

DiscreteCharacters

Brian O'Meara

07 February, 2018

For this exercise, explore looking at discrete character models. Note: they are not “discreet” character models – they’re actually pretty noisy.

Remember for homework,

`git remote -v` gives list of remotes

`git fetch upstream`

`git pull upstream master` pulls from upstream (me).

change `eval=TRUE` for all the R blocks, debug, commit, and do a pull request.

These are useful packages in this area (far from exhaustive list).

```
library(yearn)
yearn::yearn(ape)
yearn::yearn(geiger)
yearn::yearn(phytools)
yearn::yearn(phangorn)
yearn::yearn(corHMM)
```

You’ll need to get data into R in some way.

```
tree <- read.tree("___PATH_TO_TREE_OR_SOME_OTHER_WAY_OF_GETTING_A_TREE___")
discrete.data <- read.csv(file="___PATH_TO_DATA_OR_SOME_OTHER_WAY_OF_GETTING_TRAITS___", stringsAsFac
```

Data are often not right in some way. They might not match the taxa in your tree, there may be missing data, etc. `geiger::treedata` is a great function for getting a tree and data that match, but your data may need other cleaning. Do it as a function so it’s repeatable.

```
CleanData <- function(phy, data) {
  #treedata() in Geiger is probably my favorite function in R.
}
```

Now write the code to use CleanData() to actually clean your data

It’s critically important to LOOK at what you have. Are there weird values? Has the match between taxa and state gone correctly? Do you think you have binary data, but there’s actually only state 1? Especially as data sets grow (yay), and are assembled using scripts rather than by error-prone, non-reproducible hands (double yay), scientists are increasingly less likely to deeply look at our data. That’s bad – don’t be that person.

```
VisualizeData <- function(phy, data) {
  #Important here is to LOOK at your data before running it. Any weird values? Does it all make sense

  # Now write the code to use VisualizeData() to actually look at your data
}
```

First, let’s use parsimony to look at ancestral states:

```
cleaned.discrete.phyDat <- phangorn::phyDat(cleaned.discrete, type="-----") #phyDat is a data
anc.p <- phangorn::ancestral.pars(tree, cleaned.discrete.phyDat)
plotAnc(tree, anc.p, 1)
```

Do you see uncertainty? What does it mean?

Now, plot the likelihood estimates.

```
anc.ml <- ancestral.pml(pml(tree, cleaned.discrete.phyDat), type="ml")
plotAnc(tree, anc.ml, 1)
```

How does this differ from parsimony?

Why does it differ from parsimony?

What does uncertainty mean?

Now, to the biological questions. For many of these, **corHMM** will be a useful package.

1. How can you estimate transition rates between states? Do it.
2. How could you examine if transition rates are equal?
3. Think about the Lewis (2001) MKV model. Are your traits all variable? Will using this make sense for your data? Try using it. Do results change?
4. How could you test order of state evolution?