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

November 7, 2024

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 
-8.7308 -3.1969  0.4275  3.3780  6.5978 

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
-8.7308 -3.1969  0.4275  3.3780  6.5978

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] 0.412

> 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.   :65144   Min.   : 1   Min.   :0.00277
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:66794   1st Qu.:15   1st Qu.:0.08203
 Median :36.00   Median :3.000   Median :67632   Median :29   Median :0.13546
 Mean   :36.37   Mean   :3.263   Mean   :67871   Mean   :29   Mean   :0.12603
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69282   3rd Qu.:43   3rd Qu.:0.17919
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.22066
     R2.adj               AIC              AICc             BIC              AICw
 Min.   :-0.086500   Min.   : 99.35   Min.   :101.3   Min.   :105.9   Min.   :0.0001045
 1st Qu.:-0.004353   1st Qu.:102.83   1st Qu.:104.4   1st Qu.:110.3   1st Qu.:0.0009978
 Median : 0.042908   Median :105.16   Median :106.7   Median :112.7   Median :0.0082233
 Mean   : 0.037069   Mean   :106.23   Mean   :107.8   Mean   :113.0   Mean   :0.0175439
 3rd Qu.: 0.077311   3rd Qu.:109.94   3rd Qu.:110.9   3rd Qu.:116.1   3rd Qu.:0.0257698
 Max.   : 0.124869   Max.   :114.74   Max.   :115.4   Max.   :119.7   Max.   :0.1220876
      BICw
 Min.   :0.0001617
 1st Qu.:0.0009796
 Median :0.0053146
 Mean   :0.0175439
 3rd Qu.:0.0173413
 Max.   :0.1637862
```

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.   :65144   Min.   : 1   Min.   :0.00277
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:66794   1st Qu.:15   1st Qu.:0.08203
 Median :36.00   Median :3.000   Median :67632   Median :29   Median :0.13546
 Mean   :36.37   Mean   :3.263   Mean   :67871   Mean   :29   Mean   :0.12603
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69282   3rd Qu.:43   3rd Qu.:0.17919
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.22066
     R2.adj               AIC              AICc            BIC             AICw
 Min.   :-0.086500   Min.   : 99.35   Min.   :101.3   Min.   :105.9   Min.   :0.0001045
 1st Qu.:-0.004353   1st Qu.:102.83   1st Qu.:104.4   1st Qu.:110.3   1st Qu.:0.0009978
 Median : 0.042908   Median :105.16   Median :106.7   Median :112.7   Median :0.0082233
 Mean   : 0.037069   Mean   :106.23   Mean   :107.8   Mean   :113.0   Mean   :0.0175439
 3rd Qu.: 0.077311   3rd Qu.:109.94   3rd Qu.:110.9   3rd Qu.:116.1   3rd Qu.:0.0257698
 Max.   : 0.124869   Max.   :114.74   Max.   :115.4   Max.   :119.7   Max.   :0.1220876
      BICw
 Min.   :0.0001617
 1st Qu.:0.0009796
 Median :0.0053146
 Mean   :0.0175439
 3rd Qu.:0.0173413
 Max.   :0.1637862
```

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
 0.00363000  0.01107125 -0.22032000  0.60523250  1.27212125  1.77186250

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

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
65144 0.00290    0.00806 -0.23872  0.56262  0.86774   1.80372
66794 0.00204   -0.00020 -0.32282      NaN  2.00097   2.48352
66984 0.00486    0.01192      NaN  0.64980  0.71679   1.19559
67632     NaN    0.01862 -0.12379  0.54833  0.64914   1.63092
68634 0.00472    0.00422      NaN      NaN  1.86769   1.95571
69282     NaN    0.01128 -0.19595      NaN  1.73075   2.18807
69472     NaN    0.02060      NaN  0.66018  0.57761   1.08351
71122     NaN    0.01407      NaN      NaN  1.76628   1.83386
attr(,"MSE")
      mass.Kg group.size arboreal monogamy leap.pct swing.pct
65144 0.00040    0.00008  0.07939  0.09547  0.08602   0.12468
66794 0.00004    0.00002  0.00670       NA  0.00896   0.01077
66984 0.00038    0.00011       NA  0.05222  0.04045   0.05003
67632      NA    0.00003  0.02664  0.04029  0.02900   0.03787
68634 0.00003    0.00001       NA       NA  0.00318   0.00346
69282      NA    0.00001  0.00231       NA  0.00303   0.00375
69472      NA    0.00007       NA  0.02095  0.01327   0.01607
71122      NA    0.00001       NA       NA  0.00069   0.00087

> 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
 0.9521850  0.9237200  0.1300075  1.0000000  1.0000000  1.0000000

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

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
65144 1.00000    1.00000  0.00000        1        1         1
66794 0.84559    0.61189  0.12302      NaN        1         1
66984 1.00000    1.00000      NaN        1        1         1
67632     NaN    1.00000  0.00000        1        1         1
68634 0.96315    0.77787      NaN      NaN        1         1
69282     NaN    1.00000  0.39701      NaN        1         1
69472     NaN    1.00000      NaN        1        1         1
71122     NaN    1.00000      NaN      NaN        1         1
attr(,"MSE")
      mass.Kg group.size arboreal monogamy leap.pct swing.pct
65144 0.00000    0.00000  0.00000        0        0         0
66794 0.01234    0.00283  0.01097       NA        0         0
66984 0.00000    0.00000       NA        0        0         0
67632      NA    0.00000  0.00000        0        0         0
```

```
68634 0.00304    0.00140       NA       NA        0         0
69282      NA    0.00000  0.00495       NA        0         0
69472      NA    0.00000       NA        0        0         0
71122      NA    0.00000       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
35 35 3 3X35 65144       27 0.1085831 0.02231696  99.98764 101.3210 106.2090 0.1220875650
36 36 2 2X36 66984       45 0.1181154 0.06466780 101.10814 101.8581 105.8587 0.0933311623
37 37 2 2X37 69472       49 0.1562989 0.10666944 103.38713 104.1144 108.2199 0.0302054248
38 38 2 2X38 71122       57 0.1551053 0.10682562 110.22896 110.9348 115.1417 0.0009978073
          BICw
35 0.137468562
36 0.163786183
37 0.050298095
38 0.001579413
```

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 +swing(0.069) +leap(0.242) monoga(0.397) mass(0.596) group(0.604)  |  arbore(0.466) R2adj: 0.054 AICc: 105.4
2 +monog(0.073) group(0.558) mass(0.638) leap(0.781)  |  arbore(0.863) R2adj: -0.03 AICc: 106.48
3 +swing(0.149) +monog(0.163) group(0.605) mass(0.688)  |  arbore(0.714) R2adj: 0.04 AICc: 104.02
4 ++swin(0.019) ++leap(0.032) mass(0.701) group(0.882)  |  arbore(0.374) R2adj: 0.086 AICc: 109.1
5 +swing(0.137) +leap(0.225) +monog(0.243) +group(0.3) +mass(0.324)  |   R2adj: 0.077 AICc: 106.11
6 +swing(0.062) +leap(0.236) monoga(0.471) mass(0.668)  |  arbore(0.335) R2adj: 0.077 AICc: 102.63
7 +swing(0.086) +group(0.156) +leap(0.3) monoga(0.329)  |  arbore(0.538) R2adj: 0.089 AICc: 106.02
8 +monog(0.055) group(0.556) mass(0.654)  |  arbore(0.906) R2adj: 0.002 AICc: 103.65
9 +leap(0.213) mass(0.886)  |  group(0.862) arbore(0.859) R2adj: -0.066 AICc: 112.92
10 +swing(0.116)  |  mass(0.974) arbore(0.842) group(0.813) R2adj: -0.035 AICc: 111.84
11 ++mono(0.054) group(0.377) mass(0.465) leap(0.591)  |   R2adj: 0.037 AICc: 105.91
12 +monog(0.068) +swing(0.305) group(0.355) mass(0.473)  |   R2adj: 0.061 AICc: 105.01
13 ++leap(0.018) ++swin(0.039) mass(0.391) group(0.537)  |   R2adj: 0.099 AICc: 110.59
14 +monog(0.084) mass(0.727) leap(0.791)  |  arbore(0.694) R2adj: -0.008 AICc: 103.97
15 +swing(0.136) +monog(0.187) mass(0.769)  |  arbore(0.555) R2adj: 0.063 AICc: 101.42
16 ++swin(0.017) ++leap(0.028) mass(0.713)  |  -arbor(0.317) R2adj: 0.115 AICc: 106.23
17 +swing(0.162) +leap(0.261) monoga(0.375) mass(0.38)  |   R2adj: 0.073 AICc: 104.52
18 +monog(0.058) +group(0.168) leap(0.846)  |  arbore(0.951) R2adj: 0.026 AICc: 106.72
19 +monog(0.131) +swing(0.16) +group(0.162)  |  arbore(0.817) R2adj: 0.086 AICc: 104.43
20 ++swin(0.021) ++leap(0.035) +group(0.288)  |  arbore(0.482) R2adj: 0.106 AICc: 110.17
21 +group(0.09) +swing(0.168) +monog(0.226) +leap(0.29)  |   R2adj: 0.112 AICc: 106.85
22 +swing(0.09) +leap(0.321) monoga(0.582)  |  arbore(0.439) R2adj: 0.056 AICc: 105.57
23 arbore(0.918)  |  mass(0.949) group(0.731) R2adj: -0.087 AICc: 112.04
24 ++mono(0.03) group(0.386) mass(0.513)  |   R2adj: 0.058 AICc: 103.54
25 +leap(0.098) mass(0.716) group(0.969)  |   R2adj: 0 AICc: 112.89
26 +swing(0.25) mass(0.932)  |  group(0.932) R2adj: -0.044 AICc: 114.51
27 +monog(0.062) mass(0.744)  |  arbore(0.729) R2adj: 0.022 AICc: 101.32
28 +leap(0.194) mass(0.857)  |  arbore(0.902) R2adj: -0.034 AICc: 110.25
29 +swing(0.106) mass(0.995)  |  arbore(0.908) R2adj: -0.004 AICc: 109.19
30 +monog(0.081) mass(0.514) leap(0.621)  |   R2adj: 0.043 AICc: 104.12
31 +monog(0.108) swing(0.328) mass(0.523)  |   R2adj: 0.064 AICc: 103.31
32 ++leap(0.02) ++swin(0.046) mass(0.415)  |   R2adj: 0.116 AICc: 108.35
33 ++mono(0.044) +group(0.163)  |  arbore(0.994) R2adj: 0.055 AICc: 104.05
34 +leap(0.212) group(0.526) arbore(0.889)  |   R2adj: -0.026 AICc: 113.71
35 +swing(0.12) group(0.525) arbore(0.82)  |   R2adj: 0.001 AICc: 112.73
```

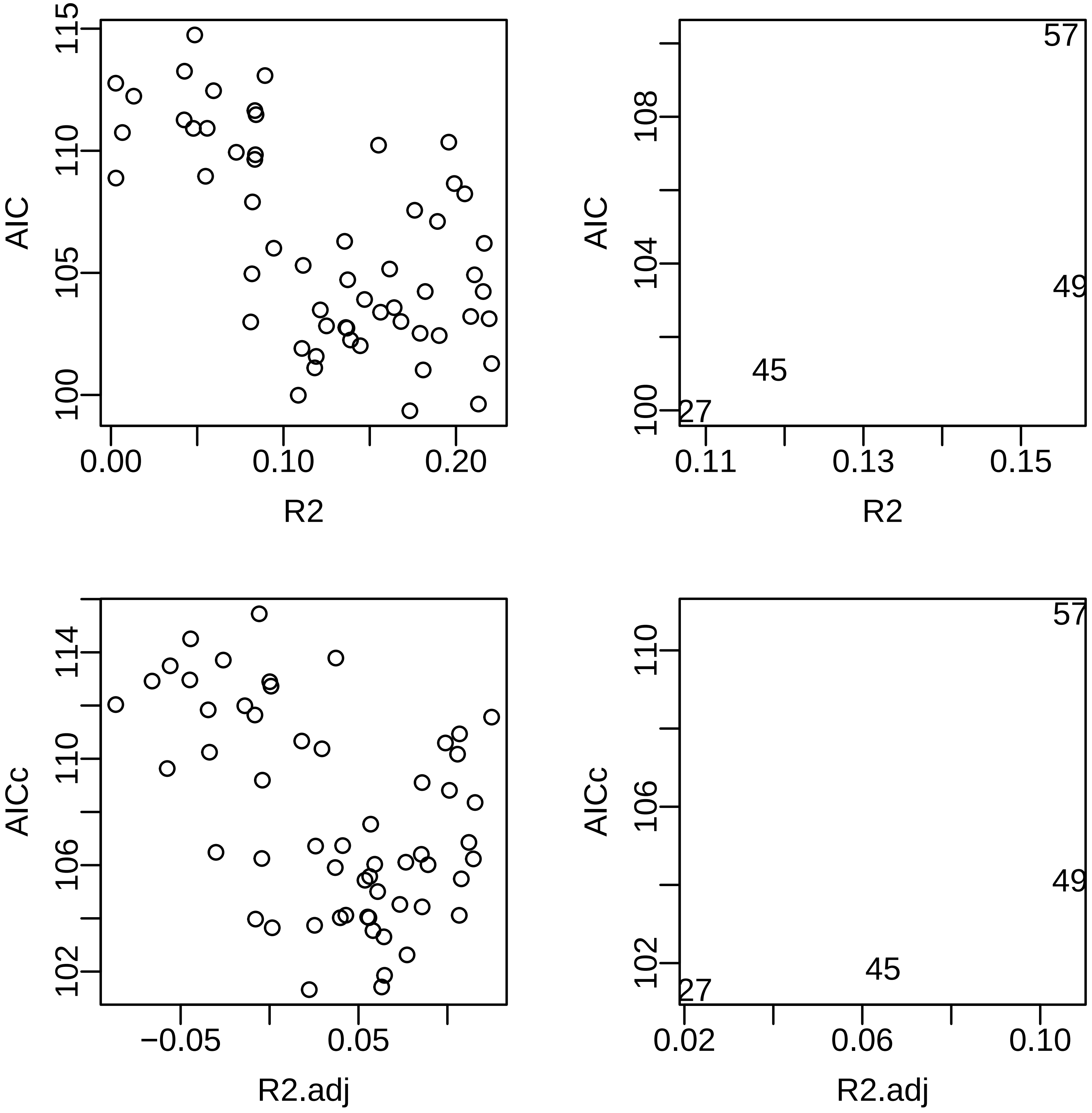

Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of 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(0.044) +group(0.113) leap(0.652)  |   R2adj: 0.085 AICc: 106.41
37 +monog(0.057) +group(0.101) +swing(0.314)  |   R2adj: 0.108 AICc: 105.48
38 ++leap(0.018) ++swin(0.041) +group(0.198)  |   R2adj: 0.125 AICc: 111.56
39 +monog(0.131) leap(0.876)  |  arbore(0.817) R2adj: -0.004 AICc: 106.25
40 +swing(0.16) +monog(0.278)  |  arbore(0.674) R2adj: 0.056 AICc: 104.03
41 ++swin(0.03) ++leap(0.05)  |  arbore(0.405) R2adj: 0.101 AICc: 108.81
42 +swing(0.219) leap(0.345) monoga(0.472)  |   R2adj: 0.057 AICc: 107.54
43   |  mass(0.956) group(0.765) R2adj: -0.056 AICc: 113.49
44 arbore(0.782)  |  mass(0.994) R2adj: -0.058 AICc: 109.63
45 ++mono(0.044) mass(0.56)  |   R2adj: 0.065 AICc: 101.86
46 +leap(0.088) mass(0.712)  |   R2adj: 0.029 AICc: 110.37
47 +swing(0.229) mass(0.931)  |   R2adj: -0.014 AICc: 111.99
48 arbore(0.56) group(0.642)  |   R2adj: -0.045 AICc: 112.96
49 ++mono(0.022) +group(0.111)  |   R2adj: 0.107 AICc: 104.11
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 +monog(0.104)  |  arbore(0.843) R2adj: 0.025 AICc: 103.74
53 +leap(0.235) arbore(0.986)  |   R2adj: -0.008 AICc: 111.64
54 +swing(0.13) arbore(0.929)  |   R2adj: 0.018 AICc: 110.66
55 +monog(0.126) leap(0.683)  |   R2adj: 0.041 AICc: 106.73
56 +monog(0.165) swing(0.37)  |   R2adj: 0.059 AICc: 106.03
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')
```

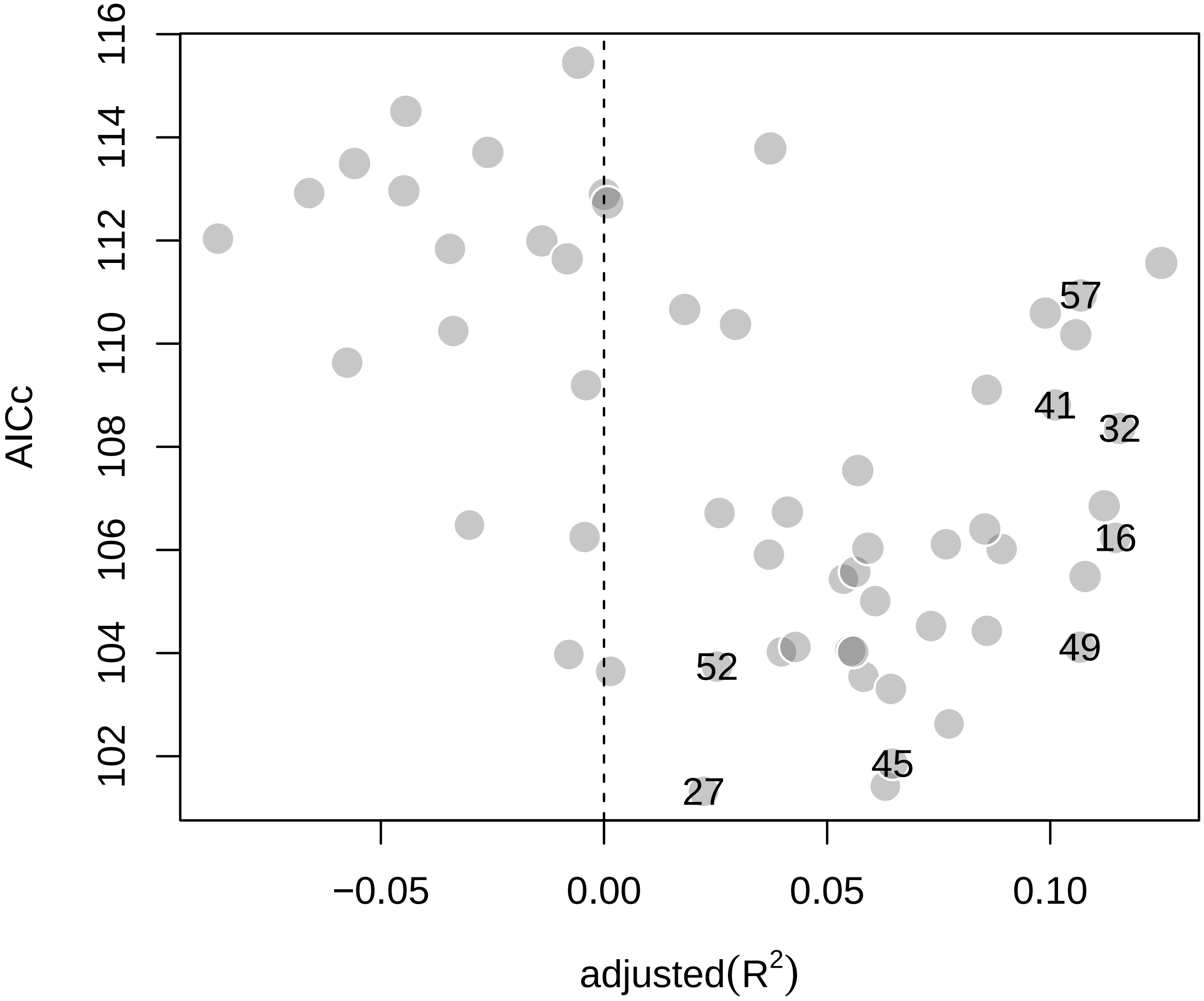

Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is not a bad idea that all "best" models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

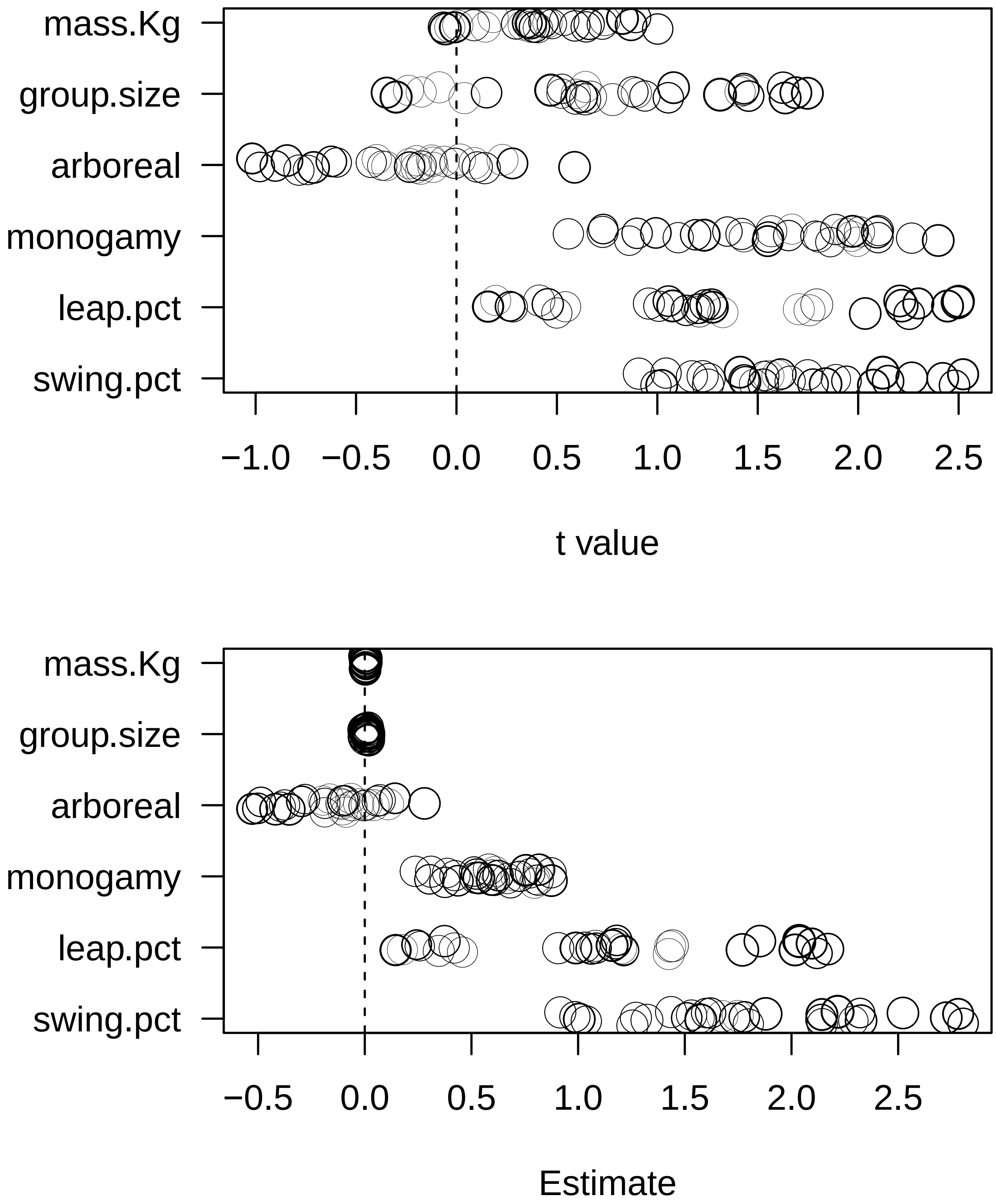

Figure 3: These horizontal parameter distribution dot-plots demonstrate how the (t-values of) coefficients from models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting in each 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.