# Dataset S12

# R code to compute a hidden Markov model that is constrained to honor user-supplied

# state probabilities. User supplies the following vectors:

# count of speciations or extinctions at each pseudolevel (“counts”),

# count of species extant at each pseudolevel (“pn”), and

# median (or mean) hidden Markov model state probabilities (“pm.med”)

# Function to initialize Pi matrix required by “dthmm”

# Initialized using random numbers from normal distribution

# Assume that values on diagonal ~0.4

# Rows must sum to 1

setPi <- function(m) {

Pi <- matrix(data=NA, nrow=m, ncol=m, byrow=T)

for(p in 1:m){

Pi[p,p] <- rnorm(1,mean=0.4,sd=0.1)

Pi[p,-p] <- (1-Pi[p,p])/(m-1)

}

return(Pi)

}

# Function to initialize delta and pm vectors required by “dthmm”

# Initialized using random numbers from normal distribution

# Delta must sum to 1

setDPm <- function(m) {

DPm <- vector(mode=”numeric”, length=m)

DPm[1] <- rnorm(1,mean=0.4,sd=0.1)

DPm[-1] <- (1-DPm[1])/(m-1)

return(DPm)

}

# Function to define new distribution; ensures that pm not overwritten

# Adapted from function “Mstep.binom”

Mstep.xyz <- function (x, cond, pm, pn)

{

if (is.null(pn))

stop(“Variable \”size\” must be specified in pn”)

prob <- pm$prob

return(list(prob = prob))

}

# Define probability functions for distribution “xyz”

# Based on binomial distribution

rxyz <- rbinom

dxyz <- dbinom

pxyz <- pbinom

qxyz <- qbinom

# Fit constrained hidden Markov model

# “m” is the number of discrete states in the model – this must be set by the user

# “counts” is the user-supplied vector of speciations or extinctions at each pseudolevel

# “pn” is the user-supplied vector of species extant at each pseudolevel

# “pm.med” is user-supplied vector of pm

library(HiddenMarkov)

m <- 3 # specify number of states in the model

x <- dthmm(counts, Pi=setPi(n.states), delta=setDPm(n.states), dist=”xyz”,   
 pm=list(prob=pm.med), pn=list(size=pn), discrete=TRUE)

y <- BaumWelch(x) # estimated parameters of the constrained HMM

n.param <- (m\*(m-1))+2\*m-1 # number of parameters in the model

log.lik.y <- logLik(y) # log likelihood of model y

AIC.y <- 2\*n.param-2\*log.lik.y # Akaike’s Information Criterion for model y

# Predict most likely sequence of states given observations and model y

v <- Viterbi(y) # predicted sequence of HMM states