A gentle introduction to Phylogenetic Generalised Linear Mixed Models

WD Pearse, MW Cadotte, J Cavender-Bares, AR Ives,
C Tucker, S Walker, & MR Helmus

February 26, 2019

Contents

1 Introduction

The following text is a (slightly) modified form of a short course given on eco-phylogenetics. It is not intended as a rigorous, comprehensive explanation of how PGLMMs work, but we hope its more conversational tone might make it a useful introduction. PGLMMs are extremely flexible: this is their greatest strength, but it can make them difficult for the beginner. Persevere, because it's worth it! For more information, read the original papers (Ives & Helmus, 2011; Rafferty & Ives, 2013) or this short overview (Pearse *et al.*, 2014).

2 PGLMMs

Fingerprint regressions (`fingerprint.regression`) are great, but some ecologists have a more fundamental question that they feel they don't answer: *what drives co-occurrence in my system*? Is it shared/divergent traits, phylogenetic (dis-)similarity, shared/divergent environmental responses (driven by traits or phylogeny), or... something else that's unique to species/sites?

A Phylogenetic Generalised Linear Mixed Model (PGLMM) is one way of answering that question. It's, quite literally, just a regression where you ask predict species' presence/absence/abundance at sites. What makes it difficult to wrap your head around are the *random effects*, which incorporate species' traits, environmental conditions, species' phylogenetic relatedness, and species' responses to environmental conditions (as a function of traits and phylogeny).

Let's step through a simple example of how to simulate some data under PGLMM, and then you can try and fit it to your own data. Be careful not to try this with the mammal dataset from earlier as-is; PGLMMs can take a very long time to fit with large datasets...

```
nspp <- 15
nsite <- 10
env <- 1:nsite
env <- as.numeric(scale(env))
```

Nothing too scary. We say how many species and sites we want to simulate, then setup a (scaled) linear environmental gradient.

```
require(pez)
require(ape)
phy <- rcoal(n=nspp)
Vphy <- vcv(phy)
Vphy <- Vphy/(det(Vphy)^(1/nspp))
```

For some reason, this part seems to scare the living daylights out of people, but it really shouldn't. First step: simulate a phylogeny. Second step: calculate the Variance Co-Variance (VCV) matrix of that phylogeny, which is just the branch length seperating species. Third step: standardise that VCV matrix. More species tends to mean larger phylogenetic distances, so we have to standardise the VCV to make the next few steps work in the same way across all phylogenies. It's like standardising variables in a regression (*e.g.*, like we did for the environmental gradient)—it keeps the effect sizes constant.

The determinant (`det`) popping up is probably what is so scary, so we're now going to explain what it is. If you don't care then (maybe) good for you and just skip this. In maths, you can turn essentially any matrix into a shape in multiple dimensions - a 2-by-2 matrix defines a parallelogram (each 'side' of the matrix is a 'side' of the parallelogram), 3-by-3 becomes a cuboid-like thing, etc. The determinant is simply the area/volume of that shape. So, by dividing all the elements of the matrix

by the matrix's 'volume', we standardise all the elements to account for the size of it. But wait! If the determinant is an area/volume, and all the elements are simply distances, then we have a unit problem - the VCV is distances (one dimension, e.g., years) yet the determinant is an area/volume (many dimensions, *e.g.*, $years^3$). So we put the determinant to the power of $\frac{1}{n.spp}$ so that the units match (*e.g.*, $\frac{years}{((years^3)(\frac{1}{3}))}$ is the same as $\frac{years}{years}$).

```
iD <- t(chol(Vphy))
intercept <- iD %*% rnorm(nspp)
slope <- iD %*% rnorm(nspp)
```

Now we must simulate set the parameters (rules) that determine how species are distributed throughout our ecosystem. First: the VCV has repeated elements across the 'diagonal' (i.e., the distance from species A to B is the same as B to A), so set all those repeated elements to 0 to avoid double-counting. This is called Cholesky decomposition; we then flip the matrix round ('**t**ranspose' it) to allow for matrix-magic in the next step. Second and third: we want to simulate species' presences and absences along the environmental gradient, which means we need an intercept and slope that determines presence/absence along the gradient for each species. Draw some random numbers, then multiply them by the transformed matrix from step one, to get single intercepts and slopes for each species *where close relatives have similar values*. The Cholesky decomposition, combined with the magic of matrix multiplication, assures this. Note that you could play around with the variance et al. on the random draw to set up different kinds of relationships...

```
prob <- rep(intercept, each=nsite)
prob <- prob + rep(slope, each=nsite) * rep(env, nspp)
prob <- prob + rnorm(nspp*nsite)
pres <- rbinom(length(prob), size=1, prob=exp(prob)/(1+exp(prob)))
```

Now we have to figure out the probabilities of species being in each community. First: add all the intercepts of the probabilities of being in a site. Second: add to the intercept the slope of each species' relationship multiplied by the environmental value in that site. Third: add some error to that relationship. Fourth: randomly draw presence (1) and absence (0) on the basis of a logit for each species in each site on the basis of the probabilities we've created. The use of `rep` might seem a bit weird, so print it out and check it by eye if you're confused.

```
site <- factor(rep(1:nsite, nspp))
species <- factor(rep(1:nspp, each=nsite))
env <- rep(env, nspp)
```

This final step is important, and getting it right is all the more important because the PGLMM function is written rather oddly. The *site* and *species* variable *must* be a factors. You will get what seem like odd error messages if the lengths of all your data points do not match up; bear in mind that a 'non-conformable argument', in maths, is something that's the wrong length. This is PGLMM's way of saying something like "you've given me ten sites and ten species, but only fifty pieces of data".

```
r.intercept.spp.indep <- list(1, sp = species, covar = diag(nspp))
r.intercept.spp.phy <- list(1, sp = species, covar = Vphy)
r.slope.spp.indep <- list(env, sp = species, covar = diag(nspp))
r.slope.spp.phy <- list(env, sp = species, covar = Vphy)
r.site <- list(1, site = site, covar = diag(nsite))
rnd.effects <- list(r.intercept.spp.indep, r.intercept.spp.phy, r.slope.spp.indep, r.
```

Now we can finally take advantage of the power of PGLMM - we can set whatever kind of model we want. In this case it's a simple one - random effects for the intercept and slope, either for each species independently or allowing for phylogenetic covariation. I make one last one for the sites, and merge them all together in one big list. We can now test whether environment has an effect (the slope), whether species have different overall means (the intercepts), whether phylogeny plays a role, and control for site-level differences in abundance. You can specify *anything you want* in these random effects - some people have put space, others time, and in the paper in your reading list there's an example of traits.

```
model <- communityPGLMM(pres ~ env, family = "binomial", sp = species, site = site, r
communityPGLMM.binary.LRT(model, re.number = 1)

## $LR
## [1] -1.933817e-05
##
## $df
## [1] 1
##
```

```
## $Pr
## [1] 0.5

communityPGLMM.binary.LRT(model, re.number = 2)

## $LR
## [1] 7.623575
##
## $df
## [1] 1
##
## $Pr
## [1] 4.715946e-05
```

Now we fit the model! We can check the significances of each of the random effect structures as shown. Note that we're using random effects because, were we to estimate fixed effects, we'd be estimating at least 20 parameters, which is way too many for 100 data points. Of course, not everyone likes random effects, and many people don't like testing for their significance... Search out the 'glmm wiki' online for more details.

We went through all that simulation because it's important to see that PGLMM is "nothing more" than a fancy way of regressing presence/absence of species against environmental variables and traits. Look at the simplicity of the formula (presence environment), and the simplicity of the model we used to simulate the data (a slope over an environmental gradient). You'll also be pleased to note that there's a simple wrapper for all this, so when you're working with real data you can just use `as.data.frame` on a `comparative.comm` object to automatically create all the variables you need. Of course, it won't create the random effects for you - because that's the fun bit where you get to decide what questions you want to answer!

References

Ives, A.R. & Helmus, M.R. (2011) Generalized linear mixed models for phylogenetic analyses of community structure *Ecological Monographs* **81**, 511–525.

Pearse, W.D., Cavender-Bares, J., Puvis, A. & Helmus, M.R. (2014) Metrics and models of community phylogenetics. *Modern Phylogenetic Comparative Methods*

and their Application in Evolutionary Biology—Concepts and Practice (ed. L.Z. Garamszegi), Springer-Verlag, Berlin, Heidelberg.

Rafferty, N.E. & Ives, A.R. (2013) Phylogenetic trait-based analyses of ecological networks *Ecology* **94**, 2321–2333.