checking identifiability for birth death model 4 taxa dataset

Question: Are the simple birth death model identifiable? The likelihood in birth death model is calculated is through diversification rate. So if we have different speciation and extinction rate combinations but same diversification rate, how does the LTT look like?

```r
# Setting up the parameters
#--------------------------------------simulation parameters
set.seed(753)
sim<-1000
n.taxa<-4 # 4 taxa tree
pars.const<-c(0.3,0.1)
lineage<-n.taxa
temp.tree<-tree.bd(pars = pars.const,max.taxa = lineage)
temp.tree
```

```
##
## Phylogenetic tree with 4 tips and 3 internal nodes.
##
## Tip labels:
##   sp1, sp2, sp3, sp4
## Node labels:
##   nd1, nd2, nd3
##
## Rooted; includes branch lengths.
```

```r
#plot(temp.tree)
# likelihood
lik.phy<-make.bd(temp.tree)
parML<-fit.bd(temp.tree)
parML
```

```
##
## Fitted birth-death model:
##
## ML(b/lambda) = 0.3681
## ML(d/mu) = 0
## log(L) = -2.2073
##
## Assumed sampling fraction (rho) = 1
##
## R thinks it has converged.
```

```r
sm1<-plot.phylo(temp.tree,edge.width = 4, font=2,direction = "rightwards",align.tip.label = TRUE,show.t:
```
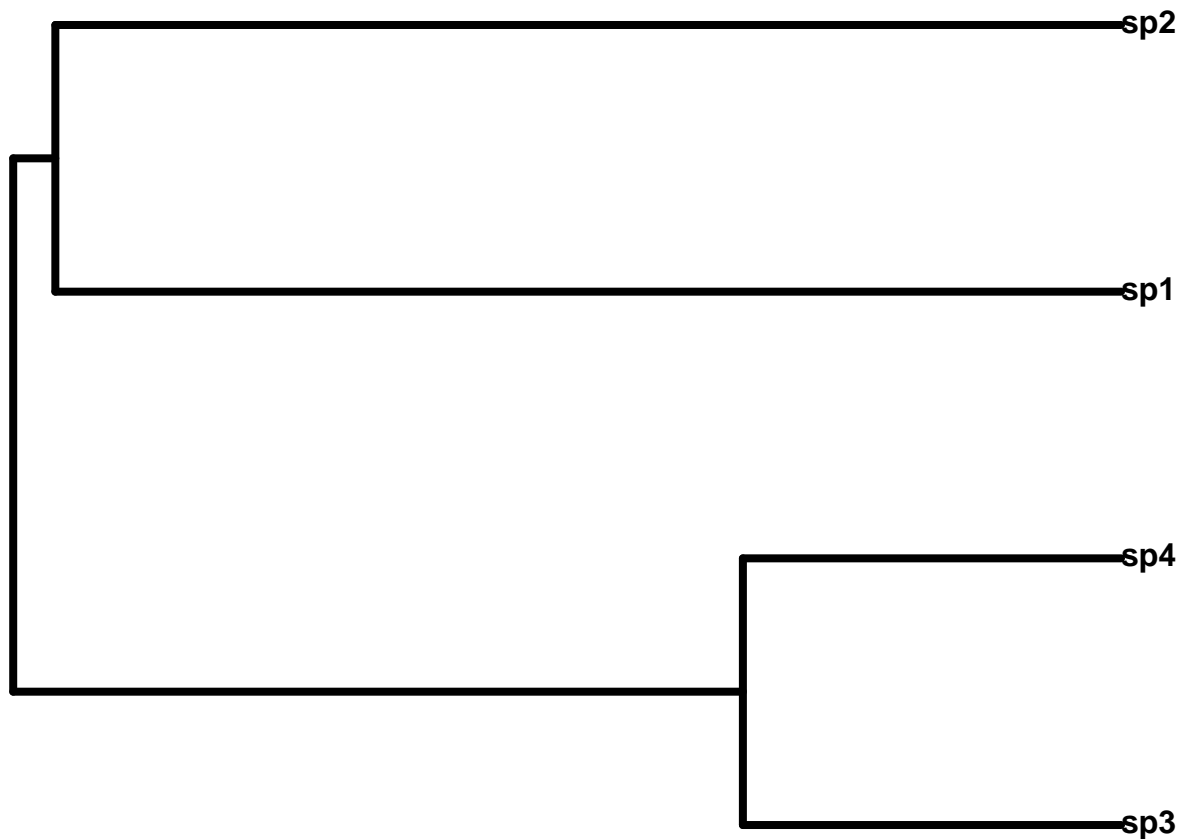
```r
# likelihood
lik.phy<-make.bd(temp.tree)
parML<-fit.bd(temp.tree)
parML

# estimating
parameter<-birthdeath(temp.tree)
lambda.hat<-parameter$para[2]/(1-parameter$para[1])
mu.hat<- lambda.hat - parameter$para[2]

p<-starting.point.bd(temp.tree)
prior.phy<-make.prior.exponential(1/ (2* (p[1] -p[2])))

# Bayesian estimation
samples.noclone<-mcmc(lik.phy, c(parML$b,parML$d), nsteps = 10000,prior=prior.phy,lower=0, w = c(0.1, 0

max.eigen.4taxa<-max(eigen(cov(samples.noclone[,2:3]))$values)

# data cloning start here:
source(file = "/Users/nhans/4days/hisse/MCMC_clone.R")
k.vec<- 2^seq(1,9,by=1)
for(k in k.vec){
  print(k)
assign(paste("samples", k, sep="_"),mcmc(lik.phy, c(parML$b,parML$d), nsteps = 10000,prior=prior.phy,lo
}

# Table 3 prep
clone.mat.4taxa<-matrix(data=NA,nrow=4,ncol=(length(k.vec)+2))
```

```r
# Naming columns pretty for table
Newcol<-vector(length = length(k.vec))
count<-0
for (k in k.vec){
  count<-count+1
  Newcol[count]<-paste("Data cloning", k, sep=" ")
}
colnames(clone.mat.4taxa)<-c("ML estimate","No clone MCMC",Newcol)
rownames(clone.mat.4taxa)<-c("lambda","mu", "lik","likD")

# ML estimates
clone.mat.4taxa[1,1]<-parML$b
clone.mat.4taxa[2,1]<-parML$d
clone.mat.4taxa[3,1]<-parML$logL
clone.mat.4taxa[4,1]<-parML$logL


# Bayesian estimates

clone.mat.4taxa[1,2]<-mean(samples.noclone[,2])
clone.mat.4taxa[2,2]<-mean(samples.noclone[,3])
clone.mat.4taxa[3,2]<-mean(samples.noclone[,4])
clone.mat.4taxa[4,2]<-mean(samples.noclone[,4]) # the likelihood is not scaled or it is 1/1*lik
#-----

# Eigen values
eig.mat<-vector(length = length(k.vec))

# Marginal variance vecs (make this into mat so it is easy to melt)
mvar.mat.lambda<-vector(length = length(k.vec))
mvar.mat.mu<-vector(length = length(k.vec))

counter<-2 # since the first two positions are already filled in the table(modify this code)
count2<-0 # using for eigen mat and mvar.mat

for(k in k.vec){
counter<-counter+1
count2<-count2+1
lambda<-eval(parse(text=paste("samples","_",k,sep="")))$lambda
clone.mat.4taxa[1,counter]<-mean(lambda)

mu<-eval(parse(text=paste("samples","_",k,sep="")))$mu
clone.mat.4taxa[2,counter]<-mean(mu)

# #temp fpr var/covariance mat and eigen values
mat<-cbind(lambda,mu)
eig.mat[count2]<-(eigen(var(mat))$values)/max.eigen.4taxa
mvar.mat.lambda[count2]<-var(lambda)
mvar.mat.mu[count2]<-var(mu)

# likelihood
op<-eval(parse(text=paste("samples","_",k,sep="")))$p
clone.mat.4taxa[3,counter]<-mean(op)
```

```
# scaled likelihood
op.mean<-mean(op)
clone.mat.4taxa[4,counter]<-(1/k)*op.mean
}
```

```
## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length

## Warning in eig.mat[count2] <- (eigen(var(mat))$values)/max.eigen.4taxa: number
## of items to replace is not a multiple of replacement length
```

```
library(knitr)
kable(round(clone.mat.4taxa,4),caption="Data cloning on 4taxa")
```

Table 1: Data cloning on 4taxa

	ML estimate	No clone MCMC	Data cloning 2	Data cloning 4	Data cloning 8	Data cloning 16	Data cloning 32	Data cloning 64	Data cloning 128	Data cloning 256	Data cloning 512
lambda	0.3681	0.5621	0.5206	0.4696	0.4299	0.4036	0.3867	0.3790	0.3733	0.3706	0.3693
mu	0.0000	0.5006	0.4040	0.2945	0.1839	0.1102	0.0619	0.0341	0.0169	0.0086	0.0043
lik	-2.2073	-3.6937	-5.9082	-10.3079	-19.0737	-36.7173	-72.0266	-142.7208	-283.9332	-566.4555	-1131.5078
likD	-2.2073	-3.6937	-2.9541	-2.5770	-2.3842	-2.2948	-2.2508	-2.2300	-2.2182	-2.2127	-2.2100

```
#library(gridExtra)
#grid.table(round(t(clone.mat.4taxa),digits = 3),theme=ttheme_minimal())
```

#Figures

```r
# Clones - separate speciation and extinction
clone512<-samples_512[2:3]
noclone<-samples.noclone[2:3]

lambda.all<-cbind(samples.noclone[2],samples_512[2])
colnames(lambda.all)<-c("No clone","Clone 512")
mu.all<-cbind(samples.noclone[3],samples_512[3])
colnames(mu.all)<-c("No clone","Clone 512")

melted.lambda<-(melt(lambda.all))
```

No id variables; using all as measure variables

```r
melted.mu<-(melt(mu.all))
```

No id variables; using all as measure variables

```r
sm2a<-ggplot(melted.lambda, aes(x=value)) +
  geom_density(alpha=.5,fill="slateblue4")+
  facet_wrap(~ variable)+
  #ggtitle("HiSSE speciation rates")+
  xlim(0,1)+
  #ylim(0,20)+
  ggtitle("Speciation rate")+
  xlab(expression(lambda))+
  ylab("Density")+
  theme_pubclean()
sm2b<-ggplot(melted.mu, aes(x=value)) +
  geom_density(alpha=.5,fill="#56B4E9")+
  facet_wrap(~ variable)+
  #ggtitle("HiSSE speciation rates")+
  xlim(0,0.05)+
  #ylim(0,20)+
   ggtitle("Extinction rate")+
  xlab(expression(mu))+
  ylab("Density")+
  theme_pubclean()

sm2 <- ggarrange(sm2a, sm2b,
                 labels = c("i", "ii"),
                  nrow = 2)
```

Warning: Removed 1041 rows containing non-finite values (stat_density).

Warning: Removed 9139 rows containing non-finite values (stat_density).

```r
# Marginal variances
Lambda<-mvar.mat.lambda
Mu<-mvar.mat.mu
bound.rates<-cbind(Lambda,Mu)
rownames(bound.rates)<-k.vec
melted.marvar<-melt(bound.rates)

cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
colnames(melted.marvar)<- c("Var1" , "Rates" ,"value")
sm3<-ggplot(data=melted.marvar, aes(x=Var1,y=value,group=Rates,colour=Rates,shape=Rates)) +
    geom_line(size=1) +
```

```
    geom_point(size=2)+
    ggtitle("Marginal variance")+
  xlab("number of clones")+
  ylab("Marginal variance ")+
   scale_colour_manual(values=cbbPalette)+
  theme_pubclean()

# Eigenvalues
dat.eigen<-data.frame(k.vec,eig.mat)
sm4<- ggplot(data=dat.eigen, aes(x=k.vec,y=eig.mat)) +
  geom_line(color="aquamarine4",size=1) +
  geom_point(color="aquamarine4",size=3)+
  ggtitle("Eigenvalues")+
  xlab("number of clones")+
  ylab("eigenvalue ")+
  theme_pubclean()

figure4taxa <- ggarrange(sm2a, sm2b, sm3, sm4,
                    labels = c("A", "B", "C","D"),
                    ncol = 2, nrow = 2)
```

```
## Warning: Removed 1041 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 9139 rows containing non-finite values (stat_density).
```

```
ggsave(filename="Figure4_high.png", plot=figure4taxa, device="png",dpi=700)
```

```
## Saving 6.5 x 4.5 in image
```