# State-dependent models: anoles and diversitree

Dan Rabosky

August 2, 2011

## 1 motivation and context

Here we will use diversitree to replicate some of the analyses from Rabosky and Glor (2010). The basic question: what diversification model best explains patterns of species richness and speciation-through-time in anoles from the Greater Antilles (Cuba, Hispaniola, Jamaica, Puerto Rico)? Even though there are many Caribbean islands with anoles, these four islands are special: it is only on these island that `Anolis` lizards have undergone evolutionary radiations. Anole communities on the other islands (e.g., the lesser Antilles) appear to be assembled mainly by inter-island dispersal. We will look at the following simple models:

- Constant speciation-through-time on all islands. This model has as single time-invariant speciation rate that applies to all anole lineages, at all points in time. This model will fit the data well if the ages of anole radiations is the primary determinant of their species richness, because the 'oldest' islands (radiations) will have had more time for their diversity to build up through time.

- Island-specific but time-constant speciation rates. This is the model proposed by Losos and Schluter (2000) to explain patterns of species richness on the Greater Antilles. Here, we have island-specific speciation rates, but these rates do not vary through time. The prediction here is that speciation rates scale with island size: larger islands have more processes that can drive speciation, so the rate of species accumulation is faster. This is also a non-equilibrium model, because it assumes that there is no diversity-dependence of speciation or extinction: lineages on large islands may accumulate faster, but there is no signal of "saturation" of the island species pool (e.g., a slowing of rates through time as the number of species increases).

- A model with island-specific, time-varying rates of speciation. Here, we will allow each island to have its own unique time-varying rate of speciation. This model should fit best if diversity-dependent or carrying-capacity dynamics influence patterns of speciation through time. Here we expect an initial high rate of speciation on each island to decline through time as total species richness increases.

This is not exactly the same analysis from Rabosky and Glor (2010), but will be similar. The analyses differ in:

- We used a more complex set of biogeographic scenarios, including time-varying rates of transition between island 'character states'. These aren't available yet in diversitree , but probably will be soon.
- We included extinction in our models, but we will ignore it here for our purposes. In any event, extinction rates as estimated from phylogenies are almost always very low.

# 2 Getting started with the anole analysis

First, we'll read our anole tree into R.

```
> library(ape)
> library(diversitree)
> anoletree <- read.tree("anolisFinalMCC.tre")
```

And we'll look at the tree:

```
> anoletree

Phylogenetic tree with 187 tips and 186 internal nodes.

Tip labels:
        agassizi, casildae, microtus, aqbl, bigblue, frenatus, ...

Rooted; includes branch lengths.
```

All good. Now we'll read in some biogeographic data for anoles:

```
> d <- read.csv("biogeography.csv", header = T)
> head(d)

   species region
1   acutus      4
2   aeneus      5
3 agassizi      6
4     ahli      0
5  alayoni      0
6  alfaroi      0
```

and we'll look at the biogeographic characters:

```
> table(d$region)

 0  1  2  3  4  5  6
49 38  6 10 11 17 56
```

This summarizes the number of species in each character state. State 0 is Cuba, state 1 is Hispaniola, state 2 is Jamaica, and state 3 is Puerto Rico. The other states are irrelevant to our purposes here (they include species found on the mainland and on islands where no speciation has occurred).

One critical point here is that here we are modeling biogeographic transitions between islands

as a simple character state change. Strictly speaking, this makes the biological assumption that a lineage can only disperse to a new island if and only if the parental lineage goes extinct. This assumption is made for analytical convenience [I will illustrate this point on the board or something....]

Now: we have species in the tree that we don't care about, namely all those with character states other than 0,1,2, or 3. So we'll get rid of them.

```
> dropset <- as.character(d$species[d$region >= 4])
> anoletree <- drop.tip(anoletree, dropset)
```

And we'll also equalize our biogeographic dataframe `d` as well, using one of my favorite R functions:

```
> d <- d[d$species %in% anoletree$tip.label, ]
```

And we'll make sure all species from `d2` are in the `anoletree` and viceversa:

```
> setdiff(anoletree$tip.label, d$species)

character(0)

> setdiff(d$species, anoletree$tip.label)

character(0)
```

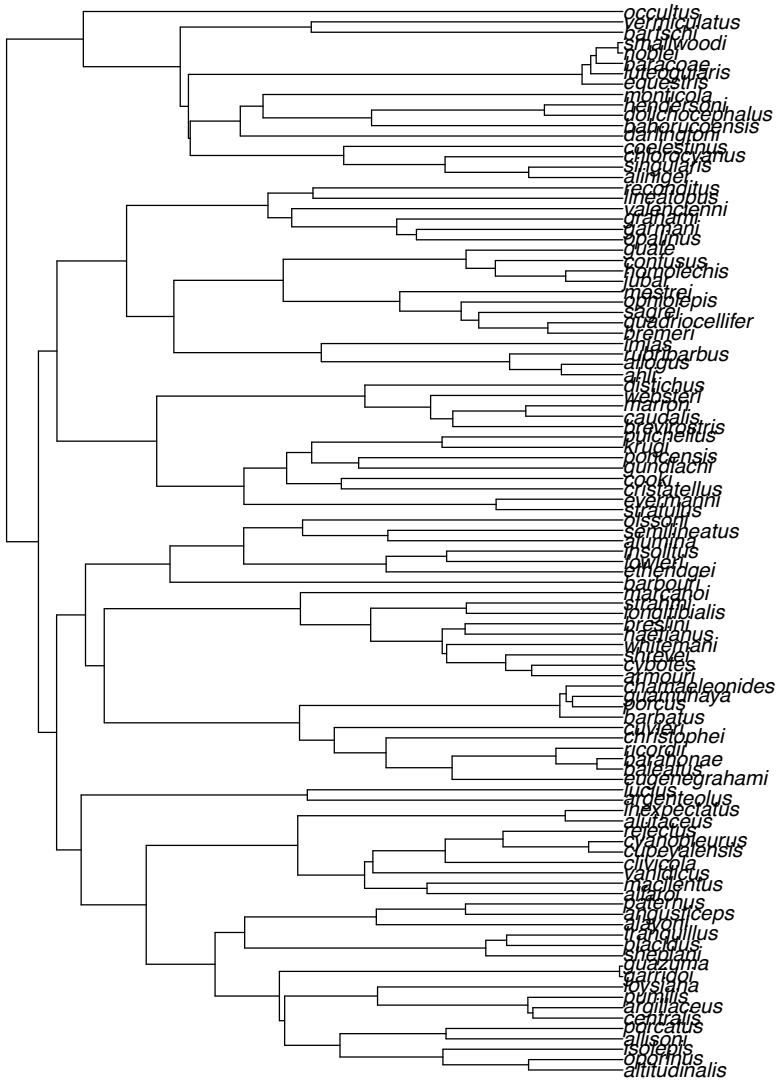
One more thing: this anole tree isn't actually calibrated on a meaningful divergence timescale. These divergence times are just relative. As such, we'll just scale the tree so that the root speciation event occurred exactly 1.0 time units ago:

```
> anoletree$edge.length <- anoletree$edge.length/max(branching.times(anoletree))
> max(branching.times(anoletree))

[1] 1
```

Now we'll actually have a look at our data:

```
> plot.phylo(anoletree)
```




Now we'll make a vector of geographic character state data. We have to add 1 to each, because multistate diversitree requires that character states be indexed from 1:n. A bit confusing, as the binary state BiSSE model uses 0 and 1 for states:

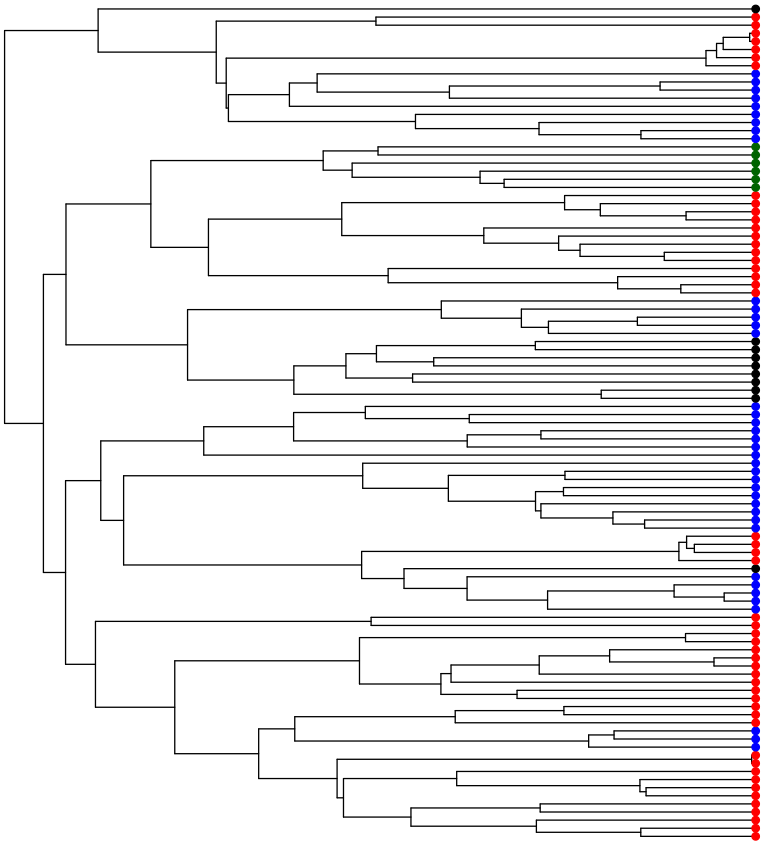
```
> biog <- d$region + 1
> names(biog) <- d$species
```

And we'll sort them to match the tree tip labels:

```
> biog <- biog[anoletree$tip.label]
```

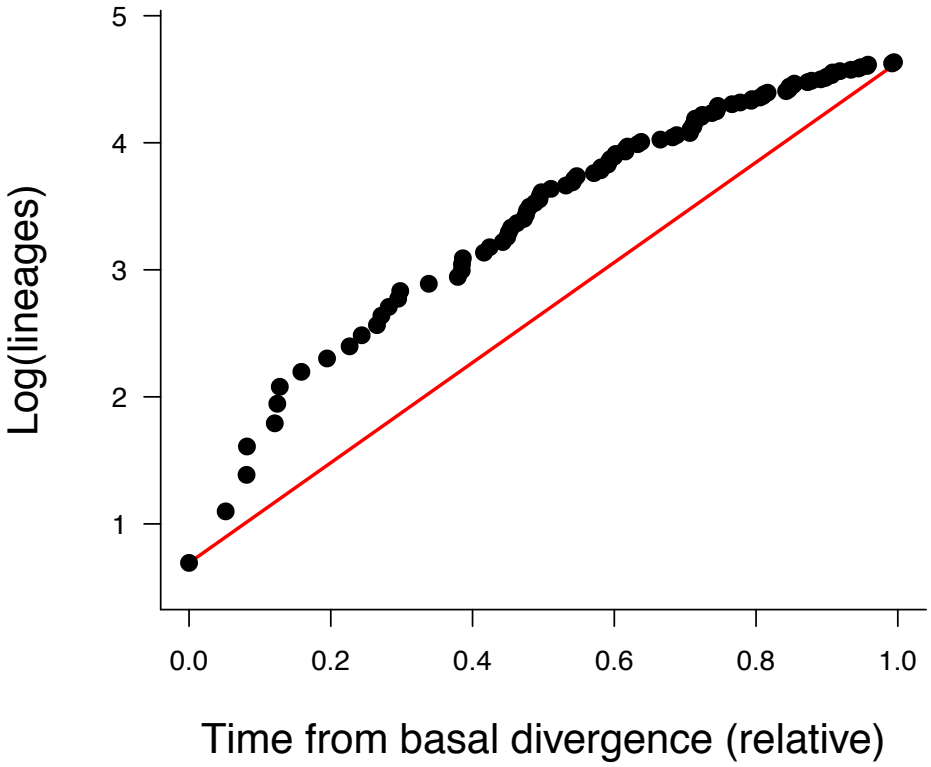
This is a fairly big tree, so now we'll just plot it with color tip states: red, Cuba; blue, Hispaniola; black, Puerto Rico; darkgreen, jamaica.

```
> plot.phylo(anoletree, show.tip.label = FALSE)
> colorvec <- c("red", "blue", "darkgreen", "black")
> statecols <- colorvec[biog]
> tiplabels(pch = 19, col = statecols, cex = 0.8)
```



And what the heck - why not generate an overall lineage-through-time plot, on a log-scale? We'll source a file with a bunch of utility functions I wrote for this workshop and plot an LTT curve:

```
> source("Rabosky_SB_functions.R")
> nicerLttPlot(anoletree)
```




Here, the red line is the 'pure birth' expectation: this is what the lineage-through-time curve should look like under a model with constant speciation through time and no extinction. Actually, the slope of the LTT plot (on a semilog scale) is an estimate of the instantaneous rate of speciation at any point in time. So - what you see here is that speciation rates in anoles - as a whole - appear to have decelerated through time. This is one interpretation of a "concave down" LTT plot.

# 3 Building and fitting complex models in diversitree

The first model we are going to fit is a simple pure-birth model. It may seem straightforward to just do what we did earlier for warblers. If all we wanted to do was fit a pure-birth model, then this would be fine. However, we are going to use a different probability model for the data: we are going to be comparing the likelihood of the data under a pure-birth model, to one where there are multiple island 'character states'. This requires using a probability model that describes the joint likelihood of both the character state parameters and the speciation-extinction parameters. Another thing we have to worry about in the anoles is

incomplete taxon sampling. Some islands (e.g., Puerto Rico) are very well sampled, but Cuba is missing a few species.

**Why is incomplete sampling important???**

Now we'll set up a vector of sampling probabilities, indicating the number of species we have in each island 'state':

```
> cuba.sprob <- 49/61
> hisp.sprob <- 38/40
> jam.sprob <- 1
> pr.sprob <- 1
> s.probs <- c(cuba.sprob, hisp.sprob, jam.sprob, pr.sprob)
> names(s.probs) <- 1:4
```

We have 4 character states, so we'll build a full 'multi-state' `musse` model as follows:

```
> lfx.global <- make.musse(anoletree, states = biog, k = 4, sampling = s.probs)
```

`lfx.global` is now our likelihood function! Let's look at it:

```
> lfx.global
```

```
MuSSE likelihood function:
function (pars, ...)
ll.musse(pars, ...)
<environment: 0x10212d948>
attr(,"k")
[1] 4
```

This doesn't help very much. But just like we did earlier, we made a function that takes a vector of parameters and returns a likelihood. What are these parameters in `lfx.global`? We can check these using the `argnames` function in diversitree as follows:

```
> argnames(lfx.global)
```

```
 [1] "lambda1" "lambda2" "lambda3" "lambda4" "mu1"     "mu2"     "mu3"
 [8] "mu4"     "q12"     "q13"     "q14"     "q21"     "q23"     "q24"
[15] "q31"     "q32"     "q34"     "q41"     "q42"     "q43"
```

OK...this is way too many parameters! there are 20 parameters in the default model constructed by `make.musse` . We need to do something to constrain some of these. Since we are fitting a pure-birth model, we'll do the following:

- We'll assume here that the 'dispersal' rate between all islands is equal - this is an 'equal rates' transition model.
- We want all speciation rates to be equal across islands (all islands have same rate)
- We want extinction set to zero (since this is a pure-birth model)

This is a little complicated in diversitree , but we have to set up a list of constraints:

```
> cons <- list(lambda2 ~ lambda1, lambda3 ~ lambda1, lambda4 ~
+     lambda1, mu1 ~ 0, mu2 ~ 0, mu3 ~ 0, mu4 ~ 0, q13 ~ q12, q14 ~
+     q12, q21 ~ q12, q23 ~ q12, q24 ~ q12, q31 ~ q12, q32 ~ q12,
+     q34 ~ q12, q41 ~ q12, q42 ~ q12, q43 ~ q12)
```

Then we use the `constrain` function to force these constraints into the likelihood function:

```
> lfx.global <- constrain(lfx.global, formulae = cons)
```

And we can check the details of the 'constrained' function:

```
> argnames(lfx.global)

[1] "lambda1" "q12"
```

So the new function only takes 2 parameters: a speciation rate, and a character transition rate. Here we'll fit the model, using diversitreeś built-in function for optimization, `find.mle`:

```
> pars.init <- runif(2, 0, 5)
> res.purebirth <- find.mle(lfx.global, pars.init, method = "optim")
```

Our fitted model is now the object `res.purebirth` , and we can access its attributes using a number of functions:

```
> attributes(res.purebirth)

$names
[1] "par"          "lnLik"        "counts"       "convergence"  "message"
[6] "optim.method" "func"         "method"

$class
[1] "fit.mle.musse" "fit.mle"
```

```
> coef(res.purebirth)

   lambda1        q12
2.59195852 0.07859546

> logLik(res.purebirth)

'log Lik.' -58.56576 (df=2)
```

Or we can access them directly:

```
> res.purebirth$lnLik

[1] -58.56576

> res.purebirth$par

   lambda1        q12
2.59195852 0.07859546
```

Now we'll make a model where we have separate island-specific but time-constant rates of speciation. To do this, we'll simply (i) make a new likelihood function, (ii) set up a new constraint list, and (iii) apply the constraints to the likelihood function. The constraints here will be similar to the previous model, but now we want separate $\lambda$ values for each island. First, the likelihood function:

```
> lfx.islandConstant <- make.musse(anoletree, states = biog, k = 4,
+     sampling = s.probs)
```

Now, the constraint list:

```
> cons <- list(mu1 ~ 0, mu2 ~ 0, mu3 ~ 0, mu4 ~ 0, q13 ~ q12, q14 ~
+     q12, q21 ~ q12, q23 ~ q12, q24 ~ q12, q31 ~ q12, q32 ~ q12,
+     q34 ~ q12, q41 ~ q12, q42 ~ q12, q43 ~ q12)
```

Now to apply the constraints:

```
> lfx.islandConstant <- constrain(lfx.islandConstant, formulae = cons)
```

And we'll check the argnames. There should be 5 (4 different $\lambda$ values and a transition rate):

```
> argnames(lfx.islandConstant)

[1] "lambda1" "lambda2" "lambda3" "lambda4" "q12"
```

And to fit this model, we'd just do

```
> pars.init <- runif(5, 0, 5)
> res.islandConstant <- find.mle(lfx.islandConstant, pars.init,
+     method = "optim")
```

This could take awhile to fit. But we'll try it anyway. Now we'll try the full model, with

island-specific and time-varying rates of speciation. This is a bit trickier, because we have to include time-varying models of speciation when we construct the likelihood function. We can do this by passing a list of functions to `make.musse` . We also have to associate a name with each function in the list:

```
> func.list <- rep(list(linear.t, constant.t), c(4, 16))
```

sets up a list of four linear time-dependent functions and 16 time-constant functions, and now we name the functions in the list using a vector of parameter names:

```
> names(func.list) <- c("lamba1", "lambda2", "lambda3", "lambda4",
+     "mu1", "mu2", "mu3", "mu4", "q12", "q13", "q14", "q21", "q23",
+     "q24", "q31", "q32", "q34", "q41", "q42", "q43")
```

And we put all of this together to make a model with time-dependent variation in speciation through time, using `make.musse.t`.

```
> lfx.islandTD <- make.musse.t(anoletree, states = biog, functions = func.list,
+     k = 4, sampling = s.probs)
```

This function has an unmanageable number of parameters:

```
> argnames(lfx.islandTD)
```

```
 [1] "lamba1.c"  "lamba1.m"  "lambda2.c" "lambda2.m" "lambda3.c" "lambda3.m"
 [7] "lambda4.c" "lambda4.m" "mu1"       "mu2"       "mu3"       "mu4"
[13] "q12"       "q13"       "q14"       "q21"       "q23"       "q24"
[19] "q31"       "q32"       "q34"       "q41"       "q42"       "q43"
```

But we don't care about extinction, and we are also assuming that there is only a single transition rate, so we can do the constraint thing again:

```
> cons <- list(mu1 ~ 0, mu2 ~ 0, mu3 ~ 0, mu4 ~ 0, q13 ~ q12, q14 ~
+     q12, q21 ~ q12, q23 ~ q12, q24 ~ q12, q31 ~ q12, q32 ~ q12,
+     q34 ~ q12, q41 ~ q12, q42 ~ q12, q43 ~ q12)
```

We now apply the constraints to the likelihood function:

```
> lfx.islandTD <- constrain(lfx.islandTD, formulae = cons)
```

This should have substantially reduced our parameter count:

```
> argnames(lfx.islandTD)

[1] "lamba1.c"  "lamba1.m"  "lambda2.c" "lambda2.m" "lambda3.c" "lambda3.m"
[7] "lambda4.c" "lambda4.m" "q12"
```

We can just fit the model as before:

```
> n.pars <- length(argnames(pars.init))
> pars.init <- runif(n.pars, 0, 5)
> res.islandTD <- find.mle(lfx.islandTD, pars.init)
```

This will probably take awhile to run. To get a sense of how slow these likelihood calculations are, we'll do 10 calculations with random parameters and get the mean runtime. You can do this using the `system.time` function:

```
> n.pars <- length(argnames(lfx.islandTD))
> timeTrial <- function() {
+     for (i in 1:10) {
+         pars.init <- runif(n.pars, 0, 5)
+         lfx.islandTD(pars.init)
+     }
+ }
> timed.results <- system.time(timeTrial())
> timed.results

   user  system elapsed
  4.465   0.006   4.497
```

Fitting a model with this many parameters could easily require 2000 or more evaluations of the likelihood function during the optimization process. how long would this take? In minutes, assuming 2000 evaluations:

```
> 200 * timed.results/60

    user   system  elapsed
14.88333  0.02000 14.99000
```

Should give an estimate of total time in minutes. Not fast! And this is still a simple model: we ignored extinction, and we used the simplest possible biogeographic transition matrix! For a model with this many parameters, we'd also want to do many optimizations with different starting parameters to ensure that we are converging on a stable solution.

This is a brief overview of multi-state, time-dependent models in diversitree .