

motani_response

2023-01-07

Response Motani et al 2022

R version 4.1.2 (2021-11-01) – “Bird Hippie”

Load packages

```
library(ape)
library(geiger)
library(phytools)
library(OUwie)
library(reshape2)
library(ggplot2)
library(tidyr)
library(dplyr)
library(ggbridges)

set.seed(2803)
```

load data

```
dat_MR <- read.table("trait.txt",header=TRUE, sep = "\t")

MetRate <- dat_MR$Calculated.MRs..mL.O2...1.h...1.g.
names(MetRate) <- dat_MR$Taxon

phy <- read.nexus("consensus_Paleotree_final.tre")

phydata <- geiger::treedata(phy, MetRate, sort=TRUE)

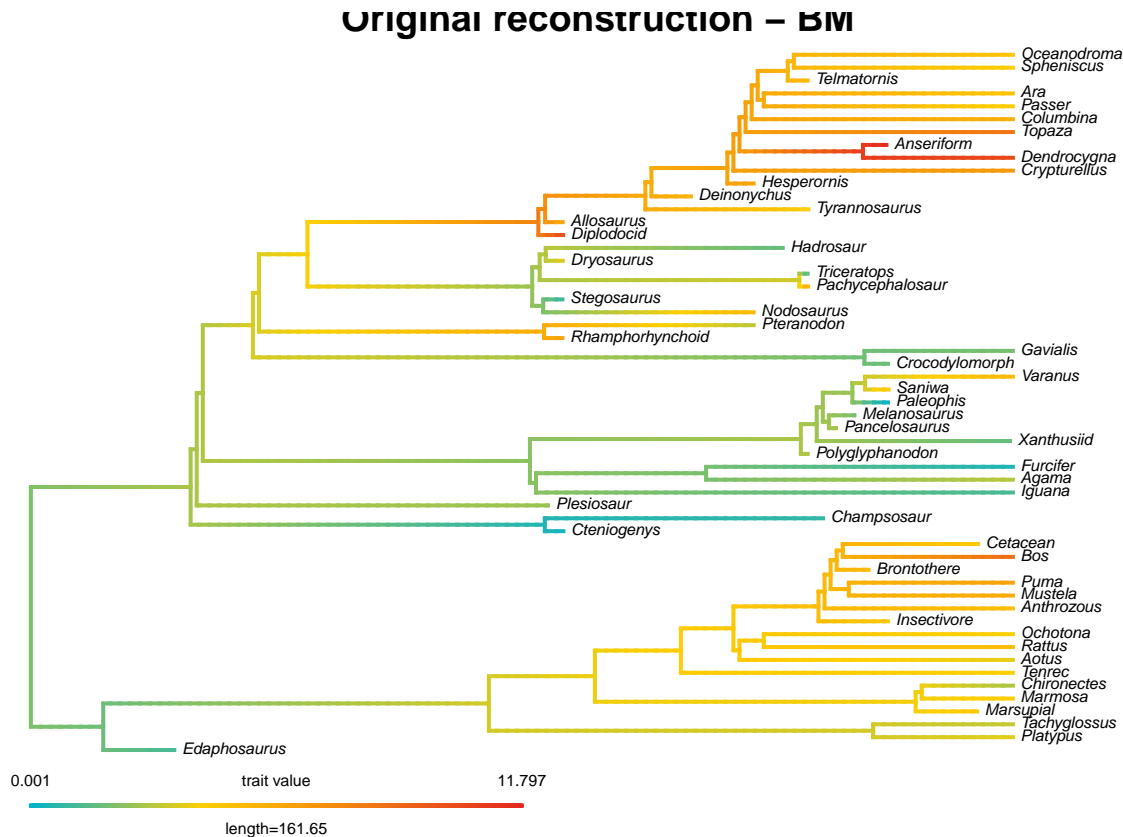
MetRate_phy <- phydata$data
colnames(MetRate_phy) <- "MetRate"
```

Original reconstruction

```
fitBM_fastanc <- fastAnc(phydata$phy, MetRate_phy, vars=TRUE, CI=TRUE)
contmap_BM <- contMap(phydata$phy, MetRate_phy[,1], method = "user",
                      anc.states = fitBM_fastanc$ace, plot = FALSE)
```

```
#change default colors
ramp_rate <- colorRampPalette(c("#02b2ce", "#ffd004", "#e52920"), bias=1.5)
contmap_BM$cols[] <- ramp_rate(1001)

plot(contmap_BM, lwd = 2, fsize=0.5, outline=FALSE)
title(main="Original reconstruction - BM")
```



How well fastAnc performs ASR —

following: <https://lukejharmon.github.io/ilhabela/instruction/2015/07/03/ancestral-states-1/> (see also Revell, L. J., Harmon, L. J. (2022). Phylogenetic Comparative Methods in R. United Kingdom: Princeton University Press)

we simulate states in our phylogeny under BM

```
MetRate_BM <- fitContinuous(phydata$phy, MetRate_phy, model = "BM")
x.BM<-fastBM(phydata$phy, internal=TRUE, sig2= MetRate_BM$opt$sigsq,
             a= MetRate_BM$opt$z0)

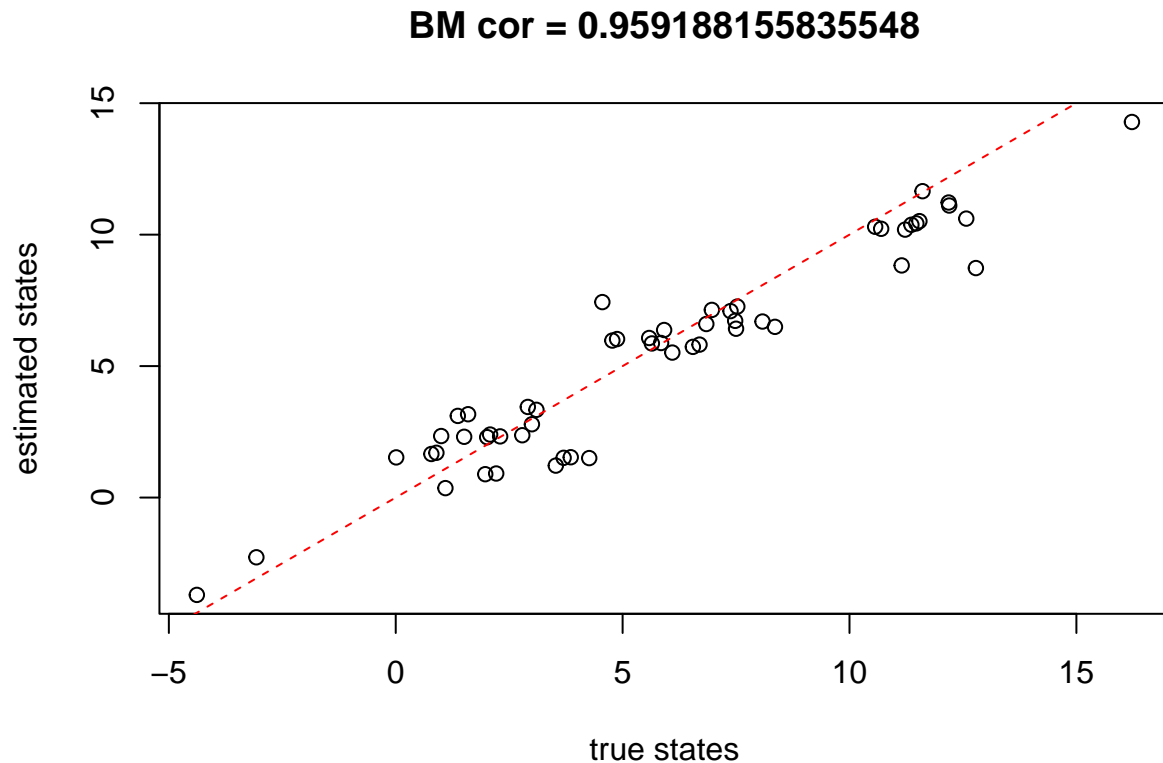
## ancestral states
a.BM<-x.BM[as.character(1:phydata$phy$Nnode+Ntip(phydata$phy))]

## tip data
x.BM<-x.BM[phydata$phy$tip.label]
```

```
fit.test.BM <- fastAnc(phydata$phy, x.BM, vars=TRUE, CI=TRUE)
```

```
correl.BM <- cor(a.BM, fit.test.BM$ace)
```

```
plot(a.BM, fit.test.BM$ace, xlab="true states", ylab="estimated states")
lines(range(c(x.BM, a.BM)), range(c(x.BM, a.BM)), lty="dashed", col="red") ## 1:1 line
title(main= paste("BM cor =", correl.BM))
```



They show high correlation, demonstrating that the function recovers credible ancestral states

BAYESIAN ANALYSIS —

```
set.seed(2803)
MetRate_phy.vec <- as.vector(MetRate_phy)
names(MetRate_phy.vec) <- rownames(MetRate_phy)
fit.bayes <- anc.Bayes(phydata$phy, MetRate_phy.vec, ngen = 99999)
```

```
## Control parameters (set by user or default):
```

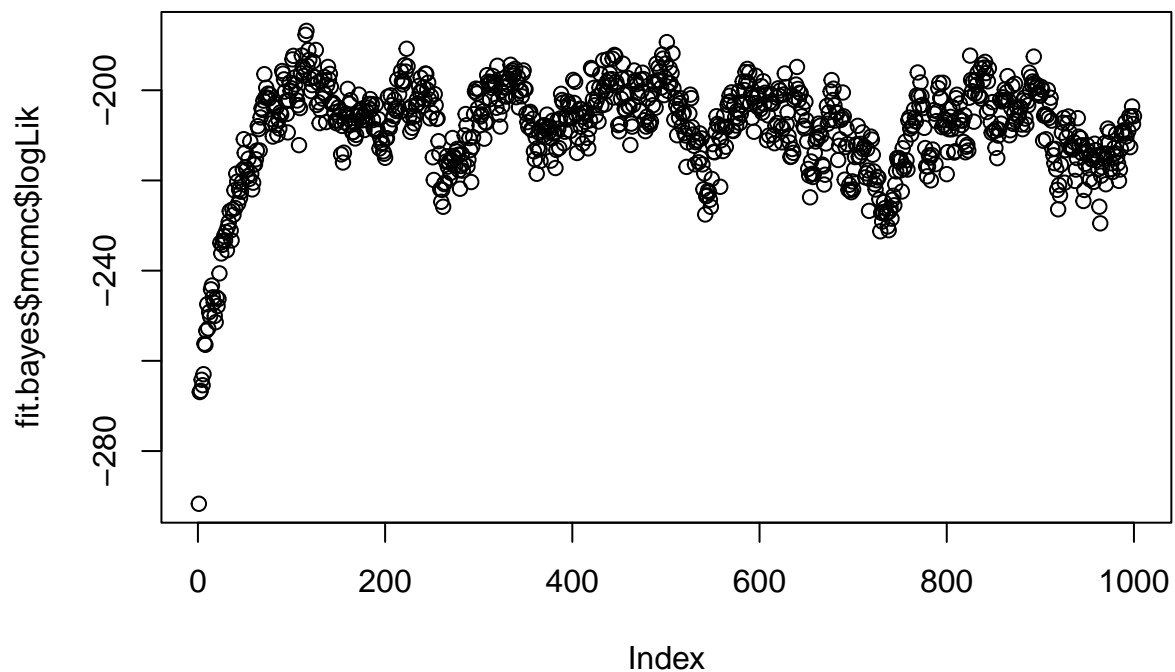
```
## List of 7
## $ sig2 : num 0.158
## $ a : num [1, 1] 2.08
## $ y : num [1:53] 2.08 2.08 2.08 2.08 2.08 ...
```

```
## $ pr.mean: num [1:55] 1000 0 0 0 0 0 0 0 0 0 ...
## $ pr.var : num [1:55] 1e+06 1e+03 1e+03 1e+03 1e+03 1e+03 1e+03 1e+03 1e+03 1e+03 ...
## $ prop : num [1:55] 0.511 0.511 0.511 0.511 0.511 ...
## $ sample : num 100
```

```
## Starting MCMC...
```

```
## Done MCMC.
```

```
plot(fit.bayes$mcmc$logLik)
```



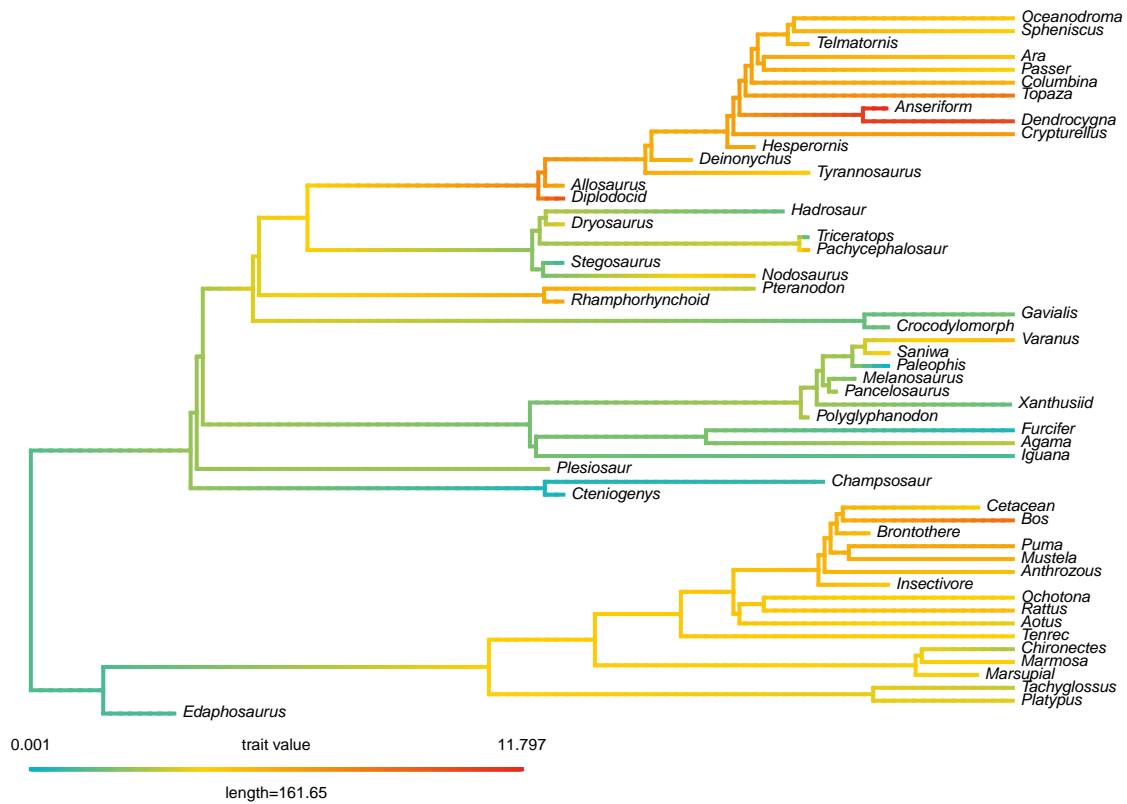
```
#burning
fit.bayes.burned <- fit.bayes$mcmc[-c(1:200),]

bayes.res <- fit.bayes$mcmc[-c(1:200),-c(1,2,ncol(fit.bayes.burned))]
nodes.inference <- colMeans(bayes.res)
nodes.range <- lapply(bayes.res, range)
nodes.CI <- sapply(bayes.res,
                  quantile, probs=c(0.025, 0.975))

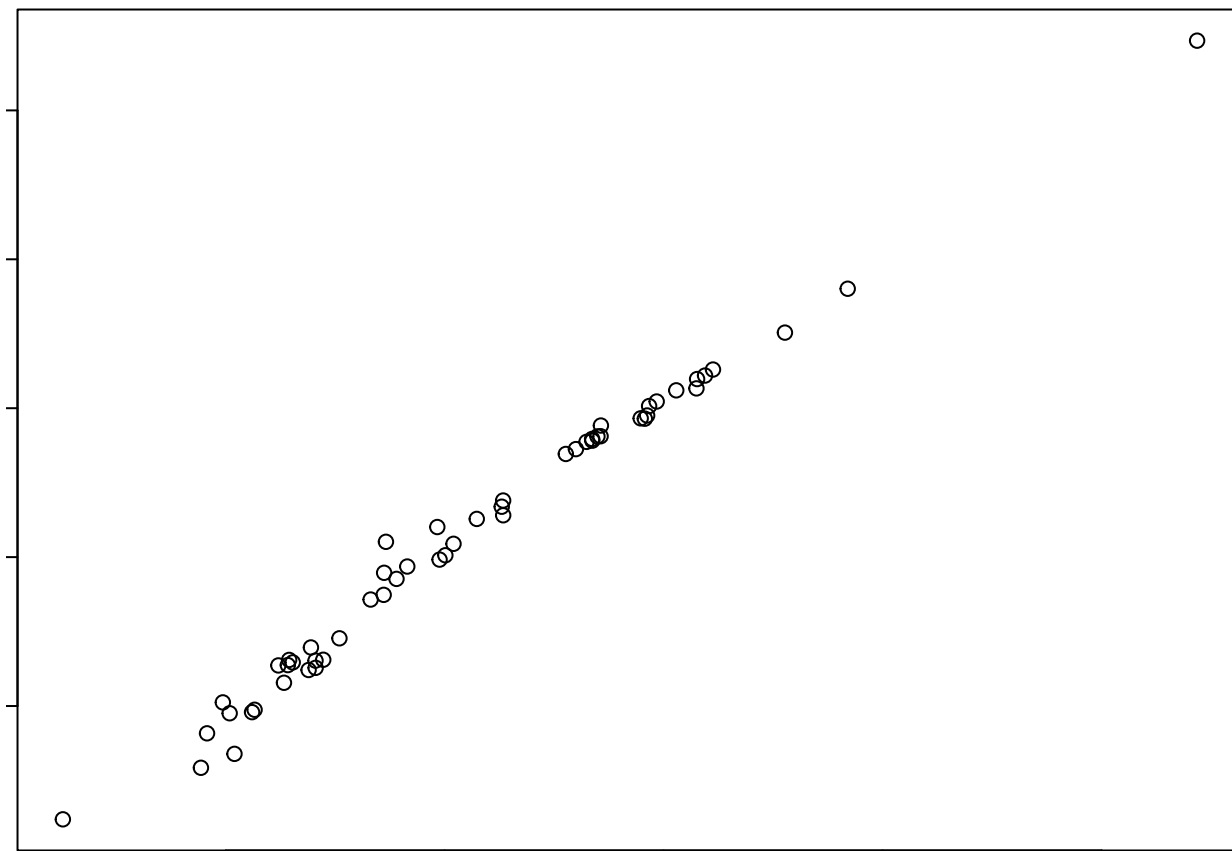
nodes.inference.df <- data.frame(variable = names(nodes.inference), Bayesian=nodes.inference,
                                CImin=nodes.CI[1,], CImax=nodes.CI[2,])

contmap_Bayes <- contMap(phydata$phy, MetRate_phy[,1], method = "user",
                        anc.states = nodes.inference, plot=F)
```

```
contmap_Bayes$cols[] <- ramp_rate(1001)
plot(contmap_Bayes, lwd = 2, fsize=0.5, outline=F)
```



```
plot(fitBM_fastanc$ace,nodes.inference)
```



```
cor(fitBM_fastanc$ace,nodes.inference)
```

```
## [1] 0.9934241
```

similar results than in our original reconstruction

plot probability of every node to show that is not the same for all the confidence interval

```
bayes.res.long <- melt(bayes.res)
```

```
## No id variables; using all as measure variables
```

```
ggplot(data=bayes.res.long, aes(x = value)) +  
  facet_wrap(facets=vars(variable), nrow=6, scales="free")+  
  geom_histogram()+  
  geom_vline(dat=nodes.inference.df, aes(xintercept=Bayesian), col="red")+  
  theme_classic()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

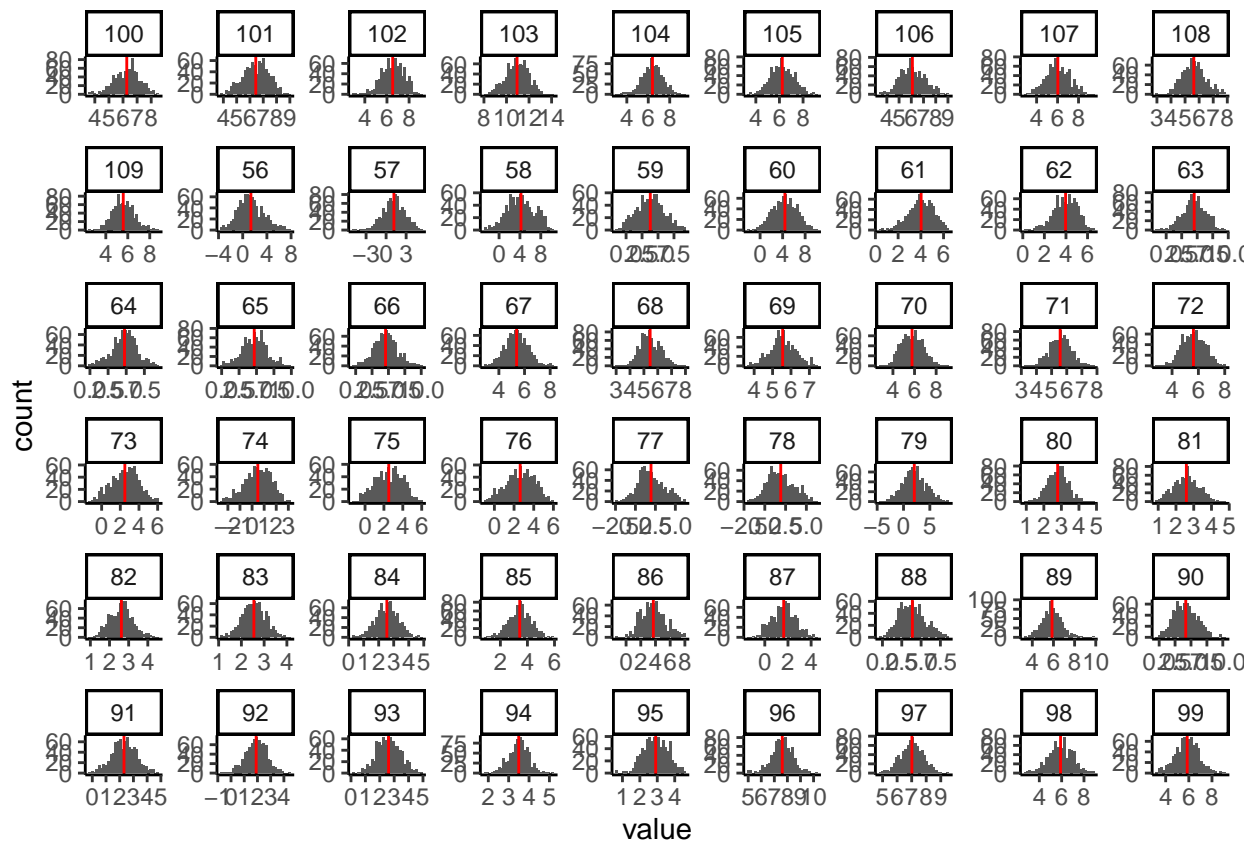


figure of Moteni et al but with probability of CI

```
bayes.res.long2 <- melt(bayes.res[,c("73","74","77","86","88", "89","90","91","96","100","58")))
```

```
## No id variables; using all as measure variables
```

```
bayes.res.long2$Clade[bayes.res.long2$variable==73] <- 'Diapsida'
bayes.res.long2$Clade[bayes.res.long2$variable == 74] <- 'Choristodera'
bayes.res.long2$Clade[bayes.res.long2$variable == 77] <- 'Squamata'
bayes.res.long2$Clade[bayes.res.long2$variable == 86] <- 'Archosauria'
bayes.res.long2$Clade[bayes.res.long2$variable == 88] <- 'Ornithodira'
bayes.res.long2$Clade[bayes.res.long2$variable == 89] <- 'Pterosauria'
bayes.res.long2$Clade[bayes.res.long2$variable == 90] <- 'Dinosauria'
bayes.res.long2$Clade[bayes.res.long2$variable == 91] <- 'Ornithischia'
bayes.res.long2$Clade[bayes.res.long2$variable == 96] <- 'Saurischia'
bayes.res.long2$Clade[bayes.res.long2$variable == 100] <- 'Aves'
bayes.res.long2$Clade[bayes.res.long2$variable == 58] <- 'Mammalia'

bayes.res.long2$Clade <- factor(bayes.res.long2$Clade, level=c('Diapsida',
  'Choristodera', 'Squamata', 'Archosauria', 'Ornithodira',
  'Pterosauria', 'Dinosauria', 'Ornithischia', 'Saurischia',
  'Aves', 'Mammalia'))
```

```

node.means <- bayes.res.long2 %>%
  group_by(Clade) %>%
  summarise(meannode = mean(value))

ggplot(data=bayes.res.long2)+
  stat_density_ridges(aes(x=value, y=Clade, fill=stat(quantile)),
    geom = "density_ridges_gradient", quantile_lines = FALSE,
    calc_ecdf = TRUE, scale=1,
    quantiles = c(0.025, 0.975))+
  scale_fill_manual(name = "Prob.", values = c("white", "gray80", "white"))+
  stat_density_ridges(aes(x=value, y=Clade),fill="transparent",scale=1,
    quantile_lines = TRUE,quantiles = c(0.5))+
  theme_classic()+
  theme(axis.text.x = element_text(angle = 90))+
  coord_flip()

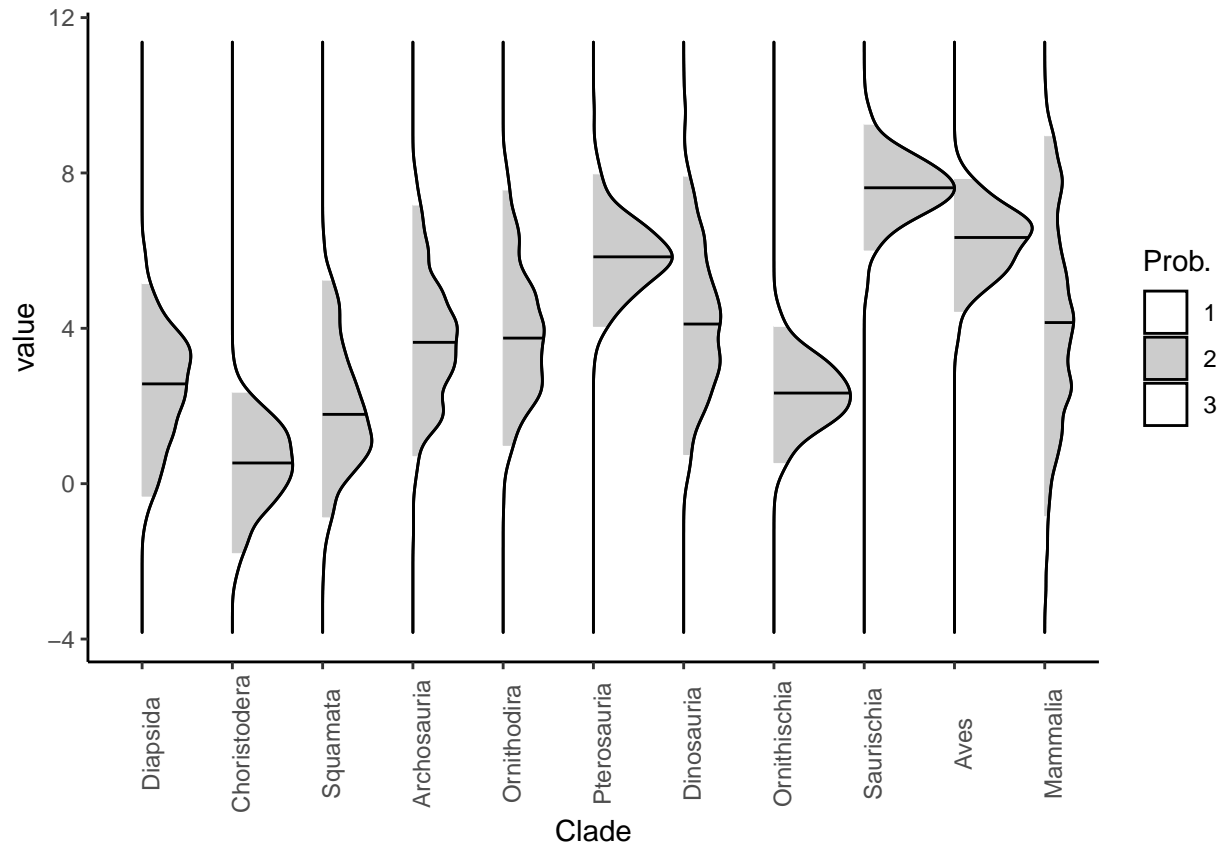
## Warning: 'stat(quantile)' was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(quantile)' instead.

## Picking joint bandwidth of 0.33

## Picking joint bandwidth of 0.33

## Warning: Using the 'size' aesthetic with geom_segment was deprecated in ggplot2 3.4.0.
## i Please use the 'linewidth' aesthetic instead.

```

MODELS

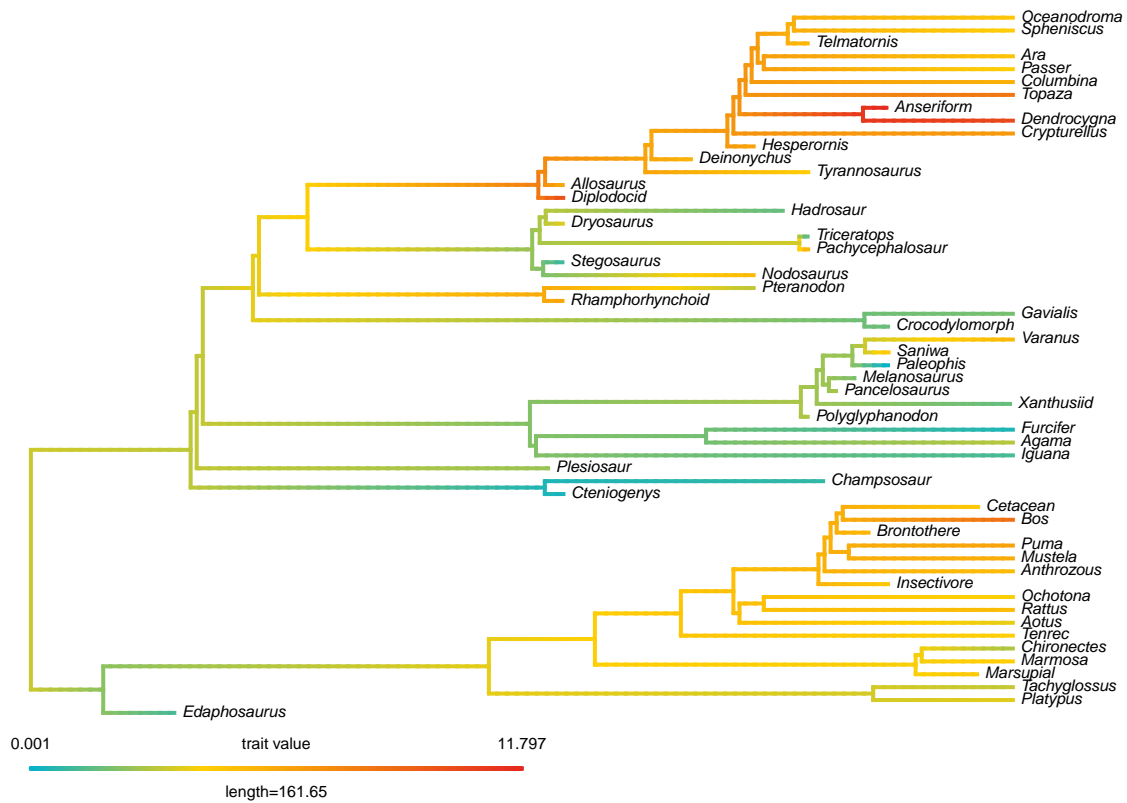
Now with an OU

```
MetRate_phy.vec <- as.vector(MetRate_phy)
names(MetRate_phy.vec) <- rownames(MetRate_phy)
fitOU_ancml <- anc.ML(phydata$phy, MetRate_phy.vec, model = "OU")

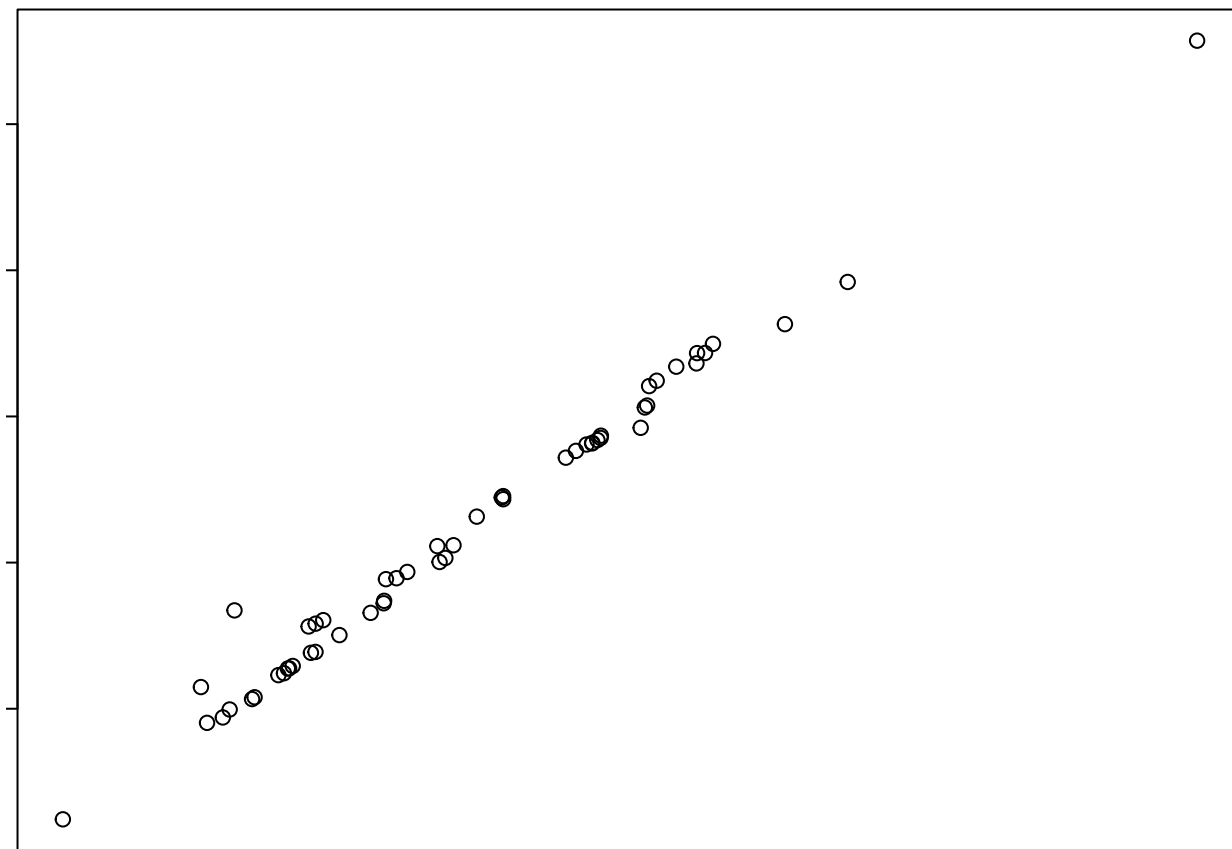
contmap_OU <- contMap(phydata$phy, MetRate_phy[,1], method = "user",
                      anc.states = fitOU_ancml$ace, plot = FALSE)
contmap_OU$cols[] <- ramp_rate(1001)

plot(contmap_OU, lwd = 2, fsize=0.5, outline=F)
title(main="FastAnc - OU")
```

FastAnc - OU



```
plot(fitBM_fastanc$ace,fitOU_ancml$ace)
```



```
cor(fitBM_fastanc$ace,fitOU_ancml$ace)
```

```
## [1] 0.9931362
```

We obtain very similar results than in our original reconstruction

Let's try with other packages, as anc.mL has not been thoroughly tested for OU models # OUwie package — first we need to do a discrete character reconstruction

```
MetRate_dis <- dat_MR$Categorical.strategy
names(MetRate_dis) <- dat_MR$Taxon
phydata_dis <- geiger::treedata(phy, MetRate_dis, sort=TRUE)
```

```
MetRate_dis <- phydata_dis$data
colnames(MetRate_dis) <- "MetRate_dis"
MetRate_dis[MetRate_dis==0] <- 2
```

```
asr_dis_ml <- make.simmap(phydata_dis$phy, MetRate_dis[,1], model = "ER", nsim=1000) #we use an equal r
```

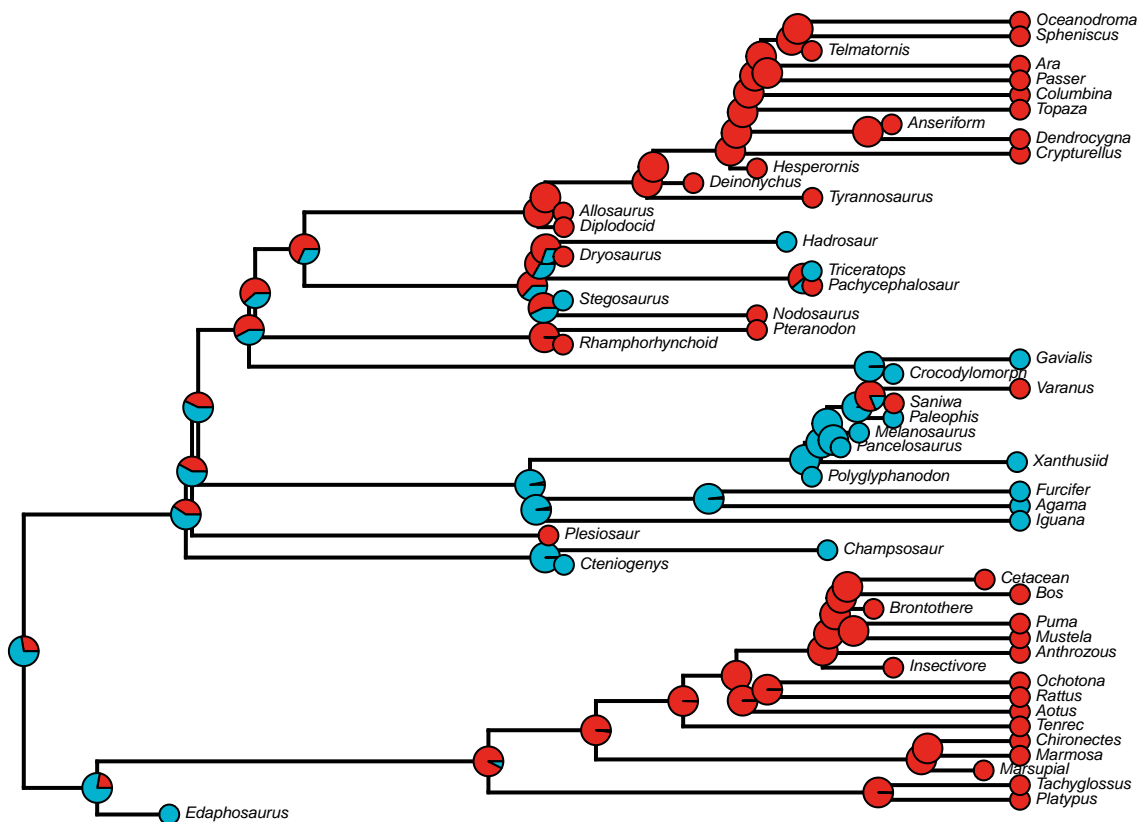
```
## make.simmap is sampling character histories conditioned on
## the transition matrix
##
## Q =
##      1      2
```

```
## 1 -0.002653359 0.002653359
## 2 0.002653359 -0.002653359
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
## 1 2
## 0.5 0.5
```

```
## Done.
```

```
asr_dis_db_ml <- describe.simap(asr_dis_ml)

col_pd <- setNames(c("#e52920", "#02b2ce"), c("1", "2"))
plot(asr_dis_db_ml, fsize=0.5, ftype="i", colors = col_pd)
```



Now we fit models with OUwie package

```
ouwie_phy <- phydata_dis$phy
ouwie_phy$node.label <- as.numeric(asr_dis_db_ml$ace[1:54,1]>0.5)
ouwie_phy$node.label[ouwie_phy$node.label==0] <- 2

data_ouwie <- data.frame(Genus_species= rownames(MetRate_phy), Reg=MetRate_dis[,1],
                        X=MetRate_phy[,1])

MetRate_BM1 <- OUwie(ouwie_phy, data_ouwie, model = "BM1", root.station=FALSE,
                    algorithm = "three.point", scaleHeight =FALSE)
```

```
## Initializing...
## Finished. Begin thorough search...
## Finished. Summarizing results.
```

```
MetRate_BMS <- OUwie(ouwie_phy, data_ouwie, model = "BMS", root.station=FALSE,
                     algorithm = "three.point", scaleHeight = FALSE)
```

```
## Initializing...
## Finished. Begin thorough search...
## Finished. Summarizing results.
```

```
MetRate_OU1 <- OUwie(ouwie_phy, data_ouwie, model = "OU1", root.station=FALSE,
                     scaleHeight = FALSE, algorithm = "three.point")
```

```
## Initializing...
## Finished. Begin thorough search...
## Finished. Summarizing results.
```

```
MetRate_OUM <- OUwie(ouwie_phy, data_ouwie, model = "OUM", root.station=FALSE,
                     algorithm = "three.point", scaleHeight = FALSE)
```

```
## Initializing...
## Finished. Begin thorough search...
## Finished. Summarizing results.
```

```
aicw(c(MetRate_BM1$AICc, MetRate_BMS$AICc, MetRate_OU1$AICc, MetRate_OUM$AICc))
```

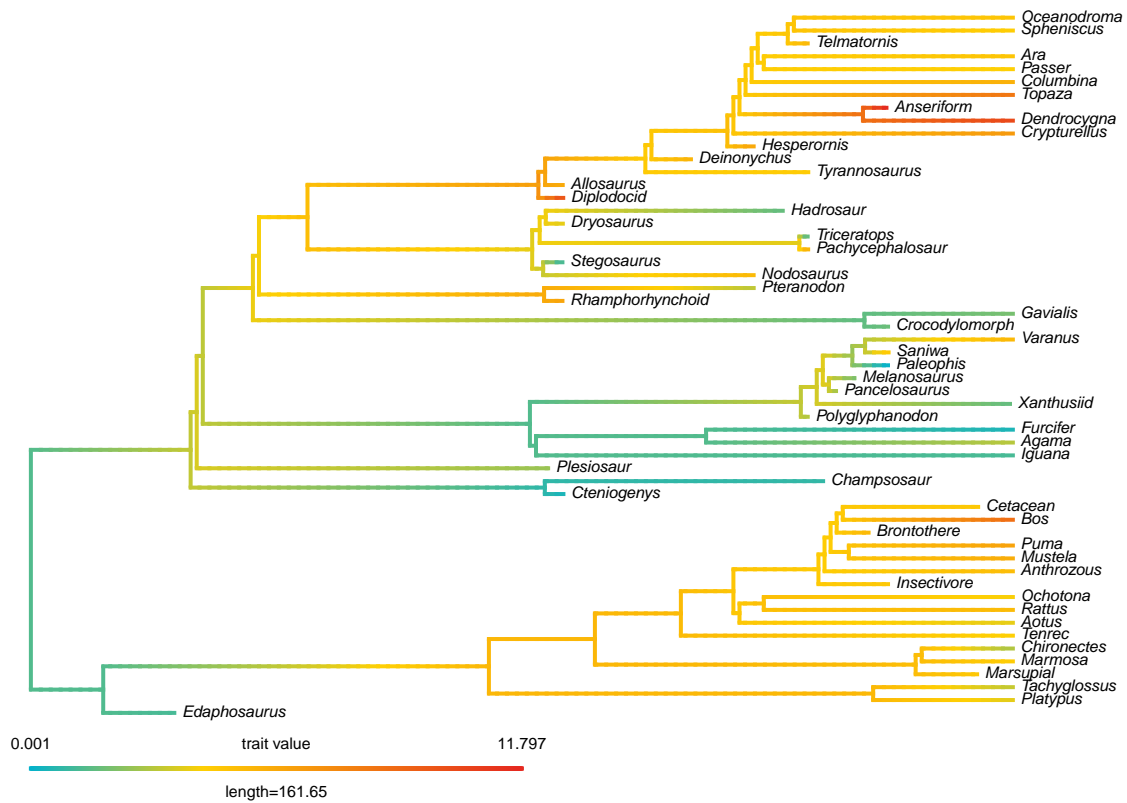
```
##      fit      delta      w
## 1 292.2293 56.11061 6.542361e-13
## 2 289.2919 53.17317 2.841786e-12
## 3 263.6573 27.53853 1.047330e-06
## 4 236.1187  0.00000 9.999990e-01
```

```
#OUM is the best fitted model
```

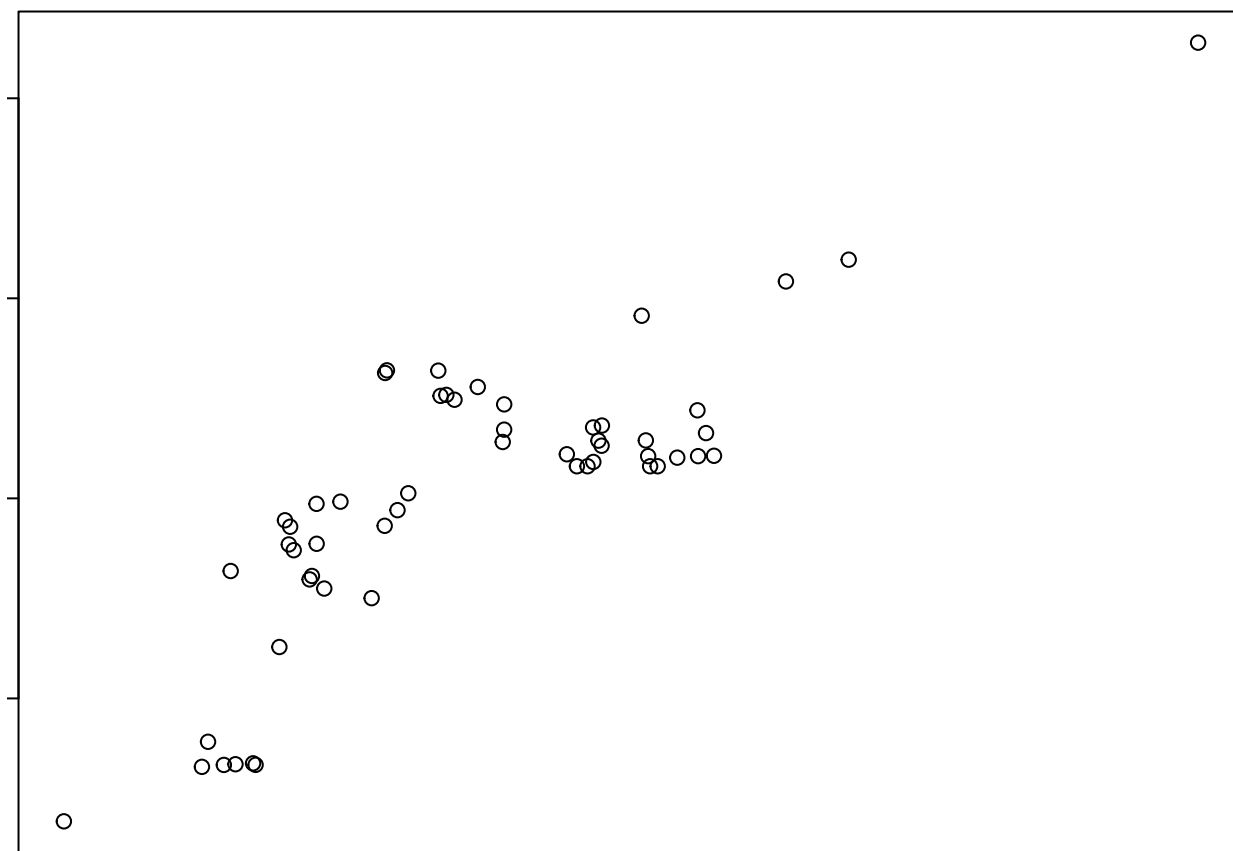
```
Ouwie_OUM_anc <- OUwie.anc(MetRate_OUM, knowledge=TRUE )
```

```
contmap_OUM_ouwie <- contMap(phydata$phy, MetRate_phy[,1], method = "user",
                             anc.states = Ouwie_OUM_anc$NodeRecon, plot = F)
contmap_OUM_ouwie$cols[] <- ramp_rate(1001)
```

```
plot(contmap_OUM_ouwie, lwd = 2, fsize=0.5, outline=F)
```



```
plot(fitBM_fastanc$ace,Ouwie_OUM_anc$NodeRecon)
```



```
cor(fitBM_fastanc$ace,Ouwie_OUM_anc$NodeRecon)
```

```
## [1] 0.8235331
```

This model fitted better, but it implies that species with different metabolic rates are subject to different regimes of natural selection, which is not the expectation. This complex models would require an underlying hypothesis. Running every available model and choosing the best fit is not always the best choice, as we need to consider their biological meaningfulness.