Biol 206/306 – Advanced Biostatistics
Lab 10 – Phylogenetically Correlation & Regression
Fall 2016

By Philip J. Bergmann

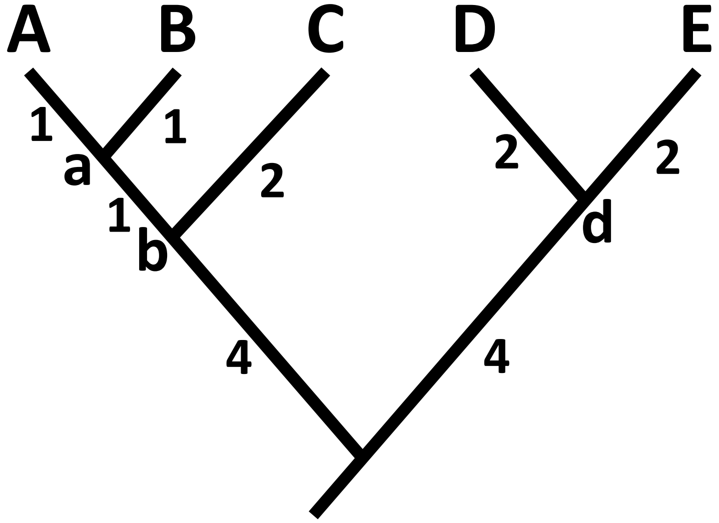
0. Laboratory Objectives

1. Learn why phylogenetic relationships need to be taken into account
2. Learn two ways of taking phylogenetic relationships into account
3. Explore the *phylo* class object and learn how to work with phylogenies in R
4. Learn to calculate an evolutionary correlation matrix
5. Learn how to do a phylogenetic generalized least squares (PGLS) regression
6. Learn how to take phylogenetic signal into account in a PGLS regression

1. Analyzing Comparative Data

Whenever a dataset includes variables that have been measured/collected for multiple taxa, we are analyzing comparative data. Most often, we think of comparative data as constituting values collected from multiple species, but this is not always the case. Comparative data could include taxa above the species level, which are called **clades** of various levels of inclusion. Comparative data could also include taxa below the species level, including sub-species, varieties, or populations. The key here is that your units of comparison are related to one another through evolution.

The problem facing an analysis of comparative data is that they are not independent, which is a key assumption of most statistical techniques. The reason for this is that taxa share part of their evolutionary history with other taxa. If you trace the branches of a phylogeny from the base towards the tips, where two taxa of interest appear, you will notice that part of the pathway you follow is the same for both taxa and part is unique to each taxon. The unique segments of the route are the time the two species were evolving independently of one another, and the common segments are where they were the same lineage. ***Try this out on the phylogeny shown.***



What are two taxa that have been evolving largely independently of one another?

What are two taxa that are minimally independent, having shared most of their evolutionary history?

A basic approach to ensuring that comparisons that you make among taxa are independent is called **sister group comparisons**. Sister groups share a **node** (branching event) on a phylogeny, and one can compare everything along one branch from the node with everything on the other branch from the same node. For example, in the phylogeny on the previous page, you can compare taxa A and B, but not taxa B and C. You can also compare taxa (A+B) and C. ***What are the two remaining sister group comparisons you can make on the phylogeny?***

Phylogenetically Independent Contrasts (PICs), is a formalized approach to sister group comparison that was developed in 1985. There are a number of components to PICs. First of all, they are **contrasts**, which is a general statistical approach that makes comparisons through looking at differences. So, a contrast is a difference. If you have two species (A & B) for which you are comparing a variable (X), you could calculate the contrast simply as X_A-X_B. Note that the order in which you do this operation is arbitrary, so there is no reason to do X_A-X_B in favor of X_B-X_A. This means that the sign (+/-) of a contrast is also arbitrary. In the PICs approach you can calculate the contrast for variable X for each of the nodes, allowing for n-1 contrasts, where n is the number of taxa on the phylogeny. ***Using the phylogeny on the previous page, list all of the contrasts that you can make. The nodes are already labeled for you for convenience.***

The contrasts that you list above are now independent of phylogeny because they only include sister group comparisons. However, PICs are a little more complicated because the branches on a phylogeny are typically of different lengths and are frequently interpreted as being proportional to evolutionary time. The PICs approach assumes that your trait of interest, X, evolves following a Brownian Motion (BM) model. In this model, the maximum amount of divergence between two taxa is proportional to time (in this case, branch lengths, and actually the sum of the two descending branch lengths). PICs are, therefore, standardized by their standard deviation, which is equivalent to the square root of the sum of their branch lengths. This means that the contrasts themselves (differences) are divided by the summed branch lengths. An important assumption of PICs is that the branch lengths of the phylogeny adequately standardize the PICs. ***How could you test this assumption?***

Contrasts are one way to take phylogeny into account, but notice that their values no longer correspond to species/taxa – there is one fewer contrast than taxa! The phylogeny can also be represented as a **phylogenetic variance-covariance (VCV) matrix** and this can be incorporated into analyses. Such a phylogenetic VCV matrix is square, with the rows and columns corresponding to taxa. The diagonal contains the phylogenetic variances, which are the distances from the root of the tree to the tip for each species. The off-diagonals are taxon phylogenetic covariances, which are the shared distances from root to tip of any two taxa. Under BM, the phylogenetic covariances represent how similar a trait is expected to be for two taxa, or how non-independent two trait values are expected to be. Notice that the branch lengths are provided on the phylogeny on page 1. ***Complete the phylogenetic VCV matrix for that phylogeny (filling out the diagonal and lower triangle is sufficient).***

	A	**B**	**C**	**D**	**E**
A					
B					
C					
D					
E					

2. Working with Phylogenies in R
The **ape** package for R treats phylogenies as objects of class *phylo*, and provides various functions necessary for reading and manipulating phylogenetic trees, and for doing some analyses. The packages **caper, geiger** and **phytools** provide added functionality, mainly in the analysis of phylogenetic and comparative data. ***If you have not done so previously, install ape, caper, geiger and phytools (don't confuse it with phylotools, which is another R package). Load these packages. Also, download the phylogeny and dataset for this week's lab.***

The most important and first step when using phylogenies in R is loading them into the work space. The following function does this very simply:

```
> read.tree("filename.tre")
```

Note that this is the appropriate function for a Newick format tree (***Take a look at the tree file you downloaded using notepad to see what Newick format looks like*** – it is akin to a Venn diagram, in that it represents the phylogeny as a nested series of brackets). If you have a Nexus format tree (this is simply another file format for storing phylogenies), then a different function, **read.nexus()** should be used. We will use only Newick trees in this class, so do not worry about Nexus files. ***Use the read.tree function to load the phylogeny for the lab into R, and assign the phylogeny to an object called "tree".***

If you just type your tree object's name, you get very basic information: the number of taxa, the number of nodes, some of the taxon names, and a few other pieces of information. To visualize the phylogeny, simply use ***plot(tree)***. ***What is the sister group of the species* Scel. spinosus*?***

You can find out more about your phylogeny using the functions:

```
> is.binary.tree(tree)
> is.ultrametric(tree)
```

A binary tree is another term for a tree that is fully resolved, meaning that all nodes are dichotomous, or have exactly two descendents. An ultrametric tree is one where all of the tips line up at the same level (all taxa are contemporary/extant). Both of these functions return TRUE if the tree is binary or ultrametric, and FALSE otherwise. If a tree is not binary, but you need it to be, you can resolve any nodes with >2 descendents randomly, producing new nodes, separated by branches of length zero. To do this, use the function:

```
> multi2di(tree)
```

Is the tree you have loaded fully resolved? Is it ultrametric?

Another useful trick for working with phylogenies is using the **str(tree)** function, which, as usual, tells you the components of the object. tree$edge provides you with a Sx2 matrix, where S is the number of branches in the phylogeny. If each node, including the tips (taxa) are numbered, then tree$edge gives you the ancestor and descendent node numbers for each branch. Note that "edge" and "branch" are synonymous. Branch is typically used in biology, edge is typically used in graph theory. tree$Nnode tells you how many nodes are on your tree. tree$tip.label is a list of your taxa. tree$edge.length is a vector of length S that gives you the length of each branch. tree$root.edge is the length of the single branch at the base (root) of the tree. ***Look at the structure (str) of your tree. How many nodes does it have?***

How many branches does it have?

What is the root branch length?

It is often desirable to have an ultrametric tree for analyses because it signifies that all species are extant. We can use the branch lengths of a non-ultrametric tree and transform them to be ultrametric. The first step is to see how long the tree is from root to tip. Then we can use that information to generate an ultrametric tree that is the same length. ***Do this now using the function:***

```
> nodeHeights(tree)
```

This function is part of the phytools package, and gives you distances of each node from the base of the tree (the base is at zero). The first column of the output is the node Height from the base of the tree to the ancestral side of a given branch, and the second is the height from the base to the descendent side of each branch. If you take the maximum of this object (the greatest number), you have the height of the entire tree. ***What is the height of the tree?***

Now we will use the chronopl function (part of the ape package) to transform the tree to a chronogram, which is another term for an ultrametric tree. The first argument is the tree you are transforming, the second is a smoothing parameter (just use 1 for this in this lab), and the third one gives the length of the tree, so age.min=max(nodeHeights(tree)).

```
> chronopl(tree, lambda, age.min=y)
```

Do this now for your tree and name it "ultra_tree". Then plot it and use the is.ultrametric() function to see if it is ultrametric. Is it?

You will likely get a warning that some of the branches in the tree were of length zero and were collapsed. Check if the tree is binary. If not, then make it binary (don't forget to name the freshly-resolved tree as an object).

4. Calculating Evolutionary Correlations Between Traits in R

You now have a tree that you can use for some phylogenetically informed analyses and are ready to load some data for the species in the trees and see how they have evolved. The dataset we will be using today is a lizard body shape dataset that we haven't used yet. ***Download and view the Excel spreadsheet.*** There are two worksheets in the file. The first, called "Raw_Data" contains measurements for adult individuals of a number of species of phrynosomatine lizards. The variables are explained to the right. The second worksheet, "Spp_Data", contains one line for each species, with the average value for each of the relative body proportions, and the maximum SVL. All of these data are calculated from the Raw_Data worksheet. ***You will be using the "Spp_Data" worksheet, so save it as a tab delimited text file and load it into R, assigning it do object "spp_data". In this dataset, what are the sampling units on which we will be doing analyses?***

Why are most of the variables in the dataset "relative" lengths? What do they represent? And why do we also include maximum SVL?

When you have a dataset that you plan to analyze using a phylogeny, the first step is to ensure that the same taxa are in the dataset and in the tree. Something as simple as a minor typo in one but not the other can result in a lot of time spent solving problems. The following function checks if the same taxa are on the tree and in the dataset (part of the Geiger package):

```
>name.check(tree,data)
```

Use the above function to confirm that the same taxa are in your tree and loaded data. Now delete one of the taxa in the tree, saving the new tree as "tree_demo":

```
> tree_demo <- drop.tip(ultra_tree, "Uro_ornatus")
```

This function is also part of the geiger package and allows you to edit trees in yet another way. ***Repeat your name.check on the new tree and your spp_data object. What do you get?***

One of the most common goals of comparative analyses is to estimate correlations among traits, given a phylogeny of relationships among species. One could do this using PICs, but, as mentioned above, the disadvantage to this approach is that the correlations are calculated for differences, as opposed to actual trait values. The latest approach involves estimating an evolutionary variance-covariance (VCV) matrix and then calculating a correlation matrix from that. The evolutionary VCV matrix assumes evolution by BM, although other models of trait evolution can be implemented (this is beyond the scope of this lab). The evolutionary VCV matrix is also useful because the diagonal elements (evolutionary variances) are actually evolutionary rate estimates for the traits, so you get an estimate of how fast different traits are evolving. Off-diagonal elements are simply evolutionary trait covariances, and easier to interpret when converted to correlations. We will use the lizard body shape dataset to calculate evolutionary VCV and correlation matrices, given your ultrametric phylogeny. A function from the Geiger package calculates your evolutionary VCV matrix given a phylogeny and dataset and a function in the pre-loaded stats package converts it to a correlation matrix:

```
> ratematrix(tree,data)
> cov2cor(VCV)
```

Assignment: (4 points)

Calculate your evolutionary VCV and correlation matrices using your ultrametric tree and species data. Note that for the ratematrix function to work, you need to call only the variables in your data frame that are numeric (including the column with clade membership will result in an error). Fill out the two matrices below to an appropriate number of decimal places.

Evolutionary VCV Matrix

	rBW	rFLL	rHLL	SVL
rBW				
rFLL				
rHLL				
SVL				

Evolutionary Correlation Matrix

	rBW	rFLL	rHLL	SVL
rBW				
rFLL				
rHLL				
SVL				

Which trait evolves the fastest and which one evolves the slowest?

What else can you say about how the traits have co-evolved, assuming that correlations above 0.2 are worth discussing?

5. Phylogenetic Regression

Traditionally, phylogenetic regression was done using PICs as well. Standardization of PICs accounts for the assumption of data independence. However, in addition to limitations of this approach described earlier in this lab, it has recently been observed that it does not account for the additional assumption of **independence of the residuals** that regression makes. An alternative approach is phylogenetic generalized least squares regression (PGLS). Like ordinary LS regression, this approach assumes no error in X, but it is generalized, so allows non-independence of the residuals. By providing a phylogeny and assuming BM model of evolution, one can model this non-independence and take it into account. The expected correlations among residuals due to non-independence in this approach are accounted for using the phylogenetic VCV matrix you practiced making at the beginning of this lab.

NB: be careful not to confuse the phylogenetic VCV, which is created from the phylogeny alone, and the evolutionary VCV matrix, which is calculated from the phylogeny and trait data. Both have been introduced in this lab.

The PGLS regression approach is particularly flexible because you can measure the amount of **phylogenetic signal** in the residuals, and adjust the analysis to take that into account. This effectively relaxes the assumption of BM model of evolution. Phylogenetic signal is the degree to which more closely related species are more alike. It is most commonly measured using the parameter lambda, λ, which ranges from zero to one. A trait or residuals evolving via BM is characterized by λ =1 – more closely related species have trait (or residual) values that are more similar because they spent more time evolving together as a single lineage. As λ decreases, species evolve more and more independently. When λ=0, species evolve completely independently, and phylogeny no longer needs to be taken into account. If we quantify lambda for our residuals, then we will take phylogeny into account and the specific amount of phylogenetic signal that they exhibit. In this lab, we will both use BM and quantify λ and take it into account. You will learn more about λ in the next lab.

We will do this using the lizard body shape dataset that you have been analyzing in this lab. We will do PGLS regressions of pairs of traits for which you found there were elevated (>0.2) evolutionary correlations. ***Which pairs of body shape variables had correlations >0.2?***

To do the PGLS regression, we will use two functions in the caper package:

```
> comparative.data(tree,data,names.col,…)
> pgls(formula,data,lambda=1,…)
```

The **comparative.data** function simply prepares your data for analysis. To sue these functions, you need one of the columns in your data frame to contain your species names. The names.col argument identifies this column. Instead of names.col, type the name of the column with species names in quotes. Finally, the … indicates that there are other arguments in the function, but we will not use these. If you want to find out what they are, simply check the caper manual or type ?comparative.data at the prompt in R.

The **pgls** function does the actual regression. The **formula** argument should follow the format of any other regression. Although we will only be doing bivariate regression here, the function is very flexible and can do multiple regression, ANOVA, and ANCOVA, while taking phylogeny into account. The **data** argument should specify the name of a comparative.data object, not your original dataframe because the comparative.data object now contains both your phylogeny and your data. The **lambda** argument specifies how to model the phylogenetic signal of the residuals. The default is λ =1, which is a BM model. You could enter any value between zero and one here. However, you can also specify **lambda='ML'** (with quotes around the ML). In this case, the function will use **maximum likelihood** to find the optimal value of λ for your analysis and use that. Finally, another powerful feature of the pgls function is that you can use summary(), residuals(), and all other functions that you could use with an aov or lm object to get the details about the model that you fitted.

Assignment: (6 points)

Today, you will compare the results of regression analyses that do not take phylogeny into account with those that do, but using different models of evolution. Specifically, you will run regular OLS regressions, PGLS regressions that assume BM (λ =1), and PGLS regressions that estimate the amount of phylogenetic signal in the residuals. You will do this on the pairs of variables for which you found elevated evolutionary correlations. Be sure to use the same variables as x and y in each case. If one variable is involved in all of the comparisons that you consider, use that variable as x in ALL cases. Add extra rows to the tables if necessary.

OLS Regression Results, λ=0

X	Y	R^2	Intercept	SE(Int)	Slope	SE(slp)

PGLS Regression, Assuming BM, λ=1

X	Y	R^2	Intercept	SE(Int)	Slope	SE(slp)

PGLS Regression that Estimates λ

X	Y	R^2	Intercept	SE(Int)	Slope	SE(slp)	λ

Do the results differ between these three approaches? How? Especially consider the results you got for R^2 and the slope.

What do the estimated lambda values tell you about the phylogenetic non-independence in the residuals and the results?

Out of these three approaches, which do you think is most justified? Why?