Ecological factors influencing primate vocal communication: a phylogenetic regression workflow for the *mmodely* R-package (Version 0.2.0)

David M. Schruth
dschruth@anthropoidea.org

November 3, 2022

1 Introduction

Perform multivariate modeling of evolved traits, with special attention to understanding the interplay of the multifactorial determinants of their origins in complex ecological settings (Stephens, 2007 <doi:10.1016/j.tree.2006.12.003>). This software primarily concentrates on phylogenetic regression analysis, enabling implementation of tree transformation averaging and visualization functionality. Functions additionally support information theoretic approaches (Grueber, 2011 <doi:10.1111/j.1420-9101.2010.02210.x>; Garamszegi, 2011 <doi:10.1007/s00265-010-1028-7>) such as model averaging and selection of phylogenetic models. There are other numerous functions for visualizing confounded variables, plotting phylogenetic trees, as well as reporting and exporting modeling results. Lastly, as challenges to ecology are inherently multifarious, and therefore often multi-dataset, this package features several functions to support the identification, interpolation, merging, and updating of missing data and outdated nomenclature.

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 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)
> # 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 appear 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

```
> 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 3 and 16 DF,  p-value: 0.4226

> 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 3 and 16 DF,  p-value: 0.4226
[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 in 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] 0.413

> 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 to obtain fixed values going forward. This approach can speed up computations for larger sets of modeling data and variable combinations. But in order to show off the model selection functionality we will only artificially sprinkle in some missing values to this small dataset for illustration purposes.

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

        l         k         d
0.2903945 0.2000000 1.3289849

> 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 (to keep things interesting for model selection)
> missing.value.ct <- 1
> for(pv in pvs){ 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

8 Model Averaging

Now we can estimate the predictor variable parameters by averaging over all possible fixed PGLS runs.

```
> calculate.weighted.means(vars=pvs, fits=PGLSi$fits, optims=PGLSi$optim)

   mass.Kg group.size   arboreal   monogamy   leap.pct
   0.00638    0.01279   -0.49280    0.84900    1.19513
 swing.pct
   1.11456
```

9 Model Selection

Plotting the coefficients of determination versus the AIC values allows selection of certain models for 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(0.067) +monog(0.085) +swing(0.2) group(0.44) mass(0.463)  |  arbore(0.426) R2adj: 0.183 AICc: 109.34
2 ++mono(0.018) +leap(0.177) group(0.454) mass(0.505)  |  arbore(0.649) R2adj: 0.164 AICc: 108.32
3 +++mon(0.005) group(0.388) mass(0.568) swing(0.798)  |  arbore(0.912) R2adj: 0.118 AICc: 112.88
4 +++lea(0.005) ++swin(0.041) +group(0.24) mass(0.475)  |  arbore(0.563) R2adj: 0.125 AICc: 113.17
5 +monog(0.07) +leap(0.089) +group(0.247) +mass(0.261) +swing(0.264)  |   R2adj: 0.193 AICc: 107.04
6 +leap(0.065) +monog(0.113) +swing(0.2) mass(0.557)  |  -arbor(0.24) R2adj: 0.194 AICc: 106.99
7 +leap(0.075) +monog(0.095) +swing(0.208) group(0.525)  |  -arbor(0.243) R2adj: 0.195 AICc: 106.92
8 +++mon(0.004) group(0.383) mass(0.559)  |  arbore(0.942) R2adj: 0.144 AICc: 110.09
9 ++leap(0.037) group(0.447) mass(0.63) arbore(0.997)  |   R2adj: 0.028 AICc: 115.38
10 swing(0.48) group(0.527) arbore(0.669) mass(0.849)  |   R2adj: -0.088 AICc: 122.55
11 ++mono(0.017) +leap(0.187) +group(0.308) mass(0.344)  |   R2adj: 0.185 AICc: 105.67
12 +++mon(0.005) +group(0.324) mass(0.488) swing(0.806)  |   R2adj: 0.146 AICc: 110.03
13 +++lea(0.005) ++swin(0.046) +group(0.182) mass(0.326)  |   R2adj: 0.143 AICc: 110.71
14 ++mono(0.023) +leap(0.17) mass(0.601)  |  arbore(0.412) R2adj: 0.175 AICc: 106.1
15 ++mono(0.007) mass(0.688) swing(0.809)  |  arbore(0.642) R2adj: 0.125 AICc: 110.92
16 ++leap(0.008) +swing(0.062) mass(0.572)  |  arbore(0.402) R2adj: 0.114 AICc: 111.98
17 +leap(0.113) +monog(0.119) +mass(0.304) swing(0.33)  |   R2adj: 0.182 AICc: 105.78
18 ++mono(0.02) +leap(0.192) group(0.531)  |  arbore(0.421) R2adj: 0.178 AICc: 105.96
19 ++mono(0.005) group(0.441) swing(0.787)  |  arbore(0.692) R2adj: 0.137 AICc: 110.41
20 ++leap(0.006) ++swin(0.045) +group(0.269)  |  arbore(0.375) R2adj: 0.139 AICc: 110.92
21 +monog(0.083) +leap(0.139) +group(0.287) swing(0.348)  |   R2adj: 0.185 AICc: 105.69
22 +leap(0.069) +monog(0.114) +swing(0.204)  |  -arbor(0.145) R2adj: 0.211 AICc: 104.52
23 arbore(0.571) group(0.573) mass(0.856)  |   R2adj: -0.072 AICc: 120.47
24 +++mon(0.003) group(0.325) mass(0.489)  |   R2adj: 0.17 AICc: 107.42
25 ++leap(0.032) group(0.423) mass(0.596)  |   R2adj: 0.058 AICc: 112.69
26 swing(0.421) group(0.584) mass(0.992)  |   R2adj: -0.062 AICc: 120.1
27 +++mon(0.005) mass(0.68)  |  arbore(0.663) R2adj: 0.15 AICc: 108.3
28 ++leap(0.043) mass(0.688)  |  arbore(0.828) R2adj: 0.04 AICc: 113.37
29 swing(0.517) arbore(0.784) mass(0.893)  |   R2adj: -0.069 AICc: 120.36
30 ++mono(0.028) +leap(0.202) mass(0.376)  |   R2adj: 0.183 AICc: 104.19
31 ++mono(0.007) mass(0.525) swing(0.87)  |   R2adj: 0.146 AICc: 108.49
32 ++leap(0.01) +swing(0.088) mass(0.371)  |   R2adj: 0.121 AICc: 110.12
33 +++mon(0.004) group(0.437)  |  arbore(0.718) R2adj: 0.161 AICc: 107.81
34 ++leap(0.039) group(0.472)  |  arbore(0.838) R2adj: 0.051 AICc: 112.97
35 swing(0.475) group(0.53) arbore(0.698)  |   R2adj: -0.057 AICc: 119.93
36 ++mono(0.022) +leap(0.239) group(0.334)  |   R2adj: 0.187 AICc: 104.02
37 ++mono(0.005) group(0.339) swing(0.833)  |   R2adj: 0.159 AICc: 107.91
38 ++leap(0.007) +swing(0.066) +group(0.2)  |   R2adj: 0.144 AICc: 109.16
39 ++mono(0.023) +leap(0.179)  |  -arbor(0.274) R2adj: 0.194 AICc: 103.72
```

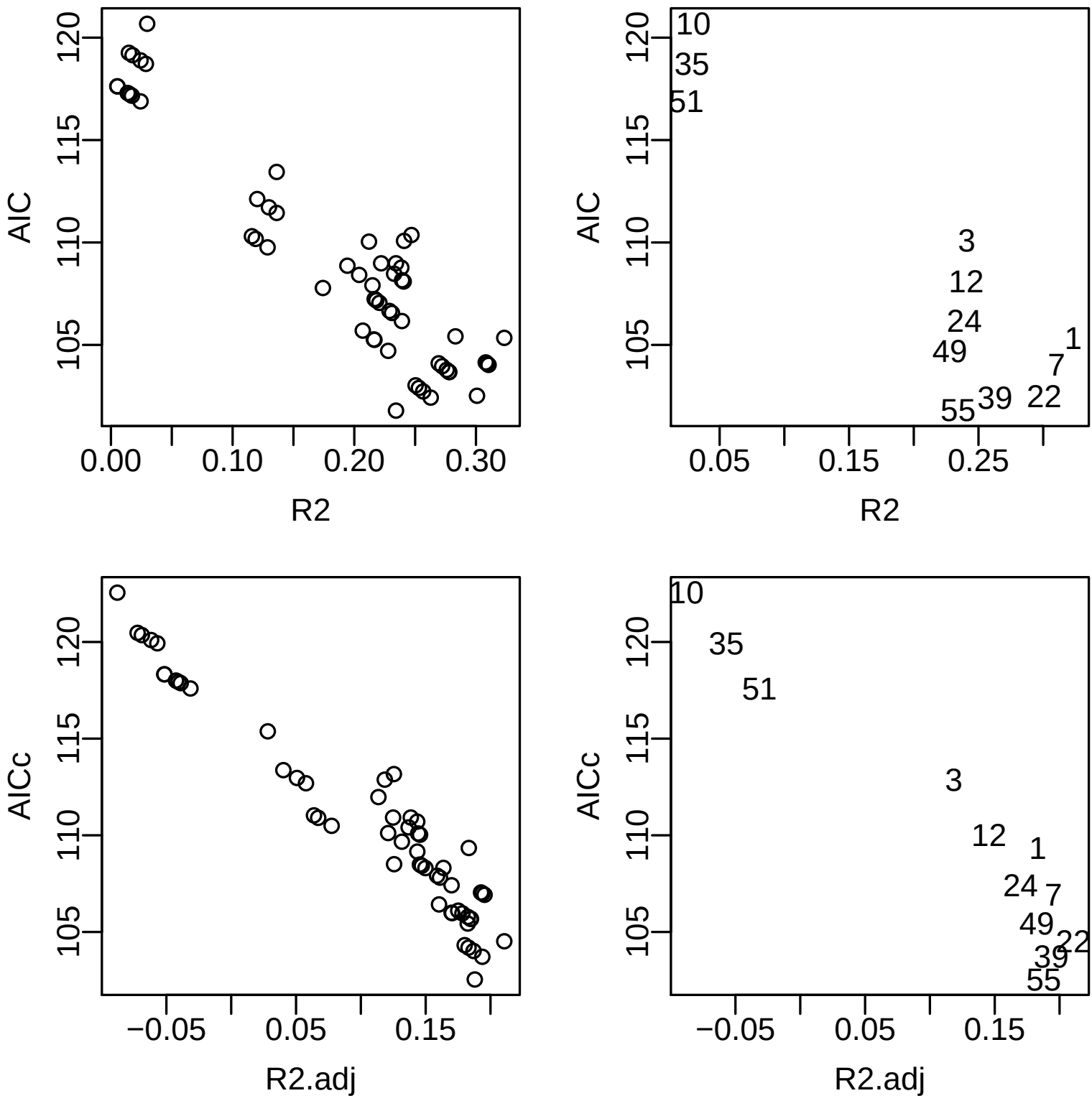

Figure 1: All possible model combinations appear as individual points above. As there is a generally negative association between AIC and the coefficient of determination, the points tend to follow a negative sloping streak to the lower right. The "best" models appear in the lower right of each streak. 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.

```
40 ++mono(0.007) swing(0.799)  |  arbore(0.497) R2adj: 0.147 AICc: 108.42
41 ++leap(0.008) +swing(0.065)  |  -arbor(0.274) R2adj: 0.132 AICc: 109.67
42 +monog(0.131) +leap(0.164) swing(0.413)  |   R2adj: 0.18 AICc: 104.32
43 group(0.673)  |  mass(0.943) R2adj: -0.052 AICc: 118.33
44 arbore(0.669) mass(0.894)  |   R2adj: -0.051 AICc: 118.33
45 +++mon(0.004) mass(0.524)  |   R2adj: 0.17 AICc: 106
46 ++leap(0.039) mass(0.596)  |   R2adj: 0.067 AICc: 110.9
47 swing(0.463) mass(0.981)  |   R2adj: -0.041 AICc: 117.94
48 group(0.577) arbore(0.583)  |   R2adj: -0.043 AICc: 118
49 +++mon(0.003) group(0.339)  |   R2adj: 0.182 AICc: 105.44
50 ++leap(0.035) group(0.418)  |   R2adj: 0.077 AICc: 110.49
51 swing(0.412) group(0.578)  |   R2adj: -0.031 AICc: 117.6
52 +++mon(0.004)  |  arbore(0.514) R2adj: 0.17 AICc: 105.97
53 ++leap(0.043)  |  arbore(0.686) R2adj: 0.064 AICc: 111.03
54 swing(0.511) arbore(0.808)  |   R2adj: -0.039 AICc: 117.87
55 ++mono(0.035) +leap(0.252)  |   R2adj: 0.188 AICc: 102.54
56 ++mono(0.008) swing(0.893)  |   R2adj: 0.16 AICc: 106.43
57 ++leap(0.015) +swing(0.117)  |   R2adj: 0.126 AICc: 108.51
```

```
> plot.pgls.R2AIC(PGLSi$optim)
```

10 Coefficient Plotting

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

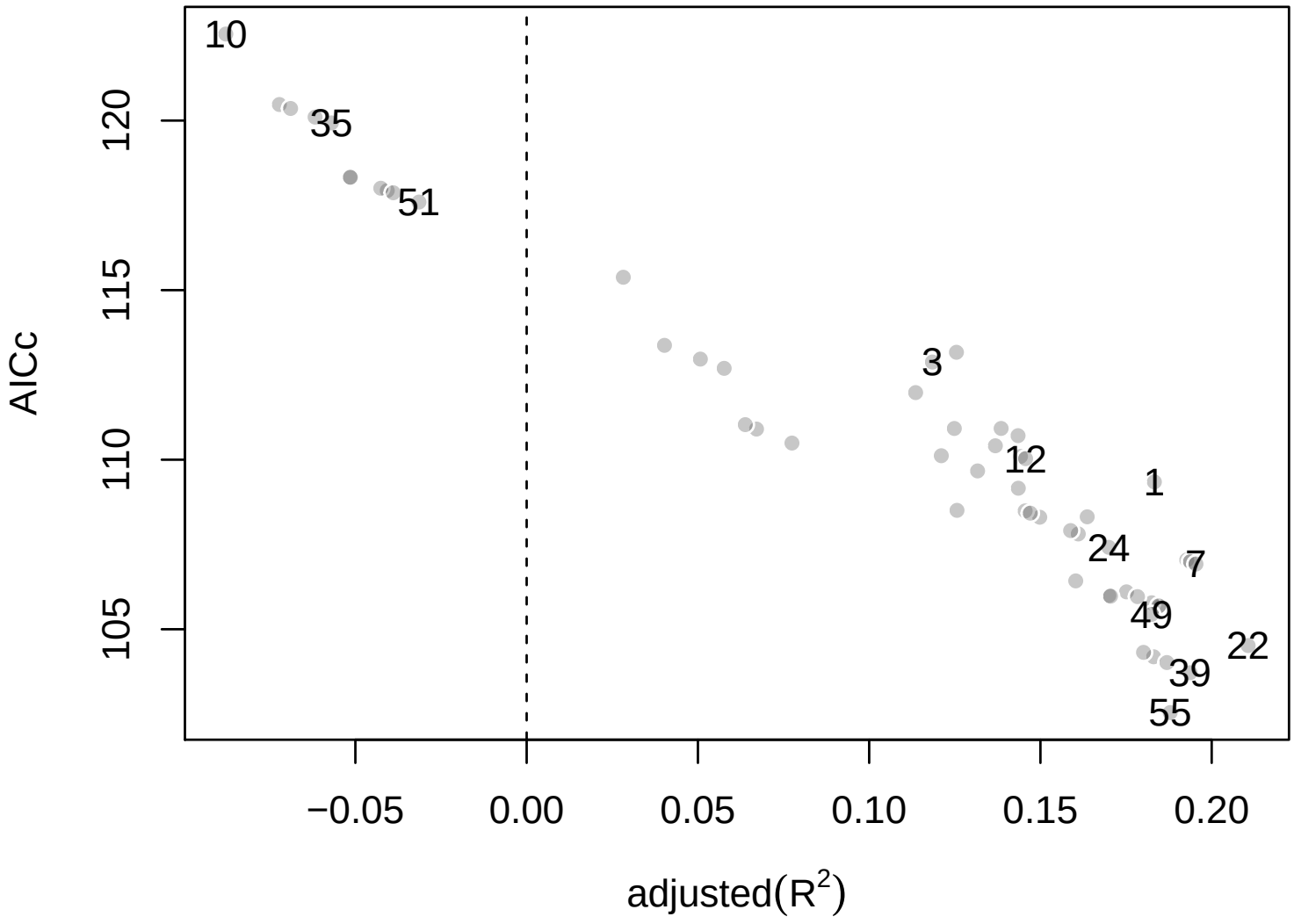

Figure 2: This is a one panel verison of the previous model selection plot. The numbered points in the lower right corner of each streak of possible models represent the best model within a sub-dataset. Since these AICc values should not strictly be compared, it is recommended that all "best" models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

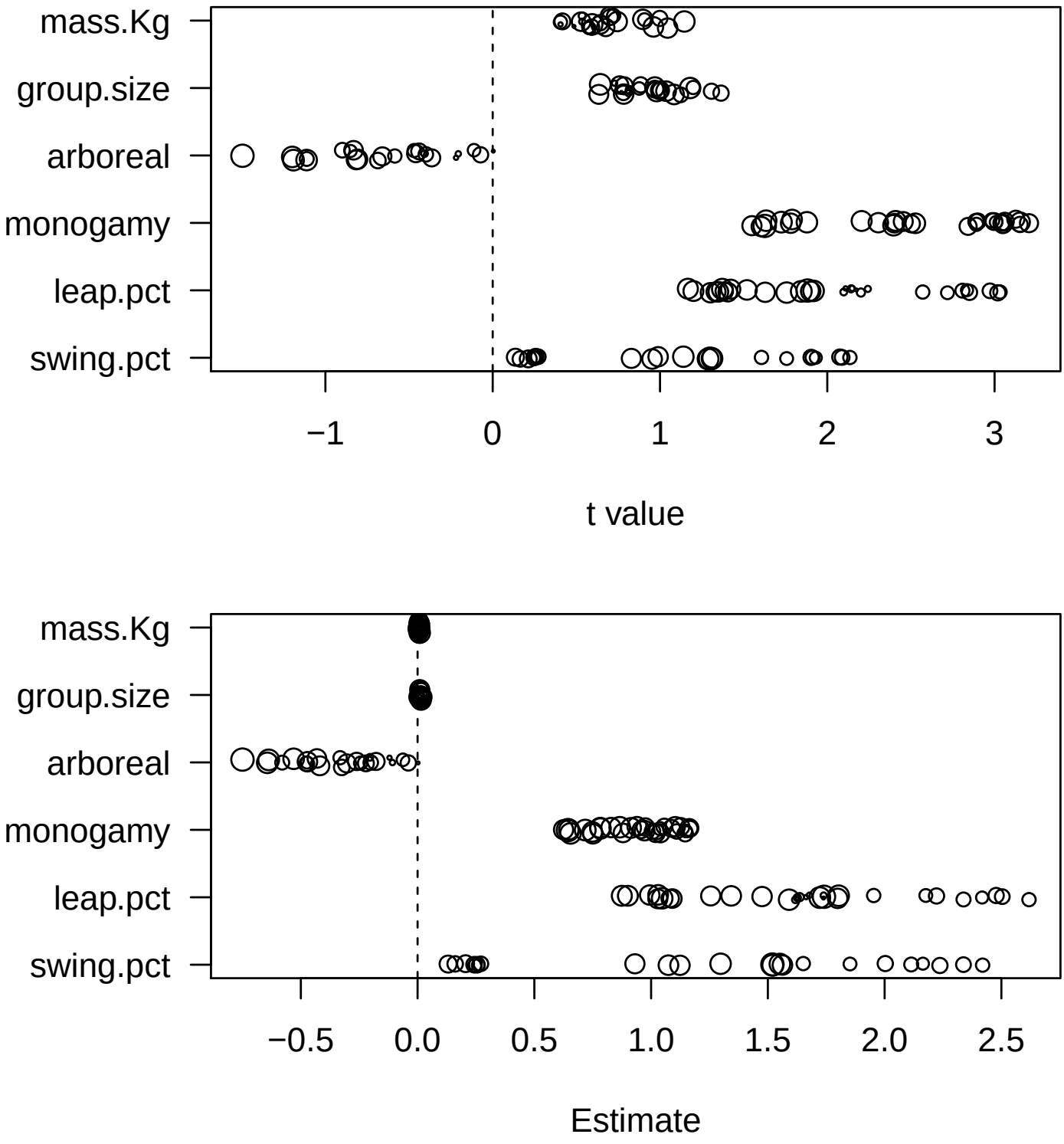

Figure 3: These 'distro dots' plots demonstrate how the (t-values of) coefficients from all "best" selected models can be simultaneously plot in order to verify consistency of estimates across the various (missing data driven) sub-datasets.

References

[1] Revell, L.J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). [CRAN]. Available at: http://cran.r-project.org/package=phytools.

[2] 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.

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

[4] Grueber, C.E., Nakagawa, .R.J., and Jamieson, I.G. (2011) 'Multimodel inference in ecology and evolution: challenges and solutions', Journal of Evolutoinary 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] 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.

[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] 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.

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

[10] Schruth, D.M. (2019a) Primate Locomotor Activity [The Center for Open Science]. Available at: https://osf.io/cd68q/.

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

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

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

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