# 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

September 8, 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



## 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] 0.427

> 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.   :64348   Min.   : 1   Min.   :0.005698
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:66131   1st Qu.:15   1st Qu.:0.074439
 Median :36.00   Median :3.000   Median :67298   Median :29   Median :0.172968
 Mean   :36.37   Mean   :3.263   Mean   :67438   Mean   :29   Mean   :0.158364
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69081   3rd Qu.:43   3rd Qu.:0.223169
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.308784
     R2.adj                AIC             AICc            BIC             AICw
 Min.   :-0.0787470   Min.   :104.3   Min.   :105.3   Min.   :109.4   Min.   :0.0003333
 1st Qu.: 0.0001832   1st Qu.:106.0   1st Qu.:108.0   1st Qu.:113.4   1st Qu.:0.0016804
 Median : 0.0947358   Median :107.3   Median :109.6   Median :115.3   Median :0.0111228
 Mean   : 0.0732322   Mean   :108.9   Mean   :110.5   Mean   :115.7   Mean   :0.0175439
 3rd Qu.: 0.1300430   3rd Qu.:112.1   3rd Qu.:113.3   3rd Qu.:118.1   3rd Qu.:0.0246387
 Max.   : 0.1935813   Max.   :114.8   Max.   :116.6   Max.   :122.5   Max.   :0.0934201
      BICw
 Min.   :0.0002069
 1st Qu.:0.0018936
 Median :0.0076668
 Mean   :0.0175439
 3rd Qu.:0.0194151
 Max.   :0.1434750
```

# 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.   :64348   Min.   : 1   Min.   :0.005698
 1st Qu.:36.00   1st Qu.:2.000   1st Qu.:66131   1st Qu.:15   1st Qu.:0.074439
 Median :36.00   Median :3.000   Median :67298   Median :29   Median :0.172968
 Mean   :36.37   Mean   :3.263   Mean   :67438   Mean   :29   Mean   :0.158364
 3rd Qu.:37.00   3rd Qu.:4.000   3rd Qu.:69081   3rd Qu.:43   3rd Qu.:0.223169
 Max.   :38.00   Max.   :6.000   Max.   :71122   Max.   :57   Max.   :0.308784
     R2.adj                AIC             AICc            BIC             AICw
 Min.   :-0.0787470   Min.   :104.3   Min.   :105.3   Min.   :109.4   Min.   :0.0003333
 1st Qu.: 0.0001832   1st Qu.:106.0   1st Qu.:108.0   1st Qu.:113.4   1st Qu.:0.0016804
 Median : 0.0947358   Median :107.3   Median :109.6   Median :115.3   Median :0.0111228
 Mean   : 0.0732322   Mean   :108.9   Mean   :110.5   Mean   :115.7   Mean   :0.0175439
 3rd Qu.: 0.1300430   3rd Qu.:112.1   3rd Qu.:113.3   3rd Qu.:118.1   3rd Qu.:0.0246387
 Max.   : 0.1935813   Max.   :114.8   Max.   :116.6   Max.   :122.5   Max.   :0.0934201
      BICw
 Min.   :0.0002069
 1st Qu.:0.0018936
 Median :0.0076668
 Mean   :0.0175439
 3rd Qu.:0.0194151
 Max.   :0.1434750
```

# 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
 0.00452500  0.01530875 -0.15083750  0.89603500  1.53045000  1.51763875

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

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
64348 0.00488    0.01974 -0.10124  0.91919  1.37623   1.24988
66131 0.00320    0.01191 -0.08939      NaN  2.06365   2.18803
66389     NaN    0.01814 -0.18631  0.86014  1.25483   1.14882
67298 0.00540    0.01931      NaN  0.92570  1.13748   0.97145
68172     NaN    0.01062 -0.22641      NaN  1.91342   2.05180
69081 0.00462    0.01245      NaN      NaN  1.84486   1.93986
69339     NaN    0.01834      NaN  0.87911  0.94232   0.78688
71122     NaN    0.01196      NaN      NaN  1.71081   1.80439
attr(,"MSE")
      mass.Kg group.size arboreal monogamy leap.pct swing.pct
64348 0.00032    0.00014  0.03567  0.03837  0.04545   0.04768
66131 0.00003    0.00002  0.00475       NA  0.00698   0.00737
66389      NA    0.00014  0.04604  0.04859  0.05642   0.06140
67298 0.00058    0.00024       NA  0.07931  0.05788   0.06493
68172      NA    0.00003  0.00525       NA  0.00839   0.00908
69081 0.00006    0.00004       NA       NA  0.00696   0.00744
69339      NA    0.00017       NA  0.07081  0.04758   0.05804
71122      NA    0.00003       NA       NA  0.00375   0.00482

> 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
  0.989205   1.000000   0.277585   1.000000   1.000000   1.000000

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

      mass.Kg group.size arboreal monogamy leap.pct swing.pct
64348 1.00000          1  0.30128        1        1         1
66131 1.00000          1  0.53928      NaN        1         1
66389     NaN          1  0.00000        1        1         1
67298 1.00000          1      NaN        1        1         1
68172     NaN          1  0.26978      NaN        1         1
69081 0.95682          1      NaN      NaN        1         1
69339     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
64348 0.00000          0  0.09846        0        0         0
66131 0.00000          0  0.00844       NA        0         0
66389      NA          0  0.00000        0        0         0
67298 0.00000          0       NA        0        0         0
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 68172 | NA | 0 | 0.00920 | NA | 0 | 0 |
| 69081 | 0.00645 | 0 | NA | NA | 0 | 0 |
| 69339 | 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
37 37 2 2X37 69339       49 0.2266769 0.18118732 104.5763 105.3036 109.4091 0.093420093
36 36 3 3X36 67298       24 0.2371923 0.16567903 104.3213 105.6116 110.6554 0.080086757
35 35 3 3X35 64348       27 0.1554944 0.07376802 105.7227 107.0560 111.9440 0.038896950
38 38 2 2X38 71122       57 0.1551053 0.10682562 110.2290 110.9348 115.1417 0.005592859
          BICw
37 0.143474999
36 0.076940222
35 0.040394000
38 0.008164924
```

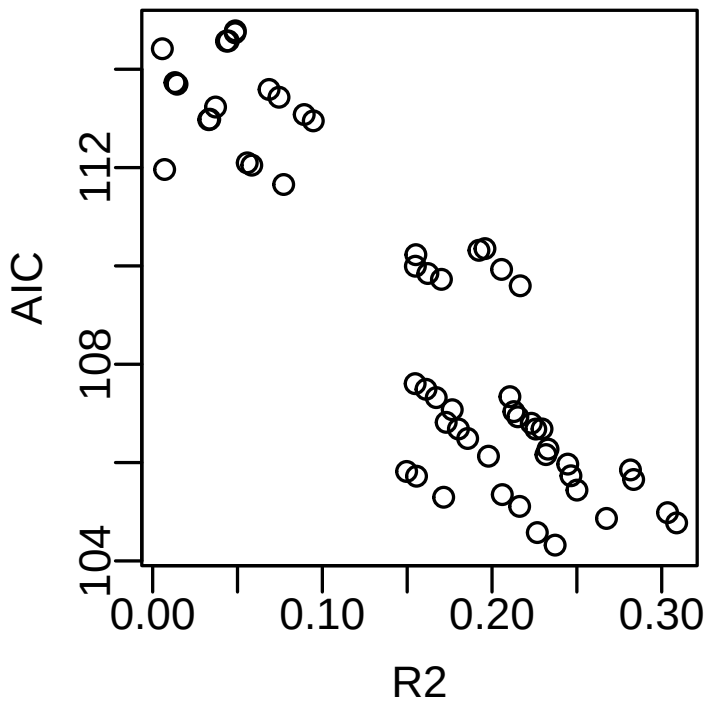
Plotting the coefficients of determination versus the AIC values allows selection of high-performing models for inspection and reporting.

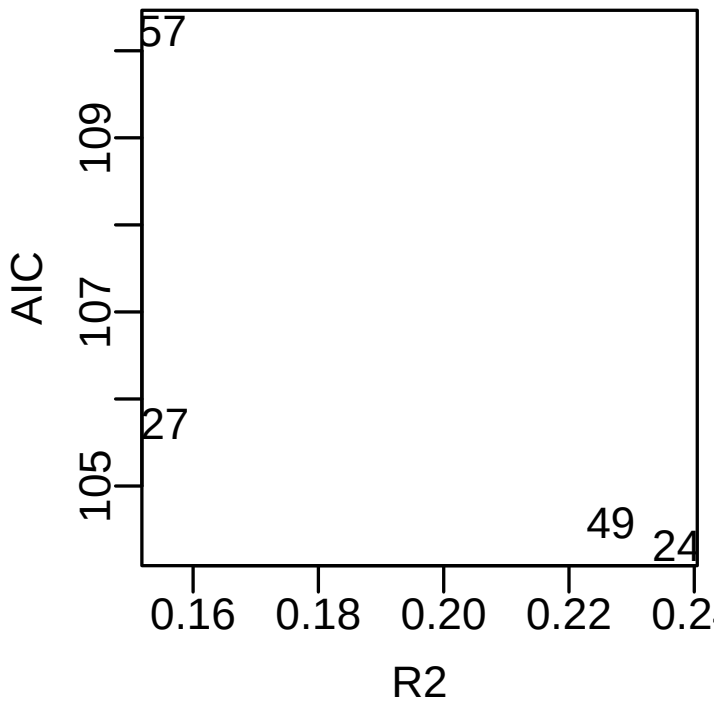
```
> plot.pgls.iters(PGLSi)
```

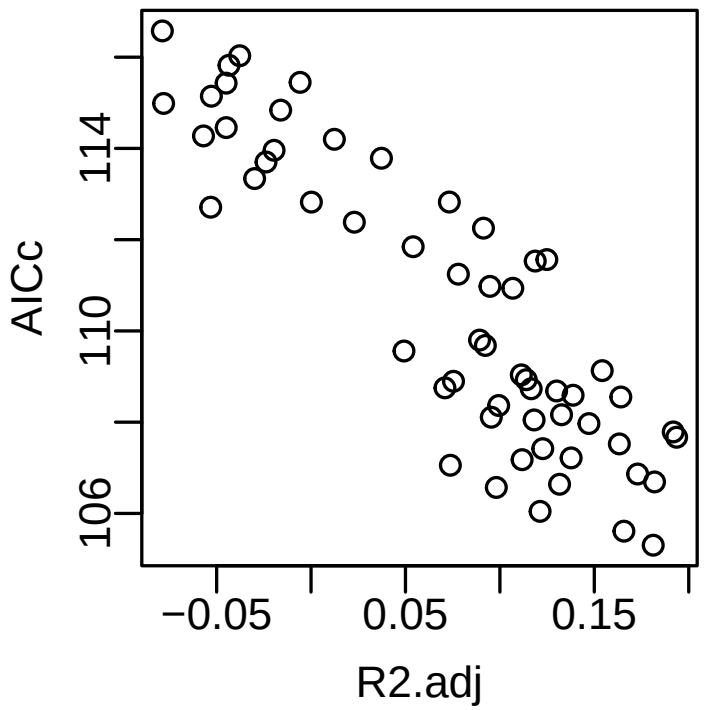
```
> sdevs.objs <- get.pgls.coefs(PGLSi$fits, est='t value')
> coefs.objs <- get.pgls.coefs(PGLSi$fits, est='Estimate')

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

```
1 +leap(0.083) +group(0.088) +monog(0.095) +swing(0.142) mass(0.378)  |  arbore(0.533) R2adj: 0.154 AICc: 109.
2 ++mono(0.02) +group(0.121) +leap(0.29) mass(0.46)  |  arbore(0.928) R2adj: 0.117 AICc: 108.73
3 ++mono(0.015) +group(0.139) mass(0.541) swing(0.615) arbore(0.969)  |  R2adj: 0.089 AICc: 109.8
4 ++leap(0.02) ++swin(0.034) +group(0.21) mass(0.49)  |  arbore(0.556) R2adj: 0.073 AICc: 112.83
5 +group(0.073) +monog(0.087) +leap(0.106) +swing(0.191) +mass(0.293)  |  R2adj: 0.194 AICc: 107.67
6 +leap(0.13) +swing(0.202) +monog(0.231) mass(0.565)  |  arbore(0.407) R2adj: 0.092 AICc: 109.68
7 +leap(0.096) +group(0.101) +monog(0.101) +swing(0.152)  |  arbore(0.348) R2adj: 0.164 AICc: 108.55
8 ++mono(0.009) +group(0.138) mass(0.536) arbore(0.894)  |  R2adj: 0.112 AICc: 107.18
9 +leap(0.166) group(0.469) mass(0.711) arbore(0.811)  |  R2adj: -0.045 AICc: 115.43
10 swing(0.33) group(0.556) arbore(0.717) mass(0.859)  |  R2adj: -0.079 AICc: 116.58
11 ++mono(0.015) +group(0.096) +leap(0.266) mass(0.404)  |  R2adj: 0.173 AICc: 106.86
12 ++mono(0.008) +group(0.105) mass(0.494) swing(0.585)  |  R2adj: 0.147 AICc: 107.97
13 ++leap(0.013) ++swin(0.033) +group(0.178) mass(0.383)  |  R2adj: 0.119 AICc: 111.53
14 +monog(0.058) leap(0.349) mass(0.634)  |  arbore(0.712) R2adj: 0.071 AICc: 108.75
15 ++mono(0.04) swing(0.658) mass(0.71)  |  arbore(0.809) R2adj: 0.049 AICc: 109.56
16 ++leap(0.037) +swing(0.059) mass(0.615)  |  arbore(0.455) R2adj: 0.054 AICc: 111.85
17 +leap(0.155) +monog(0.222) +swing(0.268) mass(0.392)  |  R2adj: 0.13 AICc: 108.68
18 ++mono(0.02) +group(0.134) +leap(0.316)  |  arbore(0.7) R2adj: 0.133 AICc: 108.16
19 ++mono(0.014) +group(0.14) swing(0.607)  |  arbore(0.845) R2adj: 0.111 AICc: 109.04
20 ++leap(0.021) ++swin(0.034) +group(0.232)  |  arbore(0.39) R2adj: 0.091 AICc: 112.25
21 +group(0.085) +monog(0.094) +leap(0.151) +swing(0.245)  |  R2adj: 0.192 AICc: 107.79
22 +leap(0.132) +swing(0.206) +monog(0.236)  |  -arbor(0.31) R2adj: 0.114 AICc: 108.93
23 arbore(0.559) group(0.63) mass(0.883)  |  R2adj: -0.078 AICc: 114.99
24 +++mon(0.004) +group(0.107) mass(0.511)  |  R2adj: 0.166 AICc: 105.61
25 +leap(0.081) group(0.431) mass(0.717)  |  R2adj: 0.012 AICc: 114.2
26 +swing(0.231) group(0.529) mass(0.937)  |  R2adj: -0.038 AICc: 116.03
27 ++mono(0.027) mass(0.704)  |  arbore(0.87) R2adj: 0.074 AICc: 107.06
28 +leap(0.196) mass(0.774) arbore(0.946)  |  R2adj: -0.03 AICc: 113.34
29 swing(0.355) arbore(0.827) mass(0.903)  |  R2adj: -0.057 AICc: 114.27
30 ++mono(0.051) +leap(0.313) mass(0.494)  |  R2adj: 0.123 AICc: 107.42
31 ++mono(0.026) mass(0.585) swing(0.661)  |  R2adj: 0.099 AICc: 108.37
32 ++leap(0.028) +swing(0.063) mass(0.444)  |  R2adj: 0.095 AICc: 110.98
33 ++mono(0.009) +group(0.141)  |  arbore(0.917) R2adj: 0.132 AICc: 106.64
34 +leap(0.17) group(0.504) arbore(0.935)  |  R2adj: -0.016 AICc: 114.84
35 +swing(0.313) group(0.552) arbore(0.745)  |  R2adj: -0.043 AICc: 115.82
```



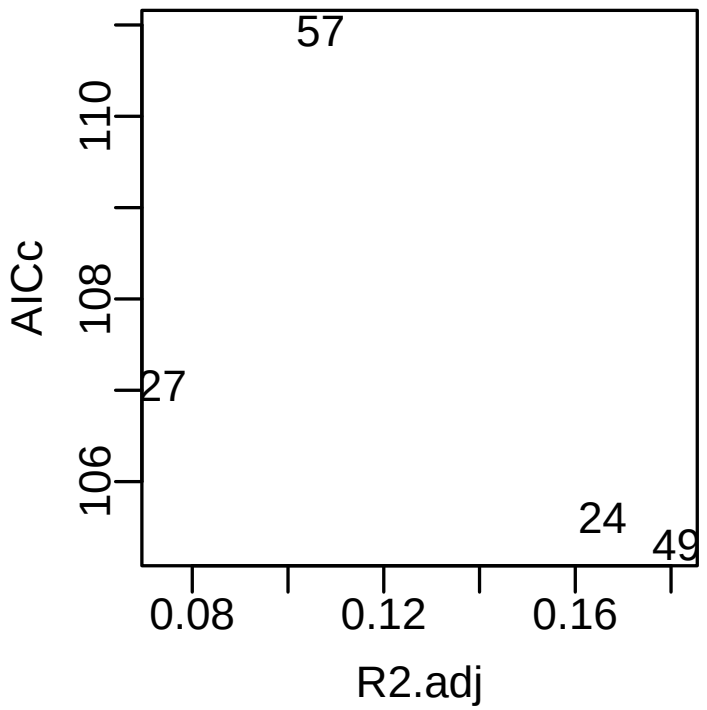




Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models typically appear in the lower right of 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(0.102) +leap(0.318)  |   R2adj: 0.182 AICc: 106.69
37 ++mono(0.008) +group(0.102) swing(0.606)  |   R2adj: 0.163 AICc: 107.52
38 ++leap(0.018) ++swin(0.041) +group(0.198)  |   R2adj: 0.125 AICc: 111.56
39 +monog(0.056) leap(0.349)  |  arbore(0.59) R2adj: 0.095 AICc: 108.11
40 ++mono(0.039) swing(0.665)  |  arbore(0.723) R2adj: 0.075 AICc: 108.9
41 ++leap(0.036) +swing(0.058)  |  arbore(0.354) R2adj: 0.078 AICc: 111.24
42 +leap(0.187) +monog(0.231) +swing(0.315)  |   R2adj: 0.139 AICc: 108.59
43 group(0.664)  |  mass(0.943) R2adj: -0.053 AICc: 115.14
44 arbore(0.639) mass(0.919)  |   R2adj: -0.053 AICc: 112.71
45 ++mono(0.013) mass(0.598)  |   R2adj: 0.121 AICc: 106.05
46 +leap(0.101) mass(0.725)  |   R2adj: 0.023 AICc: 112.38
47 +swing(0.261) mass(0.931)  |   R2adj: -0.02 AICc: 113.96
48 arbore(0.564) group(0.641)  |   R2adj: -0.045 AICc: 114.46
49 +++mon(0.004) +group(0.104)  |   R2adj: 0.181 AICc: 105.3
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(0.025)  |  arbore(0.778) R2adj: 0.098 AICc: 106.57
53 +leap(0.193)  |  arbore(0.966) R2adj: 0 AICc: 112.82
54 swing(0.343) arbore(0.836)  |   R2adj: -0.024 AICc: 113.7
55 +monog(0.054) leap(0.338)  |   R2adj: 0.138 AICc: 107.22
56 ++mono(0.026) swing(0.686)  |   R2adj: 0.118 AICc: 108.05
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')
```

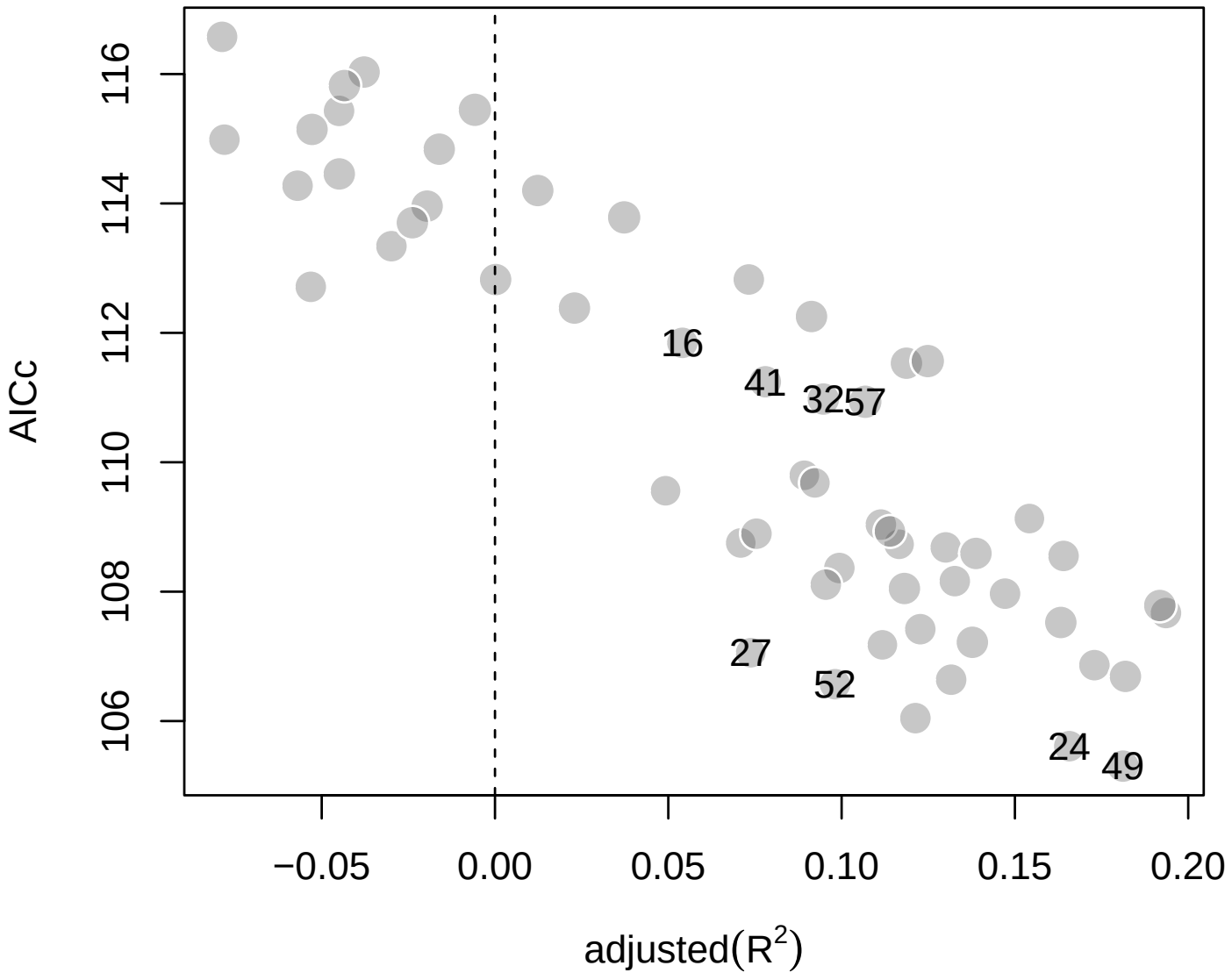



Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is not a bad idea that all "best" models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

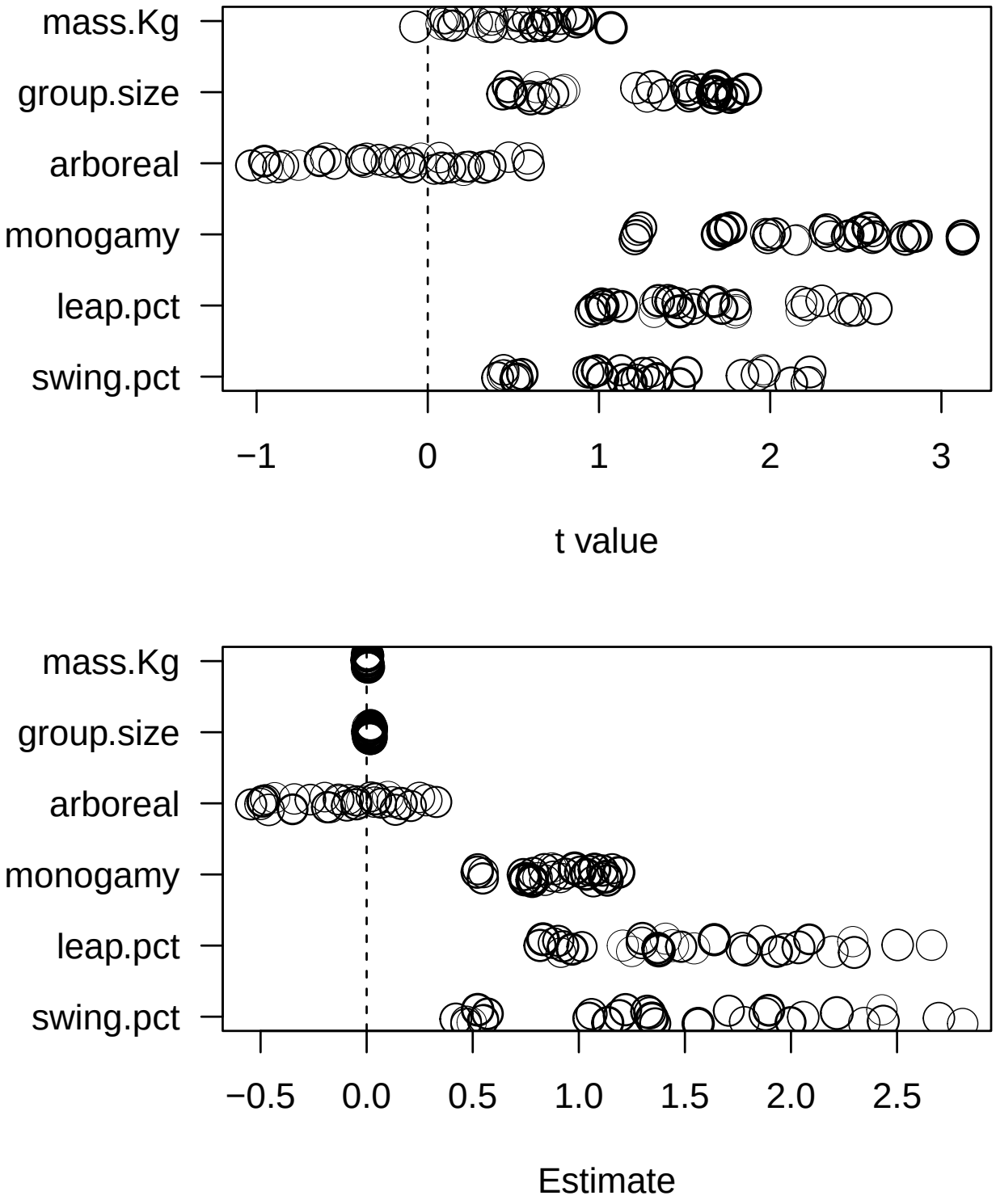



Figure 3: These horizontal parameter distribution dot-plots demonstrate how the (t-values of) coefficients from models can be simultaneously plot in order to verify consistency of estimates across the various (often missing-data driven) sub-datasets. To visually assess potential over-fitting in each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