Phylogenetic comparative methods in the lme4-verse

Michael Li and Ben Bolker

- The standard problem of *phylogenetic comparative methods* is to analyze relationships among data where the observations are gathered from nodes (usually tips) of a phylogenetic tree - for example, regression analyses of body temperature as a function of body size for animals within a clade
- More generally, we can frame this in the usual GLMM way as

$$
\begin{aligned}
y &\sim D(\mu, \phi) \\
\mu &= g^{-1}(\eta) = g^{-1}(X\beta + Zb) \\
b &\sim \mathrm{MVN}(0, \Sigma)
\end{aligned}
$$

  where the part that makes it specifically phylogenetic is that Σ captures the *phylogenetic correlation*. The PC is the correlation among observations due to relatedness; recently diverged taxa have higher correlation than more anciently diverged taxa. In the extreme case of a *star phylogeny* (all taxa diverged from each other simultaneously at some point in the past) the phylogenetic correlation collapses to a diagonal matrix and we get back to the simple, uncorrelated regression.

Various P(G)LMM (phylogenetic [generalized] linear mixed model] approaches have been proposed. Many depend on Pagel's lambda transformation, which gives the correlation matrix a particularly simple form (but has been criticized ...)

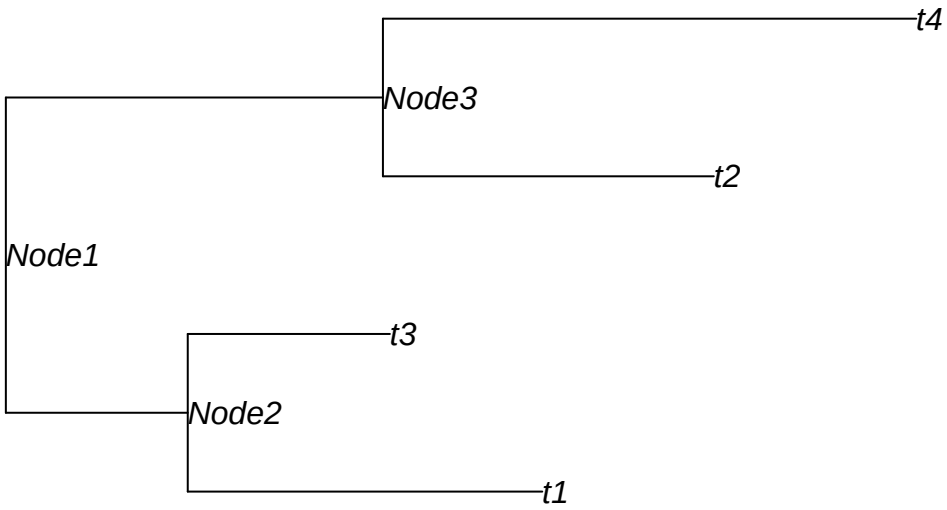
An alternative approach is to model the phylogenetic correlation as a *Gaussian process* (GP). In particular, suppose that the evolutionary process is a Brownian motion (an almost certainly incorrect/oversimplified model of evolution, but one that many phylogenetic methods are built on). In that case, the phylogenetic variability of a particular observation can be written as the sum of the evolutionary changes that occurred on all of the branches in the phylogeny in its past. If we set up the Z matrix appropriately, we can model everything with a sequence of *independent* errors, rather than having to do fancy things to impose a correlation structure on the random effects.

Nuts and bolts: from a phylogeny to a Z matrix for the GP

```
library(ape)
```

```
## Warning: package 'ape' was built under R version 3.4.4
```

```
library(Matrix)
set.seed(101)
r <- makeNodeLabel(rtree(4))
plot(r,show.node.label=TRUE)
```



Information in a `phylo` object is contained in the *edge matrix*:

> edge: a two-column matrix of mode numeric where each row represents an edge of the tree; the nodes and the tips are symbolized with numbers; the tips are numbered 1, 2, ..., and the nodes are numbered after the tips. For each row, the first column gives the ancestor.

```
t(r$edge)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    5    6    6    5    7    7
## [2,]    6    1    2    7    3    4
```

and a list of *edge lengths*

```
r$edge.length
```

```
## [1] 0.3000548 0.5848666 0.3334671 0.6220120 0.5458286 0.8797957
```

Inspecting this tree, we can figure out (see `$tip.label` and `$node.label` for label-to-number correspondences):

- tips are 1-4, nodes are 5-7
- tip 1 (`t1`) involves branches 2 ($6 \to 1$) and 1 ($5 \to 6$).
- tip 2 (`t3`) involves branches 3 ($6 \to 2$) and 1 ($5 \to 6$)
- tip 3 (`t2`) involves branches 5 ($7 \to 3$) and 4 ($5 \to 7$)
- tip 4 (`t4`) involves branches 6 ($7 \to 4$) and 4 ($5 \to 7$)

So, for example, we can say that the 'error' value corresponding to tip 1 is $\ell_1 b_1 + \ell_2 b_2$, where ℓ_i is the (square root of??) the branch length and the b_i are independent, homoscedastic Normal variates. Alternately, the Z matrix is

$$\begin{pmatrix} \ell_1 & \ell_2 & 0 & 0 & 0 & 0 \\ \ell_1 & 0 & \ell_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ell_4 & \ell_5 & 0 \\ 0 & 0 & 0 & \ell_4 & 0 & \ell_6 \end{pmatrix}$$

where ℓ_i is the length of the i^{th} branch, so that the species effects are Zb.

If we can build the corresponding Z matrix, then we can insert it in the `lme4` modular model-fitting process (see `?modular`).

Here's a (probably not very efficent) way to construct the Z matrix. (There must be a way to not walk the tree multiple times from every tip ...)

```
phylo.to.Z <- function(r,stand=FALSE){
  ntip <- length(r$tip.label)
  Zid <- Matrix(0.0,ncol=length(r$edge.length),nrow=ntip)
  nodes <- (ntip+1):max(r$edge)
  root <- nodes[!(nodes %in% r$edge[,2])]
  for (i in 1:ntip){
    cn <- i  ## current node
    while (cn != root){
      ce <- which(r$edge[,2]==cn)   ## find current edge
      Zid[i,ce] <- 1   ## set Zid to 1
      cn <- r$edge[ce,1]            ## find previous node
    }
  }
  V <- vcv(r)
  sig <- exp(as.numeric(determinant(V)["modulus"])/ntip)
  Z <- t(sqrt(r$edge.length) * t(Zid))
  if(stand){Z <- t(sqrt(r$edge.length/sig) * t(Zid))}
  rownames(Z) <- r$tip.label
  colnames(Z) <- 1:length(r$edge.length)
  return(Z)
}
phylo.to.Z(r)
```

```
## 4 x 6 sparse Matrix of class "dgCMatrix"
##            1         2         3         4         5         6
## t1 0.5477726 0.7647657 .         .         .         .
## t3 0.5477726 .         0.5774661 .         .         .
## t2 .         .         .         0.7886774 0.7388021 .
## t4 .         .         .         0.7886774 .         0.9379743
```

(This could benefit from the repeated-entry sparse matrix class that Steve Walker wrote.)

On the other hand, it only takes a few seconds to run for a 200-species phylogeny (see below).

constructing a GP PGLMM with lme4: machinery

"All" we need to do is (1) call `(g)lFormula`, with a formula that includes a `(1|phylo)` term, to build the basic (wrong) structure; (2) modify the `reTrms` component of the structure appropriately; (3) go through the rest of the modular procedure for building a (G)LMM.

```
library(ape)
library(lme4)
library(MCMCglmm)
library(MASS)
library(pez)
library(glmmTMB)
library(dplyr)
library(coda)
library(lattice)
library(broom)
library(dotwhisker)
```

The phylo–to–Z function is already in the source code.

```
source("lme4_phylo_setup.R")
```

glmmTMB fits: nuts and bolts

`glmmTMB` can be deconstructed in a similar way. In fact, we can re-use a lot of the machinery. Being able to use `glmmTMB` means we can use a broader range of distributions, zero-inflation, etc. (machinery below assumes phylogenetic structure only in the conditional distribution). This is also a little clunky, some adjustment on the `glmmTMB` side might make it a bit easier.

```
source("glmmTMB_phylo_setup.R")
```

Example

get data

From chapter 11 of Garamszegi (ed.): data are here

```
if (!file.exists("data/phylo.nex")) {
    dir.create("data")
    download.file("http://mpcm-evolution.com/OPM/Chapter11_OPM/data.zip",
                  dest="data/OPM_ch11_data.zip")
    setwd("data")
    untar("OPM_ch11_data.zip")
    setwd("..")
}
phylo <- read.nexus("data/phylo.nex")
```

Compute appropriate Z matrix up front, to measure speed (also reusable in a few places below):

```
system.time(phyloZ <- phylo.to.Z(phylo))
```

```
##    user  system elapsed
##   1.226   0.021   1.256
```

Result comparison with Gaussian example in chapter 11

```
datG <- read.table("data/data_simple.txt",header=TRUE)
datG$obs <- factor(seq(nrow(datG)))
datG <- datG %>% mutate(sp = phylo)
phylo_lmm_fit <- phylo_lmm(phen~cofactor+(1|sp)
  , data=datG
  , phylonm = "sp"
  , phylo = phylo
  , phyloZ=phyloZ
  , REML = TRUE
  , control=lmerControl(check.nobs.vs.nlev="ignore",check.nobs.vs.nRE="ignore")
)

print(summary(phylo_lmm_fit))
```

```
## Linear mixed model fit by REML ['lmerMod']
##
```

```
## REML criterion at convergence: 1550.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4487 -0.5124 -0.0311  0.5663  2.2279
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sp       (Intercept) 207.03   14.388
##  Residual              83.74    9.151
## Number of obs: 200, groups:  sp, 200
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   39.821      6.999    5.69
## cofactor       5.175      0.136   38.06
##
## Correlation of Fixed Effects:
##          (Intr)
## cofactor -0.186
```

Result comparison with Gaussian with repeated measures example in chapter 11

```
datR <- read.table("data/data_repeat.txt",header=TRUE)
datR$obs <- factor(seq(nrow(datR)))
datR <- (datR
  %>% mutate(sp = species
        , animals = phylo
        )
)
datR$spec_mean_cf <- sapply(split(datR$cofactor,datR$phylo),mean)[datR$phylo]
datR$within_spec_cf <- datR$cofactor-datR$spec_mean_cf
phylo_lmm_fit <- phylo_lmm(phen~spec_mean_cf+within_spec_cf+(1|sp) + (1|animals)
  , data=datR
  , phylonm = "sp"
  , phylo = phylo
  , phyloZ=phyloZ
  , REML = FALSE
  , control=lmerControl(check.nobs.vs.nlev="ignore",check.nobs.vs.nRE="ignore")
)

print(summary(phylo_lmm_fit))
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
##
##      AIC      BIC   logLik deviance df.resid
##   7425.8   7455.2  -3706.9   7413.8      994
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0237 -0.6058 -0.0005  0.5853  2.5184
##
```

```
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sp       (Intercept) 257.11   16.03
##  animals  (Intercept)  25.30    5.03
##  Residual              65.45    8.09
## Number of obs: 1000, groups:  sp, 200; animals, 200
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)    38.24867    7.68928   4.974
## spec_mean_cf    5.09606    0.10227  49.831
## within_spec_cf -0.05911    0.18646  -0.317
##
## Correlation of Fixed Effects:
##             (Intr) spc_m_
## spec_men_cf -0.128
## wthn_spc_cf  0.000  0.000
```

Result comparison with non-Gaussian example in chapter 11

```
dat <- read.table("data/data_pois.txt",header=TRUE)
dat$obs <- factor(seq(nrow(dat)))


dat <- dat %>% mutate(sp=phylo)
phylo_glmm_fit <- phylo_glmm(phen_pois~cofactor+(1|sp)+(1|obs)
  , data=dat
  , phylonm = "sp"
  , family = poisson
  , phylo = phylo
  , phyloZ=phyloZ
  , control=lmerControl(check.nobs.vs.nlev="ignore",check.nobs.vs.nRE="ignore")
)

summary(phylo_glmm_fit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
##
##      AIC      BIC   logLik deviance df.resid
##    699.8    713.0   -345.9    691.8      196
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0615 -0.5704 -0.3418  0.4268  5.0133
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  sp     (Intercept) 0.01224  0.1106
##  obs    (Intercept) 0.04108  0.2027
## Number of obs: 200, groups:  sp, 200; obs, 200
##
```

```
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.06628    0.18658  -11.07   <2e-16 ***
## cofactor     0.25022    0.01119   22.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## cofactor -0.926
```