Supplemental file to “Testing for heterogeneous rates of discrete character evolution on phylogenies” – error & power analysis

Liam J. Revell et al.

2023-11-30

This code reiterates the analysis of type I error and power 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.

Analysis of Type I error

```
## load packages
library(phytools)
```

```
## Loading required package: ape
```

```
## Loading required package: maps
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(future.apply)
```

```
## Loading required package: future
```

```
## set seed
set.seed(10)
```

```
## this function picks a random position in a "phylo" objects in which
## the probability that an edge is selected depends on the relative
## lenth of that edge
RP<-function(tree){
    cum.edge<-cumsum(tree$edge.length)
    index<-tree$edge[,2]
    pos<-runif(1)*sum(tree$edge.length)
    edge<-1; while(pos>cum.edge[edge]) edge<-edge+1
    return(list(node=index[edge],posn=cum.edge[edge]-pos))
}
## wrapper for phytools::sim.Mk that will exclude any data vector in
## which two states are not observed among terminal taxa
SIM.MK<-function(...,m=2){
    x<-sim.Mk(...)
    while(length(levels(x))!=2) x<-sim.Mk(...)
    x
}
```

```
## taxon sample sizes for simulation
N<-c(25,50,100,200,400,800)

## number of simulations per tree size
nsim<-200

## Q matrix for simulation
Q<-matrix(c(-0.5,0.5,0.5,-0.5),2,2,dimnames=list(letters[1:2],
  letters[1:2]))
```

```
## function for simulating a tree with a mapped regime
simtree<-function(N){
    chk<-FALSE
    while(!chk){
        tree<-pbtree(n=N,scale=1)
        obj<-NULL
        while(is.null(obj)||obj$node<=N) obj<-RP(tree)
        tree<-paintSubTree(tree,obj$node,"1","0",stem=obj$posn)
        obj<-NULL
        while(is.null(obj)||obj$node<=N) obj<-RP(tree)
        tree<-paintSubTree(tree,obj$node,"1","0",stem=obj$posn)
        colors<-setNames(c("blue","red"),0:1)
        p<-summary(factor(getStates(tree,"tips"),levels=0:1))/N
        if(var(p)<=0.125&&sum(summary(tree)$Tr)==2) chk<-T
    }
    tree
}
```

```
## object to store the results of simulation
P<-matrix(NA,nsim,length(N),dimnames=list(1:nsim,N))
TT<-list()
X<-list()
FITS<-list()
FITM<-list()
typeI<-matrix(NA,length(N),2,dimnames=list(N,c("type I",
    "P (binomial test)")))
```

```
## type I error analysis
## detect cores (for future.apply)
ncores<-min(c(parallel::detectCores()-1,nsim))
## plan multisession
plan(multisession,workers=ncores)
for(i in 1:length(N)){
    trees<-replicate(nsim,simtree(N=N[i]),simplify=FALSE)
    class(trees)<-c("multiSimmap","multiPhylo")
    x<-lapply(trees,SIM.MK,Q=Q)
    fits.single<-future_mapply(fitMk,trees,x,SIMPLIFY=FALSE,
      future.seed=TRUE)
    fits.multi<-future_mapply(fitmultiMk,trees,x,SIMPLIFY=FALSE,
      future.seed=TRUE)
```

```r
    LR.test<-suppressWarnings(mapply(lrtest,fits.single,fits.multi,
        SIMPLIFY=FALSE))
    P[,i]<-sapply(LR.test,function(x) x[["Pr(>Chisq)"]][2])
    TT[[i]]<-trees
    X[[i]]<-x
    FITS[[i]]<-fits.single
    FITM[[i]]<-fits.multi
    typeI[i,1]<-mean(P[,i]<=0.05)
        typeI[i,2]<-pbinom(sum(P[,i]<=0.05),nsim,0.05,
        lower.tail=FALSE)
}
## close parallel session
plan(sequential)
```

```r
## create multi-panel figure
par(mfrow=c(3,2))
for(i in 1:length(N)){
  h<-hist(P[,i],breaks=seq(0,1,by=0.05),plot=FALSE)
  h$counts<-h$counts/nsim
  plot(h,col="grey",xlab=expression(paste("P-value from ",chi^2,
    " test")),ylab="relative frequency",main="",ylim=c(0,0.4),
    axes=FALSE)
  axis(1,at=seq(0,1,by=0.2))
  axis(2,at=seq(0,0.4,by=0.1),las=1)
  mtext(text=paste(letters[i],") N = ",N[i],sep=""),adj=0,line=1,
    cex=1)
  lines(c(0,1),rep(0.05,2),lwd=1,col="red",lty="dotted")
}
```

```r
## print table of results
print(typeI)
```

```
##     type I P (binomial test)
## 25   0.070        0.07813442
## 50   0.055        0.30024430
## 100  0.050        0.41693282
## 200  0.050        0.41693282
## 400  0.065        0.12989223
## 800  0.035        0.78669530
```

```r
## export the same figure to a PDF
pdf(file="Figure_S1.pdf",width=7,height=8)
par(mfrow=c(3,2))
for(i in 1:length(N)){
  h<-hist(P[,i],breaks=seq(0,1,by=0.05),plot=FALSE)
  h$counts<-h$counts/nsim
  plot(h,col="grey",xlab=expression(paste("P-value from ",chi^2,
    " test")),ylab="relative frequency",main="",ylim=c(0,0.4),
    axes=FALSE)
  axis(1,at=seq(0,1,by=0.2))
  axis(2,at=seq(0,0.4,by=0.1),las=1)
  mtext(text=paste(letters[i],") N = ",N[i],sep=""),adj=0,
    line=1,cex=1)
  lines(c(0,1),rep(0.05,2),lwd=1,col="red",lty="dotted")
}
```

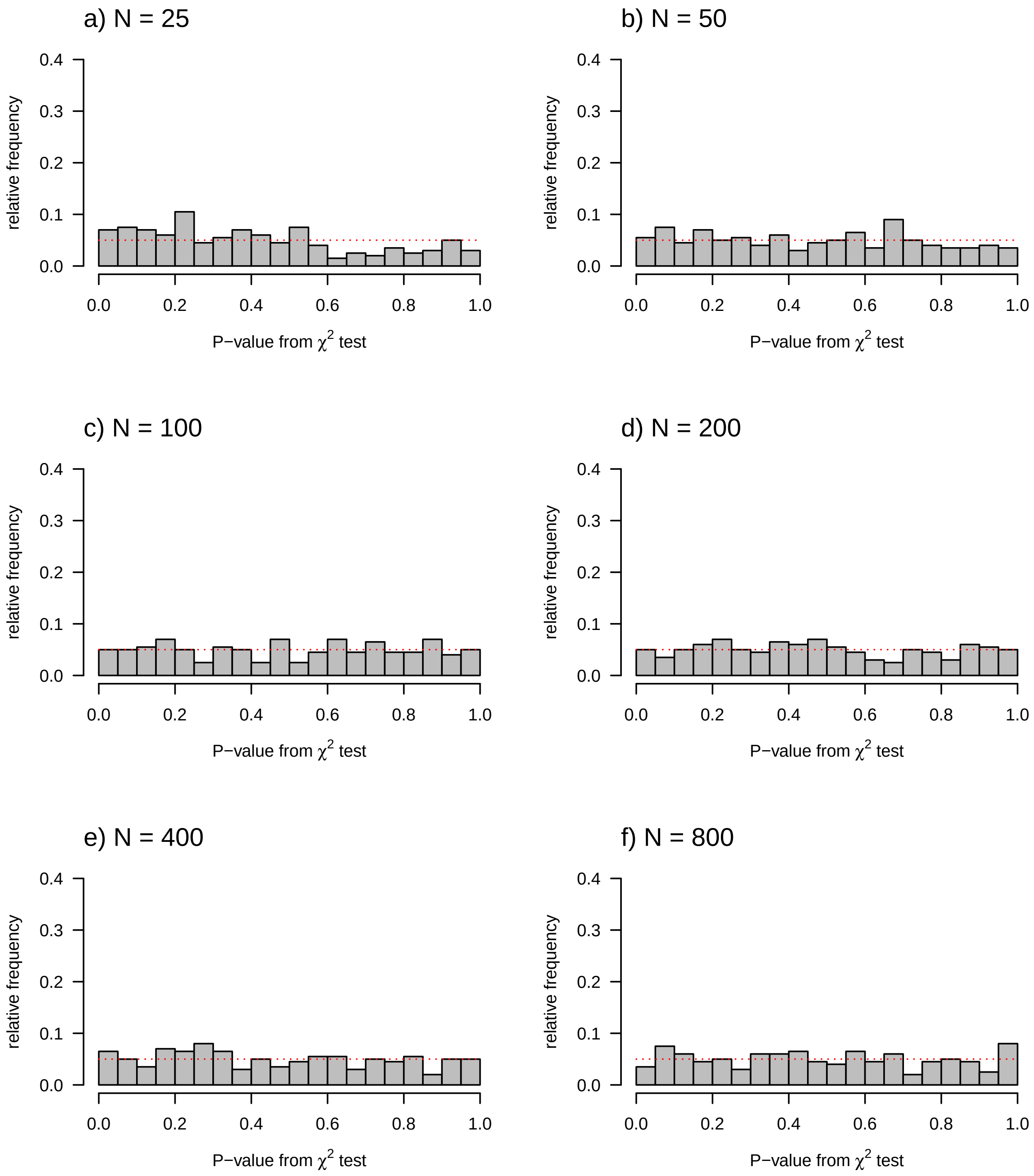

Figure 1: Manuscript Figure S1.

```
dev.off()
```

```
## pdf
##   2
```

Power Analysis

```
## extract just the 100-taxon trees
trees<-TT[[3]]

## set parameters for simulation
q0<-0.5
q1<-c(0.5,1,2,4,8)

## wrapper for phytools::sim.multiMk
SIM.MULTIMK<-function(...,m=2){
    x<-sim.multiMk(...)
    while(length(levels(x))!=2) x<-sim.Mk(...)
    x
}

## create objects to store results
P2<-Q0<-Q1<-matrix(NA,nsim,length(q1),dimnames=list(1:nsim,q1))
X2<-list()
FITS2<-list()
FITM2<-list()
power<-matrix(NA,length(q1),7,dimnames=list(q1,c("power",
    "q0","sd(q0)","median(q0)","q1","sd(q1)","median(q1)")))
```

```
## run power analysis
plan(multisession,workers=ncores)
for(i in 1:length(q1)){
    Q<-setNames(list(
        matrix(c(-q0,q0,q0,-q0),2,2,
          dimnames=list(letters[1:2],letters[1:2])),
        matrix(c(-q1[i],q1[i],q1[i],-q1[i]),2,2,
          dimnames=list(letters[1:2],letters[1:2]))),
      c("0","1"))
    x<-lapply(trees,SIM.MULTIMK,Q=Q)
    fits.single<-future_mapply(fitMk,trees,x,SIMPLIFY=FALSE,
      future.seed=TRUE)
    fits.multi<-future_mapply(fitmultiMk,trees,x,SIMPLIFY=FALSE,
      future.seed=TRUE)
    LR.test<-suppressWarnings(mapply(lrtest,fits.single,fits.multi,
        SIMPLIFY=FALSE))
    P2[,i]<-sapply(LR.test,function(x) x[["Pr(>Chisq)"]][2])
    Q0[,i]<-sapply(fits.multi,function(x) x$rates[x$regimes=="0"])
    Q1[,i]<-sapply(fits.multi,function(x) x$rates[x$regimes=="1"])
    X2[[i]]<-x
    FITS2[[i]]<-fits.single
    FITM2[[i]]<-fits.multi
    power[i,]<-c(mean(P2[,i]<=0.05),mean(Q0[,i]),sd(Q0[,i]),
      median(Q0[,i]),mean(Q1[,i]),sd(Q1[,i]),median(Q1[,i]))
}
```

```
plan(sequential)
```

```
## print table of results
print(power)
```

```
##     power        q0     sd(q0) median(q0)         q1      sd(q1) median(q1)
## 0.5  0.07 0.5162408 0.2600148  0.4785735  0.5517727   0.3995695  0.5050823
## 1    0.21 0.5279818 0.2958061  0.4623568  1.2343665   0.9669194  1.0449710
## 2    0.57 0.5210190 0.2271921  0.4815624  3.4274788  13.7843825  2.1273346
## 4    0.88 0.5032588 0.2761358  0.4660105 22.5082751 196.3185429  4.1971244
## 8    0.96 0.5032047 0.2311689  0.4862356 86.7641457 467.9836075  8.1704777
```

```
    ## figure
par(mfrow=c(2,1),lend=2,mar=c(5.1,4.1,2.1,2.1),bty="n")
boxplot(Q0,ylim=c(0,3),col="lightgrey",pch=21,cex=1.5,bg="lightgrey",
    ylab=expression(paste("estimated ",q[a],sep="")),
    xlab=expression(paste("simulated ",q[b],sep="")),las=1)
lines(rep(q0,i),type="b",pch=21,cex=1.5,bg=make.transparent("blue",0.5),
    lwd=2,col=make.transparent("blue",0.5))
legend(x="topleft",expression(paste("simulated ",q[a],sep="")),lty=1,pch=21,
    pt.bg=make.transparent("blue",1),pt.cex=1.5,col=make.transparent("blue",1),
    lwd=2,bty="n")
mtext(text="a)",adj=0,line=1,cex=1)
boxplot(Q1,ylim=c(0,35),col="lightgrey",pch=21,cex=1.5,bg="lightgrey",
    ylab=expression(paste("estimated ",q[b],sep="")),
    xlab=expression(paste("simulated ",q[b],sep="")),las=1)
lines(q1[1:i],type="b",pch=21,cex=1.5,bg=make.transparent("blue",0.5),
    lwd=2,col=make.transparent("blue",0.5))
mtext(text="b)",adj=0,line=1,cex=1)
legend(x="topleft",expression(paste("simulated ",q[b],sep="")),lty=1,pch=21,
    pt.bg=make.transparent("blue",1),pt.cex=1.5,col=make.transparent("blue",1),
    lwd=2,bty="n")
```

```
## pdf
##   2
```

```
apply(Q1,2,function(x) sum(x>35))
```

```
## 0.5   1   2   4   8
##   0   0   1   4  24
```

```
save.image(file="power-analysis.Rdata")
```

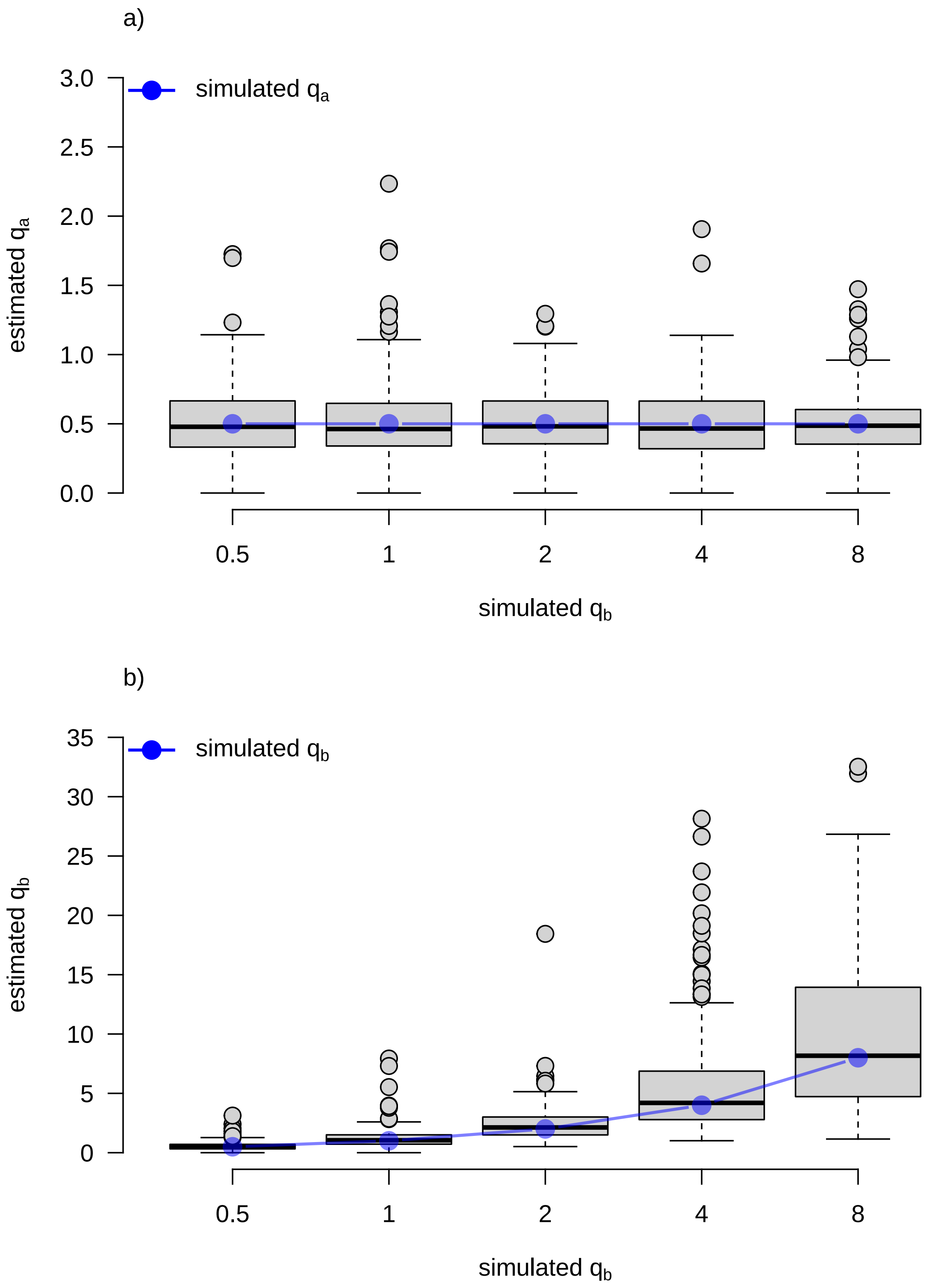

Figure 2: Manuscript Figure 2.