

Supplemental file to “Testing for heterogeneous rates of discrete character evolution on phylogenies” – analysis of *Anolis* caudal vertebrae number

Liam J. Revell et al.

2023-12-01

This code reiterates the analysis of *Anolis* caudal vertebrae evolution for the *phytools* method `fitmultiMk` as described in the study titled “Testing for heterogeneous rates of discrete character evolution on phylogenies” by Revell et al.

```
## load packages
library(phytools)

## Loading required package: ape
## Loading required package: maps
library(geiger)
library(foreach)
library(doParallel)

## Loading required package: iterators
## Loading required package: parallel
## check package version
packageVersion("phytools")

## [1] '2.0.7'
## set seed
set.seed(99)

## read tree from file
anolis_tree<-read.tree("mcc.tre")
anolis_tree

##
## Phylogenetic tree with 187 tips and 186 internal nodes.
##
## Tip labels:
## punctatus, transversalis, agassizi, casildae, microtus, aqbl, ...
##
## Rooted; includes branch lengths.
## read data
anolis_dat<-read.csv("anolis_caudal_vertebrae_data.csv",
  row.names=1)
## data columns are species labels, average caudal vertebrae,
```

```
## average caudal vertebrae rounded to nearest
head(anolis_dat)
```

```
##      Species AvgNumber Vert
## 1      ahli  40.40000  40
## 2    allogus 42.66667  43
## 3 rubribarbus 44.00000  44
## 4      imias 41.50000  42
## 5     sagrei 46.20000  46
## 6    bremeri 45.50000  46
```

```
## convert numeric vector to factor
anolis_vert<-factor(setNames(anolis_dat[,3],anolis_dat[,1]),
  levels=min(anolis_dat[,3]):max(anolis_dat[,3]))
head(anolis_vert)
```

```
##      ahli      allogus rubribarbus      imias      sagrei      bremeri
##      40      43      44      42      46      46
## 22 Levels: 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 ... 55
```

```
## prune tree to match data
pruned.anolis_tree<-drop.tip(anolis_tree,
  setdiff(anolis_tree$tip.label,anolis_dat[,1]))
## check names match
name.check(pruned.anolis_tree,anolis_vert,
  data.names=names(anolis_vert))
```

```
## [1] "OK"
```

```
## create design matrix for ordered evolution
k<-length(levels(anolis_vert))
ordered<-matrix(0,k,k,dimnames=list(levels(anolis_vert),
  levels(anolis_vert)))
for(i in 1:(k-1)){
  ordered[i,i+1]<-1
  ordered[i,i-1]<-2
}
ordered[k,k-1]<-2
```

```
## check first 5 rows & columns of matrix
ordered[1:5,1:5]
```

```
##      34 35 36 37 38
## 34  0  1  0  0  0
## 35  2  0  1  0  0
## 36  0  2  0  1  0
## 37  0  0  2  0  1
## 38  0  0  0  2  0
```

```
## check bottom 5 rows & columns of matrix
ordered[(nrow(ordered)-5):nrow(ordered),
  (ncol(ordered)-5):ncol(ordered)]
```

```
##      50 51 52 53 54 55
## 50  0  1  0  0  0  0
## 51  2  0  1  0  0  0
## 52  0  2  0  1  0  0
## 53  0  0  2  0  1  0
```

```

## 54 0 0 0 2 0 1
## 55 0 0 0 0 2 0

## convert factor to matrix
anolis_vert<-to.matrix(anolis_vert,levels(anolis_vert))

## set number of optimization iterations per model
niter<-10
## open cluster for parallelized optimization
ncores<-min(niter,parallel::detectCores()-2)
mc<-makeCluster(ncores,type="PSOCK")
registerDoParallel(cl=mc)

## run niter parallelized optimizations of single
## regime model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitMk(pruned.anolis_tree,anolis_vert,
    model=ordered,rand_start=TRUE)
}
logL<-sapply(fits,logLik)
print(logL)

## [1] -303.4417 -303.4417 -303.4417 -303.4417 -303.4417 -303.4417 -303.4417
## [8] -303.4417 -303.4417 -303.4417

## select best-fitting of niter iterations
fit.single<-fits[[which(logL==max(logL))[1]]]
print(fit.single,digits=3)

## Object of class "fitMk".
##
## Fitted (or set) value of Q:
##      34      35      36      37      38      39      40      41      42      43      44
## 34 -0.130  0.130  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 35  0.218 -0.348  0.130  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 36  0.000  0.218 -0.348  0.130  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 37  0.000  0.000  0.218 -0.348  0.130  0.000  0.000  0.000  0.000  0.000  0.000
## 38  0.000  0.000  0.000  0.218 -0.348  0.130  0.000  0.000  0.000  0.000  0.000
## 39  0.000  0.000  0.000  0.000  0.218 -0.348  0.130  0.000  0.000  0.000  0.000
## 40  0.000  0.000  0.000  0.000  0.000  0.218 -0.348  0.130  0.000  0.000  0.000
## 41  0.000  0.000  0.000  0.000  0.000  0.000  0.218 -0.348  0.130  0.000  0.000
## 42  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.218 -0.348  0.130  0.000
## 43  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.218 -0.348  0.130
## 44  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.218 -0.348
## 45  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.218
## 46  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 47  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 48  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 49  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 50  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 51  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 52  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 53  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 54  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## 55  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##      45      46      47      48      49      50      51      52      53      54      55

```

```

## 34 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 35 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 36 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 37 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 38 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 39 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 40 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 41 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 42 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 43 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 44 0.130 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 45 -0.348 0.130 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 46 0.218 -0.348 0.130 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 47 0.000 0.218 -0.348 0.130 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 48 0.000 0.000 0.218 -0.348 0.130 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## 49 0.000 0.000 0.000 0.218 -0.348 0.130 0.000 0.000 0.000 0.000 0.000 0.000
## 50 0.000 0.000 0.000 0.000 0.218 -0.348 0.130 0.000 0.000 0.000 0.000 0.000
## 51 0.000 0.000 0.000 0.000 0.000 0.218 -0.348 0.130 0.000 0.000 0.000 0.000
## 52 0.000 0.000 0.000 0.000 0.000 0.000 0.218 -0.348 0.130 0.000 0.000 0.000
## 53 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.218 -0.348 0.130 0.000 0.000
## 54 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.218 -0.348 0.130 0.000
## 55 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.218 -0.348 0.130
##
## Fitted (or set) value of pi:
##      34      35      36      37      38      39      40      41      42      43      44      45      46
## 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045
##      47      48      49      50      51      52      53      54      55
## 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045
## due to treating the root prior as (a) flat.
##
## Log-likelihood: -303.442
##
## Optimization method used was "nllminb"
##
## R thinks it has found the ML solution.
## graph regimes on pruned tree
## note this is not for publication as taxon sampling different
## from other analysis (though the tree is the same)
nodes<-c(133,113)
pruned.anolis_tree<-paintSubTree(pruned.anolis_tree,
  nodes[1],"M","I",stem=0.5)
pruned.anolis_tree<-paintSubTree(pruned.anolis_tree,
  nodes[2],"M","I",stem=0.5)
cols<-setNames(c("brown","blue"),c("M","I"))
plot(pruned.anolis_tree,cols,type="arc",arc_height=0.1,
  part=1,fs=0.7,ftype="i")

```


[illegible]

[illegible]

```

## Log-likelihood: -303.129
##
## Optimization method used was "nlminb"
stopCluster(cl=mc)

## compare models
anova(fit.single,fit.multi)

##          log(L) d.f.      AIC    weight
## fit.single -303.4417    2 610.8834 0.8439058
## fit.multi  -303.1293    4 614.2586 0.1560942

## print rates for each model
fit.single$rates

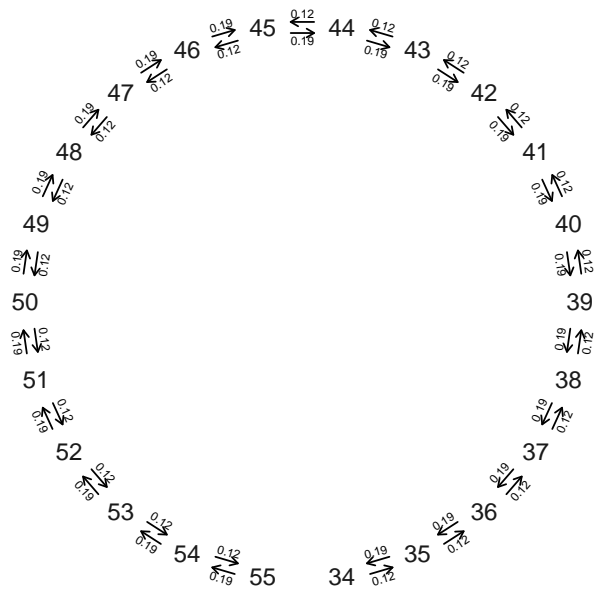
## [1] 0.1304915 0.2175916
fit.multi$rates

## [1] 0.1330670 0.2242077 0.1158850 0.1866748

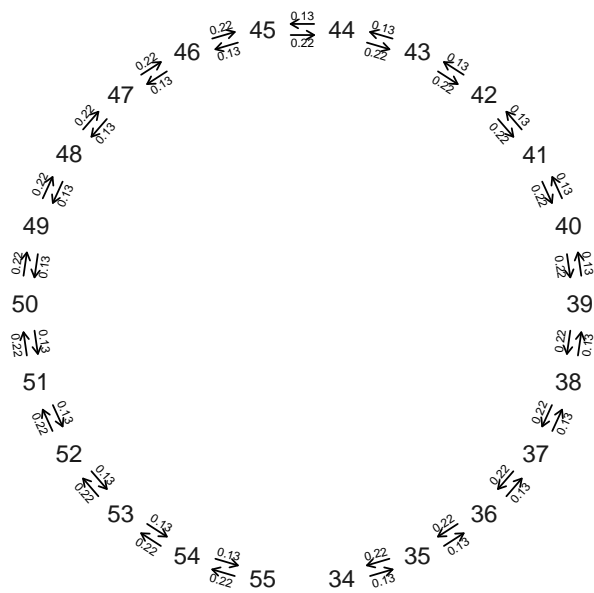
## graph fitted models
par(mfrow=c(2,1),mar=c(1.1,1.1,1.1,1.1))
obj<-fit.multi
obj$rates<-round(obj$rates[3:4],2)
obj$regimes<-NULL
class(obj)<-"fitMk"
plot(obj,show.zeros=F,mar=rep(2.1,4),show.zeros=F,
      tol=1e-3,cex.traits=0.8,cex.rates=0.4)
mtext(text="a) Mainland caudal vertebra number",
      adj=0,line=-1,cex=0.9)
obj<-fit.multi
obj$rates<-round(obj$rates[1:2],2)
obj$regimes<-NULL
class(obj)<-"fitMk"
plot(obj,show.zeros=F,mar=rep(2.1,4),show.zeros=F,
      tol=1e-3,cex.traits=0.8,cex.rates=0.4)
mtext(text="b) Caribbean caudal vertebra number",
      adj=0,line=-1,cex=0.9)

```


a) Mainland caudal vertebra number



b) Caribbean caudal vertebra number



```

## repeat graph, but to export as PDF
pdf(file="Figure_S2.pdf",width=6,height=10)
par(mfrow=c(2,1),mar=c(1.1,1.1,1.1,1.1))
obj<-fit.multi
obj$rates<-round(obj$rates[3:4],2)
obj$regimes<-NULL
class(obj)<-"fitMk"
plot(obj,show.zeros=F,mar=rep(2.1,4),show.zeros=F,
      tol=1e-3,cex.traits=0.8,cex.rates=0.4)
mtext(text="a) Mainland caudal vertebra number",
      adj=0,line=-1,cex=0.9)
obj<-fit.multi
obj$rates<-round(obj$rates[1:2],2)
obj$regimes<-NULL
class(obj)<-"fitMk"
plot(obj,show.zeros=F,mar=rep(2.1,4),show.zeros=F,
      tol=1e-3,cex.traits=0.8,cex.rates=0.4)
mtext(text="b) Caribbean caudal vertebra number",
      adj=0,line=-1,cex=0.9)
dev.off()

## pdf
## 2

save.image(file="morph-analysis.Rdata")

```