Model selection and multimodel inference

Landscape Genetics DGS 2016

Learning goal: The goal of this lab is for you to become familiar with creating and analyzing landscape genetic hypotheses using maximum likelihood population effects (MLPE; Van Strien et al. 2012) and information theory.

Section 1. In this section, you will review the concepts in the lecture and background reading to interpret AICc and BIC values from a set of landscape genetic models. In **Section 2** you will learn how to create this output from landscape genetic data.

Here are the results of a set of models run with MLPE (more information on model variables can be found in Section 2):

1. The full model: solarinso, forest, ag, shrub, dev
2. The landcover model: ag, shrub, forest, and dev
3. The human footprint model: ag, dev
4. The energy conservation model: slope, shrub, dev
5. The historical model: soils, slope, solarinso

Model	AICc	AICc weight	BIC	BIC weight
1	121.6218	0.03	132.8991	0.01
2	116.5579	0.42	126.8563	0.19
3	116.2116	0.50	124.1391	0.76
4	120.8709	0.05	130.0494	0.04
5	133.2179	0.00	144.4952	0.00

Based on information theory, generally ΔAICc values:

- ≤2 are considered to show substantial evidence
- 4-7 are considered to show considerably less evidence
- >10 are considered to show essentially no evidence

1. Considering guidelines above and the principle of parsimony, what model is the 'best' of this set? How do you know?
2. How strong is the evidence for the best model, and how does it differ between AICc and BIC?
3. Why are evidence weights from BIC less equivocal between models than AICc evidence weights?
4. What variables would contribute to a multimodel prediction that would be excluded if only the best model was used?
5. Before starting this analysis, the researcher found that slope was collinear with land cover types therefore did not include them in the same models. What does this mean for inference from the best model?

Written by Caren Goldberg
March 30, 2016

Section 2. In this section, you will re-analyze the dataset from Goldberg and Waits 2010 using MLPE. Blue text indicates code to input into R, turquoise indicates text you may be changing when you get to Section 3.

In previous labs, you have created genetic distance matrices, so here we are going to start with those data already complete, as are the landscape data. For MLPE, two vectors representing the nodes at the ends of each link must be created, code for that can be found in the Bonus code under Lab 7. For this lab, we are going to start with a fully realized dataset and focus on creating and selecting models.

First, read in the dataset. I've named it CSF (Columbia spotted frog) rather than RALU (*Rana luteiventris*) to differentiate it from Melanie's dataset you have used in previous labs. You'll have to adjust the path to the directory where you put the .csv file:

```
CSFdata <- read.csv('C:/…/CSF_network.csv', header = TRUE)
```

It's always a good idea next to take a look at the field names and make sure you and R agree on what everything is called.

```
names(CSFdata)
```

The Y in these models will be logDc.km, the genetic distance per geographic distance, log transformed to meet normality assumptions. For this exercise, you'll test 5 alternative hypotheses. For **Section 3**, you'll create your own from these variables and test support for them. Definitions are as follows:

LENGTH – Euclidean distance of the link. This is incorporated into the Y, but you may want to use it as part of an interaction in Section 3

slope – Average slope along the link.

solarinso – average solar insolation along the link.

soils – dominant soil type along the link. Note that these are written out in text rather than represented as numbers, a trick to make sure that R always interprets this as a factor (categorical variable).

ag, grass, shrub, hi, lo, dev, forest – proportion of each land cover along the link: agriculture, grassland, shrubland, high density forest, low density forest, development (buildings), and forest (the sum of hi and lo).

For this section, your *a priori* set of hypotheses (as in Section 1) is as follows:

6. The full model: solarinso, forest, ag, shrub, dev
7. The landcover model: ag, shrub, forest, and dev
8. The human footprint model: ag, dev
9. The energy conservation model: slope, shrub, dev
10. The historical model: soils, slope, solarinso

Running the models

```
require(lme4)
# Create the Zl and ZZ matrices
Zl <- lapply(c("pop1","pop2"), function(nm) Matrix:::fac2sparse(CSFdata[[nm]],"d", drop=FALSE))
ZZ <- Reduce("+", Zl[-1], Zl[[1]])
# Fit a lmer model to the data
mod1 <- lFormula(logDc.km ~ solarinso + forest + ag + shrub + dev + (1|pop1), data = CSFdata, REML = TRUE)
```

At this point an error message appears:

```
Warning message:
Some predictor variables are on very different scales: consider rescaling
```

Which variable is causing this problem? To check, get R to show you the first 5 rows of the dataset:

```
CSFdata[1:5,]
```

One of these variables is a lot larger than the others, but has a small range. This often happens with variables such as easting and can cause issues for model fit. The common solution is to z-transform the data, which is conveniently done in R using the scale function:

```
solarinsoz <- scale(CSFdata$solarinso)
```

We then can attach these data back to our dataframe:

```
CSFdata <- cbind(CSFdata, solarinsoz)
```

Check to make sure this worked:

```
names(CSFdata)
```

If solarinsoz is there, you're good.

While we're at it, let's make sure that these variables don't have issues with multicollinearity:

You'll need to install packages usdm and sp if you don't have them already.

```
require(usdm)
```

Make a dataframe of just the variables to test (note, you can't use factors here):

```
CSF.df <- data.frame(CSFdata$solarinsoz, CSFdata$forest, CSFdata$dev,
CSFdata$shrub, CSFdata$ag)
vif(CSF.df)
```

```
            Variables      VIF
1 CSFdataz.solarinsoz 1.479273
2     CSFdataz.forest 3.127104
3        CSFdataz.dev 1.800218
4      CSFdataz.shrub 1.523341
5         CSFdataz.ag 2.532198
```

Numbers less than 10, or 3, or 4, depending on who you ask, are considered to not be collinear enough to affect model outcomes. So we're good to go here. Note, though, what would happen if we added grass to this:

```
            Variables       VIF
1 CSFdataz.solarinsoz  1.541954
2     CSFdataz.forest 97.167730
3        CSFdataz.dev 55.768172
4      CSFdataz.shrub 11.657666
5         CSFdataz.ag 68.852254
6      CSFdataz.grass 45.970721
```

Now we have to remake the modeling objects with our new and improved data frame, but referring to our z-transformed data (thanks to Martin Van Strien and Helene Wagner for the base model code):

```
Zl <- lapply(c("pop1","pop2"), function(nm) Matrix:::fac2sparse(CSFdata[[nm]],"d", drop=FALSE))
ZZ <- Reduce("+", Zl[-1], Zl[[1]])
# Fit a lmer model to the data
mod1 <- lFormula(logDc.km ~ solarinsoz + forest + ag + shrub + dev + (1|pop1), data = CSFdata, REML = TRUE)
dfun <- do.call(mkLmerDevfun,mod1)
opt <- optimizeLmer(dfun)
mod_1 <- mkMerMod(environment(dfun), opt, mod1$reTrms,fr = mod1$fr)
# In the fitted model replace Zt slot
mod1$reTrms$Zt <- ZZ
# Refit the model
dfun <- do.call(mkLmerDevfun,mod1)
opt <- optimizeLmer(dfun)
mod_1z <- mkMerMod(environment(dfun), opt, mod1$reTrms,fr = mod1$fr)
summary(mod_1z)
Linear mixed model fit by REML ['lmerMod']

REML criterion at convergence: 101.9

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.86045 -0.67339 -0.06932  0.57516  1.87959

Random effects:
 Groups   Name        Variance Std.Dev.
 pop1     (Intercept) 0.02986  0.1728
 Residual             0.49166  0.7012
Number of obs: 48, groups:  pop1, 20

Fixed effects:
             Estimate Std. Error t value
(Intercept) -4.681159   0.383071 -12.220
solarinsoz  -0.001611   0.135868  -0.012
forest      -0.163231   0.591137  -0.276
ag          -1.504031   0.642904  -2.339
shrub       -1.903160   1.343940  -1.416
dev          0.176494   0.652835   0.270
```

```
Correlation of Fixed Effects:
            (Intr) slrnsz forest ag     shrub
solarinsoz -0.088
forest     -0.802  0.360
ag         -0.848 -0.020  0.626
shrub      -0.237 -0.384 -0.121  0.259
dev        -0.691 -0.020  0.533  0.551  0.195
```

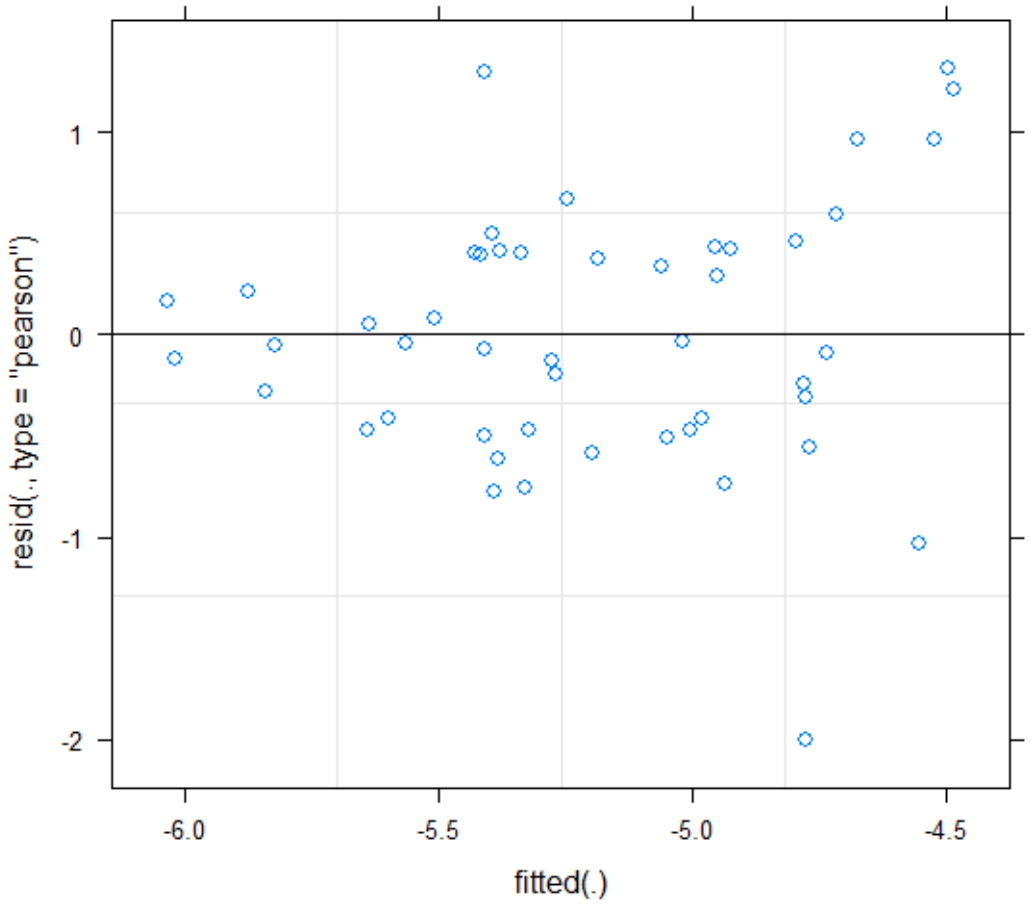
Let's get our AIC and BIC values:

```
AIC(mod_1z)
[1] 117.9295
```

```
BIC(mod_1z)
[1] 132.8991
```

This is the fullest model in our dataset (although our models are not completely nested), so let's take a look at the residuals:

```
plot(mod_1z)
```



Residuals are centered around zero and don't show large groupings or patterns, although there is some increase in variation at larger numbers (so the model is having a more difficult time predicting larger

genetic distances per km). There are many more checks that can be done at this point (referenced in the lecture), but for now we're going to leave model fit at this and fit the other models:

```
#Fitting model 2
mod2 <- lFormula(logDc.km ~ forest + ag + shrub + dev + (1|pop1), data = CSFdata, REML = TRUE)
dfun <- do.call(mkLmerDevfun,mod2)
opt <- optimizeLmer(dfun)
mod_2 <- mkMerMod(environment(dfun), opt, mod2$reTrms,fr = mod2$fr)
mod2$reTrms$Zt <- ZZ
# Refit the model
dfun <- do.call(mkLmerDevfun,mod2)
opt <- optimizeLmer(dfun)
mod_2z <- mkMerMod(environment(dfun), opt, mod2$reTrms,fr = mod2$fr)

#Fitting model 3
mod3 <- lFormula(logDc.km ~ ag + dev + (1|pop1), data = CSFdata, REML = TRUE)
dfun <- do.call(mkLmerDevfun,mod3)
opt <- optimizeLmer(dfun)
mod_3 <- mkMerMod(environment(dfun), opt, mod3$reTrms,fr = mod3$fr)
mod3$reTrms$Zt <- ZZ
# Refit the model
dfun <- do.call(mkLmerDevfun,mod3)
opt <- optimizeLmer(dfun)
mod_3z <- mkMerMod(environment(dfun), opt, mod3$reTrms,fr = mod3$fr)

#Fitting model 4
mod4 <- lFormula(logDc.km ~ slope + shrub + dev + (1|pop1), data = CSFdata, REML = TRUE)
dfun <- do.call(mkLmerDevfun,mod4)
opt <- optimizeLmer(dfun)
mod_4 <- mkMerMod(environment(dfun), opt, mod4$reTrms,fr = mod5$fr)
mod4$reTrms$Zt <- ZZ
# Refit the model
dfun <- do.call(mkLmerDevfun,mod4)
opt <- optimizeLmer(dfun)
mod_4z <- mkMerMod(environment(dfun), opt, mod4$reTrms,fr = mod4$fr)

#Fitting model 5
mod5 <- lFormula(logDc.km ~ soils + slope + solarinsoz + (1|pop1), data = CSFdata, REML = TRUE)
dfun <- do.call(mkLmerDevfun,mod5)
opt <- optimizeLmer(dfun)
mod_5 <- mkMerMod(environment(dfun), opt, mod5$reTrms,fr = mod5$fr)
mod5$reTrms$Zt <- ZZ
# Refit the model
dfun <- do.call(mkLmerDevfun,mod5)
opt <- optimizeLmer(dfun)
mod_5z <- mkMerMod(environment(dfun), opt, mod5$reTrms,fr = mod5$fr)
```

```
CSF.IC <- cbind(Model = c(1:5), AIC = c(AIC(mod_1z), AIC(mod_2z), AIC(mod_3z), AIC(mod_4z),
AIC(mod_5z)), BIC = c(BIC(mod_1z), BIC(mod_2z), BIC(mod_3z), BIC(mod_4z), BIC(mod_5z)))
CSF.IC
```

```
     Model      AIC      BIC
[1,]     1 117.9295 132.8991
[2,]     2 113.7579 126.8563
[3,]     3 114.7831 124.1391
[4,]     4 118.8222 130.0494
[5,]     5 129.5256 144.4952
```

We've got some results, great!
Now let's work with these a bit. First, because we don't have an infinite number of samples, we'll convert AIC to AICc:
First, find the k parameters used in the model and add them to the table

```
CSF.IC <- cbind(CSF.IC, k = c(attr(logLik(mod_1z), "df"), attr(logLik(mod_2z), "df"), attr(logLik(mod_3z),
"df"), attr(logLik(mod_4z), "df"), attr(logLik(mod_5z), "df")))
CSF.IC
```

```
     Model      AIC      BIC k
[1,]     1 117.9295 132.8991 8
[2,]     2 113.7579 126.8563 7
[3,]     3 114.7831 124.1391 5
[4,]     4 118.8222 130.0494 6
[5,]     5 129.5256 144.4952 8
```

Now, calculate AICc and add it to the dataframe

```
CSF.IC<- as.data.frame(CSF.IC)
AICc <- CSF.IC$AIC + 2*CSF.IC$k*(CSF.IC$k+1)/(48-CSF.IC$k-1)
CSF.IC <- cbind(CSF.IC, AICc = AICc)
CSF.IC
```

```
  Model      AIC      BIC k     AICc
1     1 117.9295 132.8991 8 121.6218
2     2 113.7579 126.8563 7 116.5579
3     3 114.7831 124.1391 5 116.2116
4     4 118.8222 130.0494 6 120.8709
5     5 129.5256 144.4952 8 133.2179
```

Next we calculate evidence weights for each model based on AICc and BIC

```
#Calculate model weights for AICc
AICcmin <- min(CSF.IC$AICc)
RL <- exp(-0.5*(CSF.IC$AICc - AICcmin))
sumRL <- sum(RL)
AICew <- RL/sumRL
CSF.IC <- cbind(CSF.IC, AICew)
#Calculate model weights for BIC
BICmin <- min(CSF.IC$BIC)
RL.B <- exp(-0.5*(CSF.IC$BIC - BICmin))
```

```
sumRL.B <- sum(RL.B)
BICew <- RL.B/sumRL.B
CSF.IC <- cbind(CSF.IC, BICew)
```

```
  Model      AIC      BIC k     AICc         AICew         BICew
1     1 117.9295 132.8991 8 121.6218 0.0333420944 9.476981e-03
2     2 113.7579 126.8563 7 116.5579 0.4193794642 1.944681e-01
3     3 114.7831 124.1391 5 116.2116 0.4986441626 7.566276e-01
4     4 118.8222 130.0494 6 120.8709 0.0485331352 3.939856e-02
5     5 129.5256 144.4952 8 133.2179 0.0001011436 2.874854e-05
```

1. Why would adding in the last land cover cause a high amount of collinearity? Consider how these data were calculated.
2. What did using AICc (rather than AIC) do to inference from these results?
3. How does k relate to the number of parameters in each model?
4. Find the best model and type its name into R to look at the beta values for the fixed effects. What does this tell you about the influence of land covers on gene flow of Columbia spotted frogs?

Section 3.
In this section, you will create your own (small) set of hypotheses to rank for evidence using the dataset provided. Modify the code in Section 2 to do this.

Hypotheses:

VIF table for full model:

Residual plot for full model:

Table of AICc and BIC weights:

What can you infer from your analysis about what influences gene flow of this species?
