Lab 12: Lineages Through Time

Today we will analyze changes in diversification that effect all the organisms in a clade at once. We will use lineage-through-time plots (LTT plots) to get a sense of the overall pattern of diversification. We will compare these to a null distribution in order to find significant deviations from our expectations. We will also explore a couple of likelihood methods for testing for shifts in diversification with time. As a first step we'll simulate trees in *R*.

Load the *ape*, *laser*, and *geiger* packages before you start.

Simulating Trees in R

You did some of this in the tree-balance lab. There are several methods for simulating trees in *R*. We will look at a few of these. The *ape* package has two functions for simulating trees, ***rtree*** and ***rcoal***. Both create trees with equal branching, but neither one gives us useful branch lengths for our current purposes.

A more useful function is ***birthdeath.tree*** in the *geiger* package. This function simulates trees by the birth death (BD) process, starting from 2 lineages. The user inputs a birth rate, a death rate and a time or final number of taxa for the process. If this function just runs the BD process for the given amount of time, it will produce trees with different numbers of final taxa. If it runs until it reaches a given number of taxa, then it will take different amounts of time. Load the *geiger* package.

```
tree1<-birthdeath.tree(0.3,0,10)
```

This will make a tree with λ=0.3, and μ=0 (speciation and extinction, respectively) by running the process for 10 time units. The expectation is that this tree will have *2*exp(0.3*10)=40* taxa, but it could have more or less. Let's see what that looks like:

```
plot(tree1)
axisPhylo()
```

Oh, it looks like crap. Wait, we can fix that:

```
tree1<-reorder(tree1)
plot(tree1)
axisPhylo()
```

Does your tree have about 40 taxa? Does it have depth 10? Let's make another tree but with a non-0 μ (extinction rate). We will set λ=0.35, and μ=0.05, so that *r* remains 0.3.

```
tree2<-reorder(birthdeath.tree(0.35,0.05,10))
plot(tree2)
axisPhylo()
```

You'll notice that this tree has both living and extinct lineages. We did not get any extinction

last time, because μ was 0. We can remove the extinct lineages as follows:

```
tree2.pruned<-prune.extinct.taxa(tree2)
plot(tree2.pruned)
axisPhylo()
```

There we are; that is the tree you would expect with only the extant taxa. We can also make a tree with a specific number of taxa.

```
tree3<-reorder(birthdeath.tree(0.35,0.05,taxa.stop=40))
plot(tree3)
axisPhylo()
```

There you go, exactly 40 taxa. You may have slightly more than 40 numbered lineages, because there are several extinct lineages, and this process stopped when there were 40 extant lineages. The amount of time this process took is randomly distributed. You may notice that the last duplication occurs in the present. This is cause the process stopped when it reached 40 living taxa. That may not give you an honest distribution of 40 taxa trees, as you could have 40 taxa again after this time. We won't deal with fixing this problem.

Now we are going to use my function ***rbdtree.n***. This function can take a λ, a μ, a time, and a final number of taxa and give you a random tree for those sets of parameters. It can also have λ and μ vary with time. You should note that this function will give you a set of branch lengths even if it is highly unlikely that the rates you chose would produce as many taxa as you chose, so it is important to pick a reasonable set of rates. Download *rbdtree.n3.R* from the website, and/or source it remotely:

```
source("http://ib.berkeley.edu/courses/ib200b/labs/lab12/rbdtree.n3.R")
```

Then, run the function *rbdtree.n*:

```
tree4<-rbdtree.n(40,10,0.35,0.05)
plot(tree4)
axisPhylo()
```

That's cool. This probably fits our null distribution better for the tests we are about to run, as it holds the final number of taxa and the time constant, and both those factors can have a large impact on an the distribution of lineages through time. Nevertheless, there may be situations in which the other assumptions are more appropriate.

Lineage-through-time Plots

Now we will make some lineage-through-time plots. These are plots of the number of lineages in a clade that have any living descendants against time. The expectation is that these plots will show exponential growth, if μ=0 and λ is constant. Significant deviations from this expectation indicate that diversification is changing with time. We will use some *ape* functions for making these plots.

```
ltt.plot(tree1)
```

It is common practice to log-transform the y-axis. We will use the general *R* plotting command *log="y"*, which will only change the way the values are plotted, and will not change the values themselves.

ltt.plot(tree1,log="y")

For a log transformed plot our expectation is a straight line, so we can add that line to our plot and compare it:

lines(c(-10,0), c(2,*total number of taxa*), lty=2)

That should look like a pretty good fit. You can also add other lineage-through-time lines to this plot.

ltt.lines(tree2,col=2)

OK, that looks a little weird, cause we have different numbers of final taxa. We can plot all our trees at once:

mltt.plot(tree1,tree2,tree3,tree4)

Null distributions for lineage-through-time plots

In order to understand how variation in the diversity parameters effects lineage-through-time plots, we have to understand what these plots would like under a null hypothesis. Let's start with the null hypothesis of a pure birth process that proceeds for a given amount of time.

trees.yule <- replicate(100, birthdeath.tree(0.3,0,10), simplify=FALSE)

This creates a list of trees by repeating our function 100 times. Let's make the lineage-through-times plot of our data.

mltt.plot(trees.yule, legend=FALSE)

The problem with that is all the trees have different numbers of taxa, and most of what you see is that those that start out with more taxa end to end up with more taxa. If you had a constant number of taxa, but variation in time, you would have a similar problem. Let's make trees with all those factors in common.

trees.yule2<-replicate(100, rbdtree.n(40,10,0.3,0), simplify=FALSE)
mltt.plot(trees.yule2, legend=FALSE, log="y")

We can see the general distribution of Yule trees from this plot, but it would be difficult to compare this distribution to another one, because all the lines would be jumbled up. So, I made another function that will plot the distribution of lineage-through-times plots for any set of parameter values.

ltt.null(40, 10, 0.3, 0)

The parameter values here are the same as they were for our *yule2* trees. The colors represent

the confidence intervals for different p-values. To see that the distributions are the same add our *yule2* trees to this plot.

lapply(trees.yule2,ltt.lines)->x

We assigned the output to the object x, so that this function would not print a bunch of useless information to the screen. As you can see, most of these lines fall within the center of our distribution. A few fall out of the 95% confidence interval. Maybe 1 falls completely out of the 99% CI, as we would expect with 100 simulations.

Let's see how trees with a higher death rate fit this distribution. We will raise μ to 0.1, and λ to 0.4, so that *r* does not change, and *N=2exp(rt)* is the same. First we'll make another plot of our null distribution, then we'll simulate 100 trees under our new parameters, and finally, we'll add those trees to our plot.

ltt.null(40,10,0.3,0)
trees.bd<-replicate(100,rbdtree.n(40,10,0.4,0.1),simplify=FALSE)
lapply(trees.bd,ltt.lines)->x

You can see that these distributions are not identical. First off the distribution with the higher extinction rate has a slightly larger spread. Secondly the plot is slightly more convex, with the majority of lineages below where you would expect them to be under the Yule distribution, and several well below the 99%CI. This implies that a significantly convex curve may be indicative of a high μ value. We would expect this result as older lineages are more likely to go extinct. However, most of the lineages are still well within the 95%CI, and so the power of this test would be relatively weak.

Comparing Real Trees to Null Distribution

Let's look at some actual trees, and see if they fit our null distribution. *laser* has several trees stored in it already, so we'll look at those. The first tree is an ultrametric tree of 69 Australian Agamid lizards from Rabosky (2006) Evolution 60:1152-1164.

data(agamids)
agamid.tree<-read.tree(text=agamids)
plot(agamid.tree)

OK, that tree looks nice. Let's compare that to a null distribution. We'll start out assuming a Yule process, so we can approximate λ by ML:

yule.agam<-yule(agamid.tree)

And identify the depth of the root:

btimes.agam<-sort(branching.times(agamid.tree),decreasing=TRUE)

Now we can use those values for our null distribution:

ltt.null(69,btimes.agam[1],yule.agam$lambda,0)

Then we can add the ltt for the tree:

ltt.lines(agamid.tree)

That is way outside our 99% confidence interval. It looks fine at the beginning, then it climbs way above the expected range and only comes back at the end. This looks nothing like any of our null distributions; when we increase μ, we get excessively convex plots, not this. The plot looks like there was a period of excessively high diversification about 0.22 time units ago, maybe another around 0.15. (I'm afraid I don't know the actual units.) We could try to fit a BD model, see if it looks like the death rate is greater than 0.

birthdeath(agamid.tree)

I still got a death rate of 0, so scratch that idea. Not that it would have mattered anyways. There are two other data sets in *laser* that have branching times only, ***plethodon*** from Highton and Larson (1979) Syst. Zool. 28:579-599 and ***warblers*** from Lovette and Bermingham (1999) *Proc. Roy. Soc. B. Lond.* 266:1629-1636. Repeat this analysis for both these data sets as follows.

data(plethodon)
yule.pleth <- pureBirth(plethodon) ##This is like *yule*, but for vector of branching times
ltt.null(length(plethodon)+1, max(plethodon), yule.pleth$r1, 0)

The functions for plotting LTT for branching times suck so we're going to have to do it ourselves:

ntaxa.pleth <- 1:length(plethodon) + 1 ## Number of lineages corresponding to times
segments(-plethodon, ntaxa.pleth, -plethodon, ntaxa.pleth-1)
segments(-plethodon, ntaxa.pleth, -c(plethodon[-1],0), ntaxa.pleth)

That looks a lot better than the Agamid tree. There does seem to be an excessively long period without a speciation from about 15 to 30 time units ago, but nothing outside our expectation. Well, maybe it would be for a lineage that had so many taxa already at 30 time units ago. We'll get to testing that in the next step.

You can fit a BD model to branching times data with:

bd.pleth <- bd(plethodon)

That one did get a death rate greater than 0, so let's try that null distribution. We should really do an LRT (likelihood ratio test) first, to see if the difference is significant, but you already know how to do that so we'll skip it. This function gave us *r* and $a=\lambda/\mu$, so we need to calculate λ and μ first.

lam.pleth <- bd.pleth$r1 / (1-bd.pleth$a)
mu.pleth <- lam.pleth - bd.pleth$r1
ltt.null(length(plethodon)+1 ,max(plethodon), lam.pleth, mu.pleth)
segments(-plethodon, ntaxa.pleth, -plethodon, ntaxa.pleth-1)
segments(-plethodon, ntaxa.pleth, -c(plethodon[-1],0), ntaxa.pleth)

That's an even more solid fit.

Question 1. Repeat this analysis for the warblers. What values of lambda and mu do you estimate?

How does this plot compare to its null distribution? (Be specific)

Simulating Trees with Varying Diversification Rates

What does it mean when a lineage-through-time plot does not fit the null distribution? The general assumption is that it means the diversification rates are changing. You can model the null distributions for this hypothesis and generate random trees using the functions I provided. You do this by providing a vector of times that represent the ends of each period. You can then set both of the rates for each period by assigning a vector for each rate in which each element of the vector sets the rate for the corresponding time. For example:

```
tree5<-rbdtree.n(40, c(3,6,10), c(0.1,1,0.05), 0)
plot(tree5)
```

…will produce a random tree with 40 taxa at the end, for which μ=0, and λ is 0.1 from time 0 to 3, 1 from time 3 to 6, and 0.05 from time 6 until the end of the process at time 3. You should see that most of the branching happens in the middle period for this tree. On the other hand:

```
tree6 <- mrbdtree.n(10, 40, c(3,6,10), c(0.1,1,0.05), c(0.05,0.05,0))
mltt.plot(tree6)
```

Will produce 10 trees in which μ also varies from one period of time to the next. You should clearly be able to see the expected pattern in the lineages through time plot.

You can also make trees and null distributions that are forced to have a given number of taxa at some time other than the end of the tree. Remember what the *plethodon* plot looked like:

```
ltt.null(length(plethodon)+1, max(plethodon), lam.pleth, mu.pleth)
segments(-plethodon, ntaxa.pleth, -plethodon, ntaxa.pleth-1)
segments(-plethodon, ntaxa.pleth, -c(plethodon[-1],0), ntaxa.pleth)
```

Everything is within the expected range, but there's that weird period of no speciation. I wonder if the overall pattern is outside our null, even if at every time is inside it just fine. Let's see what the null distribution would look like, if we forced it to have 11 taxa at about 20 time units before the present.

```
ltt.null(c(11, length(plethodon)+1), c(-20,0)+max(plethodon), lam.pleth, mu.pleth)
segments(-plethodon, ntaxa.pleth, -plethodon, ntaxa.pleth-1)
segments(-plethodon, ntaxa.pleth, -c(plethodon[-1],0), ntaxa.pleth)
```

Still looks OK to me. In fact I don't see much difference between these null distributions, except right around the time the second one is forced to have 11 taxa.

Decreasing Birth Rate or Increasing Death Rate

So, we looked at a couple of trees that seemed to have an excessive number of early speciation events relative to the number of speciations we saw later. This can not be explained by an ordinary birth-death process. This patten is often seen in large phylogenies and has been suggested to be indicative of adaptive radiations. In other words this pattern might be explained by high rates of speciation early and lower rates of speciation later after all the niches are filled. However, it could also

be explained by an increase in the extinction rate.

Let's see if our intuition about these two models is right. We'll make up a pair of arbitrary models. Let's say we want to end up with 60 species and that we will have two time periods; we will make the first one 1 time unit and the second one 9. We will also say that the diversification rate, *r*, is ten times greater in the first period (r_1), than it is in the second (r_2). The expected number of taxa can be calculated as $N=2*exp(r_1t_1)*exp(r_2t_2)$, or to fill in our assumptions, $60=2*exp(1*10r_2)*exp(9r_2)$. We can then solve for $r_2=log(30)/19=0.179$, and $r_1=1.79$.

Now we'll look at what happens to our distributions if we increase μ or decrease λ. Let's say that $\mu_1=0.05$, so that $\lambda_1=1.84$. Then in our first model with decreasing λ, we'll keep m the same and make $\lambda_2=.229$, and for the second model we'll keep λ the same and make $\mu_2=1.661$. We'll compare both of them to a Yule null distribution, in which $\lambda=log(30)/10=0.34$.

```
ltt.null(60, 10, 0.34, 0)
model.lambda <- mrbdtree.n(10, 60, c(1,10), c(1.84,0.229), 0.05)
lapply(model.lambda, ltt.lines) -> x
```

That looks pretty good, the lineages increase early, and then come back to the expected curve. Maybe if the two time periods were closer in length, we would get a slightly smoother curve. Now let's see what happens if we increase the extinction rate:

```
ltt.null(60, 10, 0.34, 0)
model.mu <- mrbdtree.n(10, 60, c(1,10), 1.84, c(0.05, 1.661))
lapply(model.mu, ltt.lines) -> x
```

Huh, that didn't work at all. This implies that the pattern we saw earlier (for example in the Agamid tree) in which you get an early increase in the number of lineages is in fact caused by a decrease in the speciation rate. We were unable to produce a concave lineage-through-time plot by increasing the extinction rate, although it may be possible for another set of parameters.

Another possible explanation for this type of lineage distribution is sampling. We would expect random sampling to reduce the number of recent speciations more than the number of old speciations, as older clades would tend to be larger and thus lest likely to be sampled out. This problem is made even worse by cryptic species, or by sampling that focuses on higher taxa. Let's do a quick test of this by simulating 80 species trees and randomly taking out 20 species.

```
ltt.null(60, 10, 0.34, 0)
make.samp <- function () prune.random.taxa(rbdtree.n(100, 10, 0.3912, 0), 40)
x <- replicate(10, ltt.lines(make.samp()))
```

Well, that didn't seem to do much of anything.

Likelihood Models for Detecting Switches in Diversification

There are tests in the *laser* package that allow us to test for changes in diversification rates using maximum likelihood. The first set of models compare a Yule model in which the diversification rate is constant throughout the entire history of the clade to one in which there are one to several points at which the diversification rate shifted (Rabosky (2006). *Evolution* 60:1152-1164). Let's test one of our simulated trees with the high early speciation rate.

First, we test the model with one rate. All these models are fit to branching times, not whole trees, so first we have to derive a vector of branching times.

times.lambda <- branching.times(model.lambda[[1]])
pureBirth(times.lambda)

That seems pretty reasonable. λ is a little lower than we might expect, but that's probably a consequence of the curve's concavity. Now we'll try a model with two rates.

yule2rate(times.lambda)

Let's look at the output from this model. *LH* is the log likelihood. *st1* is the time at which the diversification process switched. How does that compare to the actual time we made the switch at? r*1* and *r2* are the diversification rates during each of those time periods. *AIC* is the Akaike's information criterion, this is another way of comparing likelihoods that we have not discussed; it can work, even when models are not nested. The AIC is defined as *2k-2log(likelihood)*, and one is supposed to pick the model with the lowest AIC. Which model is best by this criterion? It is also possible to decide between models using the Likelihood Ratio Test.

If you picked the two rate model as best, then you can try models with more rates using *yule3rate, yule4rate,* or *yule5rate*. You can also use the function *fitdAICrc* to compare multiple models at once, although I fail to see the advantage of this function to comparing them one at a time.

Question 2. What is the best model for the Warbler tree? What are the parameters for that model?

Simulating Trees with Continuous Changes in Diversification Rates

In Rabosky and Lovette 2008. (Evolution 62-8:1866-1875) the authors suggest a model of lineage diversification in which the diversification rate decreases exponentially. They propose three models: $\lambda(t)=\lambda_0 exp(-kt)$ and μ is constant for SPVAR, λ is constant and $\mu(t)=\mu_0(1-exp(-zt))$ for EXVAR and both rates are variable for BOTHVAR. We can simulate these models by estimating the rates at hundreds of different points and using those to simulate the data.

For the SPVAR model the expected number of taxa $N=2*exp(\lambda_0[1-exp(-kt)]/k-\mu t)$. So, if we want our process to run for 10 time units, produce 60 species, have the extinction rate equal 0.05, and have the speciation rate be ten times as big at the start as it is at the end, then $exp(-10k)=0.1$ and $N=2\lambda_0 exp(-10k)/k-10\mu)$, so that $k=0.23$ and $\lambda_0=0.997$. Therefore if we break the time from 0 to 10 up into 100 intervals, we get the average λ for any period as $\lambda_0 exp(-kt)[exp(-k\Delta t)-1]/k\Delta t=1.009exp(-0.23t)$. We should create a vector of our average λs.

lambdas <- 1.009*exp(-0.23*1:100/10)

Then we can make some trees and plot them:

trees.spvar <- mrbdtree.n(10, 60, 1:100/10, lambdas, 0.05)
ltt.null(60, 10, 0.34, 0)
lapply(trees.spvar, ltt.lines) -> x

Those look a lot like the real trees we are trying to explain with a high diversification rate early. We can create the opposite model in which μ increases by assuming *r* has the same value at every interval, but that λ is constant. It can easily be shown that this can be achieved when $\lambda_2=\lambda_0-\mu_1$ and $\mu_2(t)=\lambda_0-\lambda_1(t)$. We can make these trees and compare them to our SPVAR trees.

mus <- 0.997-lambdas
trees.exvar <- mrbdtree.n(10, 60, 1:100/10, 0.947, mus)
lapply(trees.exvar,ltt.lines,col=3)->x

Question 3. What do trees simulated under EXVAR look like in general? Do they look like the trees we are trying to explain with a high diversification rate early? What does that say about the processes that could have produced these trees?

Fitting Models of Continuous Change Using Likelihood

If you noticed that the author of the last paper is the author of the *laser* package (and, now, a postdoc in the Huelsenbeck lab, soon to be faculty at Michigan – NJM), you may not be surprised to learn that *laser* has several functions for fitting continuous models of diversification change. This package has functions for fitting SPVAR, EXVAR and BOTHVAR, which I explained in the last section, as well as models in which the diversification rate declines as the number of species increases as in population growth. We will just touch on these briefly here, but if you are interested, you can explore these options further.

We'll fit SPVAR to one of the trees we simulated under SPVAR, and see how it does. First test this tree under the null hypothesis of constant diversification rates.

times.spvar <- branching.times(trees.spvar[[1]])
bd(times.spvar)

Then check under the SPVAR model.

fitSPVAR(times.spvar)

How did it do at estimating the parameters? Which model did it prefer? We won't bother with EXVAR or BOTHVAR right now, but you get the idea how to use them. We will briefly check out one of the density dependent diversification models.

DDX(times.spvar)

Does that model look like a good fit? You should check out the *laser* help files/manual if you want to understand the parameters of this model. As a matter of fact, you should look at the help file/ manual for any function that you want to use in *R*. I have only provided you with a small fraction of the capabilities for many of these functions, and you should see what other options are available.

References

Rabosky, D. L. (2006). "Likelihood methods for detecting temporal shifts in diversification rates." *Evolution* **60**(6): 1152-1164. http://dx.doi.org/10.1111/j.0014-3820.2006.tb01194.x
Link: http://onlinelibrary.wiley.com/doi/10.1111/j.0014-3820.2006.tb01194.x/abstract

Rabosky, D. L. (2006). "LASER: a maximum likelihood toolkit for detecting temporal shifts in diversification rates from molecular phylogenies." *Evolutionary Bioinformatics Online* **2**: 247-250
Link: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2674670/?tool=pubmed

Rabosky, D. L. and I. J. Lovette (2008). "Explosive evolutionary radiations: decreasing speciation or increasing extinction through time?" *Evolution* **62**(8): 1866-1875. http://dx.doi.org/10.1111/j.1558-5646.2008.00409.x
Link: http://onlinelibrary.wiley.com/doi/10.1111/j.1558-5646.2008.00409.x/full

Harmon, L. J., J. T. Weir, et al. (2008). "GEIGER: investigating evolutionary radiations." *Bioinformatics* **24**(1): 129-131. http://dx.doi.org/10.1093/bioinformatics/btm538
Link: http://bioinformatics.oxfordjournals.org/content/24/1/129.abstract