

<p>BASIC SET UP</p> <p>1. Describe what is happening in these lines of code</p>	<pre> setOption("useScaling","true") NUM_STATES = 2 NUM_HIDDEN = 2 NUM_RATES = NUM_STATES * NUM_HIDDEN observed_phylogeny <- readTrees("basicdata/poleult.tre")[1] data <- readCharacterDataDelimited("basicdata/pole_datadis.csv", stateLabels=2, type="NaturalNumbers", delimiter=",", headers=TRUE) data_exp <- data.expandCharacters(NUM_HIDDEN) taxa <- observed_phylogeny.taxa() root_age <- observed_phylogeny.rootAge() # set my move index mvi = 0 mni = 0 </pre>
<p>DIVERSIFICATION RATES</p> <p>2. Draw the fixed, stochastic and deterministic nodes determined by the RevBayes code for speciation and extinction</p>	<pre> ### Create the constant prior parameters of the diversification rates ## Number of surviving lineages is 165 mx=(ln(165/2)/observed_phylogeny.rootAge()) sx= 0.05 rate_mean <- exp(mx+sx^2) rate_sd <- sqrt(exp(2*mx+sx^2)*exp(sx^2-1)) for (i in 1:NUM_STATES) { ### Create a lognormal distributed variable for the speciation rate speciation_alpha[i] ~ dnNormal(mean=rate_mean,sd=rate_sd) moves[++mvi] = mvSlide(speciation_alpha[i],delta=0.20,tune=true,weight=3.0) ### Create a lognormal distributed variable for the extinction rate </pre>

```

extinction_alpha[i] ~ dnNormal(mean=rate_mean,sd=rate_sd)
moves[++mvi] =
mvSlide(extinction_alpha[i],delta=0.20,tune=true,weight=3.0)
}

for (i in 1:NUM_HIDDEN) {
### Create an exponential distributed variable for the speciation rate
speciation_beta[i] ~ dnExp(1.0)
moves[++mvi] =
mvScale(speciation_beta[i],lambda=0.20,tune=true,weight=2.0)

### Create an normal distributed variable for the extinction
extinction_beta[i] ~ dnNormal(0.0,1.0)
moves[++mvi] =
mvSlide(extinction_beta[i],delta=0.20,tune=true,weight=2.0)

}

for (j in 1:NUM_HIDDEN) {
for (i in 1:NUM_STATES) {
if ( j == 1) {
speciation[i] := exp( speciation_alpha[i] )
extinction[i] := exp( extinction_alpha[i] )
} else {
index = i+(j*NUM_STATES)-NUM_STATES
speciation[index] := speciation[index-NUM_STATES] * exp(
speciation_beta[j-1] )
extinction[index] := exp( extinction_alpha[i] + extinction_beta[j-1] )
}
}
}
}

```

<p>TRANSITION RATES BETWEEN OBSERVED STATES</p> <p>3. Draw the fixed, stochastic and deterministic nodes determined by the RevBayes code for transition rates between states 0 and 1</p>	<pre> shape_pr := 0.5 rate_pr := 1 #### First create a 4x4 matrix full of zeros for (i in 1:4) { for (j in 1:4) { q[i][j] := 0.0 } } #Key point: The way we define the states in RevBayes is by letter. Therefore, 1=0A, 2=1A, 3=0B, and 4=1B q_01 ~ dnGamma(shape=shape_pr, rate=rate_pr) moves[+mvi] = mvScale(q_01, weight=2) q_10 ~ dnGamma(shape=shape_pr, rate=rate_pr) moves[+mvi] = mvScale(q_10, weight=2) q[1][2] :=q_01 q[3][4] :=q_01 q[2][1] :=q_10 q[3][4] :=q_10 </pre>
<p>TRANSITION RATES BETWEEN HIDDEN STATES</p> <p>4. Draw the fixed, stochastic and deterministic nodes determined by the RevBayes code for transition rates between states A and B</p> <p>#Key point: The way we define the states in RevBayes is by letter. Therefore, 1=0A, 2=1A, 3=0B, and 4=1B</p>	<pre> hidden_rate1 ~ dnExponential(rate_pr) moves[+mvi] = mvScale(hidden_rate1,lambda=0.2,tune=true,weight=5) hidden_rate2 ~ dnExponential(rate_pr) moves[+mvi] = mvScale(hidden_rate2,lambda=0.2,tune=true,weight=5) #### Here the hidden rates should go (alpha,beta, alpha, beta) for assymetric models q[1][3] := hidden_rate1 q[2][4] := hidden_rate1 q[3][1] := hidden_rate2 q[4][2] := hidden_rate2 ### Note, we could go a step further and define even 4 different hidden rates ##### # Create the rate matrix for the combined observed and hidden states # ##### rate_matrix := fnFreeK(q, rescaled=false, matrixExponentialMethod="scalingAndSquaring") </pre>

<p>ROOT FREQUENCIES</p> <p>5. Draw the fixed, stochastic and deterministic nodes determined by the RevBayes code to calculate the root frequencies</p>	<pre> rate_category_prior ~ dnDirichlet(rep(1,NUM_RATES)) moves[+mvi] = mvBetaSimplex(rate_category_prior,tune=true,weight=2) moves[+mvi] = mvDirichletSimplex(rate_category_prior,tune=true,weight=2) ### Sampling bias ### fix this to 165/450 sampling <- observed_phylogeny.ntips()/450 </pre>
<p>FULL HiSSE MODEL</p> <p>6. As a large group we will come back to this point to determine the full graphical model for HiSSE</p>	<pre> hissemmodel ~ dnCDBDP(rootAge = root_age, speciationRates = speciation, extinctionRates = extinction, Q = rate_matrix, pi = rate_category_prior, rho = sampling) ### clamp the model with the "observed" tree hissemmodel.clamp(observed_phylogeny) hissemmodel.clampCharData(data_exp) #note the clamping on the expanded dataset </pre>

