PHYLOGR: EXAMPLES

Here we present several commented examples of the use of the PHYLOGR package, to familiarize readers with the use of both it and R in general. (A brief note on the installation of R and its libraries is included at the end). The examples we discuss here include ANCOVAs and PCAs from simulated data, independent contrasts analyses, and generalized least squares models. R commands and output are shown in courier font; ">" indicates the R prompt, and continuation lines are indicated by indentation. When output is suppressed it is indicated by "....". Comments in R are preceded by an "#". The file 'examples−code.R' under the subdirectory 'Examples' contains the code to repeat all these examples. In the 'Examples' directory we have also included the files of original data to be read for the examples used here (such as the .inp, .pdi, .fic, and .dsc files); we also include a few simulated data files, but to minimize space they are small (only 50 sets of simulated data), not the full ones used in the examples here (5,000 simulations). To make the files for these examples easily available when repeating them, it will be convenient to either start R from the 'Examples' directory, or to copy all the files in there to another directory, and launch R from there. The first examples presented include many explanations on the R code use, to familiarize the readers with the use and characteristics of R; more detailed explanation can be found in the manuals included with R.

ANCOVA with Simulations

As our first example, we show how to repeat the ANCOVA shown in Garland et al. (1993), which compared home range areas of mammalian Carnivora and ungulates after adjusting for body mass (both traits $\log_{10}$ transformed). The first step is to simulate data for log body mass and log home range area, using the PDSIMUL program (see Garland et al., 1993), with the file 49lbr.inp (included in the PDAP distribution) as input. (For this and all other examples, we used simple Brownian motion as the simulation model.) We assume that you named the simulated file "49lbr.sim", and that it is under the working directory from where R was started. (If the file is somewhere else, then you will need to provide the full path to the command "read.sim.data"; for instance, if the file is in c:\cm\pdap, then write "c:\\cm\\pdap\\49lbr.sim" or "c:/cm/pdap/49lbr.sim", instead of "49lbr.sim"). Start R, and type:

```
> library(PHYLOGR) # load our package
> ANCOVA.sim <− read.sim.data("49lbr.sim", "49lbr.inp",
      variable.names = c("body.mass","home.range"), other.variables
      = data.frame(clade=c(rep("Carnivore",19),rep("Herbivore",30))))
      # this reads in the simulated data file, and adds a column with
      the herbivore vs. carnivore indicator
```

The above command calls the function read.sim.data (with the arguments that the user specifies, such as the simulated files, the inp files, etc.). But if we just call read.sim.data, then we will see a (potentially very long) array of numbers on our screen. Instead, the common form of operation in R is to assign the output of a function (its returned value) to some object; the "<−" is the assignment operator. Thus, the command above calls read.sim.data and stores the returned value in the (just created) "ANCOVA.sim" object; we could have named this in any other way. This object can then be examined (for instance typing its name will print the object onto the screen) or manipulated further, as we do in the next command.

We will now call the function lm.phylog; one of the arguments of the function lm.phylog is "data", and in this case we pass "ANCOVA.sim", the just created object with simulated data, to the lm.phylog function. As before, we do not just call the function lm.phylog, but assign the returned value to another new object that we create, which in this case we name ancova.fit. Finally, the call "summary(ancova.fit)" uses the function in our package summary.phylog.lm (you only need to type "summary" because this is a generic function and our package contains the specific method for objects of class phylog.lm; this exploits the class oriented features of R). When calling summary, we pass as an argument the recently created ancova.fit.

```
> ancova.fit <- lm.phylog(home.range ~ body.mass + clade,
      data = ANCOVA.sim) # fit the model
>
> summary(ancova.fit)

Call:
lm.phylog(formula = home.range ~ body.mass + clade, data = ANCOVA.sim)

Model from inp data:
     Intercept      body.mass cladeHerbivore
       −0.1055         0.9970        −1.0224

F−test's from inp data:
Overall.F body.mass     clade
    24.58     47.40     23.97

P−value:
Overall.F body.mass     clade
 0.150570  0.003399  0.227355

Quantiles of F−value distribution:
      Overall.F body.mass   clade
50%       7.283     1.974   7.067
90%      31.710    12.657  44.904
95%      44.403    18.085  65.672
99%      80.688    35.868 126.087
99.9%   146.283    60.318 218.497

Number of simulations used in analyses:  5000
```

Using the conventional 5% Type I error rate, the critical values for the *F* distributions are those from the 95% quantile. The *P*−values computed based upon these null distributions

distributions are 0.003 for body mass (indicating a significant effect of body mass on home range −− this is a marginal *F*, and thus accounts for possible differences in home range between herbivores and carnivores) and 0.23 for clade, indicating no significant differences in home range between carnivores and ungulates after correcting for differences in body mass. These results closely resemble those reported in Garland et al. (1993, p. 275); identical results are not to be expected, as we used a different seed for the pseudo−random number generator (1000 instead of 2) and 5,000 instead of 1,000 simulated data sets.

Multiple Regression and Interactions between a Categorical and a Continuous Predictor

Here we show a more complex example, from Garland and Janis (1993), that illustrates analyzing subsets of data, excluding particular species, using a more complex linear model than above, and plotting results from the analyses. We first show how to analyze the data using independent contrasts (note that there are differences with the results reported in Garland and Janis [1993] because the phylogeny we use here is the more recent one of Garland et al. [1993]). As before, we assume that all the data are available under the current working directory for R; otherwise, you will need to include the full path in the calls to the functions for reading .sim and .pdi files.

Independent contrasts.−− First, we read the files with the independent contrast information. In this example, the files with independent contrasts have been obtained from the PDTREE program (part of the PDAP package) and are read with our function read.pdi.data; however, files with contrasts information in other formats could also be read and manipulated after appropriately modifying our functions.

Because the .fic files have a standard structure of variables−in−columns and cases−in−rows we read them using the built in "read.table" function. We are only interested in the third and fourth columns (the ones with the unstandardized contrasts), and thus when using read.table we only keep the third and fourth columns from the returned object: the "[,c(3,4)]" refers to all the rows and columns 3 and 4. Moreover, we actually want to read several .fic files (those that correspond to several traits); we could do it by reading one .fic file at a time and later combining their output, but it is simpler to read all the files at once. This is what we do with the built−in "cbind" function (for combine by columns), and we assign the returned value to a new object that we name "garland.janis.ic". Next, to standardize the contrasts, we read column 5 of one of the .fic files that contains the standard deviations (square roots of sums of corrected branch lengths); for

clarity, we store that as a separate object that we call "branch.lengths".

```
> garland.janis.ic <- cbind(read.table("49ms.fic")[,c(3,4)],
      read.table("49hmt.fic")[,c(3,4)]) # read fic files
> branch.lengths <- read.table("49ms.fic")[,5] # the branch lengths
```

We now want to divide each one of the unstandardized contrasts in the garland.janis.ic object by the appropriate standard deviation. We could try to do this with a loop, but in languages such as R (or Octave or Matlab or Mathematica) vectorized operations are usually much faster (and result in cleaner code) than loops. Thus, we apply the "/" operation, which in this case will divide each element in garland.janis.ic by branch.lengths. The garland.janis.ic data frame has four columns, so the same vector branch.lengths is used repeatedly (recycled) in each of the columns of garland.janis.ic. Notice also how we assign the output of that division to garland.janis.ic again; we could have created another object, but in this case we are only interested in the standardized contrasts, so there is no reason to keep around the original garland.janis.ic. As well, we rename the column names to something meaningful, using the built−in names function.

```
> garland.janis.ic <- garland.janis.ic/branch.lengths # standardize contrasts
> names(garland.janis.ic) <- c("body.mass", "running.speed",
      "hind.l.length", "mtf.ratio") # name the variables
```

Next, we "positivize" the contrasts for graphical convenience, as explained in Garland et al. (1992); we will positivize all of them with respect to the body mass contrasts. In other words, we want all the body mass contrasts to be positive, so we multiply by −1 those that are negative, while changing also the sign of all those contrasts in the other traits for that corresponding contrast. Thus, we select those rows (i.e., those cases or contrasts) where the body mass contrast is negative, subsetting with "[garland.janis.ic$body.mass<0,]": we select those rows where the variable body.mass from the garland.janis.data frame is less than 0. Those elements that fulfill that requirement have their sign changed (by multiplying by −1) and, as before, they are reassigned back into themselves.

```
> garland.janis.ic[garland.janis.ic$body.mass<0,] <- −1 *
      garland.janis.ic[ garland.janis.ic$body.mass<0, ] # to positivize
      contrasts
```

Finally, we first add the branch length column to the garland.janis.ic data frame: we assign, using the "<−" operator, the branch.lengths variable to a new column in the garland.janis.ic data frame that we also name branch.lengths (this step is best carried out after positivizing to prevent

some branch lengths from having their sign changed). To allow us to exclude certain contrasts in future analyses (e.g., outliers), we also add the names of the contrasts to the garland.janis.ic data frame (the contrasts' names, as node−node, are in the first column of the .fic file). The last step adds another column to the data frame, one that identifies the type of contrasts: the first one corresponds to the root contrast, the next 18 are contrasts within the Carnivora clade, and the final 29 are contrasts within the ungulates. We want this column to be considered a factor (instead of a character vector), and thus the use of "as.factor" to convert what is a character vector to a factor. Now, the vector could have been entered by entering "root", "Carnivore", "Carnivore", (another 17 times)..., "Herbivore", (another 29 times)... Instead of typing that, we create the vector by concatenating "root" with the output from the "rep("Carnivore", 19)", which repeated Carnivore 19 times, and "rep("Herbivore", 29).

```
> garland.janis.ic$branch.lengths <- branch.lengths # add branch lengths
      to the file, for completeness
> garland.janis.ic$names.contr <- as.factor(read.table("49ms.fic")[,1])
> garland.janis.ic$clade.contr <- as.factor(c("root",rep("Carnivore",18),
      rep("Herbivore",29))) # create indicator for clade
```

Garland and Janis (1993; see p. 140 and ff.) were interested, among other questions, in understanding the relationship between running speed and hind limb length in cursorial mammals; these relationship can be affected by differences in body mass, and thus we need to correct for the possible effects of body mass. Moreover, as their data set included members of two quite distinct clades, Carnivora and ungulates, we could ask whether the possible dependency of running speed on hind limb length (after adjusting for body mass) differs between the two lineages. Garland and Janis (1993) examined the relationship between speed and hind limb length after correcting for body size by first obtaining residuals from both traits regressed on body mass, and then examining the relationship between these residuals. The partial regression slope obtained with this procedure is equivalent to performing a multiple regression of speed on body mass and limb length (this is the rationale behind the "added variable plots"; e.g., Hocking, 1996, pp. 226 and ff.). Here, we report both correlations and regressions; the former to compare with the results in Garland and Janis (1993), the later for comparison with the GLS results (see below). To use correlation of residuals, we first obtain the residuals by calling the built−in "resid" function on a fitted linear model (which uses regression through the origin, as is mandatory with independent contrasts). We then compute the correlation through the origin of these two sets of residuals and obtain the *P*−value. Next, we repeat the analyses using regression, also examining scatterplots of the data and diagnostic plots for the fitted regression.

```
> attach(garland.janis.ic) # to eliminate the need to call the data set
      explicitly
> # correlation of residuals from regressions on body mass
> resid.hll <- resid(lm(hind.l.length ~ body.mass - 1))
      # this model formula uses a regression through the origin (the -1).
> resid.speed <- resid(lm(running.speed ~ body.mass - 1))
> rho1 <- cor.origin(resid.hll,resid.speed)
> rho1 # the correlation through the origin
          [,1]
[1,] 0.2439
>  2*(1-pt(sqrt(46)*rho1/sqrt(1-rho1^2),46)) # the p-value, using the
      standard formula but providing the correct df's.
           [,1]
[1,] 0.09481
>
> # the same, but using multiple regression
> # first, look at some scatterplots of the contrasts
> plot(body.mass,running.speed)
> plot(body.mass,hind.l.length)
> plot(hind.l.length,running.speed)
> fit1 <- lm(running.speed ~ body.mass + hind.l.length - 1)) # note also
      regression through the origin
> plot(fit1) # regression diagnostic plots
> summary(fit1)


Call:
lm(formula = running.speed ~ body.mass + hind.l.length - 1)

Residuals:
      Min        1Q    Median        3Q       Max
-6.44e-05 -1.95e-05 -4.73e-06  1.29e-05  7.45e-05

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
body.mass      -0.1053     0.0762   -1.38    0.174
hind.l.length   0.4678     0.2743    1.71    0.095 .
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Residual standard error: 3.04e-05 on 46 degrees of freedom
Multiple R-Squared: 0.0609,   Adjusted R-squared: 0.02
F-statistic: 1.49 on 2 and 46 degrees of freedom,     p-value: 0.236
```

The two procedures yield identical $P-$values (note that the correlation of the residuals is through the origin, and that we carry out the significance test using the appropriate degrees of freedom, which are the same as in the regression case, i.e., 46). However, the plot of the regression diagnostics (specifically the last plot, using Cook's distance) shows that the sixth data point (corresponding to the contrast between the grizzly and the polar bear) could be an influential point. Based upon the suggestions from that plot, and following Garland and Janis (1993), we now exclude the polar bear-grizzly contrast and repeat the analysis.

```
> # Excluding the polar bear-grizzly contrast
> # using residuals:
> resid.hll.nbear <- resid(lm(hind.l.length ~ body.mass - 1,
      subset = names.contr!="Tm-Ur"))
```

```
> resid.speed.nbear <- resid(lm(running.speed ~ body.mass - 1,
      subset=names.contr!="Tm-Ur"))
> rho2 <- cor.origin(resid.hll.nbear,resid.speed.nbear)
> rho2 # the correlation through the origin
          [,1]
[1,] 0.3857
> 2*(1-pt(sqrt(45)*rho2/sqrt(1-rho2^2),45)) # the p-value, using the standard
formula but providing the correct df's.
           [,1]
[1,] 0.007416
>
> # with multiple regression
> fit2 <- lm(running.speed ~ body.mass + hind.l.length - 1,
      subset=names.contr!="Tm-Ur")
> plot(fit2) # regression diagnostic plots
....
> summary(fit2)

Call:
lm(formula = running.speed ~ body.mass + hind.l.length - 1, subset =
names.contr !=
    "Tm-Ur")

Residuals:
      Min        1Q    Median        3Q       Max
-5.48e-05 -2.27e-05 -2.80e-06  1.65e-05  8.00e-05

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
body.mass      -0.1921     0.0797   -2.41   0.0200 *
hind.l.length   0.8268     0.2948    2.80   0.0074 **
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Residual standard error: 2.87e-05 on 45 degrees of freedom
Multiple R-Squared: 0.15,     Adjusted R-squared: 0.112
F-statistic: 3.98 on 2 and 45 degrees of freedom,     p-value: 0.0257
```

As before, the results using correlation and regression agree; the diagnostic residual plots do not indicate any problematic pattern. Additionally, Garland and Janis (1993) were interested in understanding whether metatarsal/femur ratio was an important explanatory variable of variation in running speed, after accounting for effects of hind limb length (and body size).

```
> # is MT/F relevant after including hindl.lenght?
      remember: need to look only at coefficient for mtf.

> summary(lm(running.speed ~ body.mass + hind.l.length + mtf.ratio - 1,
      subset=names.contr!="Tm-Ur"))

Call:
lm(formula = running.speed ~ body.mass + hind.l.length + mtf.ratio -
    1, subset = names.contr != "Tm-Ur")

Residuals:
      Min        1Q    Median        3Q       Max
-5.59e-05 -1.84e-05 -3.95e-06  1.81e-05  8.75e-05

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
body.mass      -0.1304     0.0913   -1.43     0.16
hind.l.length   0.6024     0.3366    1.79     0.08 .
```

```
mtf.ratio       0.1693     0.1260    1.34     0.19
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Residual standard error: 2.84e-05 on 44 degrees of freedom
Multiple R-Squared: 0.184,    Adjusted R-squared: 0.128
F-statistic:  3.3 on 3 and 44 degrees of freedom,     p-value: 0.0289
```

The question of whether MT/F ratio is a significant predictor is answered by noting that the *P*−value is 0.1860. In this case, the *P*−values of body mass and hind limb length have also decreased, which is not unexpected as those are all partial−*t* statistics, and thus they are adjusted for the possible effects of MT/F ratio; in other words, these are equivalent to marginal *F*−tests, or the tests that would be obtained by comparing the full model (with all variables) with a reduced model (one with all the variables except the one examined). We can also obtain a summary from the above model using an ANOVA−type display, with *F*−statistics instead of *t*−values. Notice, however, that these are sequential *F*−statistics: the sequential *F*−statistic, and associated *P*−value, tests the relevance of a particular variable after the preceding variables (but not the subsequent ones) have been included in the model; this is different from the *t*−values and marginal *F*−statistics, which refer to one variable when all the other variables are in the model. Thus, except for MT/F ratio, the *P*−values are not the same as those from the *t*−values (marginal *F*−statistics, however, would yield the same *P*−values as the *t*−statistics).

```
> anova(lm(running.speed ~ body.mass + hind.l.length + mtf.ratio - 1,
      subset = names.contr!="Tm-Ur")) # anova summary

Analysis of Variance Table

Response: running.speed
              Df   Sum Sq  Mean Sq F value Pr(>F)
body.mass      1 1.00e-10 1.00e-10    0.09  0.763
hind.l.length  1 6.50e-09 6.50e-09    8.01  0.007 **
mtf.ratio      1 1.50e-09 1.50e-09    1.81  0.186
Residuals     44 3.56e-08 8.00e-10
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
```

Monte Carlo simulations.−− We will now use simulations to compare some of the above results; in addition, we will examine a model that allows for differences in the slope of speed on hind limb length among clades. We first read the simulated data sets, give meaningful names to the variables, and add a column, clade, which indicates whether a given species is a carnivore or ungulate.

```
> GarlandJanis.sim <- read.sim.data(c("49ms.sim","49hmt.sim"),
      pdi.files = c("49ms.pdi","49hmt.pdi"), variable.names =
      c("body.mass","running.speed","hind.l.length","mtf.ratio"),
      other.variables = data.frame(clade =
      c(rep("Carnivore",19),rep("Herbivore",30))))
```

With the data read, we can now use the function lm.phylog. We first examine the relationship between running speed and hind limb length, correcting for body mass. We then repeat the analyses after excluding the grizzly and polar bears. To further illustrate the simulation approach, we test for a possible clade difference in the slopes of speed on hind limb length (still excluding the grizzly and polar bears). Next, we examine if MT/F ratio is a relevant predictor in the presence of hind limb length and body mass. Finally, we illustrate how to fit the model (running speed on hind limb length correcting for body mass) when we want to exclude one of the clades (the Ungulates) and some species of the other clade (the grizzly and polar bears).

```
> plot(lm(running.speed ~ body.mass + hind.l.length, data = GarlandJanis.sim,
      subset = sim.counter==0)) # regression residual plots
> speed.hll <- lm.phylog(running.speed ~ body.mass + hind.l.length,
      data = GarlandJanis.sim)
> summary(speed.hll)

Call:
lm.phylog(formula = running.speed ~ body.mass + hind.l.length,
    data = GarlandJanis.sim)

Model from inp data:
    Intercept     body.mass hind.l.length
      0.00202      -0.31651       1.29429

F-test's from inp data:
    Overall.F     body.mass hind.l.length
        19.07         24.93         36.55

P-value:
    Overall.F     body.mass hind.l.length
      0.06779       0.04879       0.01960

Quantiles of F-value distribution:
      Overall.F body.mass hind.l.length
50%       4.268     2.447         2.472
90%      15.849    16.186        16.478
95%      21.896    24.379        24.258
99%      36.356    45.583        47.067
99.9%    57.531    83.088        84.646

Number of simulations used in analyses:  5000

> postscript(file="Figure1.ps") # to generate a postscript file
      (shown in Fig.1); you won't see the figure in the screen.
> par(mfrow=c(2,3)) # 2x3 figures in the page, so that all fit in one window
> par(cex.lab=1.5) # to make labels larger
> plot(speed.hll) # plot the analyses from lm.phylog
> dev.off() #this closes the device, generating the file
```

We first produce standard residual plots from the original data set. Next, we analyze the simulations and provide a summary of the results, including the *P*−values for each of the *F*−tests. Here (as in the examples in Garland et al., 1993), the critical *F*−values at the 5% level from the simulations are much larger than the conventional tabulated *F*−values (for either independent variable, approximately 24.3 vs. 4.05 from a conventional table of *F* statistics with 1 and 46 d.f.). The results of the analyses can be plotted using the plot method on the fitted object. The output generated from that command is in Figure 1 (that is the figure as directly produced from R). If you want to see the figure on the screen instead of directly generating the postscript file, then use the commands above except for the "postscript" and the "dev.off" ones; if you desire, you can then obtain a the figure as a postscript file (by using, for instance, the built−in dev.copy and dev.copy2eps functions; see their help for details).

As in Garland and Janis (1993), we now exclude the grizzly and polar bears.

```
> speed.hll.nbear <- lm.phylog(running.speed ~ body.mass + hind.l.length,
      data=GarlandJanis.sim, exclude.tips=c("Tm","Ur"))
> summary(speed.hll.nbear)
....

Model from inp data:
    Intercept     body.mass hind.l.length
     −0.03118      −0.32246       1.31924

F−test's from inp data:
    Overall.F     body.mass hind.l.length
        18.84         23.86         35.59

P−value:
    Overall.F     body.mass hind.l.length
      0.06519       0.04939       0.01900
....
```

In this case, exclusion of the grizzly and polar bears has minimal effects on the results. (For simplicity, we have omitted the comparisons with the *F* statistics for simulated data, which would need to be recalculated.)

```
> speed.hll.nbear.int <− lm.phylog(running.speed ~ body.mass +
      hind.l.length*clade, data=GarlandJanis.sim, exclude.tips=c("Tm","Ur"))
      # note that R automatically adds both the interaction and each of the
      single terms when we use a "*" between terms (in R, and S, the
      interaction is denoted as ":"; thus, A*B expands into A + B + A:B).
```

```
> summary(speed.hll.nbear.int)

....
Model from inp data:
                  Intercept                    body.mass
                    −0.3039                      −0.2726
               hind.l.length                cladeHerbivore
                     1.4543                       0.9872
hind.l.length.cladeHerbivore
                    −0.5693

F−test's from inp data:
          Overall.F           body.mass hind.l.length.clade
             11.744              16.786               5.546

P−value:
          Overall.F           body.mass hind.l.length.clade
            0.25995             0.04659              0.24215
....
```

We have examined the difference in slopes of speed on hind limb length by including an interaction term between hind limb length and clade (and, thus, we are assuming that the slope of speed on body mass is the same in the two clades). The results provide no evidence of such interaction. Because our function builds upon the drop1 function, when reporting marginal *F* −tests we respect the usual marginality principle (see help for lm.phylog, drop1, and references therein), and thus we do not report the *F*−test for either clade or hind.l.length in the presence of the interaction term.

Finally, we can add MT/F ratio to the model.

```
> # Testing whether MT/F ratio adds anything
> speed.mtf <− lm.phylog(running.speed ~ body.mass + hind.l.length
      + mtf.ratio,data=GarlandJanis.sim, exclude.tips=c("Tm","Ur"))
> summary(speed.mtf) # we are only interested here in the F value of mtf.ratio

....

Model from inp data:
    Intercept     body.mass hind.l.length     mtf.ratio
       0.1447       −0.2874        1.1585        0.0733

F−test's from inp data:
    Overall.F     body.mass hind.l.length     mtf.ratio
      12.7457       13.8093       16.1974        0.7662

P−value:
    Overall.F     body.mass hind.l.length     mtf.ratio
      0.13317       0.10378       0.08298       0.69046
....
```

In summary, the results clearly show that, once hind limb length and body mass are in the model, MT/F ratio does not explain any additional variation in maximal sprint running speed.

In the foregoing analyses, note that if one wanted to analyze only one clade or the other

(and exclude particular species [e.g., the polar bear and grizzly bear]), the following code could be used.

```
> attach(GarlandJanis.sim) # to allow subseting of Tips for excluding
      either Carnivores or Herbivores
> speed.mass.length.carn <- lm.phylog(running.speed ~ body.mass +
      hind.l.length, data=GarlandJanis.sim, exclude.tips=
      c(as.character(Tips[clade=="Herbivore" & sim.counter==0]),"Tm","Ur"))
      # only carnivores, except polar bear and grizzly
> summary(speed.mass.length.carn)
....
```

An Example of Principal Components Analysis

Bauwens and Díaz−Uriarte (1997) carried out a PCA of seven life history traits for a set of 18 species of lacertid lizards. One of the questions asked was whether the variation in life history traits could be summarized by a smaller set of (orthogonal) axes; in other words, whether the PCA would reveal that one or a few axes accounted for most of the variance (larger and distinct eigenvalues), whereas the rest of the dimensions could be considered essentially "noise" (i.e., smaller and roughly equal eigenvalues) (see Morrison [1990] and especially Krzanowski [1990, pp. 256 and ff.] for discussion of interpretation of PCA). Bauwens and Díaz−Uriarte (1997) performed the PCA on the correlation matrix from independent contrasts, and to decide how many principal components to retain they used the broken−stick model, which usually yielded similar results to scree−plots and the eigenvalue > 1 rule (see Jackson, 1993).

We present an alternative approach, not using independent contrasts, which involves simulating phylogenetically the independent evolution of the seven traits and comparing the eigenvalues from a PCA of the original data with the eigenvalues from PCAs performed on many sets of simulated data. If the original data can be summarized by a smaller set of orthogonal axes, then we would expect to find that the first few eigenvalues are larger than most of the eigenvalues from PCAs on simulated data.

We compute the "*P*−value" of a particular eigenvalue in two different ways. First, we count the number of simulated data sets that equal or exceeded the observed eigenvalue (this is what we denominate 'component−wise' in the output of the summary.phylog.prcomp function). However, when judging the significance of any eigenvalue λi, where $i>1$, we probably will not want to apply the above approach directly. Examination of, say, the third component is not sensible unless the first and second components do explain significant amounts of variance. It is easy, especially in data with no structure, to obtain smaller *P*−values for, say, λ_2 or λ_3 than λ_1. To

minimize this problem, we provide a second method to compute the "$P-$value" that accounts for the value of previous eigenvalues. We can illustrate this method with an example: when examining the second eigenvalue, we only consider that there is support against the null hypothesis (no relevant second component) if both the first and the second eigenvalues from the observed data set are simultaneously larger than most of the (first and second) eigenvalues from the simulated data sets; thus, we compute the "$P-$value" for the second eigenvalue as the number of simulations in which the second simulated eigenvalue is larger than the second observed eigenvalue, or the first simulated eigenvalue is larger than the first observed eigenvalue, or both (so that the only cases that count against the null hypothesis are those where both the first and the second simulated eigenvalues are smaller than the observed ones). The "$P-$value" computed this way ('multiple eigenvalue') also guarantees that the $P-$values of successive eigenvalues form an increasing sequence.

Another approach to determining the number of principal components to retain is the 'parallel analysis' criterion (e.g., Horn, 1965; Zwick and Velicer, 1986; Lautenschlager, 1989). Briefly, we retain those components that have an eigenvalue larger than the mean value of that eigenvalue from the simulated data sets. This approach has been shown in simulations (in non−phylogenetic contexts) to perform as well as or better than other alternative approaches (Zwick and Velicer, 1986; Lautenschlager, 1989), although it can sometimes slightly overestimate the number of components. The observed eigenvalue for each component, as well as the mean eigenvalue for each component for the simulated data sets, are returned by our function summary.phylog.prcom (see output below) and thus we can easily apply this rule. This method can also be used graphically (see Horn, 1965), by adding to the scree plot (a plot of the value of each successive eigenvalue against its rank order) from the original data a scree plot based on the average of the simulated data sets; this plot is automatically produced by our function plot.phylog.prcomp.

First, we simulated 5,000 times the independent Brownian motion evolution of the seven traits along the phylogeny (Fig. 1 A in Bauwens and Díaz−Uriarte, 1997), using PDSIMUL. (As PDSIMUL works with pairs of traits, we used four .pdi files as input; this gave eight traits, so we simply ignored the last one.) The simulated data and the input .pdi files were then read into R using the read.sim.file function of our package.

```
> library(phylog) # load the package, if you haven't already done so.
> LacertidSim <- read.sim.data(c("ifsmi.sim", "ihshw.sim", "iclag.sim",
      "icfxx.sim"), pdi.files=c("ifsmi.pdi", "ihshw.pdi", "iclag.pdi",
      "icfxx.pdi"), variable.names= c("svl","svl.matur", "hatsvl",
      "hatweight", "clutch.size", "age.mat","cl.freq", "xx"))
> LacertidSim <- LacertidSim[,-10] # exclude last column
```

```
> LacertidPCA <- prcomp.phylog(LacertidSim)
Loading required package: mva
>

> summary(LacertidPCA)

Call:
prcomp.phylog(data = LacertidSim)

Eigenvalues from inp data:
 lambda1  lambda2  lambda3  lambda4  lambda5  lambda6  lambda7
5.304034 0.963079 0.418638 0.229861 0.063511 0.015286 0.005592

'Component-wise' P-value for eigenvalues:
  lambda1   lambda2   lambda3   lambda4   lambda5   lambda6   lambda7
0.0002000 0.9992002 0.9998000 1.0000000 1.0000000 1.0000000 1.0000000

'Multiple eigenvalue' P-value:
  lambda1   lambda2   lambda3   lambda4   lambda5   lambda6   lambda7
0.0002000 0.9992002 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000

Mean of simulated eigenvalues:
lambda1 lambda2 lambda3 lambda4 lambda5 lambda6 lambda7
 2.6091  1.6451  1.1087  0.7429  0.4718  0.2797  0.1426

Quantiles of eigenvalues' distributions:
      lambda1 lambda2 lambda3 lambda4 lambda5 lambda6 lambda7
50%     2.559   1.636   1.110  0.7430  0.4651  0.2709  0.1346
90%     3.181   1.954   1.333  0.9392  0.6375  0.3981  0.2226
95%     3.372   2.042   1.400  0.9910  0.6917  0.4360  0.2544
99%     3.798   2.206   1.521  1.0975  0.7806  0.5096  0.3103
99.9%   4.246   2.373   1.651  1.2114  0.8502  0.5925  0.3663

Number of simulations used in analyses:  5000

> par(mfrow=c(2,4))
> par(cex.lab=1.2)
> plot(LacertidPCA) # shown in Figure 2
```

The function prcomp.phylog returns, among other output, the eigenvalues for each of the data sets and, using the function summary.phylog.prcomp, we computed the "*P*−values". The results coincide with those of Bauwens and Díaz−Uriarte (1997; table 4), where only the first principal component is retained. The first eigenvalue of the original data set was larger than any of the 5,000 simulated data sets; however, the rest of the eigenvalues are very small, smaller than those from most of the simulated data sets (recall that, in a PCA, the sum of the eigenvalues is equal to the trace of the correlation matrix, and thus the number of dimensions, in this case seven; thus, after a first eigenvalue of 5.3, none of the remaining eigenvalues can possibly be larger than 1.7). Parallel analysis leads to the same conclusion (see the scree plot in Fig.2).

For completeness, here we also show how to perform the analyses using independent contrasts, which reproduces Table 4 in Bauwens and Díaz−Uriarte (1997). Note that many of the commands are used only to obtain nicer output (with explicit names, etc.). The first step is to obtain the matrix of correlation coefficients (based upon regressions through the origin) from the

independent contrasts. (Again, independent contrasts were computed and output to an ASCII file with the PDTREE program −− a .fic file.) Then, we carry out the PCA by simply performing a singular value decomposition of this correlation matrix (this has the advantage, over the "eigen" function, of providing normalized eigenvectors). The correlation between the original variables and the factors is obtained by applying standard formulas (e.g., Morrison, 1990). (Explanation of the syntax of most of the following commands is given above in "An Example of Multiple Regression and Interactions between a Categorical and a Continuous Predictor", under "Independent contrasts").

```
> LacertidIC <− cbind(read.table("ifsmi.fic")[,c(3,4)],
      read.table("ihshw.fic")[,c(3,4)],
      read.table("iclag.fic")[,c(3,4)], read.table("icfxx.fic")[,3])
      # the (unstandardized) contrasts
> stand <− read.table("ifsmi.fic")[,5] # the square root of the sum of
      branch lengths
> LacertidIC <− LacertidIC/stand # compute the standardized contrasts
> LacertidIC$contr <− read.table("ifsmi.fic")[,1] # name of the contrast
> names(LacertidIC) <− c("svl","svl.matur", "hatsvl", "hatweight",
      "clutch.size", "age.mat","cl.freq","ICcontr")
> cor.for.ic.pca <− matrix(nrow=7,ncol=7)
> for (i in 1:7) for (j in 1:7) cor.for.ic.pca[i,j]
      <− cor.origin(LacertidIC[[i]],LacertidIC[[j]]) # to obtain the
      correlation matrix based on regressions through the origin
> ic.pca <− svd(cor.for.ic.pca) #so that the eigenvectors are normalized
> cor.with.factors <− t(sqrt(ic.pca$d) * t(ic.pca$u))
> cor.with.factors <− as.data.frame(cor.with.factors)
> ic.pca$d # these are the eigenvalues
[1] 5.155831 0.796477 0.677368 0.257998 0.085071 0.022309 0.004947
> 100*ic.pca$d/length(ic.pca$d) # percentage of variance
[1] 73.65473 11.37825  9.67668  3.68568  1.21530  0.31870  0.07066
> names(cor.with.factors) <− paste("PC",seq(1:7),sep="")
      # nicer names for output
> row.names(cor.with.factors)<−names(LacertidIC)[−8]
> cor.with.factors # the correlation between the variables and the components
               PC1     PC2     PC3     PC4     PC5      PC6       PC7
svl         −0.970 −0.1184 −0.1276 −0.0117 −0.1588  0.01323  0.048264
svl.matur   −0.971 −0.0995 −0.1500 −0.0384 −0.1376 −0.04760 −0.046745
hatsvl      −0.923  0.1902 −0.2887 −0.1038  0.0757  0.10799 −0.013055
hatweight   −0.929  0.2435 −0.1988 −0.0800  0.1502 −0.08956  0.015745
clutch.size −0.792 −0.4934  0.0444  0.3433  0.0960  0.00847 −0.002483
age.mat     −0.716 −0.2634  0.5856 −0.2711  0.0408  0.00780 −0.000455
cl.freq      0.644 −0.6035 −0.4133 −0.2188  0.0420 −0.00719  0.002706
```

The results are identical to those in Bauwens and Díaz−Uriarte (1997), except for the sign (which is arbitrary, as we can define the new principal axes as pointing in one direction or its opposite).

Two Examples of Generalized Least Squares

As a first example of a fit using GLS, we can use again the data set in Bauwens and Díaz−Uriarte (1997); here, we mainly illustrate the function fitting embedded within repeated function calls. Table 2 in Bauwens and Díaz−Uriarte (1997) shows the slope and standard error of the bivariate regressions of several life−history traits on female SVL (snout−vent length). They obtained these statistics using IC; we can reproduce their results using GLS instead (for simplicity, we will only reproduce part of the table, corresponding to the same variables as in the PCA example above). The first step is to read the original data set (contained in LacertidOriginal). We also need to obtain the phylogenetic variance−covariance matrix for use with GLS. This matrix was obtained from the program PDDIST (using ifsmi.pdi as input file; for output, we used option 5, with shared branch lengths [see Garland and Ives, 2000], in matrix form, with header) and read into R with the read.phylog.matrix function (the output of this function, in ascii format, is included in the 'data' directory of our package with the name Lacertid.varcov). Now, we only need to call the function phylog.gls.fit to obtain the fit for the appropriate pair of dependent and independent variables. Because calling that function repeatedly becomes tedious, the commands below show how to obtain, in table form, the slope and standard error of the fit for the variables SVL at maturity, hatchling SVL, hatchling mass, clutch size, age at maturity, and clutch frequency.

```
> LacertidOriginal <− LacertidSim[LacertidSim$sim.counter==0,−1]
      # we could read the pdi data again, but this is another simple way,
      since the original data are in the file with the simulated data
> Lacertid.varcov <− read.phylog.matrix("ifsmi.dsc")
      # from pddist, option 5, in matrix form, with header and scaled
>
> # attach the data set, so variables can be accessed directly
> # without reference to the data set
> attach(LacertidOriginal)
>
> t(apply(LacertidOriginal[,−c(1,2)], 2, function(y)
      {summary(phylog.gls.fit(svl, y, Lacertid.varcov))[[4]][2,c(1,2)]}))
            Estimate Std. Error
svl.matur     0.9529    0.02909
hatsvl        0.5724    0.06922
hatweight     1.8041    0.24922
clutch.size   1.2363    0.23006
age.mat       0.7460    0.21922
cl.freq      −0.4931    0.21051
```

As should be the case (see Garland and Ives, 2000), the results are identical to the ones reported in Table 2 of Bauwens and Díaz−Uriarte (1997), who used independent contrasts. The last command is the one that carries out the computations, and serves to illustrate some of the advantages of working in R. First, to reproduce the table in Bauwens and Díaz−Uriarte (1997), we are only interested in the fitted coefficient for the slope and its standard error: for this example, we are not

interested in the rest of the information returned by the fitting function (such as the intercept) or by the summary.lm function (such as *t*−value, global *F*, degrees of freedom, etc.). Thus, we call the fitting function, phylog.gls.fit, and ask for the summary of the fitted model (which will provide the standard error), but we only want the fourth component of the list (coefficients) and from those, only the second row (slope, instead of intercept), and the first and second columns (fitted coefficient and standard error, but not *t*−value or the *P*−value). In other words, the call to

```
> summary(phylog.gls.fit(x,y,varcov))[[4]][2,c(1,2)] (x1)
```

returns the fitted slope and its standard error for a GLS model with variance−covariance matrix varcov, dependent variable y and independent variable(s) x.

Now, if we want to repeat the fitting with several dependent variables, we could use a for loop; for instance, a call such as:

```
> for (i in 3:8) print( summary( phylog.gls.fit( svl,
      eval( parse( text = names( LacertidOriginal)[i])),
      Lacertid.varcov))[[4]] [2, c(1,2)])
```

But the use of for loops (and other looping) is discouraged in languages such as R and S (and in other languages such as Octave, Scilab, Matlab, Mathematica, etc.), where vectorized operations are generally faster and result in simpler and easier−to−understand code than loops. We thus use the built−in function apply; in this case, apply applies a function (the one on x1) over the columns (i.e., the second dimension, and thus the 2) of the data frame (in this case, we are not interested in the first two columns of the data frame, which are the names and SVL, and thus those are excluded). The function applied in this case is what is called an "anonymous" function; this is a function that we define inside the call to another function (in this case apply). As an alternative we might have used:

```
> summary.table.in.bd <−
      function(y){summary(phylog.gls.fit(svl,y,Lacertid.varcov))
                  [[4]][2,c(1,2)] }
```

and called apply as:

```
> t(apply(LacertidOriginal[,−c(1,2)],2,summary.table.in.bd(y)))
```

but that seems an unnecessary step as we are unlikely to need such an operation again. Finally, the call to apply is embedded inside a t(x) function, where t is the transpose; in this case, we use the transpose to obtain tabular output where the variables are the rows and the columns are the slope

and the standard error.

As a second example of fitting a GLS model, we repeat some of the independent contrasts analyses from Garland and Janis (1993) reported above. Here, we show the use of the fitting function with multiple predictors and the exclusion of certain tips. We first need to read in the data and the variance−covariance matrix. As in other examples, we assume that the files are in the current working directory. The .pdi files contain the original data (plus the phylogenetic information, which is not used here); the file 49ms.dsc is the phylogenetic variance−covariance matrix obtained from the PDDIST program (using 49ms.pdi as input file; for output, we used option 5, with shared branch lengths, in matrix form, with header). Please refer to the help file for additional information on the fitting function (in particular, at the time of this writing, the *F* statistic returned for the whole model is wrong; this function will be re−written once a reporting problem in the nlme package is fixed −−−this problem was present in the latest version, 3.1−10, of the nlme package).

```
> Garland.Janis.Orig <− read.pdi.data(c("49ms.pdi","49hmt.pdi"),
      variable.names = c("body.mass", "running.speed", "hind.l.length",
      "mtf.ratio"))
> Garland.Janis.Orig$clade <− as.factor(c(rep("Carnivore",19),
      rep("Herbivore",30)))
> Garland.Janis.Cov <− read.phylog.matrix("49ms.dsc")
>
> detach(LacertidOriginal)
> detach(garland.janis.ic) # just in case it is attached
> attach(Garland.Janis.Orig)
>
> # all data
>
> fit.gls.gj <− phylog.gls.fit(cbind(body.mass,hind.l.length),
      running.speed, Garland.Janis.Cov) # the model fitting call

> summary(fit.gls.gj) # summary of the gls model; same as with IC
....

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.0522     0.3803    2.77   0.0081 **
body.mass      −0.1053     0.0762   −1.38   0.1739
hind.l.length   0.4678     0.2743    1.71   0.0948 .
....
```

Following Garland and Janis (1993) we next exclude the grizzly and polar bear.

```
> # without the grizzly and polar bear
> summary(phylog.gls.fit(cbind(body.mass,hind.l.length), running.speed,
      Garland.Janis.Cov, exclude.tips=c("Tm","Ur"))) # as expected, not
      identically equal to IC
....

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.5772     0.4119    1.40   0.1682
```

```
body.mass      −0.1928     0.0818   −2.36   0.0230 *
hind.l.length   0.8283     0.2999    2.76   0.0083 **

....
```

Our conclusions are the same as from the IC analyses. Note, however, that when we exclude the two bears results should not exactly agree with the IC analysis done above (after deleting polar−grizzly), because to make them agree you would have to delete the tip species, redraw the tree, and then redo the contrasts.

References

Bauwens, D., and R. Díaz−Uriarte. 1997. Covariation of life−history traits in lacertid lizards: a comparative study. American Naturalist 149:91−111.

Garland, T., Jr., A. W. Dickerman, C. M. Janis, and J. A. Jones. 1993. Phylogenetic analysis of covariance by computer simulation. Syst. Biol. 42:265−292.

Garland, T., Jr., P. H. Harvey, and A. R. Ives. 1992. Procedures for the analysis of comparative data using phylogenetically independent contrasts. Syst. Biol. 41:18−32.

Garland, T. Jr., and A. R. Ives. 2000.Using the past to predict the present: confidence intervals for regression equations in phylogenetic comparative methods. American Naturalist 155:346−364.

Garland, T. Jr., and C. M. Janis. 1993. Does metatarsal/femur ratio predict maximal running speed in cursorial mammals? Journal of Zoology (London) 229:133−151.

Garland, T., Jr., P. E. Midford, and A. R. Ives. 1999. An introduction to phylogenetically based statistical methods, with a new method for confidence intervals on ancestral values. Am. Zool. 39:374−388.

Hocking, R. R. 1996. Methods and applications of linear models. John Wiley & Sons, New York.

Horn, J. L. 1965. A rationale and test for the number of factors in factor analysis. Psychometrica 30:179−185.

Jackson, D. A. 1993. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. Ecology 74:2204−2214.

Krzanowski, W. J. 1990. Principles of Multivariate Analysis. Oxford University Press, Oxford.

Lautenschlager, G. J. 1989. A comparison of alternatives to conducting Monte Carlo analyses for determining parallel analysis criteria. Multivariate Behavioral Research 24:365−395.

Morrison, D. F. 1990. Multivariate Statistical Methods, 3r Ed. McGraw−Hill, New York.

Zwick, W. R., and W. F. Velicer. 1986. Comparison of five rules for determining the number of components to retain. Psychological Bulletin 99:432−442.

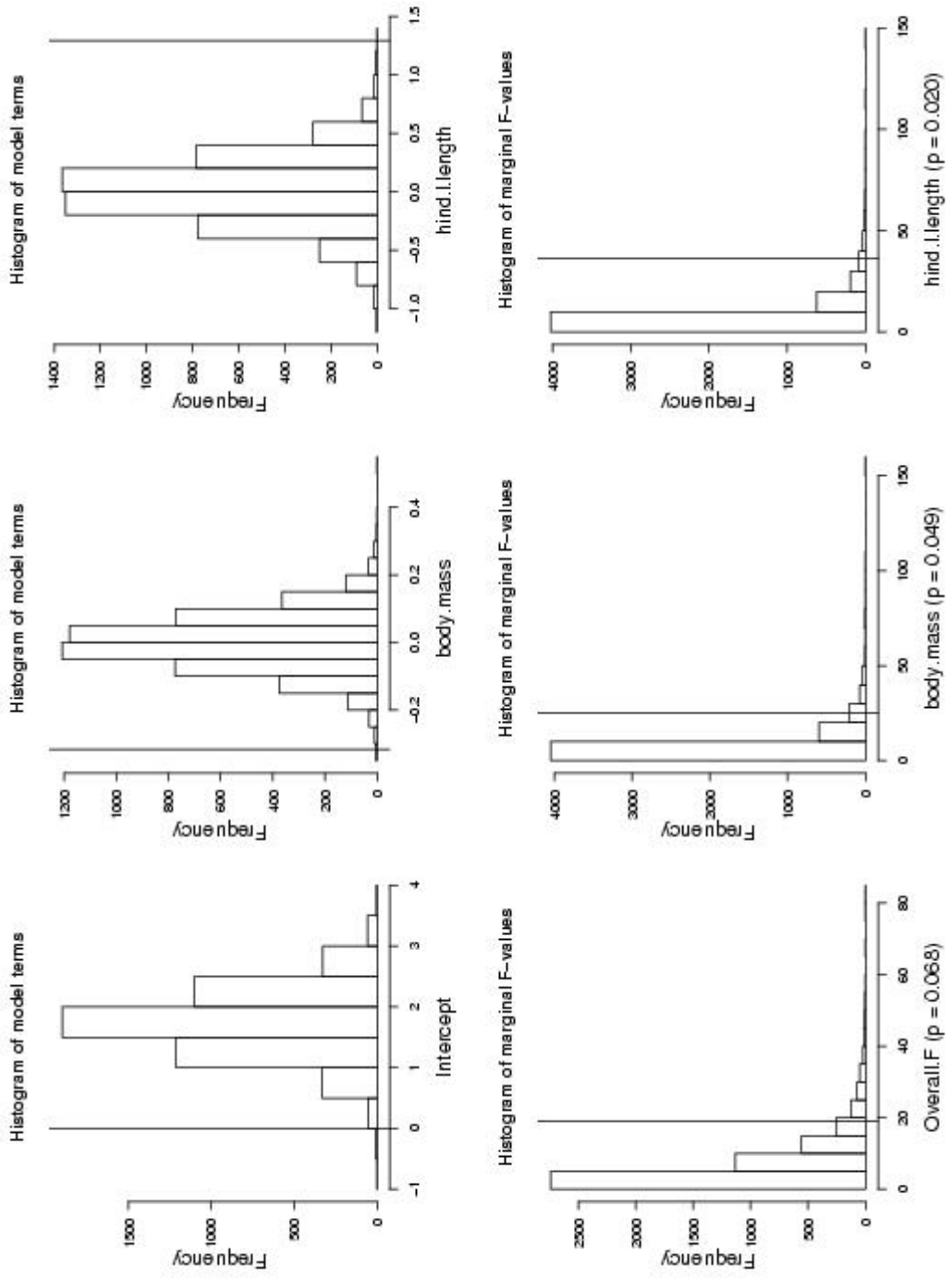

Figure 1. Result of calling "plot" on an object output from lm.phylog (see text for complete sequence of commands). This plot presents summaries of the linear model fitted to a set of simulated data. The histograms show the fitted coefficients (intercept, two partial regression slopes −− i.e., the slopes of fitting a particular term after all others are in the model) and marginal *F*−tests (the F−value corresponding to a term after all others are in the model) from the simulated data (N = 5,000) for the specified linear model. The vertical bars indicate the values for that model from the original (non−simulated) data. For the *F*−tests, the values in parentheses are the proportion of simulations with an *F*−value greater than the *F*−value of the original data (i.e., the *P*−value).

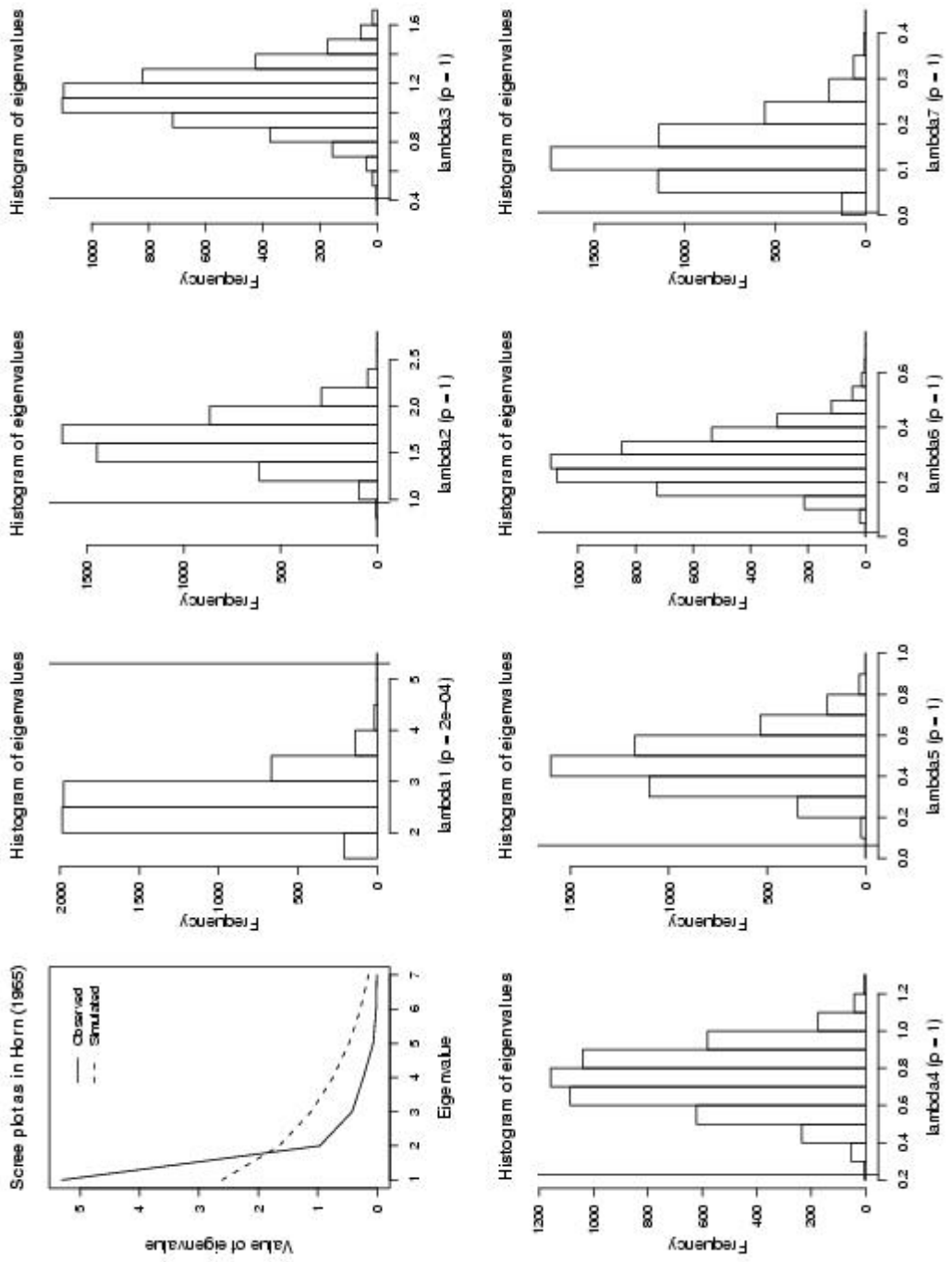

Figure 2. Output from calling "plot" on an object output from prcomp.phylog (see text for complete sequence of commands). This plot presents summaries of a principal components analysis performed on a set of simulated data. In the first plot, the solid lined is a scree plot from the original data and the dotted line is a scree plot based on the average value of each eigenvalue from the simulated data (N = 5,000). This plot is based on Horn's (1965) method of determining the number of components to retain (also called parallel analysis); see references in text. The rest of the plots show histograms of eigenvalues from the simulated data, with the vertical bar indicating the value of the eigenvalue from the original (non−simulated) data; the values in parentheses are the proportion of simulations with an eigenvalue larger than the eigenvalue of the original data (what we denominate "component−wise P−value" in the text).

Appendix 1

Downloading and Installing R

R can be obtained from CRAN and mirror sites. For Windows users, there are compiled binaries available, which can be accesses by going to "Sources and binaries" and then "Binary distributions"; choose "Windows". The base R package is available under "base". R can be installed in several ways. A simple way is to download all the needed files (if you have a fast network connection, you might want to download everything except maybe the sources −− *sp.zip; if you have a slow connection, you can decide which of the help file formats you are not interested −−maybe the LaTeX or HTML ones); download also rwinst.exe. You might want to download all these files to, say, "c:\Program Files\ R". After downloading the base files, go to "contrib", and download the packages you want, including PHYLOGR, and maybe some others, such as nlme. Start Explorer and click on rwinst, which has a wizard−like interface that will ask you a few simple questions and allow to install both the base R packages and the contributed packages you have downloaded. (Install the base package before the contributed libraries). Contributed packages can also be installed without using rwinst by unzipping them under the \library directory. If using Unix, including GNU/Linux, binaries might be available for your particular architecture/operating system. First, download the base *.rpm file. If it exists, also download the rpm file for the package you want; otherwise, go to "Contributed extensions" and download the tar.gz file. Once you have the files, start by installing the base system, using the rpm manager of your choice (and possibly after changing privileges to root). If the package is available as rpm, install it now too using the rpm manager. Otherwise, type R INSTALL name_of_package.tar.gz. If no binaries are available for your particular Linux or Unix system, then obtain the tgz for the R−base, as well as the tar.gz files for the packages you want. Most of the time, installing R from the tgz file is as simple as unpacking it, typing ./configure, typing make, and if that is successful, typing make install. For instance, this is what R.D−U does in his Linux system: first, change permissions to root. Next, change to /usr/lib and unpack with GNU's tar "tar −zxvf R−x.y.z.tgz". cd to the created directory, and type "./configure" and "make". Next, "make prefix=/usr/local install" and "make install−pdf" for the help. The complete process can take a few minutes (as the program needs to be compiled from sources); note also that with the approach above we end up with the program in /usr/local/lib/R and the sources in /usr/lib/R−x.y.z. Then install the packages with R INSTALL name_of_package.tar.gz. The "INSTALL" file, available on the web site or in the tgz file, provides detailed explanation on building R from

sources.

If you plan to use R a lot, then you will want to consider installing ESS (for Emacs Speaks Statistics), another free, open−source, GNU program. ESS uses GNU Emacs or XEmacs (both open−source, free programs released under the GNU GPL license) to provide a common interface to many statistical packages (including R, S, SAS, XLisp−Stat, with future additions planned such as Stata, Fiasco, SPSS). You will almost certainly want to use ESS if you do development and programming in R. ESS provides, among other things, powerful command−line editing, searcheable command history, command−line completion, several hot−keys, transcript recording, an interface to the help system, a specialized editing mode for R code (including syntax highlighting, automatic indentation of code, information on mismatched parentheses and brackets, etc.) and facilities for evaluating and loading R code. The last two alone make ESS invaluable. To use ESS, you will first need to install Emacs or XEmacs (either or both come with most Linux distributions); next, ESS can be obtained from CRAN (under "Miscellaneous"). Installation in Linux systems is straightforward; if you are most lucky, rpm files will be available −− install them and modify your .emacs file. Otherwise, download the tar.gz file, unpack, and follow the README (you might need to modify the ess−site.el file to specify where the R binary is located; you will also want to modify .emacs file so that ESS is loaded when Emcas/XEmacs starts). ESS and Emacs and XEmacs are also available for use under Windows, but we have limited experience with them; information is available from CRAN (and in the R−help email list there is a message, on 7−March−2001, by Emmanuel Paradis, that summarizes the downloading and installation of Emacs and ESS under Windows; its URL is http://www.ens.gu.edu.au/robertk/R/help/01a/0749.html).