

Untitled

Heath Blackmon

June 11, 2016

Stochastic mapping is a popular tool that can help us to understand the evolution of discrete traits on phylogenies. While most ancestral state reconstruction approaches only provide states or probabilities at each node in the tree stochastic mapping provides us with actual histories of our traits evolution with states at every point along the tree (not just the nodes). This approach was first developed by Nielson (2002). However, most citations you see are to the paper that came out a year later (Huelsenbeck et al. 2003). This paper is bit more accessible and articulates the method quite nicely.

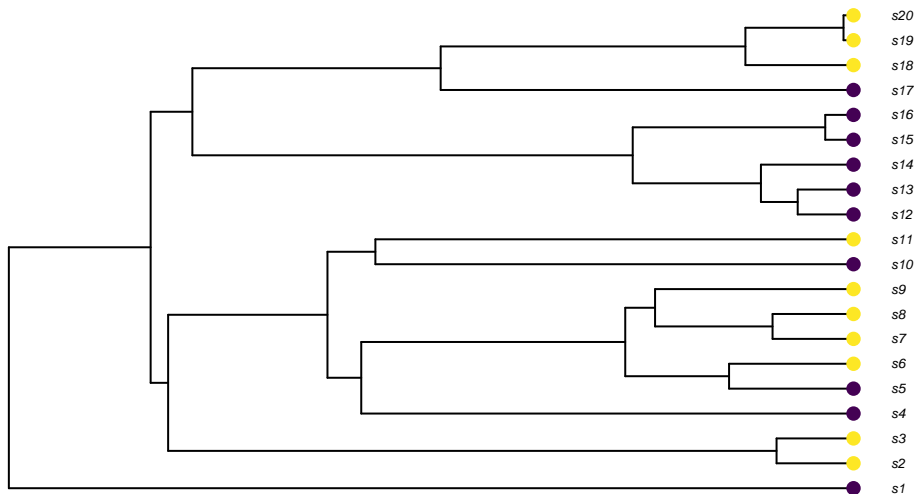
To work out an example of this lets go ahead and first simulate a small dataset:

```
library(geiger)

## Loading required package: ape

## Warning: package 'ape' was built under R version 3.2.5

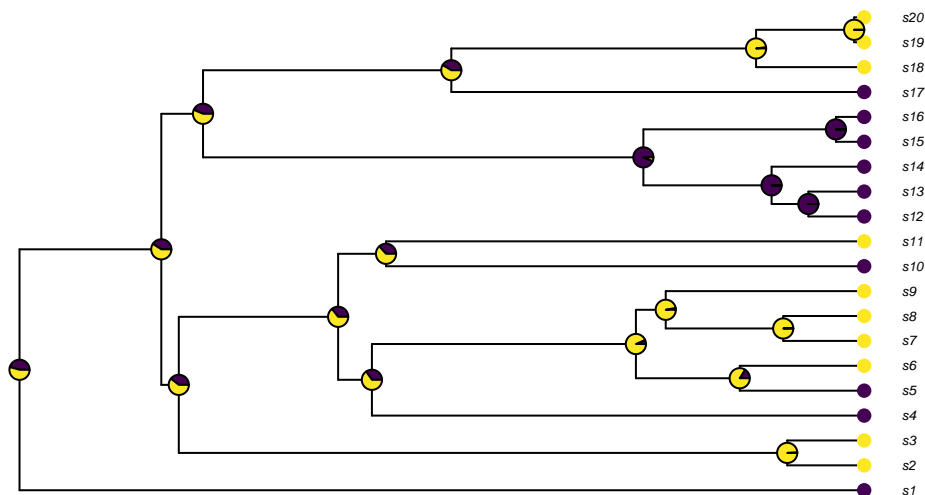
tree <- sim.bdtree(b=0.1, d=0, stop="taxa", n=20, seed=1)
par <- rbind(c(-.1, .1), c(.1, -.1))
set.seed(1)
data <- sim.char(tree, par, model = "discrete")
plot(tree, show.tip.label = T, cex=.4, label.offset=1)
library(viridis)
tiplabels(col=viridis(2)[data[,1]], pch=16)
```



Now with these observed tip states we need to calculate probabilities for the nodes in the tree.

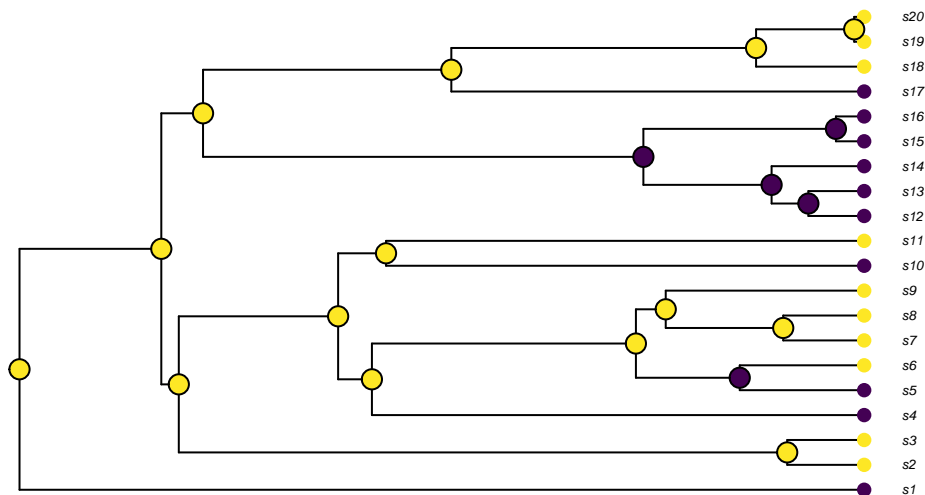
Steps in stochastic mapping 1) calculate the probabilities of each character state at each node in the tree.

```
asr <- ace(x=data[,1], phy=tree, type="discrete", model= "ARD",)
plot(tree, show.tip.label = T, cex=.4, label.offset=1)
tiplabels(col=viridis(2)[data[,1]], pch=16)
nodelabels(piecol=viridis(2), pie=asr[[5]], cex=.5)
```



- 2) Next we must fix each node of the tree into one of the two states. This is accomplished by working our way from the root of the tree to the tips. This yields a tree that now looks like this.

```
plot(tree, show.tip.label = T, cex=.4, label.offset=1)
library(viridis)
tiplabels(col=viridis(2)[data[,1]], pch=16)
nodelabels(piecol=viridis(2), pie=c(0,0,0,0,0,0,0,0,1,0,0,
0,0,1,1,1,1,0,0,0), cex=.5)
```



- 3) we now have to work our way between each of these nodes simulating the history along each branch. The parameters of Q-matrix that we estimated during the ancestral state reconstruction provide us with all the information that we need to do this. Specifically the diagonal elements describe the distribution of expected waiting times until next change is expected to occur. We use this value $T = \frac{-1}{\lambda \ln(1-U)}$ where λ is the diagonal element corresponding to the state at the start of the branch and U is a draw from a uniform distribution on the interval (0,1). If T is greater than the branch length then there is no character change on that branch and we move on to the next branch. If T is less than the branch length then we choose the type of change based on the transition probabilities from our model and repeat the above process for the branch length remaining.

In the end this generates a history like this:

```
library(phytools)
```

```
## Loading required package: maps
```

```
##
## # maps v3.1: updated 'world': all lakes moved to separate new #
## # 'lakes' database. Type '?world' or 'news(package="maps")'. #
x <- make.simmap(tree, x=data[,1])

## make.simmap is sampling character histories conditioned on the transition matrix
##
## Q =
##      1      2
## 1 -0.0556337  0.0556337
## 2  0.0556337 -0.0556337
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##    1    2
## 0.5 0.5
## Done.
cols<-setNames(viridis(2),1:2)
plotSimmap(x, colors=cols, lwd=3)
```

