Ancestral_reconstruction_final

```
library(phytools)
```

```
## Loading required package: ape
```

```
## Warning: package 'ape' was built under R version 3.5.3
```

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

```
library(geiger)

data <- read.csv("review_diapause_all_species.csv", row.names = 1)
tree <- read.tree("tree_all_species.phy.phy")

treeace = multi2di(tree)
treeace$edge.length[treeace$edge.length==0]<-max(nodeHeights(treeace))*1e-6

diapause <- setNames(data[,2],rownames(data))
diapause <- as.factor(diapause)
#cols<-setNames(c("red","blue","yellow","black"),levels(diapause))
```

I used AIC and likelihood to compare the different models to run ace. Highest likelihood and lowest AIC was considered the best model.

```
fitERt<-fitDiscrete(treeace,diapause,model="ER", type="discrete")
fitERt
```

```
## GEIGER-fitted comparative model of discrete data
##  fitted Q matrix:
##                  1            2            3            4            5
##     1 -0.004133914  0.001033478  0.001033478  0.001033478  0.001033478
##     2  0.001033478 -0.004133914  0.001033478  0.001033478  0.001033478
##     3  0.001033478  0.001033478 -0.004133914  0.001033478  0.001033478
##     4  0.001033478  0.001033478  0.001033478 -0.004133914  0.001033478
##     5  0.001033478  0.001033478  0.001033478  0.001033478 -0.004133914
##
##  model summary:
##  log-likelihood = -148.942801
##  AIC = 299.885603
##  AICc = 299.910919
##  free parameters = 1
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  frequency of best fit = 1.00
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
fitARDt<-fitDiscrete(treeace,diapause,model="ARD", type="discrete")
fitARDt
```

```
## GEIGER-fitted comparative model of discrete data
##  fitted Q matrix:
##                    1             2             3             4
##     1 -2.381535e-03  6.864555e-12  1.928153e-04  1.293879e-03
##     2  5.953718e-04 -2.983428e-03  1.240881e-11  1.804988e-03
##     3  1.346021e-11  1.476574e-09 -7.636308e-03  7.636306e-03
##     4  5.826644e-08  1.410374e-02  2.876406e-09 -1.692405e-02
##     5  2.210362e-02  1.806007e-02  5.058207e-12  6.526436e-12
##                    5
##     1  8.948408e-04
##     2  5.830674e-04
##     3  1.004814e-11
##     4  2.820252e-03
##     5 -4.016369e-02
##
##  model summary:
##  log-likelihood = -115.424996
##  AIC = 270.849993
##  AICc = 276.893158
##  free parameters = 20
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  frequency of best fit = 0.02
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
fitSYMt<-fitDiscrete(treeace,diapause,model="SYM", type="discrete")
fitSYMt
```

```
## GEIGER-fitted comparative model of discrete data
##  fitted Q matrix:
##                    1             2             3             4
##     1 -3.515672e-03  2.176075e-03  1.783639e-17  0.0007424942
##     2  2.176075e-03 -6.332385e-03  6.469926e-26  0.0041563094
##     3  1.783639e-17  6.469926e-26 -5.739415e-03  0.0057394153
##     4  7.424942e-04  4.156309e-03  5.739415e-03 -0.0131415306
##     5  5.971024e-04  2.553079e-17  5.591208e-19  0.0025033118
##                    5
##     1  5.971024e-04
##     2  2.553079e-17
##     3  5.591208e-19
##     4  2.503312e-03
##     5 -3.100414e-03
##
##  model summary:
##  log-likelihood = -133.078256
##  AIC = 286.156512
##  AICc = 287.633022
##  free parameters = 10
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  frequency of best fit = 0.01
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

Best model ARD.

Bayesian approach for ancestral reconstruction. make.simmap function - “performs stochastic mapping using several methods.

For Q=“empirical”, it first fits a continuous-time reversible Markov model for the evolution of x and then simulates stochastic character histories using that model and the tip states on the tree. This is the same procedure that is described in Bollback (2006), except that simulation is performed using a fixed value of the transition matrix, Q, instead of by sampling Q from its posterior distribution.

For Q=“mcmc”, it first samples Q nsim times from the posterior probability distribution of Q using MCMC, then it simulates nsim stochastic maps conditioned on each sampled value of Q."

ARD - Q=empirical

```
mtree <- make.simmap(tree,diapause,model="ARD", Q="empirical", pi="estimated")
```

```
## Using pi estimated from the stationary distribution of Q assuming a flat prior.
## pi =
##        1        2        3        4        5 
## 0.342284 0.539783 0.008672 0.088056 0.021205 
## 
## make.simmap is sampling character histories conditioned on the transition matrix
## 
## Q =
##              1            2             3            4             5
## 1 -0.002392278  0.000000000  0.0001933054  0.001299446  0.0008995265
## 2  0.000625409 -0.002967009  0.0000000000  0.001810166  0.0005314336
## 3  0.000000000  0.000000000 -0.0076423098  0.007642310  0.0000000000
## 4  0.000000000  0.013916033  0.0000000000 -0.016923046  0.0030070129
## 5  0.022347890  0.017966745  0.0000000000  0.000000000 -0.0403146350
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##           1           2           3           4           5 
## 0.340550138 0.541568925 0.008613909 0.087967997 0.021299031
```

```
## Done.
```

```
mtrees<-make.simmap(tree,diapause,model="ARD",nsim=100, Q="empirical", pi="estimated")
```

```
## Using pi estimated from the stationary distribution of Q assuming a flat prior.
## pi =
##        1        2        3        4        5 
## 0.342068 0.539935 0.008667 0.088022 0.021307 
## 
## make.simmap is sampling character histories conditioned on the transition matrix
## 
## Q =
##              1            2             3            4             5
## 1 -0.002392288  0.000000000  0.0001933048  0.001299433  0.0008995503
## 2  0.000625411 -0.002967023  0.0000000000  0.001810173  0.0005314391
## 3  0.000000000  0.000000000 -0.0076424334  0.007642433  0.0000000000
## 4  0.000000000  0.013916162  0.0000000000 -0.016923099  0.0030069372
## 5  0.022348169  0.017966439  0.0000000000  0.000000000 -0.0403146081
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##           1           2           3           4           5
## 0.340552374 0.541567164 0.008613661 0.087967631 0.021299170
```

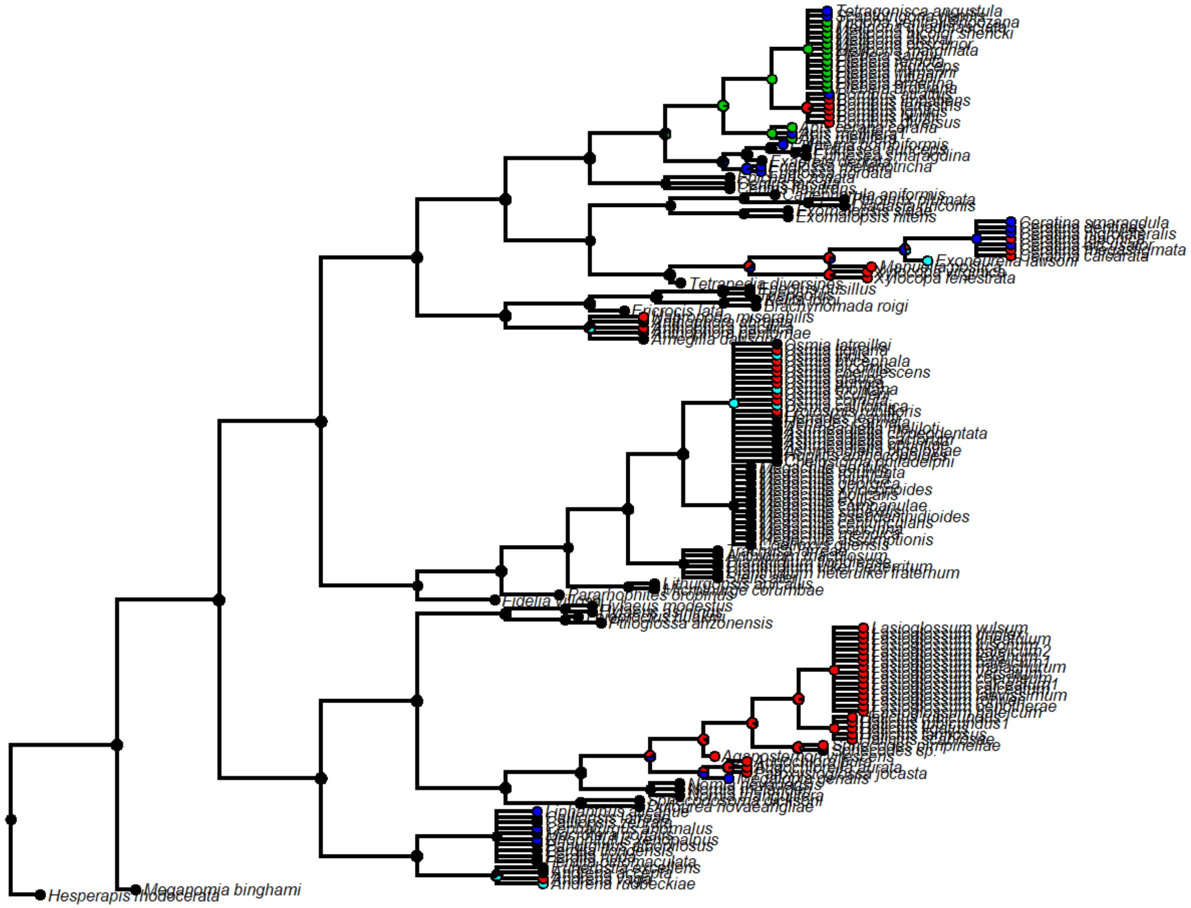
```
## Done.
```

```
pd<-summary(mtrees,plot=FALSE)
pd
```

```
## 100 trees with a mapped discrete character with states:
##  1, 2, 3, 4, 5
##
## trees have 65.84 changes between states on average
##
## changes are of the following types:
##      1,2  1,3  1,4  1,5  2,1 2,3  2,4  2,5 3,1 3,2  3,4 3,5 4,1  4,2 4,3
## x->y   0 1.32 8.68 5.57 1.23   0 3.74 0.85   0   0 4.53   0   0 9.98   0
##      4,5   5,1   5,2 5,3 5,4
## x->y 2.1 15.56 12.28   0   0
##
## mean total time spent in each state is:
##                 1            2            3            4            5
## raw  6561.8277981 1964.7814621 591.35329979 705.00979930 691.02764072
## prop    0.6241038    0.1868729   0.05624437   0.06705438   0.06572452
##      total
## raw  10514
## prop     1
```

```
plot(pd,fsize=0.5,ftype="i", cex=0.20)
cols<-setNames(c("black","red","green","blue","cyan"),levels(diapause))
```

```
add.simmap.legend(leg=c("Development","Adult","Reproductive","NoDiapause","Plastic"), colors=cols,    prompt=FALS
E,x=0.3*par()$usr[1],y=-max(nodeHeights(tree)),fsize=0.7)
```



```
dev.off()
```

```
## null device
##           1
```