

Section 2: Checking implementation

#Checking FBD likelihood

A simple way to check that the likelihoods returned by `MiSSE()`, `hisse()`, and `MuHiSSE()` are correct is to take the likelihood calculated by `MiSSE()` and add this to the likelihood of a simple Markov transition model. As described in Caetano et al. (2018), an SSE model jointly maximizes the probability of the observed states at the tips *and* the observed tree, given the model. Thus, if the character is completely disassociated from the rate differences in the tree (with what we call a character independent model) then this test should work.

First step is load the following packages:

```
suppressPackageStartupMessages(library(hisse))
suppressPackageStartupMessages(library(corrHMM))
suppressPackageStartupMessages(library(versitree))
```

We will calculate the likelihood of a simple BiSSE model, where the diversification rates are the same for states 0 and 1, on a simulated dataset that contains fossils:

```
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- NULL
while( is.null( phy ) ){
  phy <- tree.bisse(pars, max.t=30, x0=0, include.extinct=TRUE)
}
k.samples <- data.frame(taxon1="sp12", taxon2="sp12", timefrompresent=3.164384,
  state=1, stringsAsFactors=FALSE)
hidden.states=FALSE
phy.k <- hisse::AddKNodes(phy, k.samples)
fix.type <- hisse::GetKSampleMRCA(phy.k, k.samples)
nb.tip <- Ntip(phy.k)
nb.node <- phy.k$Nnode
gen <- hisse::FindGenerations(phy.k)

data <- data.frame(taxon=names(phy$tip.state), phy$tip.state,
  stringsAsFactors=FALSE)
data <- hisse::AddKData(data, k.samples)
data.new <- data.frame(data[,2], data[,2], row.names=data[,1])
data.new <- data.new[phy.k$tip.label,]

dat.tab <- hisse::OrganizeDataHiSSE(data.new, phy=phy.k, f=c(1,1),
  hidden.states=FALSE)
edge_details <- hisse::GetEdgeDetails(phy=phy.k,
  intervening.intervals=strat.cache$intervening.intervals)
fossil.taxa <- edge_details$tipward_node[which(edge_details$type ==
  "extinct_tip")]
pars.bisse <- c(0.1+0.03, 0.1+0.03, 0.03/0.1, 0.03/0.1, 0.01, 0.01)

model.vec <- numeric(48)
model.vec[1:6] = pars.bisse
```

```

phy$node.label = NULL
cache <- hisse::ParametersToPassfHiSSE(model.vec, hidden.states=hidden.states,
  nb.tip=Ntip(phy.k), nb.node=Nnode(phy.k), bad.likelihood=-300, f=c(1,1),
  ode.eps=0)
cache$psi <- 0.01
hisse.full <- hisse::DownPassHiSSE(dat.tab, gen, cache, root.type="madfitz",
  condition.on.survival=TRUE, root.p=NULL, node=fix.type$node,
  state=fix.type$state, fossil.taxa=fossil.taxa, fix.type=fix.type$type)

```

Next we will calculate the likelihood of *just* the tree using MiSSE():

```

dat.tab <- hisse::OrganizeDataMiSSE(phy=phy.k, f=1, hidden.states=1)
model.vec <- c(0.1+0.03, 0.03/0.1, rep(0,51))
cache = hisse::ParametersToPassMiSSE(model.vec=model.vec, hidden.states=1,
  fixed.eps=NULL, nb.tip=nb.tip, nb.node=nb.node, bad.likelihood=
  exp(-500), ode.eps=0)#
cache$psi <- 0.01
gen <- hisse::FindGenerations(phy.k)
MiSSE.logL <- hisse::DownPassMisse(dat.tab=dat.tab, cache=cache, gen=gen,
  condition.on.survival=TRUE, root.type="madfitz", root.p=NULL,
  fossil.taxa=fossil.taxa, node=fix.type$node, fix.type=fix.type$type)

```

Finally, I will use corHMM to calculate the likelihood of *just* character data:

```

library(corHMM)
char.logL <- corHMM(phy.k, data, rate.cat=1, model = "ER", node.states = "none",
  fixed.nodes=FALSE, p=0.01, root.p="maddfitz")

```

```

## You specified 'fixed.nodes=FALSE' but included a phy object with node labels. These node labels have
## State distribution in data:
## States: 1 2
## Counts: 2 17
## Calculating likelihood from a set of fixed parameters

```

We can compare the likelihoods obtained from hisse() against the sum of the tree *and* the character:

```

tot.logL <- char.logL$loglik + MiSSE.logL
comparison <- identical(round(hisse.full,3), round(tot.logL,3))
print(comparison)

```

```
## [1] TRUE
```

This confirms that the calculations are correct. I will show the same using MuHiSSE():

```

pars <- c(.1, .15, .2, .1,
  .03, .045, .06, 0.03,
  .05, .05, .00,
  .05, .00, .05,
  .05, .00, .05,
  .00, .05, .05)
set.seed(2)
phy <- NULL
while( is.null( phy ) ){
  phy <- tree.musse(pars, 30, x0=1, include.extinct=TRUE)
}
k.samples <- data.frame(taxon1="sp20", taxon2="sp37", timefrompresent=8.54554,
  state1=0, state2=1, stringsAsFactors=FALSE)

```

```

phy.k <- hisse::AddKNodes(phy, k.samples)
fix.type <- hisse::GetKSampleMRCA(phy.k, k.samples)
nb.tip <- Ntip(phy.k)
nb.node <- phy.k$Nnode
gen <- hisse::FindGenerations(phy.k)

states <- phy$tip.state
states <- data.frame(phy$tip.state, phy$tip.state,
row.names=names(phy$tip.state))
states <- states[phy$tip.label,]
states.trans <- states
for(i in 1:Ntip(phy)){
  if(states[i,1] == 1){
    states.trans[i,1] = 0
    states.trans[i,2] = 0
  }
  if(states[i,1] == 2){
    states.trans[i,1] = 0
    states.trans[i,2] = 1
  }
  if(states[i,1] == 3){
    states.trans[i,1] = 1
    states.trans[i,2] = 0
  }
  if(states[i,1] == 4){
    states.trans[i,1] = 1
    states.trans[i,2] = 1
  }
}

data <- data.frame(taxon=names(phy$tip.state),
  states.trans[,1], states.trans[,2], stringsAsFactors=FALSE)
data <- hisse::AddKData(data, k.samples, muhisse=TRUE)
data.new <- data.frame(data[,2], data[,3], row.names=data[,1])
data.new <- data.new[phy.k$tip.label,]

pars.muhisse <- c(rep(0.1+0.03,4), rep(0.03/.1, 4), 0.05,0.05,0, 0.05,0,0.05,
  0.05,0,.05, 0,0.05,.05)
model.vec = rep(0,384)
model.vec[1:20] = pars.muhisse
cache <- hisse::ParametersToPassMuHisSE(model.vec=model.vec, hidden.states=FALSE,
  nb.tip=Ntip(phy.k), nb.node=Nnode(phy.k), bad.likelihood=exp(-500),
  f=c(1,1,1,1), ode.eps=0)
cache$psi <- 0.01
gen <- hisse::FindGenerations(phy.k)
dat.tab <- hisse::OrganizeData(data.new, phy.k, f=c(1,1,1,1),
  hidden.states=FALSE, includes.fossils=TRUE)
fossil.taxa <- which(dat.tab$branch.type == 1)

muhisse.full <- hisse::DownPassMuHisSE(dat.tab, gen=gen, cache=cache,
  root.type="madfitz", condition.on.survival=TRUE, root.p=NULL,
  node=fix.type$node, state=fix.type$state,
  fossil.taxa=fossil.taxa, fix.type=fix.type$type)

```

```
## Trait independent model should be loglik_tree + loglik_character ##
#Part 1: MiSSE loglik:
dat.tab <- hisse::OrganizeDataMiSSE(phy=phy.k, f=1, hidden.states=1)
model.vec <- c(0.1+0.03, 0.03/0.1, rep(0,51))
cache = hisse::ParametersToPassMiSSE(model.vec=model.vec, hidden.states=1,
    fixed.eps=NULL, nb.tip=nb.tip, nb.node=nb.node,
    bad.likelihood=exp(-500), ode.eps=0)#
cache$psi <- 0.01
edge_details <- hisse::GetEdgeDetails(phy=phy.k,
    intervening.intervals=strat.cache$intervening.intervals)
fossil.taxa <- edge_details$tipward_node[which(edge_details$type == "extinct_tip")]
gen <- hisse::FindGenerations(phy.k)
MiSSE.logL <- hisse::DownPassMisse(dat.tab=dat.tab, cache=cache, gen=gen,
    condition.on.survival=TRUE,
    root.type="maddfitz", root.p=NULL, fossil.taxa=fossil.taxa,
    node=fix.type$node, fix.type=fix.type$type)

#Part 2: corHMM loglik:
char.logL <- corHMM(phy.k, data, rate.cat=1, model = "ER", node.states = "none",
    fixed.nodes=FALSE, p=0.05, root.p="maddfitz")
```

```
## You specified 'fixed.nodes=FALSE' but included a phy object with node labels. These node labels have
## State distribution in data:
## States: 1 2 3 4
## Counts: 12 21 6 3
## Calculating likelihood from a set of fixed parameters
```

```
tot.logL <- char.logL$loglik + MiSSE.logL

comparison <- identical(round(muhisse.full,3), round(tot.logL,3))
print(comparison)
```

```
## [1] TRUE
```

#Checking branch calculations for FBD with stratigraphic ranges

Here we will demonstrate that the additional branch calculations necessary to compute the likelihood of the FBD with stratigraphic ranges conforms to the analytical calculations provided by Stadler et al. (2018). The first, are edge segments that represents a stratigraphic range. As described in the main text, we modify $D_i(t)$ to remove the possibility of an unobserved speciation and subsequent extinction event within the interval given that the entire stratigraphic range records the duration of species on branches. The following example assumes tip stratigraphic range, starting at time 0 and ending at time 10:

```
compE <- numeric(26)
compD <- numeric(26)
compD[1] <- 1

yini <- c(EOA = compE[1], EOB = compE[2], EOC = compE[3], EOD = compE[4],
    EOE = compE[5], EOF = compE[6], EOG = compE[7], EOH = compE[8],
    EOI = compE[9], EOJ = compE[10], EOK = compE[11], EOL = compE[12],
    EOM = compE[13], EON = compE[14], EOO = compE[15], EOP = compE[16],
    EOQ = compE[17], EOR = compE[18], EOS = compE[19], EOT = compE[20],
    EOU = compE[21], EOV = compE[22], EOW = compE[23], EOX = compE[24],
    EOY = compE[25], EOZ = compE[26], DOA = compD[1], DOB = compD[2],
    DOC = compD[3], DOD = compD[4], DOE = compD[5], DOF = compD[6],
```

```

    DOG = compD[7], DOH = compD[8], DOI = compD[9], DOJ = compD[10],
    DOK = compD[11], DOL = compD[12], DOM = compD[13], DON = compD[14],
    DOO = compD[15], DOP = compD[16], DOQ = compD[17], DOR = compD[18],
    DOS = compD[19], DOT = compD[20], DOU = compD[21], DOV = compD[22],
    DOW = compD[23], DOX = compD[24], DOY = compD[25], DOZ = compD[26])

times=c(0, 10)

pars <- numeric(55)
pars[55] <- 0.05
pars[1] <- .2
pars[2] <- .1
pars[54] <- 1

stad.1 <- hisse:::q_sym_ratio(si=10, ei=0, lambda=.2, mu=.1, psi=0.05, rho=1)

## Now do same calculation using SingleChildCode
rho = 1
turnover = .2 + .1
eps = .1 / .2
psi = 0.05
model.vec <- c(turnover, eps, rep(0,51))
cache <- hisse:::ParametersToPassMisSE(model.vec=model.vec, hidden.states=1,
    fixed.eps=NULL, nb.tip=1, nb.node=1, bad.likelihood=exp(-300), ode.eps=0)
cache$psi <- psi
strat.range.calc <- hisse:::SingleChildProbMisSE(cache, pars, compD,
    compE, 0, 10, 3)
comparison <- identical(round(unname(strat.range.calc[27]),4), round(stad.1,4))

print(comparison)

## [1] TRUE

```

When an edge segment represents intervening time intervals between two stratigraphic ranges without an observed speciation events, we must account for the fact that there must have been at least one speciation event along this interval. In the following example, we assume the intervening interval occurs for the interval [10,12], but first we must calculate the branch probability for the interval [0,10]:

```

compE <- numeric(26)
compD <- numeric(26)
compD[1] <- 1

yini <- c(EOA = compE[1], EOB = compE[2], EOC = compE[3], EOD = compE[4],
    EOE = compE[5], EOF = compE[6], EOG = compE[7], EOH = compE[8],
    EOI = compE[9], EOJ = compE[10], EOK = compE[11], EOL = compE[12],
    EOM = compE[13], EON = compE[14], EOO = compE[15], EOP = compE[16],
    EOQ = compE[17], EOR = compE[18], EOS = compE[19], EOT = compE[20],
    EOU = compE[21], EOV = compE[22], EOW = compE[23], EOX = compE[24],
    EOY = compE[25], EOZ = compE[26], DOA = compD[1], DOB = compD[2],
    DOC = compD[3], DOD = compD[4], DOE = compD[5], DOF = compD[6],
    DOG = compD[7], DOH = compD[8], DOI = compD[9], DOJ = compD[10],
    DOK = compD[11], DOL = compD[12], DOM = compD[13], DON = compD[14],
    DOO = compD[15], DOP = compD[16], DOQ = compD[17], DOR = compD[18],
    DOS = compD[19], DOT = compD[20], DOU = compD[21], DOV = compD[22],
    DOW = compD[23], DOX = compD[24], DOY = compD[25], DOZ = compD[26])

```

```
times=c(0, 10)

pars <- numeric(55)
pars[55] <- 0.05
pars[1] <- .2
pars[2] <- .1
pars[54] <- 1

stad.1 <- hisse:::q_sym_ratio(si=10, ei=0, lambda=.2, mu=.1, psi=0.05, rho=1)
```

Next we use the equation from Corollary 13 in Stadler et al. (2018) to account for at least one unobserved speciation event along this interval, which we then multiply the probability above:

```
## Now test an intervening branch calculations using Stadler eqs:
first.ratio <- hisse:::q_t(t=10, lambda=.2, mu=.1, psi=0.05, rho=1) /
  hisse:::q_sym_tilde(t=10, lambda=.2, mu=.1, psi=0.05, rho=1)
second.ratio <- hisse:::q_sym_tilde(t=12, lambda=.2, mu=.1, psi=0.05, rho=1)/
  hisse:::q_t(t=12, lambda=.2, mu=.1, psi=0.05, rho=1)
stad.unobs <- 1 - (first.ratio * second.ratio)
stad.2 <- stad.unobs * hisse:::q_ratio(12,10,lambda=.2, mu=.1, psi=0.05, rho=1)* stad.1
```

Finally, we can perform the same calculation using an ordinary differential equation in MiSSE, by first integrating along the interval [0,10] using the standard branch calculation, and then using these values as the starting conditions to integrate along the intervening interval. We will then compare to the probability from the analytical calculation:

```
# Now do it using ODE from MiSSE:
rho = 1
turnover = .2 + .1
eps = .1 / .2
psi = 0.05
model.vec <- c(turnover, eps, rep(0,51))
cache <- hisse:::ParametersToPassMiSSE(model.vec=model.vec, hidden.states=1,
  fixed.eps=NULL, nb.tip=1, nb.node=1, bad.likelihood=exp(-300), ode.eps=0)
cache$psi <- psi
strat.range.calc <- hisse:::SingleChildProbMiSSE(cache, pars, compD, compE,
  0, 10, 3)
strat.intervene.calc <- hisse:::SingleChildProbMiSSE(cache, pars,
  strat.range.calc[27:52], strat.range.calc[1:26], 10, 12, 4)

comparison <- identical(round(unname(strat.intervene.calc[27]),4),
  round(stad.2,4))
print(comparison)
```

```
## [1] TRUE
```

References

- Caetano D.S., O'Meara B.C., Beaulieu J.M. 2018. Hidden state models improve state-dependent diversification approaches, including biogeographic models. *Evolution*, 72:2308-2324.
- Stadler T., Gavryushkina A., Warnock R.C.M., Drummond A.J., Heath T.A. 2018. The fossilized birth-death model for analysis of stratigraphic range data under different speciation modes. *Journal of Theoretical Biology* 447:41-55.