the "pie" nodes to get proportions in the likelihood mode. These will appear in the Trace Character palette.

Q: What residence pattern is reconstructed at the root of the tree under parsimony? Under likelihood?

Q: By eyeballing these data, do you think there is any evidence for coevolution?

SECTION C. BAYESTRAITS FOR CHARACTER EVOLUTION

We would like to be able to estimate ancestral states, test for correlated evolution, etc, and take account of both phylogenetic and character uncertainty. A way to do this is to use Bayesian Markov-chain Monte Carlo methods (see the Sidebar on the next page). Some of the mathematical ideas may be new, but as we do not have time to go into all the complexities of this approach, we can black-box the theory a little in order to take advantage of the methodology. Should you wish to use these approaches in your own analyses, we can recommend further reading and guidance on matters such as models, posteriors and priors, and hypothesis testing as appropriate.

Two important points to remember:

(1) A Bayesian MCMC analysis integrates our parameters of interest over a statistically-derived sample of trees AND over many possible models of evolution. It thus addresses both phylogenetic and character uncertainty in a statistically principled way - and doesn't take a lifetime to compute.

(2) We gain a distribution (called a posterior probability distribution) of the value of the parameters of interest at a node, rather than a point value.

A single phylogeny is just one hypothesis about evolution. Increasingly, biologists use approaches that account for uncertainty in historical reconstruction. Given that many skeptics consider uncertainty about the nature of linguistic and cultural transmission processes to be the biggest problem for cultural phylogenetics, anthropologists should take advantage of these new approaches as much as possible.

Ronquist (2004; a very good paper!) describes two sources of error that can produce uncertainty in the results obtained in comparative analyses:

1. **Phylogenetic uncertainty**: errors attributable to the phylogeny itself, caused by homoplasy/horizontal transmission in the characters used to construct the phylogeny, multiple plausible reconstructions, multiple character histories etc.

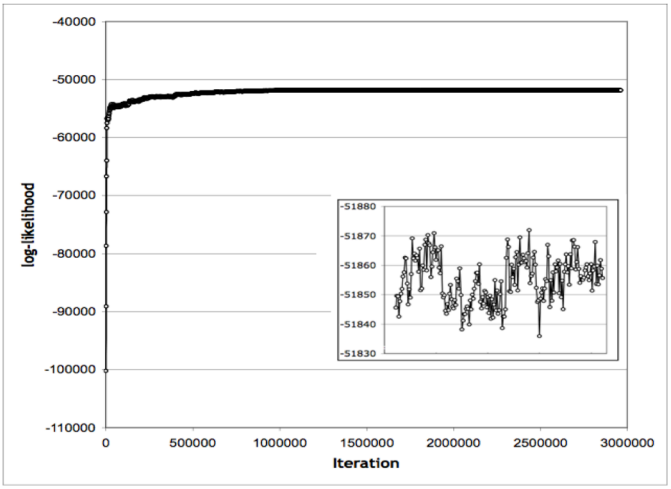
2. **Mapping/character uncertainty**: this is error associated with the reconstructing the evolution of a character on any given phylogeny, caused by different rates of evolution, character specification, the likelihood of changes between states etc.

Bayesian methods allow us to use a sample of trees rather than a single tree in our comparative analyses, thus addressing in a principled fashion the first source of error. Bayesian comparative methods also address the second source of error, by providing a probability distribution of character states across the sample of trees as well. Whether you use parsimony, likelihood, or Bayesian tree-building methods, we strongly encourage you to test your hypotheses on more than one tree.

Sidebar: BAYESIAN MCMC METHODS

Likelihood methods have many desirable properties: they use all the information in the data (e.g. branch lengths) and they incorporate explicit and complex models of evolution. However, ML methods are hampered by computational inefficiency (calculation times). Bayesian inference is a statistical approach where the model of evolution and the model parameters are treated as random variables, and the data treated as fixed observations (Ronquist 2004). Applied to the phylogenetic context, these methods simulate a "universe of solutions" which can be sampled to obtain phylogenies or outcomes in proportion to their likelihood. The Markov chain Monte Carlo algorithm allows likelihood methods to be computationally tractable on large data sets (Archibald et al. 2003).

Bayesian phylogenetic methods employ a Markov chain Monte Carlo (MCMC) algorithm to take a "random walk" through a parameter space that approximates a probability distribution[a]. Each step in the walk, or chain, involves a random modification of parameters such as tree topology or branch length (when building trees), or node state and transition rates (when doing a comparative analysis). At each iteration (see figure) we sample these values and calculate the likelihood[b]. The Markov chain thus only jumps to a new state as a function of the current state and does not, like parsimony, hill-climb along a gradient imposed by previous states (Pagel and Meade 2005).



Plot of the Markov chain. At first the likelihoods fluctuate wildly but after a burn-in period settle into stationarity where they fluctuate about a mean (inset).

The chain thus visits areas of "parameter-space" in proportion to their posterior probability. At length, the chain reaches an equilibrium distribution where it is not seeking an optimally best result but is sampling better and worse trees/parameters into a sample used to approximate the posterior distribution of all results.

(a) Markov models use a rate matrix which describes the transition between states of the data (in binary-coded models, the change from 0 →1 and 1→0) in an infinitely small period of time. To gain the transition rates for a given data set, the rate matrix is integrated over time, and this matrix is used to estimate the transition rates for the observable data (Pagel 1994; Ronquist 2004).
(b) The likelihood depends on the configuration of parameters obtained at each rearrangement. We call this the posterior probability. If this is larger than the prior probability (a value specified by our model), the step is taken. If smaller, then the action depends on the ratio of the new to the current posteriors.

BayesTraits

If you want to run BayesTraits from wherever you are on your computer, you can place the executable file in your ~bin folder, add it to your path in your .bash_profile, and make it executable. If that was gobbledegook to you, ignore that for now – we'll assume that the program is in the same folder as tree and trait files.

GETTING FAMILIAR WITH THE TERMINAL

Skip this section if you are already familiar with a command-line environment for Unix.

Click the Terminal (or similar) application icon on the desktop. On a Mac, Terminal lives in **Applications > Utilities**. On a PC, use a Command Prompt or something like PowerShell.

Commands that you will type in are shaded like this in **`> courier font`** in the directions below. Filenames/directories and outputs that will appear on the screen are in **`plain bold courier font`**. You can also get help from within the Terminal itself by using the "manual" command `man`. For our purposes, the commands you will be likely to use today are:

```
cd ................. change current directory
ls ................. show contents of directory, in alphabetical order
logout ............. logs off system
rm ................. remove files
mkdir .............. make a directory
man [command] ...... shows help on a specific command
pico ............... opens a basic text editor
ctrl + c ........... cancels a process
./[program files] .. executes a program and files
```

When you have started up a **Terminal "shell"**, you are always located in a particular directory. The default starting location is your home directory. To see where you are, enter the **`pwd`** (print working directory) command. Navigate from there to the directory "**`qmss/PCM_discrete/exercises`**" which you have placed somewhere accessible like your working folder.

`> cd qmss/PCM_discrete/exercises/ase`

To go back up one level (in this case to the folder "**exercises**"), you can type

`> cd ..`

To go back to the ase folder

`> cd ase`

Check that the files you will need are in the directory "**`ase`**" by listing the contents:

`> ls`

You should have the following:

BayesTraitsV2 (the program file)
ancestralstates.txt (instructions for the ancestral states section)

AN83.trees (a treefile)
res.PMP/ (nodefiles)
res.POC/ (nodefiles)
res83_binary.trait (a trait file where variables are coded as binary)
res83_multi2.trait (a trait file where variables are coded as multistate)

res_fish.mesquite.nex
res83_multi.trait (a trait file)

pico.txt (a very brief introduction to a text editor)

Use the Pico text editor to read in the file "**pico.txt**". Your in-shell text editor might also be called something like Nano or Vi.

> pico pico.txt (alternatively, you could just type **pico** and then open the file using **^R**)

Read through the file and exit pico (**CTRL+X**). Remove the file from the directory (don't worry, we'll give you a copy later if you want it).

> rm pico.txt

Have a look at the Nexus treefile **AN83.trees** in a text editor. The format should be familiar to you now.

1. Inferring ancestral states

There are many different ways to test evolutionary scenarios using programs like **Mesquite** and **BayesTraits**. Inferring the model of evolution and ancestral states generally go hand-in-hand, as the conclusions we can drawn from one aspect depend on the other. Here we are going to look at a subset of data on Austronesian postmarital residence as similarly reported in Jordan et al (2009) and Fortunato and Jordan (2011).

Multistate data on postmarital residence from 83 Austronesian societies in the Ethnographic Atlas has been recoded into a binary coding scheme. In the Inferring Models exercise we will create a multistate coding scheme from the same data.

This variable in the Ethnographic Atlas has the following coding assignments:

*0 = missing
*1 = avunculocal
2 = AMBILOCAL: residence can be with either the kin of husband or wife, fairly even split
*3 = optionally avunculocal/uxurilocal
*4 = optionally avunculocal/virilocal
5 = MATRILOCAL: residence with matrilineal kin group of the wife
6 = NEOLOCAL: residence in a place not determined by kin ties of either spouse
*7 = no common residence
8 = PATRILOCAL: residence with patrilineal kin group of the husband
9 = UXORILOCAL: residence with wife / wife's kin
10 = VIRILOCAL: residence with husband / husband's kin

Note: Matrilocal is a subset of uxorilocal; similarly, patrilocal is a subset of virilocal.
* a small number of societies with these variables have been removed for this demonstration

These data have been recoded from the original assignments to the following in "**ase/res83_binary.trait**":

matri/uxo = 5/9 = 0
patri/viri = 8/10 = 1
ambi = 2 = 01
neo = 6 = missing (-)

You can open this trait file in a text editor to examine the assignments. This coding takes advantage of the fact that taxa can be assigned more than one state, so ambilocality can be given both uxorilocal and virilocal codes. The disadvantage is that the model will only reconstruct the probability of single states, not 01 (=ambilocality). Also, as we will see, tests of correlated evolution are confined to two strictly discrete states and dual assignments are not possible. Nonetheless, it is an effective way to code polymorphic or linguistic traits: for example, a language with two possible word orders could be coded in this fashion.

Note that any information about neolocality is lost as we code it as missing. We return to this issue in the Inferring Models exercises.

The treefile for this exercise is **ase/AN83.trees**. You may wish to open this and the trait file in Mesquite to see how the data looks on the trees. Follow the instructions as for previous exercises.

>> *It is usually a good idea to start any Bayesian analysis with a quick look at ML solutions to give you a feel for some of the values of the parameters.*

You will need to have the program BayesTraitsV2 in the same directory as the tree and taxa file. Start the program by typing in the Terminal window

```
./BayesTraitsV2 AN83.trees res83_binary.trait
```

The syntax here is always

```
[./program] [treefile] [traitfile]
```

Press return. ⏎

>> **./** *("dot slash) is a Unix command which tells the shell to look in the current directory for the command whose name you are typing. If you have set up your computer such that BayesPhylogenies is executable from your ~bin, then you won't need the dot-slash. For more info:* http://www.linfo.org/dot_slash.html

Here's what you'll see.

```
BayesTraits V2.0 (Jul 11 2014)
Mark Pagel and Andrew Meade
www.evolution.reading.ac.uk


Please select the model of evolution to use.
1)    MultiState
4)    Continuous: Random Walk (Model A)
5)    Continuous: Directional (Model B)
7)    Independent Contrast
8)    Independent Contrast: Correlation
```

When prompted for the model of evolution choose **Multistate** by typing **1**. Choose **ML** as the analysis method by also typing **1**.

>> *Note that there are 8 different options to choose from, and not all are displayed. MultiState will accept both binary and multistate discrete data.*

You'll get something like the following output and the program will wait for your input. We'll go through these options and talk about what they mean.

```
Model:                          MultiState
Tree File Name:                 AN83.trees
Data File Name:                 res83_binary.trait
Log File Name:                  res83_binary.trait.log.txt
Summary:                        False
Seed                            4120938273
Analsis Type:                   Maximum Likelihood
ML attempt per tree:            10
Precision:                      64 bits
Cores:                          1
No of Rates:                    2
Base frequency (PI's)           None
Character Symbols:              0,1
Using a covarion model:         False
Restrictions:
    q01                         None
    q10                         None
Tree Information
     Trees:                     50
     Taxa:                      83
     Sites:                     1
     States:                    2>
```

You can get this output at any time by typing "`info`".

>>*The BayesTraits manual explains all of the syntax and commands in the glossary. When you do your own analyses it is likely you'll need to have the manual open until you're familiar with the commands.*

Give the analysis logfile a meaningful name using the **`lf`** (logfile) command

```
lf res83_ancestral.log
```

```
run
```

Multistate runs an ML analysis on each tree and infers the transition rates for the parameters and the root node. You'll get something like the below. We can't average or integrate over these trees, but we can can record the ranges.

```
Tree No    Lh    q01   q10   Root P(0)  Root P(1)
1     -31.573138 0.953710   0.255761   0.637091   0.362909
2     -31.530091 0.856847   0.228325   0.722856   0.277144
3     -31.264389 0.910596   0.257854   0.542191   0.457809
```

Q: What (roughly) is the range of the Lh? What are the transition rates and what is your interpretation of them? What do you think the most probable root reconstruction is?

Start another new analysis (remember, **CTRL+C** to restart). Lets now restrict the transition rates to be equal by typing

```
restrict q01 q10
```

The syntax here is always the same. If we wished to set q01 to zero (or any other value), we would type `restrict alpha1 0.`

Check the restriction has been made by typing "`info`" and examining the Restrictions.

Now, give the analysis logfile a meaningful name

```
lf res83_eqrates.log
```

```
run
```

Q: What does setting the rates of change to be equal do to the Lh? How would you interpret this?

We might want to infer ancestral states for nodes other than the root. To do this we tell the program what taxa are descendants of the nodes we wish to infer. These are listed in the files res.PMP and res.POC. Re-run your analysis as above, but this time before typing run, paste the node information in as instructed. Make sure you give your logfile a meaningful name!

Q: What states can you infer for PMP? What about POc?

If we have time, we'll also cover how to "fossilize" ancestral nodes to be one state or another, in order to test our inferences for robustness.

Recall that you have only been doing maximum likelihood analyses here! In the Correlated Evolution section we'll do these in a Bayesian context as well, and the principles will be the same such that you can apply them to ancestral states and models of evolution.

2. Inferring a model of evolution

Our data coding has been binary so far, which causes us to lose some information in our data. We can be a bit more meaningful by inferring a multistate model of sequential evolution to understand how one trait changes between states. This is something we often wish to do: infer a diachronic process from synchronic data, using a phylogeny.

Switch to the **exercises/models** folder in your Terminal.

```
cd ../models
ls
```

AN83.trees (a treefile)
res_fish.mesquite.nex
res83_multi.trait (a trait file)
res83_multi2.trait (a trait file where variables are coded as multistate)

The trait file **res83_multi.trait** is coded for postmarital residence as follows:

0 = 5/9 uxorilocal
1 = 8/10 virilocal
2 = 2 ambilocal
3 = 6 neolocal

Start off with a basic analysis.

```
./BayesTraits AN83.trees res83_multi.trait
```

When prompted for the model of evolution choose **Multistate** by typing `1`. Choose **ML** as the analysis method by also typing `1`.

```
lf res83_multi.log
```

```
run
```

This'll take some time to go through the fifty trees, as we're estimating a much more complex model with four states. You could also use **res83_multi2.trait** which has ambilocal societies coded as polymorphic (01) if you wish. As before, you could estimate different ancestral states other than the root by using the node designations.

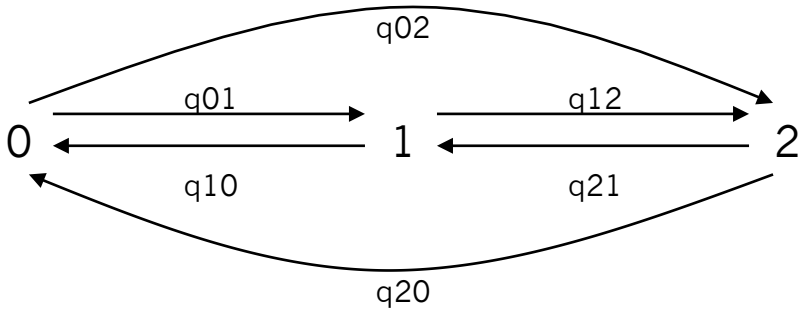
Q: What are the rates of each transition? What are the ancestral states for the root?

We'll use the simpler coding to infer our model, so start a new analysis with

```
./BayesTraits AN83.trees res83_multi2.trait
```

Run the analysis and annotate the figure below with the transition rates.

Q: Are any of them zero, or close to zero? Are any of them much higher than the others? Identify which ones are close to zero and so might be "taken out" of the model.



To exclude these transitions we set them to zero.

```
restrict q12 0
```

Check the restriction has been made by typing "`info`" and examining the Restrictions.

```
run
```

Run your analysis and note the likelihood.

Q: Does excluding these transitions make a difference to the likelihood? Now, pick another non-zero transition and repeat. What about the likelihood this time? How would you go about constructing a set of restrictions to test a stepwise model versus one where all transitions were possible?

We'll talk about more general ways to investigate directional or sequential models of evolutionary change, as this is a very useful set of techniques for cultural and linguistic evolution.

D. CORRELATED EVOLUTION

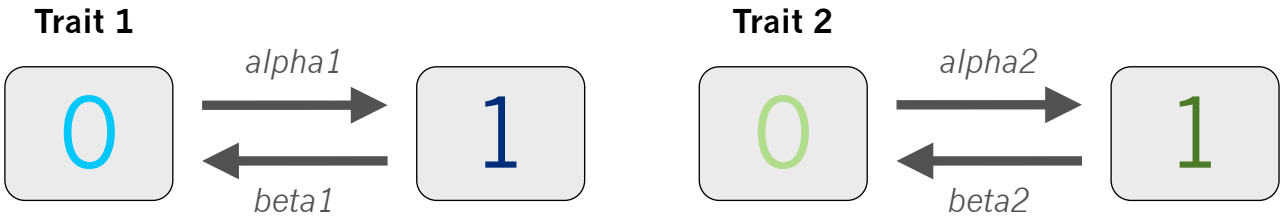
1. Coevolution: Discrete in Mesquite

CORRELATED EVOLUTION
Recall that in any situation where taxa may be hierarchically related, we can't treat data points as independent instances of evolution. Phylogenies impose a correlation on characters, and so by controlling for history, a comparative method allows us to test if two traits are co-evolving together more than would be expected against that background.

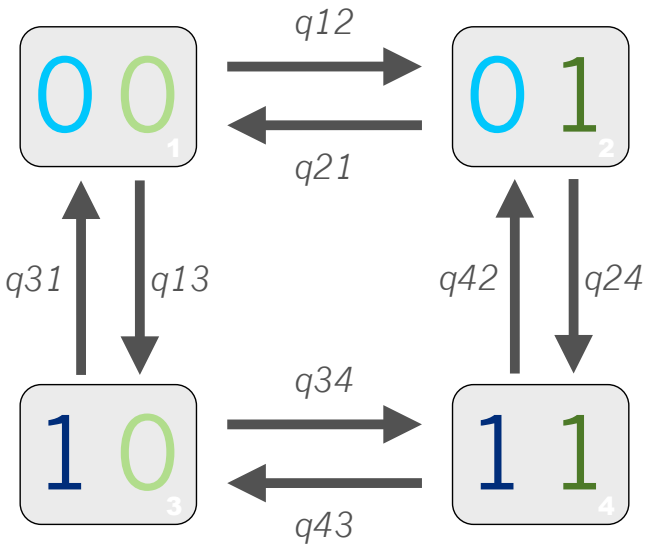
0. Independent and Dependent models

At the core of these approaches is the formalisation of the hypothesis about correlated evolution into (a) independent and (b) dependent models of evolution.

(a) **Independent**: a model in which the two traits evolve independently. With two binary traits, this gives us four parameters. This can be represented as:



(b) **Dependent**: a model in which the state of one trait can influence the probability of change in the other trait. With two binary traits, this gives us eight parameters, as each transition between states depends on the state in the other trait.



You'll see these diagrams quite often, so it's worth taking the time to understand what they are saying.

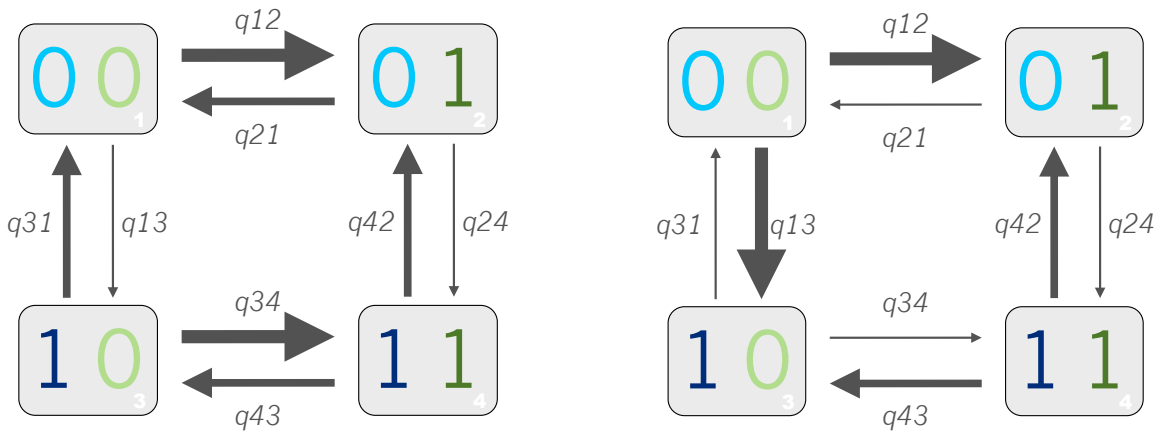
In this framework, we are estimating the rate of change from one state (in the grey boxes) to another. These parameters are the transition-rates. A rate matrix is often denoted as a Q-matrix, so you will see these parameters sometimes denoted as (e.g.) q01 or q42.

We can do useful things like fix these rates to certain values, or make them equal, or set them to zero. Outside of the correlated evolution paradigm, we can also use transition rate parameters to estimated multistate models of evolutionary change. If we have time, we can cover this functionality.

In general, whether we are operating in the ML or Bayesian framework, we obtain the likelihood for each of the Independent and Dependent models and compare them using an appropriate test. We can then make a statement about the evidence we have for correlated evolution, and, if we do, interpret the "dynamics" of the dependent model according to our hypothesis and other information such as ancestral state estimates etc.

EXERCISE:

The Independent is a special case of the Dependent model (i.e., the Dependent can be reduced to the Independent). In the following Dependent models, are they equivalent to the Independent?



Q: If the blue trait represents social organization (light = extended family households, dark = nuclear family households) and green represented main mode of subsistence (light = fishing, dark = agriculture), what might you conclude about the second model?

There is a version of Pagel's Discrete test implemented in Mesquite, so we'll do that before before moving on to BayesTraits.

Make sure you have the residence and fishing data open and in the **Tree** tab go:

> Analysis:Tree > Correlation Analysis

Check the box for **Present p value**. Leave the other options as they are. Choosing 100 simulations will allow us to estimate $p < 0.05$. To get more accurate, enter a larger number of simulations. Leave it at 100 for now. Depending on the speed of your processor, this analysis could take a few minutes.

Mesquite will ask you to specify X and Y as the first two characters in the matrix and will do the analysis on the active tree. The likelihood analysis will start immediately and be displayed to the right of the tree. Progress indicators will show how far through the analysis you are - it will take a few minutes. In this "old-school" version of Discrete, the program simulates data against which we can compare our observed likelihood ratio (that is, the difference in the dependent versus the independent model). From this simulation, we can tell if our observed LR is significant. As it is running, look at the likelihoods of each model.

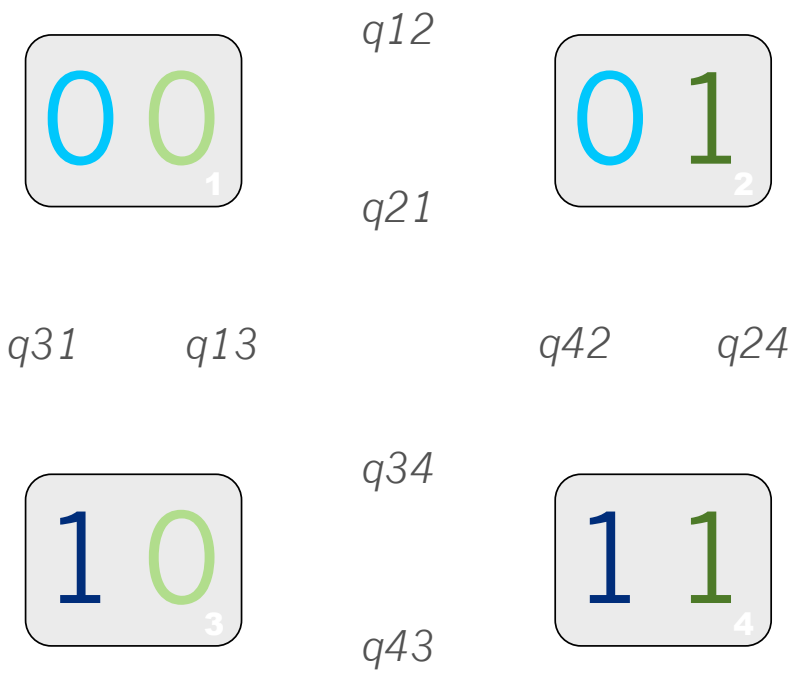
Your output should look something (with some random variation) like this:

```
X: RESIDENCE
Y: FISHING
Calculation: Pagel's 1994 test of correlated (discrete) character evolution

For four parameter model :
q12(alpha1) =1.24969820859169E-5
q13(alpha2) = 0.4782037335839831
q21(beta1) = 0.1861818065107821
q31(beta2) = 0.3749437876570352

log Likelihood is -58.667583497548435

For eight parameter model :
q12 = 0.15994672878773994
q13 = 4.309813961332921E-5
q21 = 0.13104732362816127
q31 = 3.438043494614995E-5
q24 = 6.118925355211004E-7
q34 = 0.6647441054710936
q42 = 0.5031226645117804
q43 = 0.9083689609004805

log Likelihood is -53.11937277014913

Difference is 5.548210727399308
p-value from 100 simulations is 0.0

p-value = 0.0 (from 100 simulations)
```

This tells us that there is a significant difference between the independent (4-parameter) and dependent (8-parameter) model. However, remember that this is just on one single phylogeny with one model of evolution.

Q: Which is better, the dependent or the independent?

Q: Sketch the results on the flow diagram below. Label the traits, and indicate your ancestral state estimation.



Q: What would this result mean in words?

Close the **Correlation Analysis** tab.

2. Using BayesTraits for correlated evolution

We will be using comparative cultural data from the Ethnographic Atlas on postmarital residence and descent form in 37 Austronesian societies. These data are a reduced set of those that were used in:

> Jordan, F.M. and Mace, R. 2007 Changes in post-marital residence precede changes in descent systems in Austronesian societies. *European Human Behaviour and Evolution Conference 2007* London School of Economics, London, UK. [http://eprints.ucl.ac.uk/14488]

where we tested GP Murdock's hypothesis that changes in residence drove change in descent to align, e.g. so patrilocal societies became patrilineal.

You will now work in the **/correlation** folder. Before getting started, open the treefile **an_37.trees** in a tree viewing program (TreeView, Mesquite etc) to get a feel for the data you're looking at.

Similarly, open the comparative data in a text file to make sure you know what is going in to your analysis. The file **resdesData.md** tells you what you're looking at.

For now we'll use the **resdes.trait** file in which the coding scheme was as follows:
The first column is the norm of postmarital residence: 0 = matrilocality, 1 = patrilocality.
The second column is the descent system norm: 0 = matrilineal, 1 = patrilineal.

You can see these plotted on a consensus tree in the file **rd37.pdf** The tips are labelled with two columns of dots indicating the residence and descent system norms for the cultural groups associated with each of the languages.

Column 1, residence: maroon = matrilocality, blue = patrilocality
Column 2, descent: maroon = matrilineality, blue = patrilineality

>> As before, we start any Bayesian analysis with a quick look at ML solutions to give you a feel for some of the values of the parameters.

Start the program by typing in the Terminal window

```
./BayesTraitsV2 an_37.trees resdes.trait
```

When prompted for the model of evolution choose **Discrete: Independent** by typing **2**.
Choose **ML** as the analysis method by also typing **1**.

You'll get the following output and the program will wait for your input:

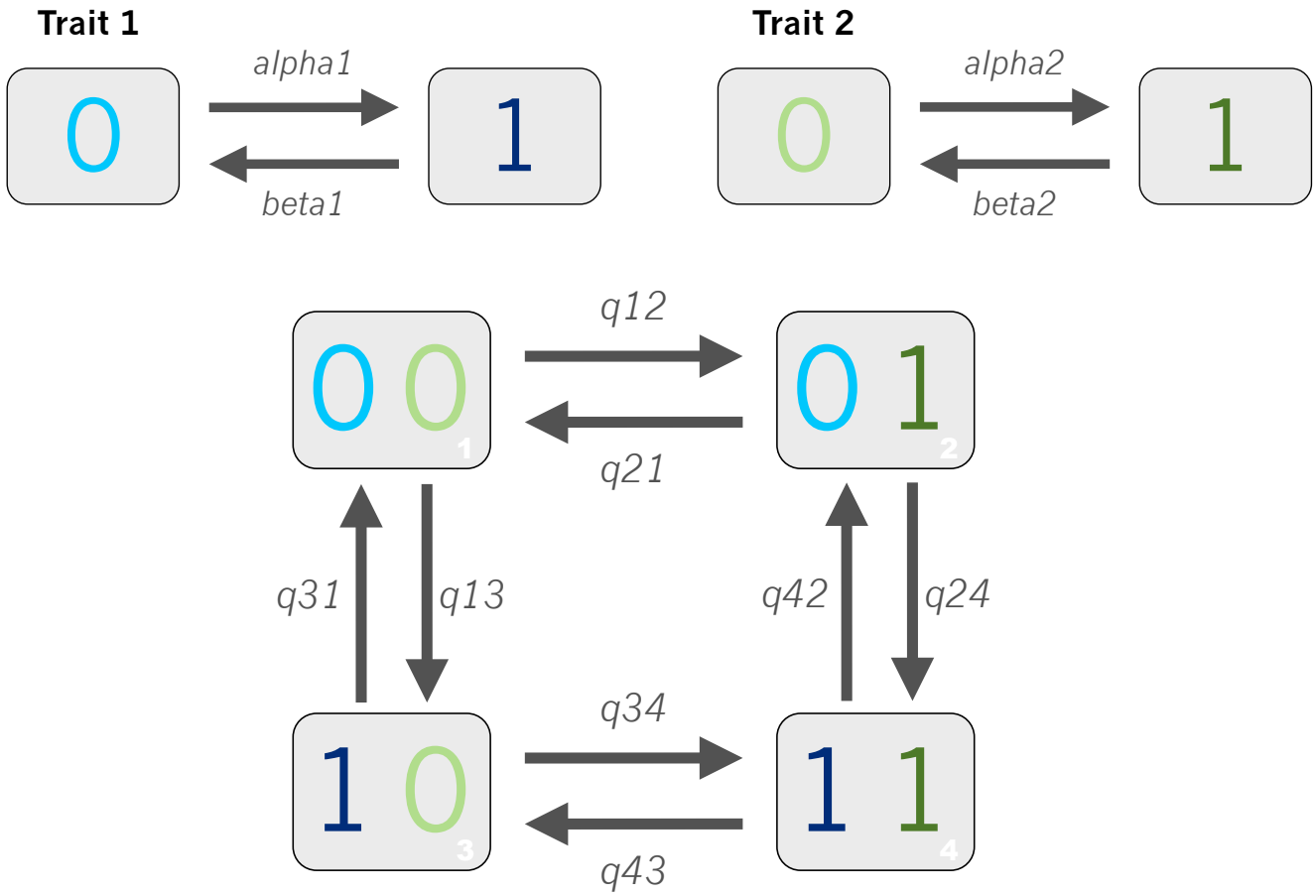
```
Options:
Model:                          Discrete: Independent
Tree File Name:                 an_37.trees
Data File Name:                 resdes.trait
Log File Name:                  resdes.trait.log.txt
Summary:                        False
Seed                            1163954356
Analsis Type:                   Maximum Likelihood
ML attempt per tree:            10
Precision:                      64 bits
Cores:                          1
No of Rates:                    4
Base frequency (PI's)           None
```

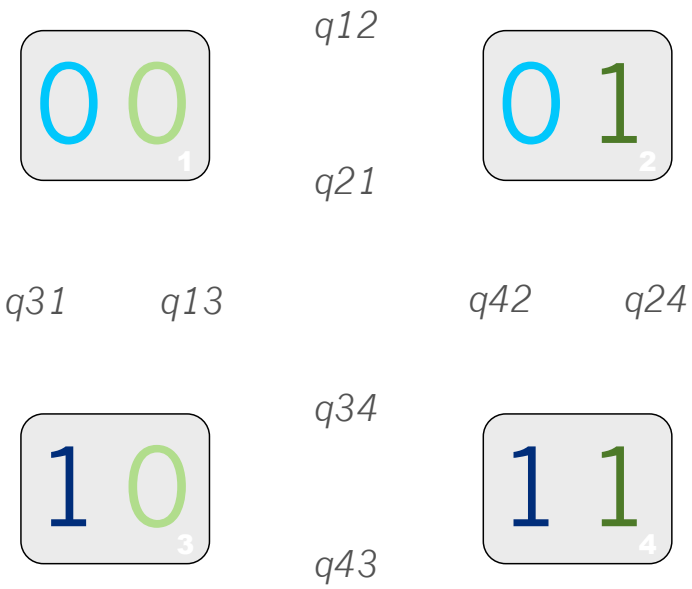
```
Character Symbols:                00,01,10,11
Using a covarion model:           False
Restrictions:
     alpha1                       None
     beta1                        None
     alpha2                       None
     beta2                        None
Tree Information
      Trees:                      50
      Taxa:                       37
      Sites:                      1
      States:                     4
>
```

You can get this output at any time by typing "`info`".

We have two binary traits listed in our **`resdes.trait`** file, so the Independent model will estimate two rates for each trait separately. Recall from earlier that for each trait we estimate these transition rate parameters as the change from one state to the other state, and we denote them here in the Independent Model as alpha1/beta1 for Trait 1 and alpha2/beta2 for Trait 2.

You might want to label these for jotting down the traits and results as you go:




q12
q21
0 0
1
0 1
2
q31
q13
q42
q24
q34
1 0
3
1 1
4
q43

Independent model under ML

To use the program defaults, just type "**run**". The ML calculations should zip through quickly. You can stop the run at any time by pressing **CTRL+C**. Following is some output (note that yours will be slightly different as ML will start from a different random seed and is fitting a probabilistic model). It will appear on screen but will also be written to a file with "**log.txt**" appended. Invoke the program again, and this time change the name of the logfile with **lf**.

```
./BayesTraits an_37.trees resdes.trait
2
1
lf resdes_MLind.log
```

>> *If you are running multiple analyses off the same input files it is good to have a regular system for your logfiles, as they will always revert to the defaults and may write over one another.*

Below is the first 20 lines of the output. Yours should look similar. The columns are, in order, the tree number in the file, the likelihood, the four rate parameters, and the likelihood of each state for each trait (four in all) for the root.

Tree No	Lh	alpha1	beta1	alpha2	beta2	Root-T1-P(0)	RootT-1-P(1)	Root-T2-P(0)	Root-T2-P(1)
1	-44.962520	0.547562	0.213409	0.431028	0.457483	2.993356	1.006644	2.652080	1.347920
2	-44.982773	0.541715	0.206148	0.443404	0.471973	3.226194	0.773806	2.736489	1.263511
3	-45.097890	0.549302	0.236177	0.415107	0.479618	2.333833	1.666167	1.922090	2.077910
4	-45.056142	0.524128	0.210662	0.409956	0.464472	3.105800	0.894200	2.569830	1.430170
5	-45.082789	0.575753	0.226554	0.441762	0.466294	3.004397	0.995603	2.894180	1.105820
6	-45.097010	0.568696	0.232818	0.429875	0.485395	2.948173	1.051827	2.406618	1.593382
7	-45.376068	0.632028	0.272317	0.438903	0.466993	2.580194	1.419806	2.741586	1.258414
8	-44.575631	0.511226	0.190482	0.423008	0.459044	3.396160	0.603840	2.718583	1.281417
9	-44.925419	0.567948	0.240352	0.428607	0.490394	2.613920	1.386080	2.189708	1.810292
10	-44.949700	0.579877	0.227899	0.442710	0.478659	3.035910	0.964090	2.621318	1.378682
11	-45.141794	0.535802	0.210647	0.454632	0.489486	2.971432	1.028568	2.583154	1.416846
12	-44.752648	0.496336	0.198291	0.421984	0.467739	3.209729	0.790271	2.725696	1.274304
13	-45.214233	0.572554	0.244238	0.443801	0.488389	2.846785	1.153215	2.659329	1.340671
14	-45.040654	0.583031	0.227692	0.467614	0.509288	2.995974	1.004026	2.650338	1.349662
15	-44.775332	0.525794	0.202960	0.445576	0.461137	3.041795	0.958205	2.599052	1.400948
16	-44.774813	0.483972	0.195866	0.456165	0.504189	3.032254	0.967746	2.465502	1.534498
17	-45.274376	0.562448	0.233476	0.436466	0.481816	2.824700	1.175300	2.628719	1.371281
18	-45.067781	0.556116	0.207320	0.432269	0.472393	3.222786	0.777214	2.757240	1.242760
19	-45.019058	0.507766	0.211219	0.431966	0.471774	3.048137	0.951863	2.529818	1.470182
20	-45.239997	0.567811	0.233744	0.429783	0.474631	3.055156	0.944844	2.721915	1.278085

The likelihood is a relative measure and indicates the fit of the model to the data. The greater it is (closer to zero) it is, the better.

Q: The rate parameters fluctuate a little but there is a general trend for each character – how would you describe this in words?

Q: Look at the estimates for the root (remember this is a probability for those two states being ancestral). What do you think these results indicate? How much confidence do you have in them?

IMPORTANT: the ML results illustrate the variation in ML solutions across the trees. However, because these are not independent re-runs of evolution, we can't meaningfully combine them to get an average or a distribution for (for example) the root values or the rate parameters. For that, we should do a Bayesian MCMC analysis.

The point of looking at the ML results is to get a feel for the values of the parameters. Your Bayesian posterior probability distribution should (ideally!) include values close to ML as part of the distribution, so this is one diagnostic you should look out for in your analysis workflow.

Dependent model under ML

```
./BayesTraitsV2 an_37.trees resdes.trait
```

When prompted for the model of evolution choose **Discrete: Dependent** by typing **3**. Choose **ML** as the analysis method by also typing **1**.

When your info screen comes up you will see that the following lines are different from what you saw with the Independent model:

```
Model:                          Discrete Dependent
No of Rates:                     8
Restrictions:
    q12                          None
    q13                          None
    q21                          None
    q24                          None
    q31                          None
    q34                          None
    q42                          None
    q43                          None
```

There are eight rates (listed below Restrictions) because each state change in this model is dependent on the state in the other character (refer to the flow diagrams on p. 21)

Give the logfile a name and set the analysis running

```
lf resdes_MLdep.log
run
```

The first line of your output will now have eight parameters estimated instead of four, and the four possible root assignments.

Q: How do likelihoods compare tree-to-tree? Based on this, would you investigate the coevolutionary hypothesis further?

Q: How do the root estimates compare to those under the Independent model?

Q: Sketch out what you think is going on with the Dependent model. How would you describe this in words?

Independent model under Bayesian analysis

We'll now move on to Bayesian Markov-chain Monte Carol approaches to correlated evolution. Start a new analysis of the data.

```
./BayesTraitsV2 an_37.trees resdes.trait
```

When prompted for the model of evolution choose **Discrete: Independent** by typing **2**. This time choose **MCMC** as the analysis method by typing **2**.

The default settings will be printed to the screen. We'll discuss those that are different (in bold), and you may wish to annotate the printout below.

```
Options:
Model:                          Discete Independant
Tree File Name:                 an_37.trees
Data File Name:                 resdes.trait
Log File Name:                  resdes.trait.log.txt
Summary:                        False
Analysis Type:                  MCMC
Sample Period:                  100
Iterations:                     5050000
Burn in:                        50000
Rate Dev:                       2.000000
No of Rates:                    4
Base frequency (PI's)           None
Character Symbols               00,01,10,11
Using a covarion model:         False
Restrictions:
    alpha1                      None
    beta1                       None
    alpha2                      None
    beta2                       None
Prior Information:
    Prior Categories:           100
    alpha1                      uniform 0.00 100.00
    beta1                       uniform 0.00 100.00
    alpha2                      uniform 0.00 100.00
    beta2                       uniform 0.00 100.00
Tree Information
     Trees:                     50
     Taxa:                      37
     Sites:                     1
     States:                    4
```

Give your MCMC analysis a logfile name and run the program.

```
lf resdes_MCMC_ind.log
```

```
run
```

You will have a brief pause (this is called "burn-in") and then output will flash past very quickly. This is because the program is writing out every 100th iteration. **CTRL+C** for now. Scroll back up the screen to look at the parameter estimates.

Your first line will have the familiar columns from your ML analysis along with two new ones, **Harmonic Mean** and **Acceptance**.

Note that there are *two* columns for the likelihood. The first is the **Lh** associated with that particular step in the chain. The second, **Harmonic Mean**, is an approximation of the *marginal likelihood* and is the one to take note of here. Ideally, our harmonic mean will come to *stationarity*, which means that it won't be getting progressively better or worse but wander about some value. As a result, the parameters and node estimates will constitute a proper statistical sample of all possible results: a posterior probability distribution. More on Harmonic Means on p. 29.

Q: How do these MCMC parameter estimates (transition rates and root estimates) differ from our ML estimates?

Look at the last column called **`Acceptance`**. This specifies the percentage of proposed models being accepted as the next step in the Markov chain. The values here are really large - we want to aim for around 20-60%. We can alter this by setting a more strict **ratedev** (**rd**) parameter (at present it is set to 2).

```
rd 10
```

This may still not be enough, and you will need to play around with values of **ratedev** to get the Acceptance rates where they should be. You can interrupt the run at any time with **CTRL-C**, invoke the program, and go again.

Using the defaults, we're also sampling at very close intervals (every 100 steps) of the Markov chain. Change the sample period to something higher like 5000. We'll run the analysis for 5,000,000 iterations.

```
sample 5000
iterations 5000000
```

or for short

```
sa 5000
it 5000000
```

We want to see the whole chain, so we set burn-in to zero

```
bi 0
```

Name the logfile again

```
lf resdes_MCMC_ind_bi.log
```

Check these changes by typing "`info`". Run, and let it run to the end.

```
run
```

Run another independent analysis (you can open a new Terminal window for this), but this time don't give the burn-in command. Name this one **resdes_MCMC_ind.log**
They will take a few minutes to run. Analyses with hundreds of trees in the sample and upwards of 100 taxa can take days and weeks to reach stationarity!

>> Parallel (cluster) computers are often necessary for Bayesian phylogenetics. Speak to your IT folk to see whether you have access to a cluster facility through your university.

>> Typing commands in each time we want to run an analysis can be tedious, so we can be programmatic in our approach and use ***command files****. As well, when you have many different analyses to run, using command files becomes necessary to manage your input and output. The box that follows describes how you can use command files.*

Using a command file

A command file is simply a short textfile that delivers all of the programs commands at once. You will find the example below in **resdes.cmd** in the **correlation** folder

2 (do an Discrete Independent analysis)
2 (use MCMC)
lf resdes_MCMC_ind.log (name the logfile)
sa 5000 (sample every 5000 steps)
it 5000000 (run chain for 5 million steps)
 # rd 50 (set the acceptance parameter to 50)
run

A line beginning with a hash (#) is commented out i.e. not read by the program.

Add **< commandfile** to your syntax, and this will sends the command file to the process started by the previous instructions, e.g.

```
./BayesTraits treefile traitfile < commandfile
```

Dependent model under Bayesian analysis

Open a new Terminal window, and proceed as before except this time choose **Dependent Model** by typing **3** and **MCMC** by typing **2.** As when we ran the ML Dependent model, you will see that the program will estimate eight parameters, because it is testing to see if a change in one character depends on the state in the other character. Type in the same commands as above, except change your logfile name and put the **rd** at a value (make a note of it!) such that it matches the acceptance rates of your independent model analysis.

```
./BayesTraits an_37.trees resdes.trait
3
2
sa 5000
it 5000000
rd ??
run
```

Those analyses may take some time to run so while we are waiting we will view the first logfile, where we set our burn-in to zero, so that we become familiar with the output.

8. Viewing the output of an MCMC run

Open Excel, R, or some other statistical analysis package. Navigate to your results log **`resdes_MCMC_ind_bi.log`** and open it. It is tab-delimited, and with a bit of poking about should import into your stats program of choice.

In Excel
The Text Import Wizard will guide you through importing the file. Choose the following options and it should all import beautifully.

- Original Data Type: Delimited
- Start import at Row 32 (gets rid of the Options in the logfile, but retain a copy!)
- Delimiters: Tab

8a. Examining the Harmonic Mean Likelihood

First, we need to check that the run reached stationarity, and we do this by plotting the harmonic mean against the length of the MCMC chain (the iteration number).

In Excel
Plot the Iterations against the Harmonic mean using the Chart Wizard.

>> *In general, harmonic means can be very unstable, so you will need to run your analyses for millions of iterations depending on your data and number of taxa.*

What is the Harmonic mean?
The harmonic mean is the reciprocal of the arithmetic mean (average) of the reciprocals.

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} 1/x_i}, \qquad x_i > 0 \text{ for all } i.$$

Why do we use it?
It is a rough approximation of the marginal likelihood. It gives more weight to small values and minimises the effect of large ones. It is still, like all means, sensitive to outliers, so we have to calculate it over a large number of iterations.

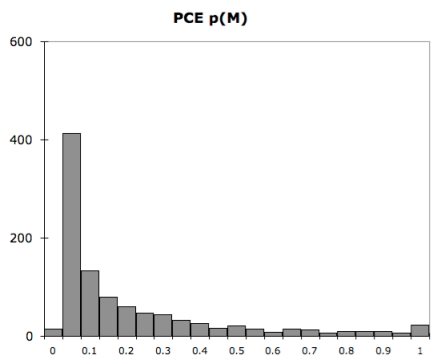
Can I calculate it?
Usually, you should take the last value in the Harmonic Mean column as the marginal likelihood for that run. If you want to calculate the harmonic mean for a subset of your data, or perhaps you are pooling post-burn-in samples from many runs, you can calculate the harmonic mean of the Likelihood column using the functions in Excel or R. Note: Be aware that you need to use absolute values of the Lh in Excel.

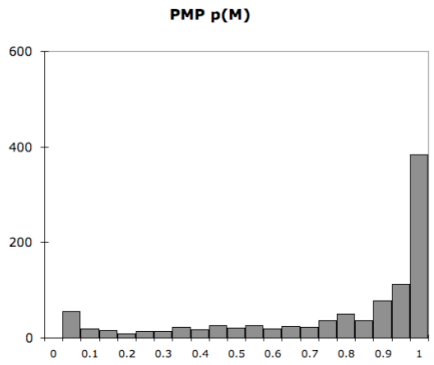
Q: What does your plot of Iteration x harmonic mean look like? Sketch it out. Where would you say is the burn-in for the marginal likelihood? From what point onwards would you take your equilibrium sample?

If your Independent model run with burn-in set to 50000 (default) is finished, you should open this and compare to the full chain. From now on, we'll refer to this second (burn-in = 50000) independent model run.

8b. Posterior Probability Distributions

Remember that we are interested in the posterior probability distribution (PPD) of our parameters rather than a single value. After discarding the portion of the chain that is not at equilibrium (the *burn-in*), we can use the values remaining to obtain a median, mean and standard error, 95% credibility interval, etc. We can then report these and/or show a histogram of the reconstructions for each node we are interested in. Here's a couple of examples of PPDs, this time reconstructing matriliny at two nodes across 1000 Austronesian language trees. At the first, matriliny is probably not the ancestral state, whereas in the second, we have good evidence that it might be.





Importantly, we must multiply our PPD for the character by the probability that the node exists in the tree. In the example above, the node PCE is only present in 85% of the tree sample, whereas PMP is present in all. Therefore, our character inferences about PCE would need to be multiplied by the phylogenetic uncertainty at that node to give the correct "combined probability".

In an analysis of correlated evolution we're most interested in the transition rate parameters (although we might also be interested in ancestral state estimates, of course). So for now, we'll examine these parameters.

Q: At approximately what iteration would you start to take the post-burn-in sample from the chain for the Independent Model? Was 50000 iterations enough for burn-in? Do you think the chain has been run for long enough?

When you have decided what will constitute your convergence sample, write the marginal Lh value in the box, and calculate the mean value of each of the parameters for the sample.

marginal lh	alpha1	beta1	alpha2	beta2

Repeat this with the Dependent Model results. Import the file, examine the plot of Iteration x Harmonic Mean.

Q: At approximately what iteration would you start to take the post-burn-in sample from the chain for the Independent Model? Was 50000 iterations enough for burn-in? Do you think the chain has been run for long enough?

Calculate the means of the eight parameters for the Dependent Model and copy them and the marginal likelihood to the table below.

marginal lh	q12	q13	q21	q24	q31	q34	q42	q43

If time permits, you should choose one or two parameters and draw histograms for each of them compare the shape of the distributions, and calculate their mean and standard error. Discuss with your neighbours what you think you can interpret from these PPD plots.

9. Bayes Factors and model choice

Bayesian statistics are a bit different to the frequentist statistics (such as ANOVA or regression) that you're probably familiar with. Great academic controversies exist over these different approaches, but they boil down to "model comparison, given the data and plausibility" (Bayesian) versus "null or alternative hypothesis, given a benchmark for elimination of chance" (frequentist). The important point for us is that we don't use p-values to reject a hypothesis, we compare the two models and take their ratio as saying something about the strength of evidence in favour of one or the other.

We use the Bayes Factor to do this. Though mathematically it differs, it is in-practice analogous to the LRT. However, we don't need to adjust for the difference in parameters, as we have averaged over the model score (the Lh) rather than maximised it. The Bayes Factor is:

$$\mathbf{BF = 2(M_1 - M_2)}$$

where M_1 and M_2 are models and M_1 is the focal model.

Interpreting values of the BF (from Raftery 1996)	
< 0	Negative (supports M_2)
0 to 2	Barely worth mentioning
2 to 5	Positive evidence in favour of M_1
> 5	Strong evidence in favour of M_1

9a. Determining evidence for the Dependent or Independent model

Q: Which should be M_1*, the Dependent or Independent? Why?*

Q: Calculate the Bayes Factor. According to the guide above, what does this mean for our hypothesis of correlated evolution between residence and descent?

Q: Plot out the Dependent model on the flow diagram below. Can you think of any interesting hypotheses you might like to test here?

10. Priors

In a Bayesian context, priors are mathematical expressions of the information you can bring to the problem. In practice, in phylogenetics we usually have very little prior knowledge, and we want to be as non-commital as possible. Usual practice is to use as non-informative a prior as possible, where we don't place constraints on the possible values our parameters can take. However, sometimes the data might be quite weak, and we might, after careful consideration, want to boost the signal with a prior. We might also have some theoretical perspective on our data, for example, we might think that in most cases our rate parameters take low values, only occasionally taking high values (for example – and then we might choose an exponential prior).

>> You are strongly encouraged to refer to the BayesTraits manuals for more on the types of priors and their implementation. You should always be able to justify your choice of prior.

Lets look at our residence and descent data again.

```
./BayesTraitsV2 an_37.trees resdes.trait
3
2
sa 5000
it 5000000
rd 30
```

There's no such thing as a prior-free Bayesian analysis. Although we didn't talk about it, previously we were using a **UNIFORM prior**, where the values of the parameters were sampled from the distribution {0,100} with equal probability. This time we're going to use a different prior, a **GAMMA prior**. We have to specific the shape of the distribution by giving a scale and shape parameter.

```
priorall gamma 2 3
run
```

Note that the **Acceptance** rate is now far too low. Restart, and play around with combinations of **rd** with these priors.

When you hit on a ratedev value that gives you a reasonable **Acceptance** rate, set a run going. Then try another prior, this time the **EXPONENTIAL prior**. This one simply needs the mean of the distribution

```
priorall exp 5
```

>> Wikipedia is pretty good for probability and statistics. Refer to the pages on each distribution for more information, e.g. http://en.wikipedia.org/wiki/Gamma_distribution

>> Refer to the manual and to Pagel & Meade 2006 and for a very cool "hyperprior" approach that allows us to integrate over model uncertainty using a reversible-jump algorithm.