

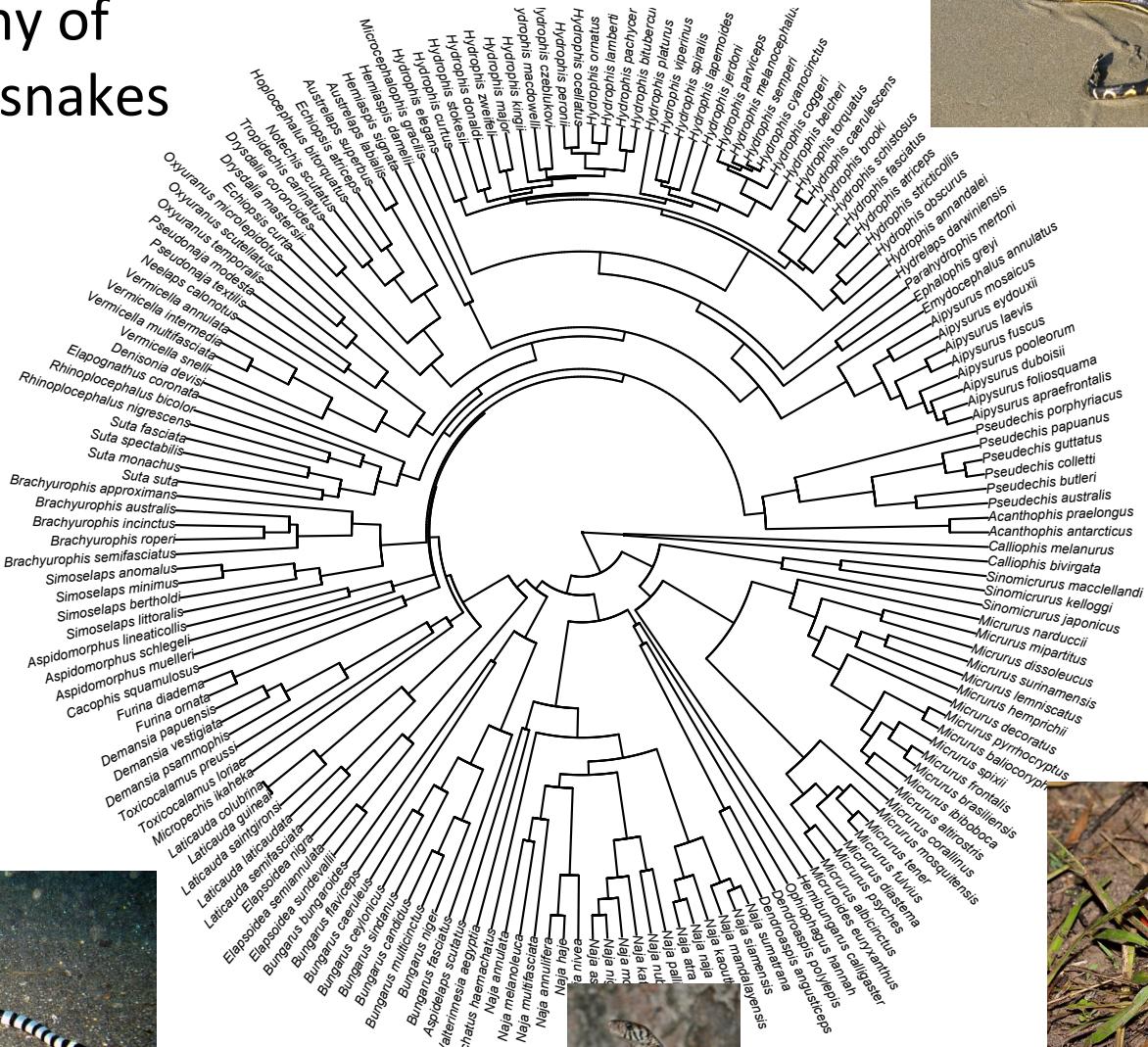
R workshop

- Does adaptation to the marine environment affect the speciation rate of elapid snakes?



- This will take place in R or R studio
- Save the BiSSE_elapids folder and set working directory

Input – a molecular phylogeny of venomous snakes



```
install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
```

Load packages and data

- BiSSE requires a nexus tree and the character data for each species
- The last two lines of code are simply to get the habitat data in the correct format

```
install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
Tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
```

Set up a BiSSE model

- The first line sets up a full 6 parameter BiSSE model but does not run it
- Starting point provides initial parameter values for the likelihood search
- find.mle initiates a run to find the maximum likelihood estimate for the parameters. This will take a few seconds to run

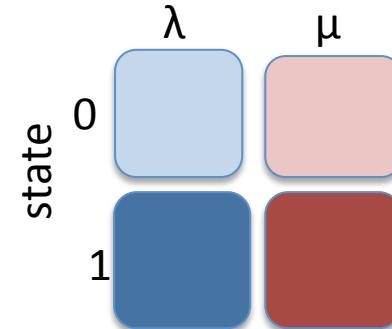
```

install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
Tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
Start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
bisse5 <- constrain(bisse6, mu1 ~ mu0)
bisse4 <- constrain(bisse6, lambda1 ~ lambda0, mu1 ~ mu0)
ml5 <- find.mle(bisse5, start)
ml4 <- find.mle(bisse4, start)

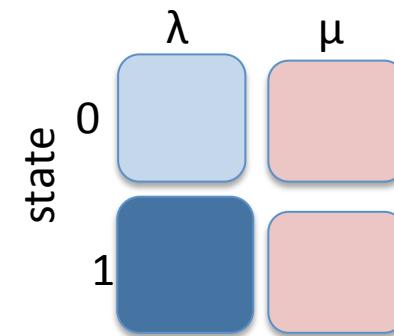
```

Set up two constrained models

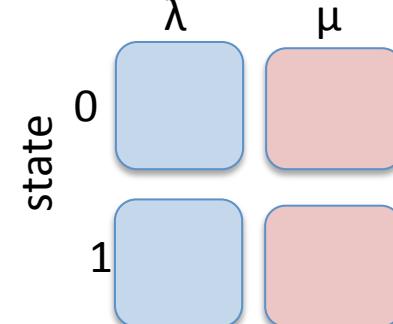
**6 parameter
(full BiSSE model)**



**5 parameter model
(equal extinction)**



**4 parameter model
(equal speciation and extinction)**



```
install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
Tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
Start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
bisse5 <- constrain(bisse6, mu1 ~ mu0)
bisse4 <- constrain(bisse6, lambda1 ~ lambda0, mu1 ~ mu0)
ml5 <- find.mle(bisse5, start)
ml4 <- find.mle(bisse4, start)
ml6$lnLik
ml5$lnLik
ml4$lnLik
```

Now have a look at the likelihood value of the results

- The maximum likelihood for the 5 parameter model is lower than for the 4 parameter model
- Why is that?

Try alternative starting parameters

- Here we use the parameters from the full run as the starting point for another set of runs
- Have the answers changed?
- If we wanted to publish, we should probably try more starting values. But for now we move on

```
install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
Tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
Start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
bisse5 <- constrain(bisse6, mu1 ~ mu0)
bisse4 <- constrain(bisse6, lambda1 ~ lambda0, mu1 ~ mu0)
ml5 <- find.mle(bisse5, start)
ml4 <- find.mle(bisse4, start)
ml6$lnLik
ml5$lnLik
ml4$lnLik
start <- ml6$par
ml6.1 <- find.mle(bisse6, start)
ml5.1 <- find.mle(bisse5, start)
ml4.1 <- find.mle(bisse4, start)
ml6.1$lnLik
ml5.1$lnLik
ml4.1$lnLik
```

```
install.packages("diversitree")
install.packages("phangorn")
library(diversitree)
library(phangorn)
Tree <- read.nexus("elapids.tree")
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
Start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
bisse5 <- constrain(bisse6, mu1 ~ mu0)
bisse4 <- constrain(bisse6, lambda1 ~ lambda0, mu1 ~ mu0)
ml5 <- find.mle(bisse5, start)
ml4 <- find.mle(bisse4, start)
ml6$lnLik
ml5$lnLik
ml4$lnLik
start <- ml6$par
ml6.1 <- find.mle(bisse6, start)
ml5.1 <- find.mle(bisse5, start)
ml4.1 <- find.mle(bisse4, start)
ml6.1$lnLik
ml5.1$lnLik
ml4.1$lnLik
```

Model comparison

$$AIC = 2 \times (\text{number of parameters}) - 2 \times \ln(\text{likelihood})$$

Penalty for number of parameters

Raw model fit

(Why ml4 not ml4.1????)

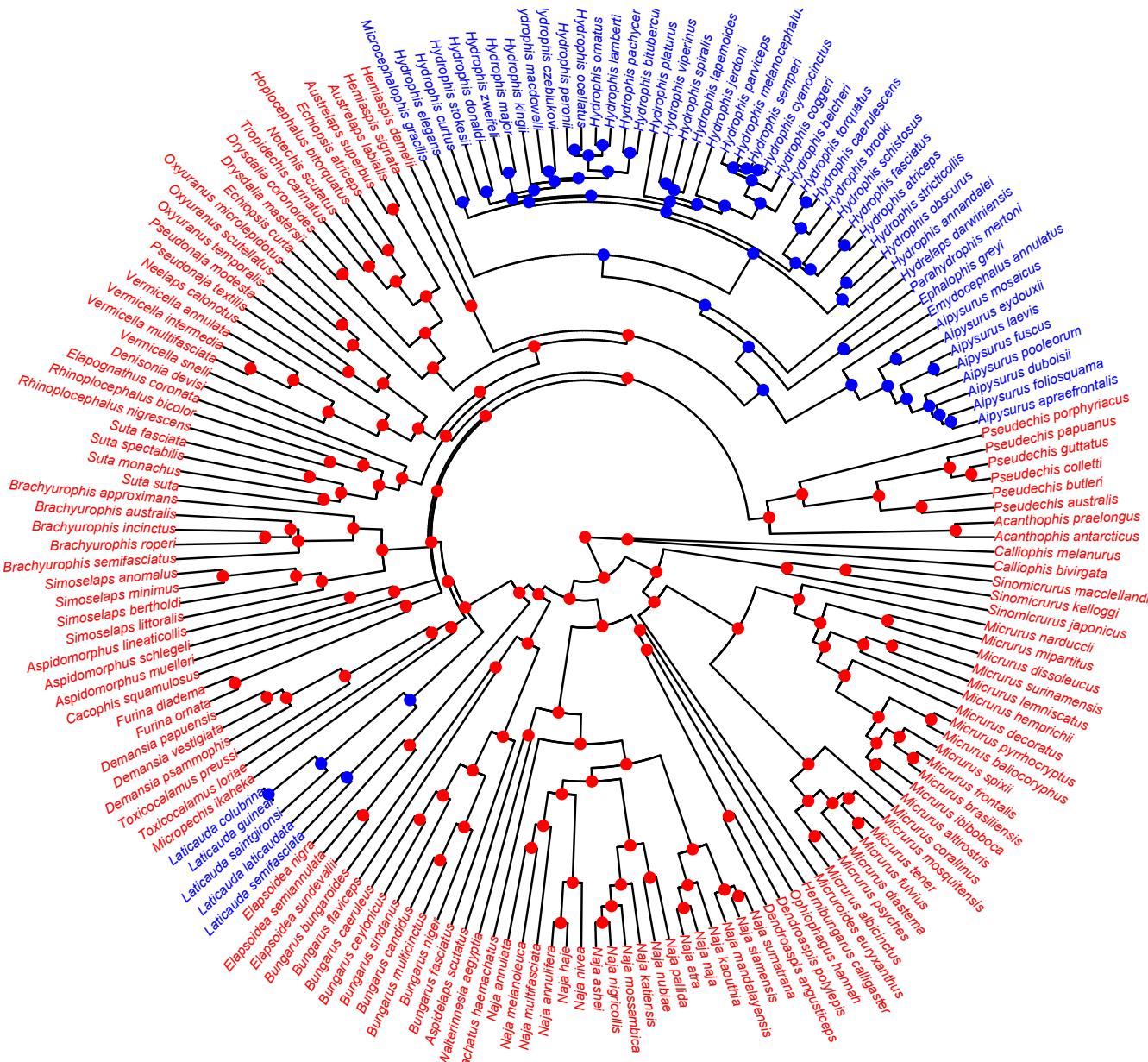
- Model with lowest AIC wins
- Check parameters of best model
 - What should we conclude?
 - More importantly, do you believe the conclusion?

Ancestral state reconstructions from BiSSE

```
df <- read.csv("habitats.csv")
v <- df[,2]
names(v) <- df[,1]
bisse6 <- make.bisse(tree, v)
Start <- starting.point.bisse(tree)
ml6 <- find.mle(bisse6, start)
bisse5 <- constrain(bisse6, mu1 ~ mu0)
bisse4 <- constrain(bisse6, lambda1 ~ lambda0, mu1 ~ mu0)
ml5 <- find.mle(bisse5, start)
ml4 <- find.mle(bisse4, start)
ml6$lnLik
ml5$lnLik
ml4$lnLik
start <- ml6$par
ml6.1 <- find.mle(bisse6, start)
ml5.1 <- find.mle(bisse5, start)
ml4.1 <- find.mle(bisse4, start)
ml6.1$lnLik
ml5.1$lnLik
ml4.1$lnLik
asr <- asr.marginal(bisse5, coef(ml5.1))
plot(tree, cex=0.4, tip.col=rgb(1-v, 0, v), type="fan")
nodelabels(frame="circle", col=rgb(1-asr[2,], 0, asr[2,]), pch=20, cex=1)
source("nicetree.R")
```

- To get a better feel for our data, we can perform ancestral state reconstructions using the BiSSE model

Results



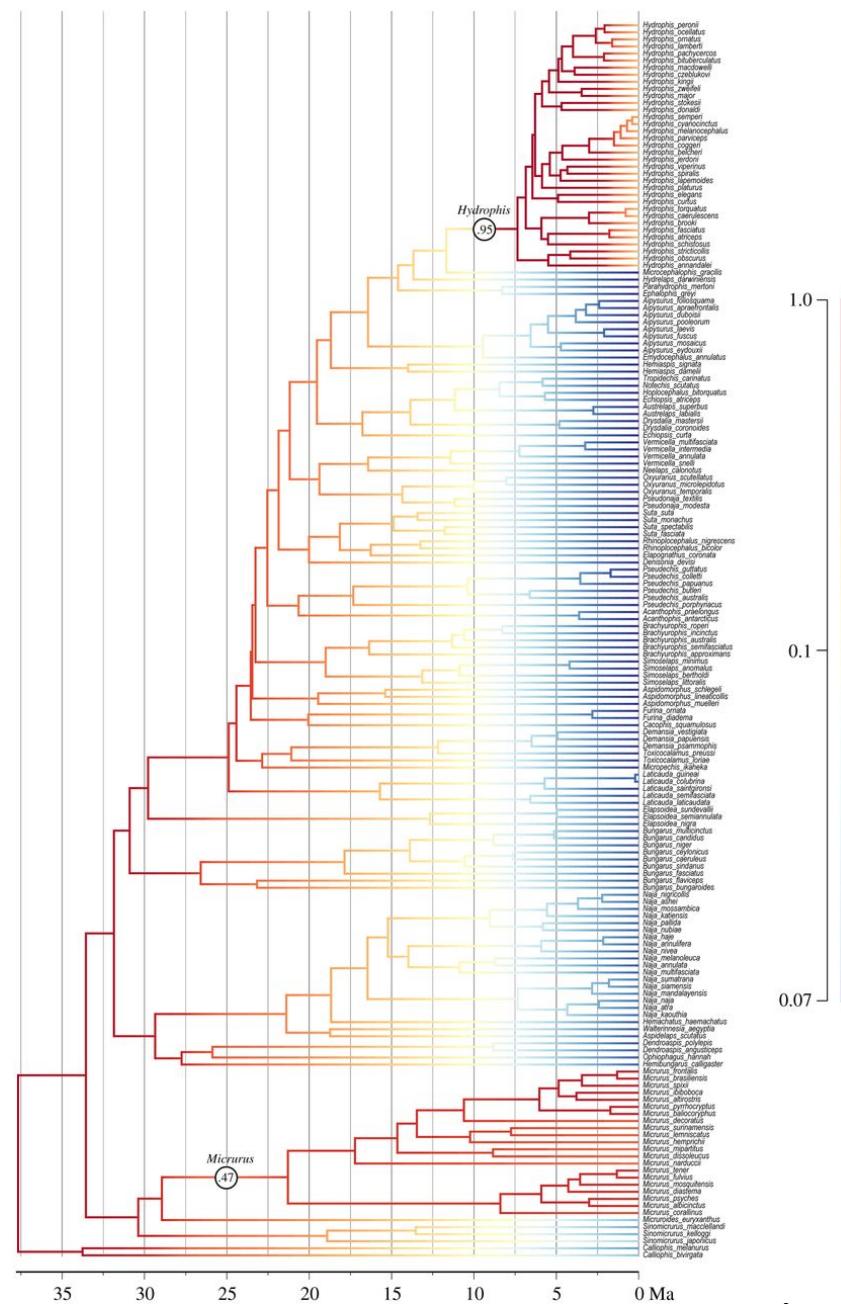
Can you spot any potential problems for our interpretation of the BiSSE output here

Which parts of the tree do you think have a high speciation rate just by looking at it

Let's take a look at speciation rate without considering habitat

(Results from BAMM)

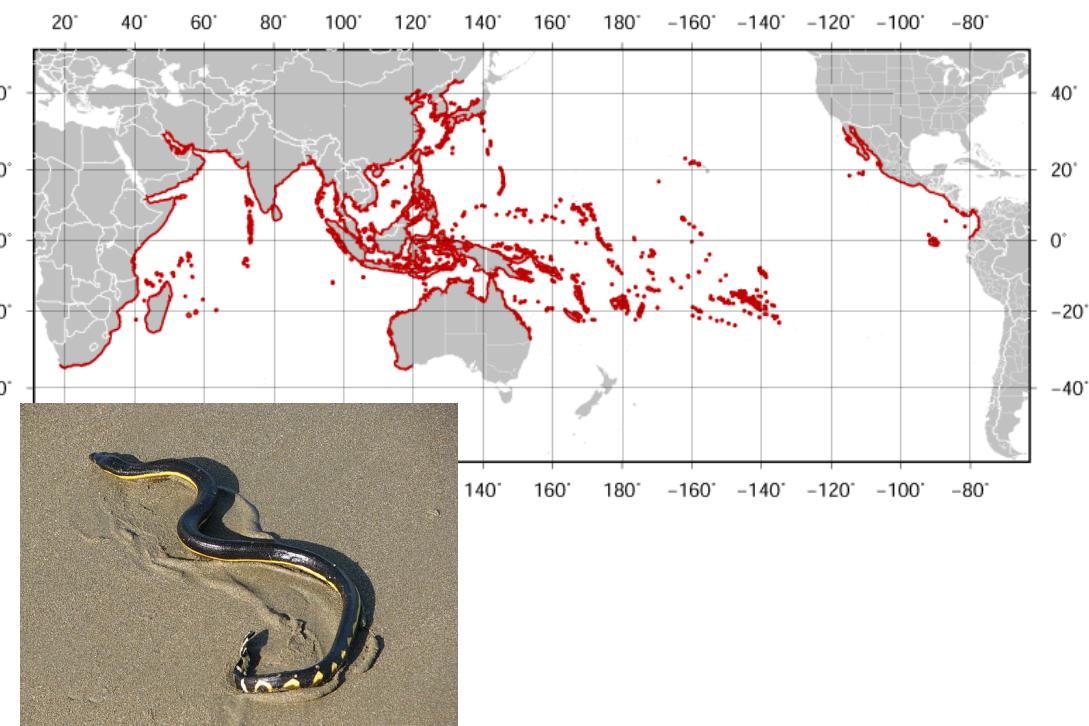
Evidence for rapid speciation in *Hydrophis*, but not other seasnakes



Lee et al. 2016

Elevated rates of speciation in *Hydrophis*

- Elevated rates only in a subgroup of seasnakes - *Hydrophis*
- Sea kraits also entered the marine environment independently. This is an earlier event but led to only 8 species.
- Does this mean the BiSSE results represent a false positive?



Hydrophis

Some members have wide ranges

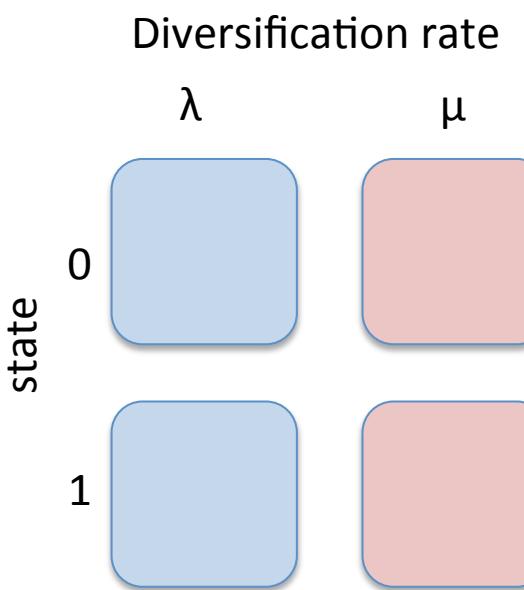


Laticauda

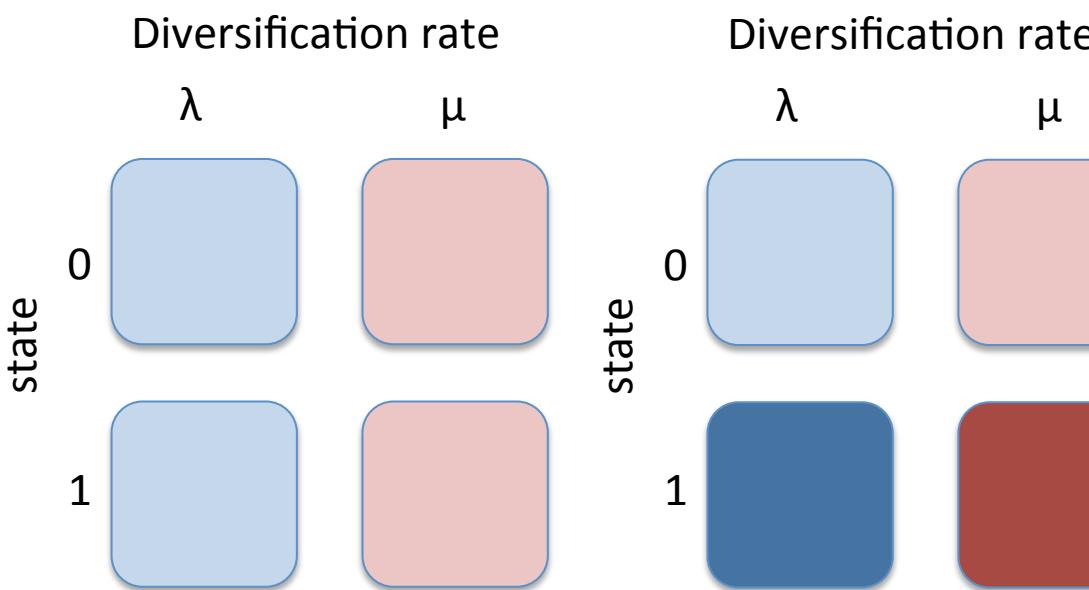
Still lay eggs, less extreme adaptation to the marine environment

Hidden state models can be used to construct appropriate null models for BiSSE

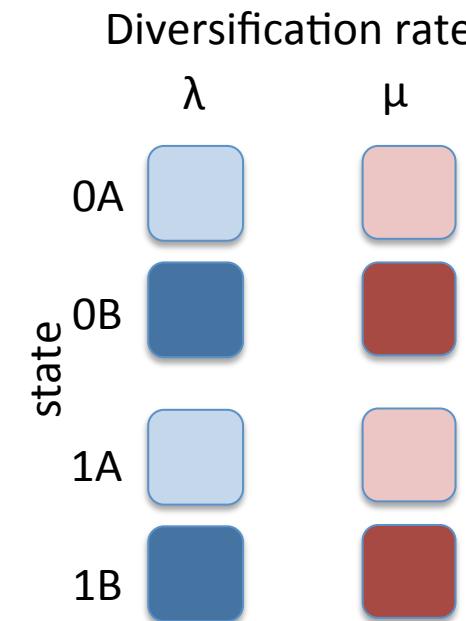
- BiSSE might be supported over a null model without state dependent diversification simply because it contains at least **some** diversification rate heterogeneity
- A better comparison is to a model which contains the same amount of diversification rate heterogeneity, but where this is **not linked to the traits**



Standard null model



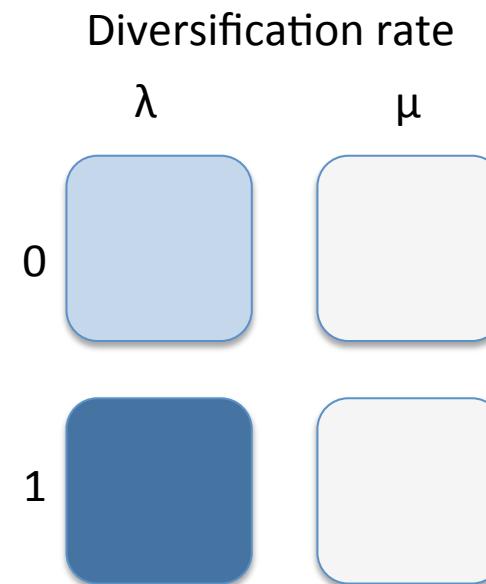
Alternative model (BiSSE)



Null model with hidden states

```
library(hisse)
trans.rates.bisse <- TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
```

Here we set up a very simple 3 parameter BiSSE model with no extinction and equal transition rates

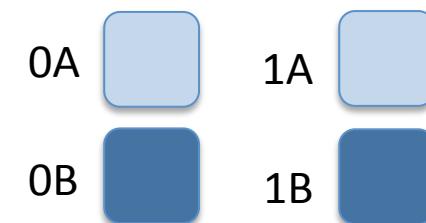


*Note that here the model is parameterised using net diversification and extinction fraction rather than using speciation and extinction directly

```
library(hisse)
trans.rates.bisse <- TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse <- hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                    turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                    trans.rate=trans.rates.bisse.eq, output.type="net.div")
```

Run the model

Have a look at the
parameters and AIC



```
library(hisse)
trans.rates.bisse <- TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse <- hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                    turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                    trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
                                pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
```

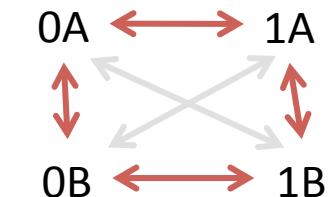
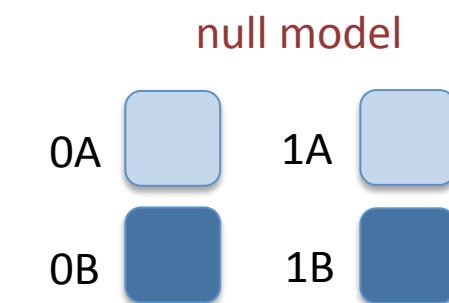
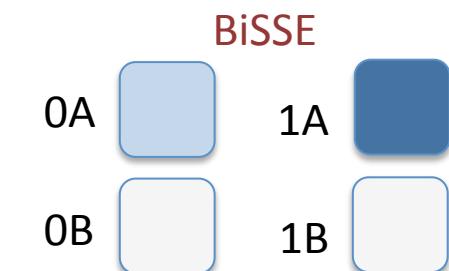
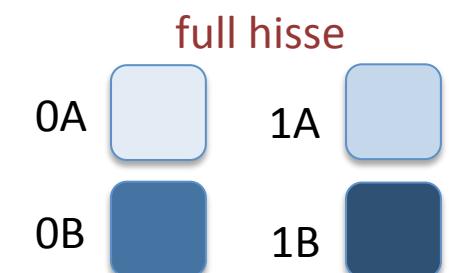
Run an ancestral state reconstruction using this model and plot it

mac users can set n.cores>1

```

library(hisse)
trans.rates.bisse =TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse = hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                  turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                  trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
                                pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
source("trans.rates.null.R")
trans.rates =TransMatMaker(hidden.states=TRUE)
trans.rates.nodual =ParDrop(trans.rates,c(3,5,8,10))
trans.rates.nodual.allequal =
    ParEqual(trans.rates.nodual,c(1,2,1,3,1,4,1,5,1,6,1,7,1,8))
trans.rates.nodual.allequal
turnover.anc.null <- c(?, ?, ?, ?)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div")

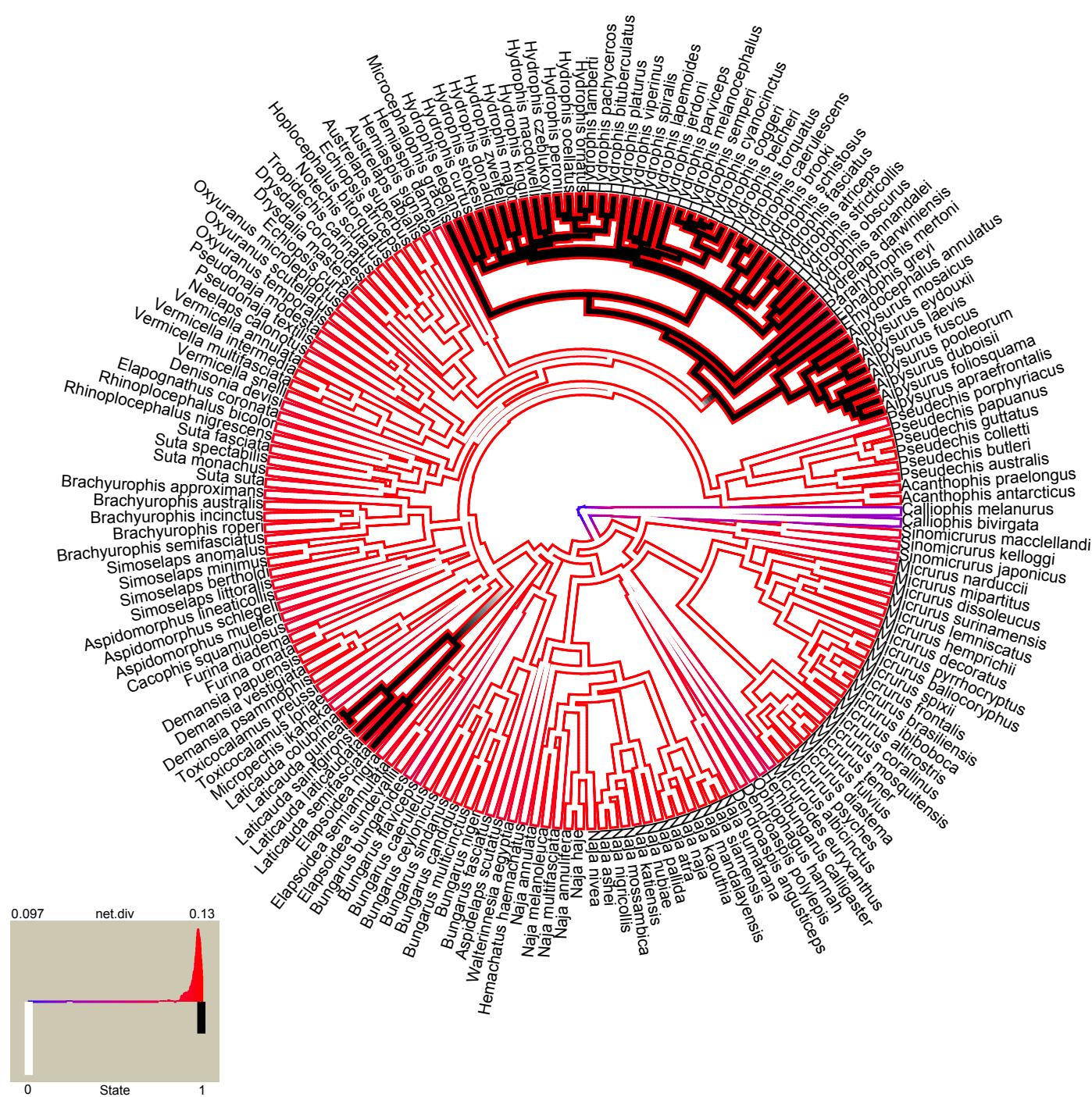
```



```
library(hisse)
trans.rates.bisse =TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse = hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                  turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                  trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
                                pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
source("trans.rates.null.R")
turnover.anc.null <- c(1,1,2,2)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div")
pp.null.recon <- MarginRecon(phy=tree, data=df, f = pp.null$f,
                                pars = pp.null$solution, aic = pp.null$AIC, n.cores=1)
plot(pp.null.recon, fsize=0.5)
```

Run an ancestral state reconstruction using this model and plot it

mac users can set n.cores>1



```
library(hisse)
trans.rates.bisse =TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse <- hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                    turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                    trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
                                pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
source("trans.rates.null.R")
turnover.anc.null <- c(1,1,2,2)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div")
starting.vals <- c(0.01, 0, 0.002)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div",
                  starting.vals=starting.vals)
```

```
trans.rates.bisse =TransMatMaker(hidden.states=FALSE)
trans.rates.bisse
trans.rates.bisse.eq <- ParEqual(trans.rates.bisse, c(1,2))
turnover.anc.bisse <- c(1,2,0,0)
eps.anc <- c(0,0,0,0)
pp.bisse <- hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
                    turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
                    trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
                                pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
source("trans.rates.null.R")
turnover.anc.null <- c(1,1,2,2)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div")
starting.vals <- c(0.01, 0, 0.002)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
                  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
                  trans.rate=trans.rates.nodual.allequal, output.type="net.div",
                  starting.vals=starting.vals)
pp.null.recon <- MarginRecon(phy=tree, data=df, f = pp.null$f,
                               pars = pp.null$solution, aic = pp.null$AIC, n.cores=1)
plot(pp.null.recon, fsize=0.5)
```

```
eps.anc <- c(0,0,0,0)
pp.bisse <- hisse(tree, df, f=c(0.42,0.68), hidden.states=FALSE,
    turnover.anc=turnover.anc.bisse, eps.anc=eps.anc,
    trans.rate=trans.rates.bisse.eq, output.type="net.div")
pp.bisse.recon <- MarginRecon(phy=tree, data=df, f = pp.bisse$f,
    pars = pp.bisse$solution, aic = pp.bisse$AIC, n.cores=1)
plot(pp.bisse.recon, fsize=0.5)
source("trans.rates.null.R")
turnover.anc.null <- c(1,1,2,2)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
    turnover.anc=turnover.anc.null, eps.anc=eps.anc,
    trans.rate=trans.rates.nodual.allequal, output.type="net.div")
starting.vals <- c(0.01, 0, 0.002)
pp.null <- hisse(tree, df, f=c(0.42,0.68), hidden.states=TRUE,
    turnover.anc=turnover.anc.null, eps.anc=eps.anc,
    trans.rate=trans.rates.nodual.allequal, output.type="net.div",
    starting.vals=starting.vals)
pp.null.recon <- MarginRecon(phy=tree, data=df, f = pp.null$f,
    pars = pp.null$solution, aic = pp.null$AIC, n.cores=1)
plot(pp.null.recon, fsize=0.5)
hisse.results.list <- list()
hisse.results.list[[1]] <- pp.bisse.recon
hisse.results.list[[2]] <- pp.null.recon
plot.hisse.states(hisse.results.list, rate.param="net.div", fsize=0.5)
```

Model averaging

We can produce an ancestral state reconstruction that averages over the models, weighted by their AIC scores

FiSSE

```
trans.rates.nodual = ParDrop(trans.rates,c(3,5,8,10))
trans.rates.nodual.allequal =
  ParEqual(trans.rates.nodual,c(1,2,1,3,1,4,1,5,1,6,1,7,1,8))
trans.rates.nodual.allequal
turnover.anc.null <- c(1,1,2,2)
pp.null <- hisse(tree, data, f=c(0.42,0.68), hidden.states=TRUE,
  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
  trans.rate=trans.rates.nodual.allequal, output.type="net.div")
c(0.01, 0, 0.002) -> starting.vals
pp.null <- hisse(tree, data, f=c(0.42,0.68), hidden.states=TRUE,
  turnover.anc=turnover.anc.null, eps.anc=eps.anc,
  trans.rate=trans.rates.nodual.allequal, output.type="net.div",
  starting.vals=starting.vals)
pp.null.recon <- MarginRecon(phy=tree, data=data, f = pp.null$f,
  pars = pp.null$solution, aic = pp.null$AIC, n.cores=1)
plot(pp.null.recon, fsize=0.5)
hisse.results.list = list()
hisse.results.list[[1]] = pp.bisse.recon
hisse.results.list[[2]] = pp.null.recon
plot.hisse.states(hisse.results.list, rate.param="net.div", fsize=0.5)
source("traitDependent_functions.R")
res <- FISSE.binary(tree, v)
pval <- min(res$pval, 1-res$pval)*2
pval
```

- Do the results from this simple test agree with those from BiSSE?
- What do we conclude now about the effect of entering the marine environment on seasnake diversification

Take-home messages

- Implementations of models (e.g. BiSSE in R) are not always perfect, for example do not find maximum likelihood
- BiSSE and other methods can only choose amongst the models available to it
- It is possible that none of the models tested are good
- Critical thinking must be applied to all comparative methods
- Common sense and biology still have a role to play!