Ecological factors influencing primate vocal signaling: a phylogenetic regression workflow for the *mmodely* R-package (Version 0.2.5)

David M. Schruth
dschruth@anthropoidea.org

May 16, 2023

1 Introduction

The historical relationships between evolved traits of organisms and the ecological settings that shape these traits are complicated systems that can be challenging to untangle [2]. The origins of behavioral traits can particularly difficult to understand as they tend to also be mediated through the behaviors of other organisms, which are themselves constantly in flux and considerably labile [3]. A perfect example of such a trait is that of vocal signal complexity. Animals use complex calls to assert obscured position, unique identity, special status, or emotive state to conspecifics over interference from other calls or distortions from background noise [1]. Here, using the *mmodely* package on a primate vocalization dataset [15], I demonstrate how the origins of complex call structure, such as syllablic diversity [17], can be elucidated from a range of environmental and behavioral covariates from disparate datasets [10]. Model averaging [MA] [6] and model selection [MS] [7] results primarily highlight locomotion [16] and mating system [13] as important factors driving complex calling, as well as the trophic security [18] variables of mass, group size, and arboreality. The *mmodely* package enables implementation of a combination of phylogenetic controlled regression [8] and information theoretic [9] (MA and MS) examination to simultaneously compare (weighted) predictor coefficients across the numerous sub-datasets generated during exploration of all possible model combinations.

2 Licensing

The *mmodely* package is licensed under the Apache License v2.0: it is therefore free to use and redistribute, however, we, the copyright holders, wish to maintain primary control over any further development. Please be sure to cite *mmodely* if you use the package in presentations or work leading to publication.

3 Installation

This package largely depends upon the *caper* package, but most functions do not require any particular library. It is recommended that you have *caper*, *ape*, and the *caroline* package installed as a minimum.

```
> # wget https://cran.r-project.org/src/contrib/Archive/caroline/caroline_0.8.0.tar.gz
> # wget https://cran.r-project.org/src/contrib/Archive/caper/caper_0.5.tar.gz
> # wget https://cran.r-project.org/src/contrib/Archive/ape/ape_3.0-5.tar.gz
> # R CMD INSTALL caroline_0.8.0.tar.gz
> # R CMD INSTALL caper_0.5.tar.gz
> # R CMD INSTALL ape_3.0-5.tar.gz
```

Building the *mmodely* package from source requires that you have the proper dependency packages, *caroline*, installed from CRAN. This can typically be accomplished via the following commands from within the R command line environment:

```
install.packages(c('caroline','ape','caper'))
```

After a successful installation the *mmodely* package can be loaded in the normal way: by starting R and invoking the following `library` command:

```
> library(caper)
> library(mmodely)
```

4 Reading in Data

Read in the tree [14] and datasets then merge them together.

```
> data.path <- system.file("extdata","primate-example.data.csv", package="mmodely")
> data <- read.csv(data.path, row.names=1)
> data$gn_sp <- rownames(data)
> #multiply two vocalization metrics together to create "vocal complexity"
> data$VC <- apply(data[,c('syllables_max','rhythm_max')], 1, prod)
> data <- subset(data, !is.na(VC))
> # merge data sets here if applicable
>
> tree.path <- system.file("extdata","primate-springer.2012.tre", package="mmodely")
> phyl <- ape::read.tree(tree.path)[[5]]
> #5. RAxML phylogram based on the 61199 bp concatenation of 69 nuclear and ten mitochondrial genes.
>
> phyl <- trim.phylo(phylo=phyl, gs.vect=data$gn_sp) # prune unused nodes and branches
> comp <- comp.data(phylo=phyl, df=data)
```

Typically there will be some missing data (species) in certain sources that do not occur in others. A merge of these will result in NA values for some cells. The more missing cells and merges there are, the more sub-datasets will be possible, due to case-wise deletion in the process of combinatorics underlying model iteration, averaging, and selection. The above example has little if any missing data, but the examples below introduce some artificially.

5 Basic Reporting

First, for illustration purposes, we perform a simple analysis of a single model using 'pgls' directly from the *caper* package, then show-off the 'pgls.report' functionality of the *mmodely* package. ANOVA, AIC, and one-line model reports can be output via this function.

```
> model <- as.formula('VC ~ mass.Kg + group.size')
> fit <- caper::pgls(formula=model, data=comp)
> summary(fit)

Call:
caper::pgls(formula = model, data = comp)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.9014 -0.9478  0.0030  1.2281  8.6394 

Branch length transformations:

kappa  [Fix]  : 1.000
lambda [Fix]  : 1.000
delta  [Fix]  : 1.000

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  2.2345483  1.1333734  1.9716   0.0662 .
mass.Kg     -0.0079678  0.0082070 -0.9709   0.3461  
group.size   0.0071381  0.0144792  0.4930   0.6287  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.61 on 16 degrees of freedom
Multiple R-squared: 0.1101,	Adjusted R-squared: -0.001146 
F-statistic: 0.9897 on 2 and 16 DF,  p-value: 0.3933 

> pgls.report(comp, f=model, anova=TRUE, QC.plot=TRUE)
```

```
Call:
pgls(formula = f, data = cd, lambda = l, kappa = k, delta = d,
    bounds = bounds)

Residuals:
    Min      1Q  Median      3Q     Max
-7.9014 -0.9478  0.0030  1.2281  8.6394

Branch length transformations:

kappa  [Fix]  : 1.000
lambda [Fix]  : 1.000
delta  [Fix]  : 1.000

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.2345483  1.1333734  1.9716   0.0662 .
mass.Kg     -0.0079678  0.0082070 -0.9709   0.3461
group.size   0.0071381  0.0144792  0.4930   0.6287
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.61 on 16 degrees of freedom
Multiple R-squared: 0.1101,        Adjusted R-squared: -0.001146
F-statistic: 0.9897 on 2 and 16 DF,  p-value: 0.3933
[1] "AIC = 58"
Analysis of Variance Table
Sequential SS for pgls: lambda = 1.00, delta = 1.00, kappa = 1.00

Response: VC
           Df Sum Sq Mean Sq F value Pr(>F)
mass.Kg     1  36.89  36.894  1.7363 0.2062
group.size  1   5.16   5.164  0.2430 0.6287
Residuals  16 339.96  21.248
 group(0.629)  |  mass(0.346)

Call:
pgls(formula = f, data = cd, lambda = l, kappa = k, delta = d,
    bounds = bounds)

Coefficients:
(Intercept)      mass.Kg   group.size
   2.234548    -0.007968     0.007138
```

6 Multivariate Combinatoric Iteration

The *mmodely* package's chief contribution is enabling approaches that utilize multi-model iteration averaging. Using a smaller subset of variables can speed up the (slower) maximum likelihood computation step and still achieve the desired result of fixed tree transformation parameters.

```
> pv0 <- c("mass.Kg","arboreal","home.range","monogamy") #"swing.pct"
> est.mods <- get.model.combos(predictor.vars=pv0, outcome.var='VC', min.q=2)
> ps <- get.phylo.stats(phylo=phyl, data=data, trait.clmn='VC');

$lambda
[1] 0.2903945

$logL
[1] -55.25736

$P
[1] 0.7103404

$K
[1] 0.1886703

$P
[1] 0.388

> lambda <- ps$lambda$lambda ; print(lambda)

[1] 0.2903945

> PGLSi <- pgls.iter(models=est.mods, phylo=phyl, df=data, l=lambda, k='ML', d='ML')

1 VC~mass.Kg+arboreal+home.range+monogamy
2 VC~mass.Kg+arboreal+home.range
3 VC~mass.Kg+arboreal+monogamy
4 VC~mass.Kg+home.range+monogamy
5 VC~arboreal+home.range+monogamy
6 VC~mass.Kg+arboreal
7 VC~mass.Kg+home.range
8 VC~mass.Kg+monogamy
9 VC~arboreal+home.range
10 VC~arboreal+monogamy
11 VC~home.range+monogamy
```

7 Tree Transformation Averaging and Re-iteration

After running PGLS on a test-subset of predictor-variable combinations using maximum likelihood, we can average the tree transformation parameters [20]to obtain fixed values going forward. This approach can speed up computations for larger sets of modeling data and variable combinations.

```
> tt.avgs <- apply(PGLSi$params, 2, mean, na.rm=TRUE) # tree transformation averages
> print(tt.avgs)
```

```
        l         k         d 
0.2903945 0.5281747 1.2349216 
```

Next we use the full set of variables and our tree transform averages. For demonstration, we sprinkle in some missing values to our dataset so as to artificially boost the number of sub-datasets. The subsequent fixed tree parameter itteration run should now generate more diverse output upon which the *mmodely* can demonstrate it unique model averaging and model selection functionality.

```
> pvs <- c("mass.Kg","group.size","arboreal","monogamy","leap.pct","swing.pct")
> all.mods <- get.model.combos(predictor.vars=pvs, outcome.var='VC', min.q=2)
> # randomly sprinkle in some missing values (for more interesting for model selection)
> missing.value.ct <- 1
> for(pv in pv0){ data[sample(x=1:nrow(data),size=missing.value.ct),pv] <- NA}
> PGLSi <- pgls.iter(models=all.mods, phylo=phyl, df=data, l=lambda, k=tt.avgs['k'], d=tt.avgs['d'])
```

```
1 VC~mass.Kg+group.size+arboreal+monogamy+leap.pct+swing.pct
2 VC~mass.Kg+group.size+arboreal+monogamy+leap.pct
3 VC~mass.Kg+group.size+arboreal+monogamy+swing.pct
4 VC~mass.Kg+group.size+arboreal+leap.pct+swing.pct
5 VC~mass.Kg+group.size+monogamy+leap.pct+swing.pct
6 VC~mass.Kg+arboreal+monogamy+leap.pct+swing.pct
7 VC~group.size+arboreal+monogamy+leap.pct+swing.pct
8 VC~mass.Kg+group.size+arboreal+monogamy
9 VC~mass.Kg+group.size+arboreal+leap.pct
10 VC~mass.Kg+group.size+arboreal+swing.pct
11 VC~mass.Kg+group.size+monogamy+leap.pct
12 VC~mass.Kg+group.size+monogamy+swing.pct
13 VC~mass.Kg+group.size+leap.pct+swing.pct
14 VC~mass.Kg+arboreal+monogamy+leap.pct
15 VC~mass.Kg+arboreal+monogamy+swing.pct
16 VC~mass.Kg+arboreal+leap.pct+swing.pct
17 VC~mass.Kg+monogamy+leap.pct+swing.pct
18 VC~group.size+arboreal+monogamy+leap.pct
19 VC~group.size+arboreal+monogamy+swing.pct
20 VC~group.size+arboreal+leap.pct+swing.pct
21 VC~group.size+monogamy+leap.pct+swing.pct
22 VC~arboreal+monogamy+leap.pct+swing.pct
23 VC~mass.Kg+group.size+arboreal
24 VC~mass.Kg+group.size+monogamy
25 VC~mass.Kg+group.size+leap.pct
26 VC~mass.Kg+group.size+swing.pct
27 VC~mass.Kg+arboreal+monogamy
28 VC~mass.Kg+arboreal+leap.pct
29 VC~mass.Kg+arboreal+swing.pct
30 VC~mass.Kg+monogamy+leap.pct
31 VC~mass.Kg+monogamy+swing.pct
32 VC~mass.Kg+leap.pct+swing.pct
33 VC~group.size+arboreal+monogamy
34 VC~group.size+arboreal+leap.pct
35 VC~group.size+arboreal+swing.pct
36 VC~group.size+monogamy+leap.pct
37 VC~group.size+monogamy+swing.pct
38 VC~group.size+leap.pct+swing.pct
```

```
39 VC~arboreal+monogamy+leap.pct
40 VC~arboreal+monogamy+swing.pct
41 VC~arboreal+leap.pct+swing.pct
42 VC~monogamy+leap.pct+swing.pct
43 VC~mass.Kg+group.size
44 VC~mass.Kg+arboreal
45 VC~mass.Kg+monogamy
46 VC~mass.Kg+leap.pct
47 VC~mass.Kg+swing.pct
48 VC~group.size+arboreal
49 VC~group.size+monogamy
50 VC~group.size+leap.pct
51 VC~group.size+swing.pct
52 VC~arboreal+monogamy
53 VC~arboreal+leap.pct
54 VC~arboreal+swing.pct
55 VC~monogamy+leap.pct
56 VC~monogamy+swing.pct
57 VC~leap.pct+swing.pct
```

```
> pgls.iter.stats(PGLSi)
```

```
models: 57
dimensions of sub-datasets:
    q     n   qXn rwGsm
    5     4    13     8
               q        n
Min.    2.000000 35.00000
1st Qu. 2.000000 36.00000
Median  3.000000 36.00000
Mean    3.263158 36.36842
3rd Qu. 4.000000 37.00000
Max.    6.000000 38.00000
tree transformation parameter averages:
        l         k         d
0.2903945 0.5281747 1.2349216
distributions of optimization parameters:
       n               q              rwGsm          model.no        R2
 Min.   :35.00   Min.   :2.000   Min.   :66208   Min.   : 1   Min.   :0.01109
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:67391   1st Qu.:15   1st Qu.:0.10176
 Median :36.00   Median :3.000   Median :68338   Median :29   Median :0.20633
 Mean   :36.37   Mean   :3.263   Mean   :68449   Mean   :29   Mean   :0.17562
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69521   3rd Qu.:43   3rd Qu.:0.23777
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.32544
     R2.adj              AIC             AICc             BIC             AICw
 Min.   :-0.07473   Min.   :103.4   Min.   :104.8   Min.   :108.8   Min.   :0.0002774
 1st Qu.: 0.01808   1st Qu.:105.2   1st Qu.:107.4   1st Qu.:112.8   1st Qu.:0.0018433
 Median : 0.11970   Median :107.6   Median :109.2   Median :114.8   Median :0.0097902
 Mean   : 0.09256   Mean   :108.4   Mean   :110.0   Mean   :115.2   Mean   :0.0175439
 3rd Qu.: 0.15113   3rd Qu.:111.8   3rd Qu.:112.6   3rd Qu.:117.8   3rd Qu.:0.0240160
 Max.   : 0.21302   Max.   :114.7   Max.   :116.4   Max.   :122.3   Max.   :0.0920487
      BICw
 Min.   :0.0001794
 1st Qu.:0.0016746
 Median :0.0075783
 Mean   :0.0175439
 3rd Qu.:0.0200342
 Max.   :0.1552189
```

8 Fixed iteration run statistics

We should briefly inspect how this fixed iteration run performed and how many sub-datasets we need to investigate. It is recommended to try *mmodely* using 'rwGsm.' This abbreviation stands for 'raw *Genus species* sums.' It represents a sum of the (concatenated) raw character values of all species constituting the underlying dataset (which has all rows with any missing data removed) for a particular combination of model predictor variables. While this default is preferred, the number of species 'n' [default] or number of model variables 'q' can also be used.

```
> pgls.iter.stats(PGLSi)
```

```
models: 57
dimensions of sub-datasets:
    q     n   qXn rwGsm
    5     4    13     8
               q        n
Min.    2.000000 35.00000
1st Qu. 2.000000 36.00000
Median  3.000000 36.00000
Mean    3.263158 36.36842
3rd Qu. 4.000000 37.00000
Max.    6.000000 38.00000
tree transformation parameter averages:
        l         k         d
0.2903945 0.5281747 1.2349216
distributions of optimization parameters:
       n               q             rwGsm          model.no        R2
 Min.   :35.00   Min.   :2.000   Min.   :66208   Min.   : 1   Min.   :0.01109
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:67391   1st Qu.:15   1st Qu.:0.10176
 Median :36.00   Median :3.000   Median :68338   Median :29   Median :0.20633
 Mean   :36.37   Mean   :3.263   Mean   :68449   Mean   :29   Mean   :0.17562
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69521   3rd Qu.:43   3rd Qu.:0.23777
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.32544
     R2.adj              AIC             AICc            BIC             AICw
 Min.   :-0.07473   Min.   :103.4   Min.   :104.8   Min.   :108.8   Min.   :0.0002774
 1st Qu.: 0.01808   1st Qu.:105.2   1st Qu.:107.4   1st Qu.:112.8   1st Qu.:0.0018433
 Median : 0.11970   Median :107.6   Median :109.2   Median :114.8   Median :0.0097902
 Mean   : 0.09256   Mean   :108.4   Mean   :110.0   Mean   :115.2   Mean   :0.0175439
 3rd Qu.: 0.15113   3rd Qu.:111.8   3rd Qu.:112.6   3rd Qu.:117.8   3rd Qu.:0.0240160
 Max.   : 0.21302   Max.   :114.7   Max.   :116.4   Max.   :122.3   Max.   :0.0920487
      BICw
 Min.   :0.0001794
 1st Qu.:0.0016746
 Median :0.0075783
 Mean   :0.0175439
 3rd Qu.:0.0200342
 Max.   :0.1552189
```

9 Model Averaging

Now we can estimate the predictor variable parameters by averaging over all possible fixed PGLS runs, using the AICc differences (from the lowest AICc) as weights. By default this AICw weighted average is performed per sub-dataset using 'rwGsm' or 'n' [default] as mentioned in the preceding section. While model averaging is not recommended under high multicolinearity, as denominators of regression coefficients change across models, it is possible to rescale these using 'standarize' [12]. A slightly more conservative alternative to MA uses 'variable importance' which is equivalent to an AIC-weighted MA of binary indicators of presence or absence of covariate model inclusion [11].

```
> w.means.pds <- average.fit.models(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim, by='rwGsm', standardize=TRU
> #
> apply(w.means.pds, 2, mean, na.rm=T) #average of weighted means over all sub-datasets

   mass.Kg group.size   arboreal   monogamy   leap.pct  swing.pct
 0.0197875  0.0139975 -0.0751200  0.9198450  1.4733825  1.4690650

> w.means.pds                                    # weighted means    per   sub-dataset

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
66208 0.02228    0.01844  0.23057  0.97197  1.13352   0.92342
67391 0.01762    0.00951 -0.24452      NaN  2.19048   2.30204
67809     NaN    0.01828  0.03065  0.87552  0.90958   0.87484
68338 0.02050    0.01619      NaN  0.94623  1.16789   0.99047
68992     NaN    0.01064 -0.31718      NaN  1.88644   2.06998
69521 0.01875    0.01005      NaN      NaN  1.97116   2.01011
69939     NaN    0.01728      NaN  0.88566  0.82964   0.78281
71122     NaN    0.01159      NaN      NaN  1.69835   1.79885
attr(,"MSE")
      mass.Kg group.size arboreal monogamy leap.pct swing.pct
66208 0.00069    0.00005  0.06311  0.06288  0.05128   0.05838
67391 0.00012    0.00002  0.00696       NA  0.01045   0.01171
67809      NA    0.00003  0.03013  0.03178  0.02325   0.02776
68338 0.00149    0.00017       NA  0.08949  0.07458   0.08301
68992      NA    0.00002  0.00485       NA  0.00641   0.00726
69521 0.00015    0.00004       NA       NA  0.00957   0.01057
69939      NA    0.00009       NA  0.03711  0.02464   0.03007
71122      NA    0.00002       NA       NA  0.00278   0.00359

> w.import.pds <- variable.importance(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim, by='rwGsm')
> #
> apply(w.import.pds, 2, mean, na.rm=T) #average of weighted means over all sub-datasets

   mass.Kg group.size   arboreal   monogamy   leap.pct  swing.pct
  1.000000   1.000000   0.505985   1.000000   1.000000   1.000000

> w.import.pds                                    # weighted means    per   sub-dataset

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
66208       1          1  0.93388        1        1         1
67391       1          1  0.22462      NaN        1         1
67809     NaN          1  0.71300        1        1         1
68338       1          1      NaN        1        1         1
68992     NaN          1  0.15244      NaN        1         1
69521       1          1      NaN      NaN        1         1
69939     NaN          1      NaN        1        1         1
71122     NaN          1      NaN      NaN        1         1
attr(,"MSE")
      mass.Kg group.size arboreal monogamy leap.pct swing.pct
66208       0          0  0.10686        0        0         0
67391       0          0  0.01280       NA        0         0
67809      NA          0  0.09782        0        0         0
68338       0          0       NA        0        0         0
```

```
68992       NA         0  0.00795       NA        0         0
69521        0         0       NA       NA        0         0
69939       NA         0       NA        0        0         0
71122       NA         0       NA       NA        0         0
```

10 Model Selection

We can select the best model by sorting each subset (e.g. by AICc) or by using visualization methods.

```
> select.best.models(PGLSi, using='AICc')
```

```
    n q  qXn rwGsm model.no        R2    R2.adj      AIC     AICc      BIC        AICw
36 36 2 2X36 68338       45 0.2063335 0.1582325 104.0005 104.7505 108.7510 0.092048666
35 35 3 3X35 66208       27 0.2084916 0.1318940 103.9130 105.2463 110.1344 0.071836928
37 37 2 2X37 69939       49 0.2228344 0.1771187 105.1064 105.8337 109.9391 0.053556133
38 38 2 2X38 71122       57 0.1551053 0.1068256 110.2290 110.9348 115.1417 0.004179294
          BICw
36 0.155218953
35 0.077724220
37 0.085694404
38 0.006356653
```

Plotting the coefficients of determination versus the AIC values allows selection of high-performing models for inspection and reporting.

```
> plot.pgls.iters(PGLSi)
```

```
> sdevs.objs <- get.pgls.coefs(PGLSi$fits, est='t value')
> coefs.objs <- get.pgls.coefs(PGLSi$fits, est='Estimate')
```

```
> report.vect <- sapply(1:length(PGLSi$fits), function(i) fit.1ln.rprt(PGLSi$fits[[i]], rtrn.line=FALSE, mn=i)
```

```
1 +monog(0.084) +leap(0.139) +group(0.146) +mass(0.146) +swing(0.23) arbore(0.924)  |   R2adj: 0.18 AICc: 108.5
2 ++mono(0.015) +mass(0.155) +group(0.155) leap(0.329) arbore(0.595)  |   R2adj: 0.165 AICc: 107.2
3 ++mono(0.008) +group(0.145) +mass(0.216) arbore(0.535) swing(0.679)  |   R2adj: 0.142 AICc: 108.16
4 ++leap(0.01) ++swin(0.029) +mass(0.236) group(0.348)  |  arbore(0.557) R2adj: 0.116 AICc: 111.23
5 +monog(0.074) +leap(0.101) +mass(0.129) +group(0.131) +swing(0.182)  |   R2adj: 0.213 AICc: 107.04
6 +mass(0.122) +leap(0.138) +monog(0.181) +swing(0.251)  |  arbore(0.806) R2adj: 0.145 AICc: 108.06
7 +monog(0.112) +group(0.12) +leap(0.195) +swing(0.241)  |  arbore(0.751) R2adj: 0.151 AICc: 109.69
8 +++mon(0.005) +group(0.143) +mass(0.197) arbore(0.471)  |   R2adj: 0.166 AICc: 105.44
9 +leap(0.084) mass(0.345) group(0.619) arbore(0.947)  |   R2adj: -0.006 AICc: 114.16
10 +swing(0.289) mass(0.584) group(0.628) arbore(0.807)  |   R2adj: -0.069 AICc: 116.36
11 ++mono(0.012) +mass(0.165) +group(0.167) +leap(0.263)  |   R2adj: 0.191 AICc: 106.32
12 ++mono(0.006) +group(0.159) +mass(0.249) swing(0.577)  |   R2adj: 0.165 AICc: 107.44
13 ++leap(0.009) ++swin(0.03) +mass(0.195) +group(0.269)  |   R2adj: 0.14 AICc: 110.54
14 ++mono(0.038) +mass(0.129) +leap(0.308) arbore(0.862)  |   R2adj: 0.134 AICc: 106.75
15 ++mono(0.021) +mass(0.183) swing(0.728) arbore(0.783)  |   R2adj: 0.107 AICc: 107.84
16 ++leap(0.013) ++swin(0.039) +mass(0.191)  |  arbore(0.421) R2adj: 0.118 AICc: 109.41
17 +mass(0.092) +leap(0.12) +monog(0.159) +swing(0.236)  |   R2adj: 0.177 AICc: 106.93
18 ++mono(0.021) +group(0.128) leap(0.445) arbore(0.892)  |   R2adj: 0.139 AICc: 108.46
19 ++mono(0.012) +group(0.122) swing(0.607) arbore(0.83)  |   R2adj: 0.13 AICc: 108.84
20 ++leap(0.016) ++swin(0.033) +group(0.257)  |  arbore(0.436) R2adj: 0.111 AICc: 111.69
21 +monog(0.094) +group(0.096) +leap(0.194) +swing(0.248)  |   R2adj: 0.18 AICc: 108.67
22 +leap(0.201) +monog(0.245) +swing(0.267)  |  arbore(0.569) R2adj: 0.108 AICc: 109.74
23 mass(0.569) arbore(0.663) group(0.73)  |   R2adj: -0.075 AICc: 114.98
24 +++mon(0.003) +group(0.166) +mass(0.237)  |   R2adj: 0.183 AICc: 105.09
25 +leap(0.063) mass(0.333) group(0.59)  |   R2adj: 0.03 AICc: 113.41
26 +swing(0.239) mass(0.611) group(0.631)  |   R2adj: -0.034 AICc: 115.78
27 ++mono(0.012) +mass(0.168) arbore(0.719)  |   R2adj: 0.132 AICc: 105.25
28 +leap(0.086) +mass(0.301)  |  arbore(0.938) R2adj: 0.018 AICc: 111.74
29 +swing(0.308) mass(0.521) arbore(0.908)  |   R2adj: -0.044 AICc: 113.93
30 ++mono(0.029) +mass(0.117) +leap(0.265)  |   R2adj: 0.165 AICc: 105.87
31 ++mono(0.015) +mass(0.18) swing(0.662)  |   R2adj: 0.137 AICc: 107.07
32 ++leap(0.015) ++swin(0.048) +mass(0.143)  |   R2adj: 0.133 AICc: 109.29
33 ++mono(0.007) +group(0.121) arbore(0.744)  |   R2adj: 0.15 AICc: 106.44
34 +leap(0.108) group(0.481)  |  arbore(0.989) R2adj: 0.004 AICc: 114.34
35 +swing(0.267) group(0.514) arbore(0.83)  |   R2adj: -0.039 AICc: 115.88
```

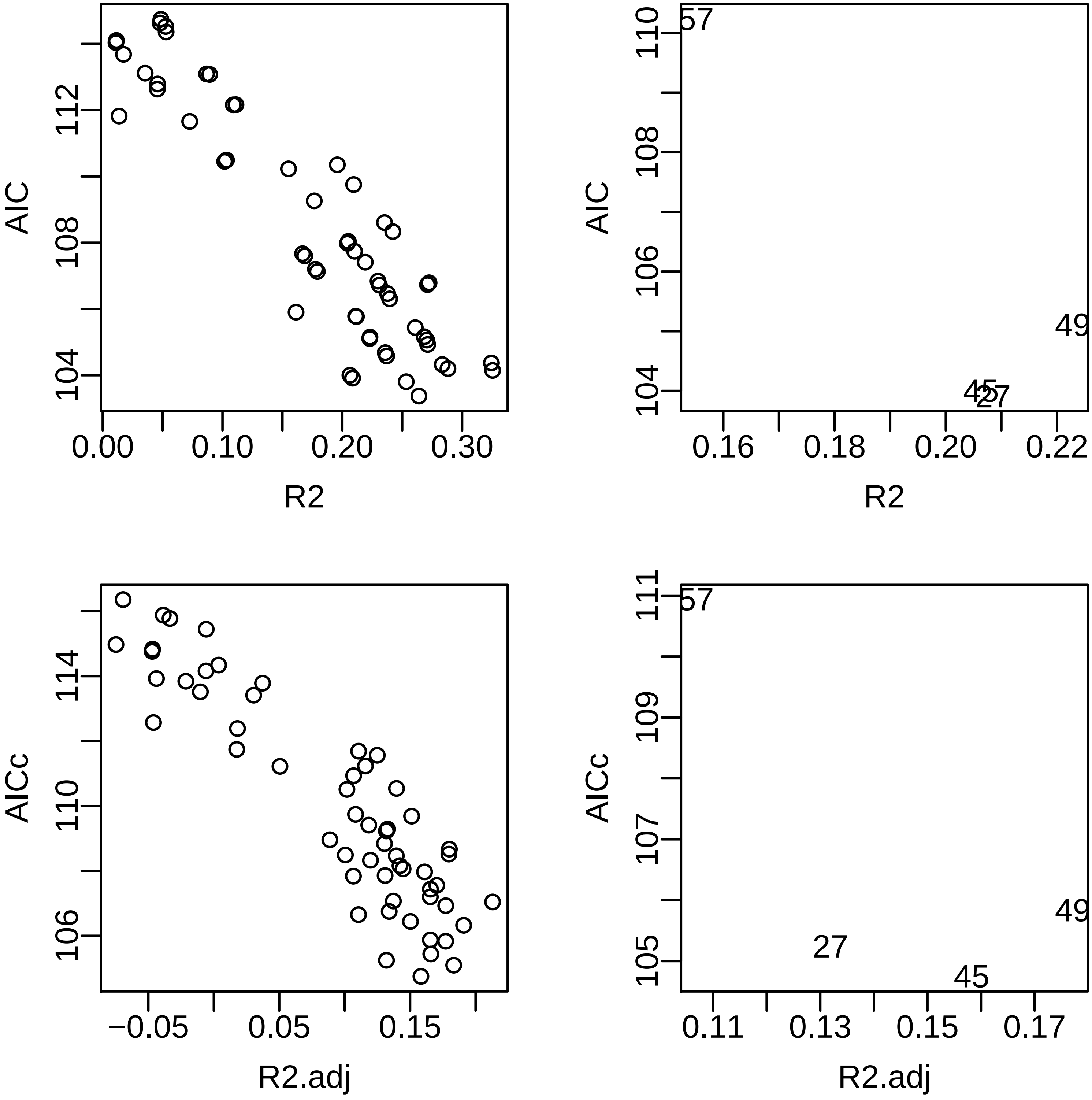

Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of each streak. Therefore, minimizing AIC tends to also maximize the coefficient of determination, but not necessarily. This four panel plot looks at correct and adjusted versions of each model assessment measure. All points are scaled by subdataset sample size by default if 'n' is used in grouping.

```
36 ++mono(0.017) +group(0.115) leap(0.402)  |   R2adj: 0.17 AICc: 107.56
37 ++mono(0.008) +group(0.112) swing(0.56)  |   R2adj: 0.161 AICc: 107.97
38 ++leap(0.018) ++swin(0.041) +group(0.198)  |   R2adj: 0.125 AICc: 111.56
39 +monog(0.059) leap(0.433)  |  arbore(0.88) R2adj: 0.1 AICc: 108.49
40 ++mono(0.035) swing(0.651)  |  arbore(0.952) R2adj: 0.089 AICc: 108.96
41 ++leap(0.023) ++swin(0.049)  |  arbore(0.333) R2adj: 0.102 AICc: 110.51
42 +monog(0.219) +leap(0.233) +swing(0.315)  |   R2adj: 0.132 AICc: 109.23
43 mass(0.63) group(0.789)  |   R2adj: -0.047 AICc: 114.83
44 mass(0.521) arbore(0.725)  |   R2adj: -0.046 AICc: 112.57
45 ++mono(0.007) +mass(0.172)  |   R2adj: 0.158 AICc: 104.75
46 +leap(0.068) +mass(0.273)  |   R2adj: 0.051 AICc: 111.22
47 +swing(0.261) mass(0.523)  |   R2adj: -0.01 AICc: 113.52
48 group(0.603) arbore(0.646)  |   R2adj: -0.047 AICc: 114.77
49 +++mon(0.004) +group(0.115)  |   R2adj: 0.177 AICc: 105.83
50 +leap(0.081) group(0.442)  |   R2adj: 0.037 AICc: 113.78
51 +swing(0.217) group(0.509)  |   R2adj: -0.006 AICc: 115.45
52 ++mono(0.021) arbore(0.972)  |   R2adj: 0.111 AICc: 106.65
53 +leap(0.12)  |  arbore(0.846) R2adj: 0.018 AICc: 112.39
54 +swing(0.293) arbore(0.96)  |   R2adj: -0.021 AICc: 113.84
55 ++mono(0.051) leap(0.42)  |   R2adj: 0.131 AICc: 107.86
56 ++mono(0.027) swing(0.637)  |   R2adj: 0.12 AICc: 108.33
57 ++leap(0.033) +swing(0.075)  |   R2adj: 0.107 AICc: 110.93
```

```
> par(mar=c(5,5,3,3))
> plot.pgls.R2AIC(PGLSi$optim)
```

11 Coefficient Plotting

Finally, the resulting model fits from the PGLS runs can be be plotted out horizontally as distributions so the influence of each ecological predictor variable can be compared.

```
> par.old <- par(mar=c(5,8,1,4),mfrow=c(2,1))
> sparge.modsel(sdevs.objs, R2x=7, xlab='t value')
> sparge.modsel(coefs.objs, R2x=7, xlab='Estimate')
```

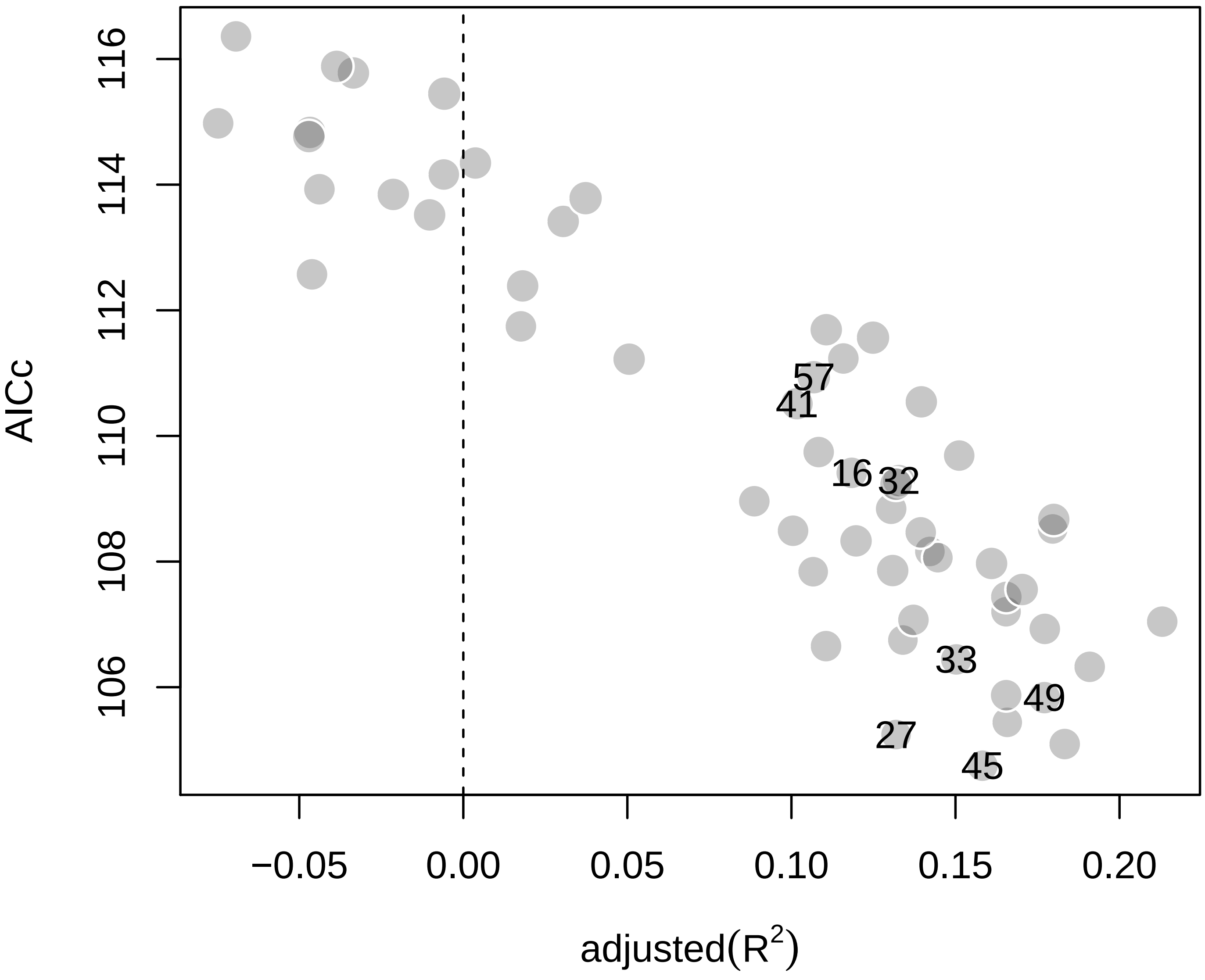

Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is not a bad idea that all "best" models selected from each sub-dataset should get reported, such as in the form of the 'sparge' plot below.

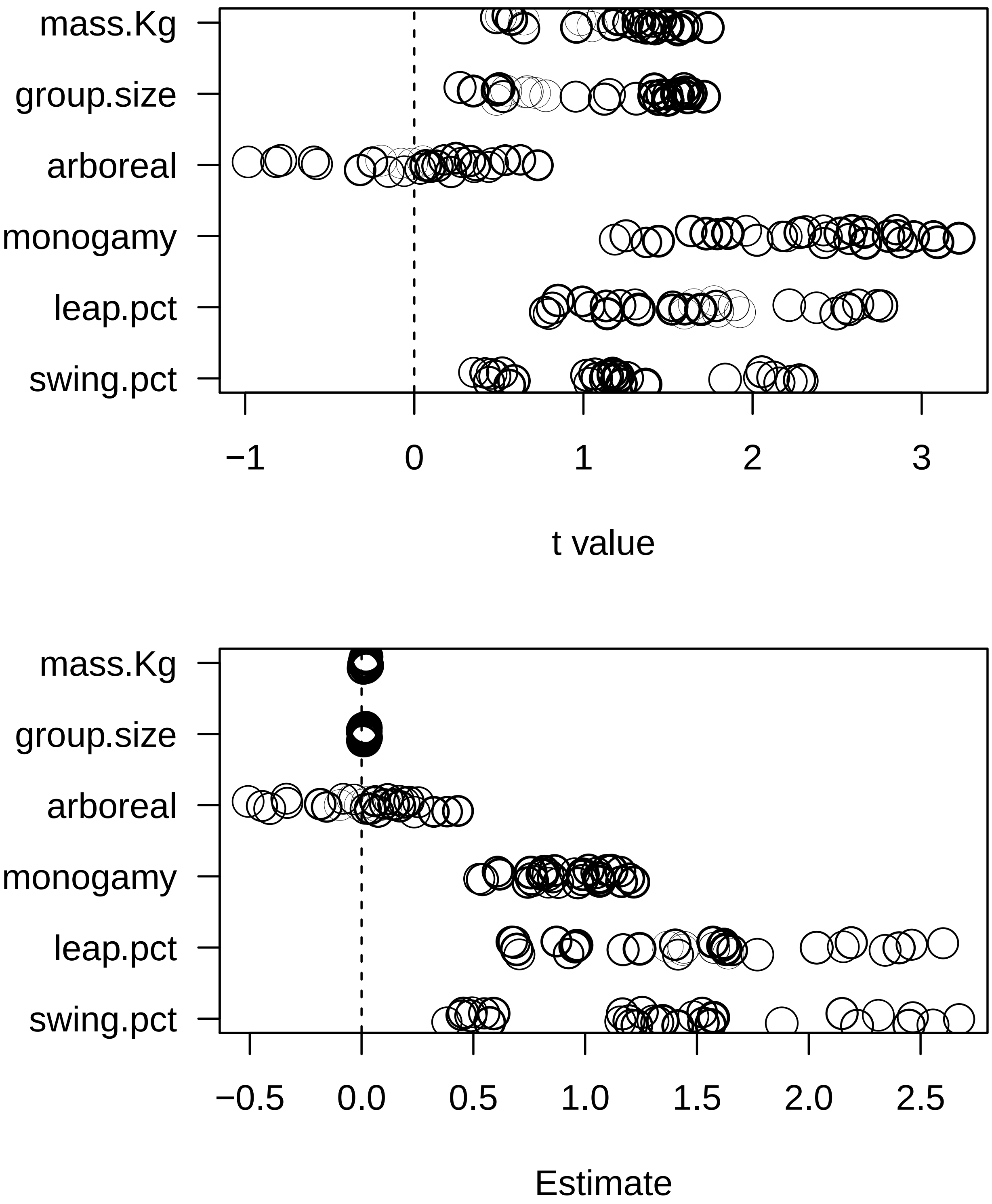

Figure 3: These horizontal parameter distribution dot-plots demonstrate how the (t-values of) coefficients from models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting in each model, point sizes represent underlying sample sizes and circle thickness corresponds coefficient of determination values. Note that mate choice, locomotion, and statural factors drive complex (here rhythmically syllabic) calling in primates.

References

[1] Rogers, L.J. and Kaplan, G. (2000) *Songs, Roars, and Rituals: communication in birds, mammals, and other animals.* Harvard University Press, Cambridge.

[2] Pagel, M. (1999) 'Inferring the historical patterns of biological evolution' *Nature*, 401, pp. 877–884. Available at: `https://doi.org/10.1038/44766`.

[3] Blomberg, S.P., Garland Jr, T., and Ives, A.R. (2003) 'Testing for phylogenetic signal in comparative data: behavioral traits are more labile' Evolution, 57(4) pp. 717–745. Available at: `https://doi.org/10.1111/j.0014-3820.2003.tb00285.x`.

[4] Grueber, C.E., Nakagawa, .R.J., and Jamieson, I.G. (2011) 'Multimodel inference in ecology and evolution: challenges and solutions' *Journal of Evolutionary Biology*, 24, pp. 699-711. Available at: `https://doi.org/10.1111/j.1420-9101.2010.02210.x`

[5] Stephens, P.A., Buskirk, S.W., and del Rio, C.M. (2007) 'Inference in ecology and evolution' *Trends in Ecology and Evolution*, 22(4). Available at: `https://doi.org/10.1016/j.tree.2006.12.003`.

[6] Symonds, M.R.E., and Moussalli, A. (2011) 'A brief guide to model selection, multimodel inference and model averaging in behavioral ecology using Akaike's information criterion' *Behavioral Ecology and Sociobiology*, 65, pp. 13–21. Available at: `https://doi.org/10.1007/s00265-010-1037-6`.

[7] Johnson, J.B., Omland, K.S. (2004) 'Model selection in ecology and evolution' *Trends in Ecology and Evolution*, 19(2). Available at: `https://doi.org/10.1016/j.tree.2003.10.013`.

[8] Revell, L.J. (2014) phytools: An R package for phylogenetic comparative biology (and other things). [CRAN]. Available at: `http://cran.r-project.org/package=phytools`.

[9] Garamszegi L.Z. (2011) 'Information-theoretic approaches to statistical analysis in behavioral ecology: an introduction' *Behavioral Ecology and Sociobiology*, 65, pp. 1–11. Available at: `https://doi.org/10.1007/s00265-010-1028-7`.

[10] Rowe, N. and Meyers, M. (2017) *All the World's Primates.* Charlestown, RI: Pogonias Press.

[11] Burnham, P.B. and Anderson, D.R. (2000) *'Model Selection and Inference: A Practical Information-Theoretic Approach'*

[12] Cade, B.S. (2015) 'Model averaging and muddled multimodel inferences' *Ecology*, 96(9), pp. 2370–2382. Available at: `https://doi.org/abs/10.1890/14-1639.1`.

[13] Feuntes, A. (1998) 'Re-evaluating primate monogamy' *American Anthropology*, 100(4) pp. 890-907.

[14] Springer, M.S., et.al. (2012) 'Re-evaluating primate monogamy' *PLoS ONE* 7(11) p. e49521 `http://doi.org/10.1371/journal.pone.0049521`.

[15] Schruth, D.M. (2019a) 'Structural acoustic features of human musicality scored on primate vocalizations' *The Center for Open Science.* Available at: `https://doi.io/bvsfz/`.

[16] Schruth, D.M. (2019b) 'Primate Locomotor Activity' *The Center for Open Science.* Available at: `https://osf.io/cd68q/`.

[17] Schruth, D.M., Templeton, C.N. and Holman, D.J. (2021a) 'On reappearance and complexity in musical calling' *PLoS ONE.* Available at: `https://doi.org/10.1371/journal.pone.0218006`.

[18] Schruth, D.M. (2021b) Arboreal locomotion and trophic security at the dawn of Euprimate vision. EcoEvoRxiv. Available at: `http://doi.org/10.32942/osf.io/d6wk2`.

[19] Schruth, D.M. (2021c) 'A global variable-permutation based approach for estimating tree transformation parameters used in phylogenetically controlled multivariate regression' *Protocols.io.* Available at: `http://doi.org/10.17504/protocols.io.bzdhp236`.

[20] Schruth, D.M. (2021d) 'Primates evolved spectrally complex calls in compensation for reduction in olfactory cognition' *Proceedings of the Annual Meeting of the Cognitive Science Society* 43. Available at: `https://escholarship.org/uc/item/0jw446s9`.