**Exercise 1: Testing the ABC model on Anoles data**

Statistical models involving hypotheses about how (mostly environmental) selection affects trait evolution of species evolving from same lineage. However, the trait evolution for a species is almost always independent of other sister species. Such cases of coupled selection may be observed if species are interaction with each other and thus affecting each other’s evolutionary trajectories.

I have derived a simplistic linear model of trait change based on Nuismer and Harmon’s (2016) analysis. This model describes the trait change over evolutionary time when subjected to three evolutionary processes: Genetic drift, Stabilizing selection due to abiotic environment and Selection imposed by species interactions.

I plan to use data from Mahler et.al. (2013),where the authors created a well-resolved phylogeny of Carribean speices (and ecolmorphs) from Anolis lizards family. The data on various traits can also be found in this study. Different models of selection have been applied to this data (Mahler et.al. 2013,Mahler et.al. 2010,Losos & Ricklefs, 2009). I wish to test the hypotheses at a crude level to discern the relative influences of selective forces namely, stabilizing selection model (Ornstein-Uhlenbeck model) and selection imposed due to coexisting species of Anoles interacting (competing) with each other.

**Methods:**

**Model**:

Every generation, mean trait change for a given species can be described as follows:

Where,

= Variance due to genetic drift

zi =Mean trait value of species *i*

θ= Optimum trait value given the environmental conditions

= Intensity of stabilizing selection due to abiotic factors

S = Intensity of selection imposed by species interactions

µ= Mean trait value of all interacting species.

I used a method of Sequential Monte-Carlo Approximate Bayesian Computation (SMC-ABC) to estimate the posterior distributions for the parameters highlighted in the equation above.

As we can see here, mean trait change is subject to change every generation as it depends on current trait values of all interacting species even if all the other parameters are kept unchanged. Therefore, it is impractical to derive likelihood functions for overall trait changes in given (evolutionary) time for these species. Thus, using ABC method can be ideal in this case.

The specifics of the model are as below:

Priors:

I will use following non-informative priors for above mentioned parameters:

|  |  |  |
| --- | --- | --- |
| Parameter | Prior distribution | Brief explanation |
|  | LogNormal(0,2) | Must have positive values, however should be skewed towards values closer to zero |
| S | LogNormal(0,2) | Must have positive values, however should be skewed towards values closer to zero |
| θ | Uniform(1,8) | Environmental optimum. Thus should be limited by trait value constraints. |
|  | Uniform(0,10) | Variance induced by drift |
| A (Ancestral trait value) | Uniform(1,8) | Same constraints as θ |

Data:

Data on Snout-to-Vent length (SVL) of all species is available and can be a good proxy for body size, an ecologically relevant trait.

Simulation: Simulation of changes in trait values will be run for ~107 generations (Derived from information by Mahler et.al. 2013)

It will start with single common ancestral having a trait value A. It will evolve according the equation above. Species will be added one-by-one based on speciation events derived from the phylogeny. Final trait values of extant species (100) will be recorded.

ABC model algorithm:

1. 50000 combinations of 5 parameter values will be taken from the non-informative priors listed above.
2. Simulation of trait evolution will be run and final trait values will be recorded.
3. Euclidean distance between the set of simulated trait values and actual trait values of extant species will be calculated.
4. If the distance is less than 20, the set of parameters will be accepted for the further filtering.
5. In a second iteration of filtering, the jump function, N(mean, 0.1\*mean) will be used to tweak the parameter values to create a neighborhood around the accepted values.

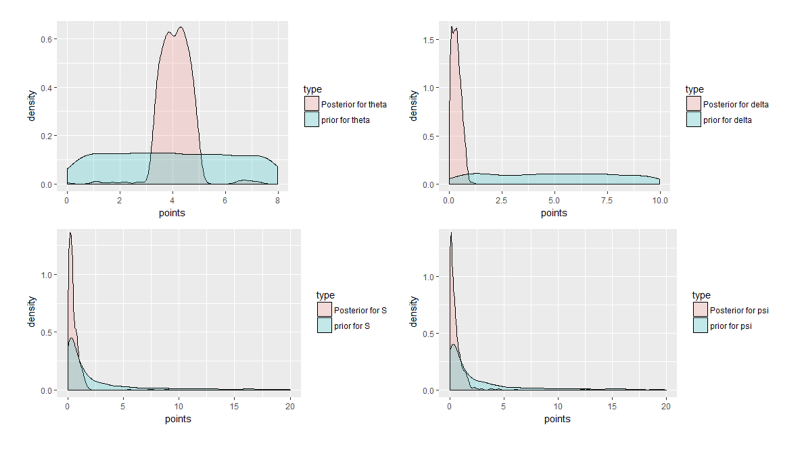
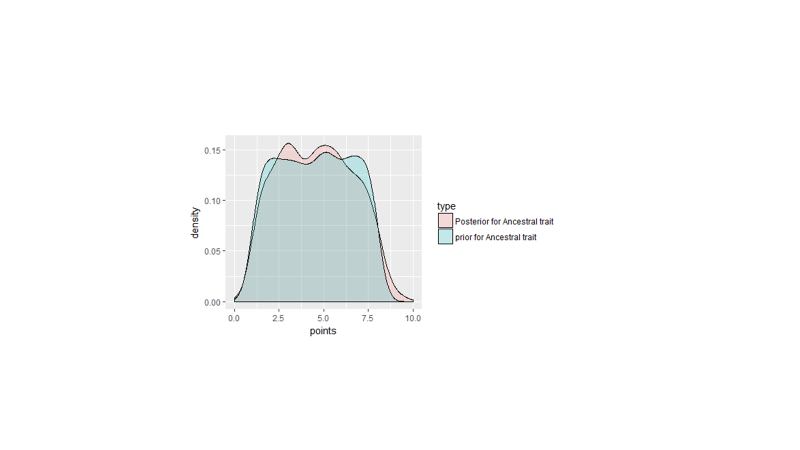
\*This departure from standard SMC ABC was forced because of the computational constraints. I actually did not end up using seawulf.

1. This new set of parameters will be tested against the data with lower tolerance (10).

Convergence:

Since, I did not use a standard MCMC model, measuring convergence is not straightforward. I judged the model based on the rejection rate and the overlap between priors and posteriors. Although I cannot strongly conclude about how well the model worked but these two measures gave me some idea. Acceptance rate was ~4% (2100 accepted out of 50000) for the first iteration (filtering). The next iteration had acceptance of ~20%. Overlaps between priors and posteriors are shown below

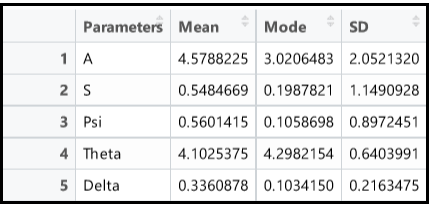
**Results:**



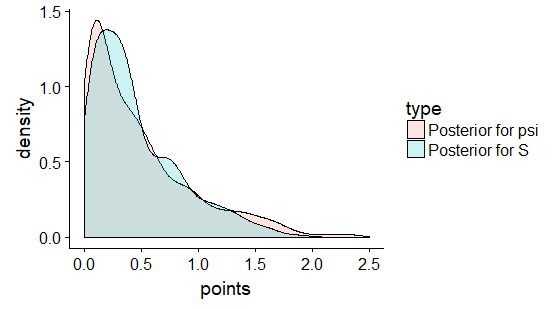
**Fig.1 Overlaps between priors and posteriors of all parameters.**

From these plots, it is clear that parameter A did not converge at all. However, all the other parameters achieved some level of departure from their priors. The theta parameter as well as delta which represent, environmental optima and a variance induced by drift, have gained a sharp and narrow distribution.

Parameters S and φ, the coefficients for tow selection processes had narrower distributions indicating convergence.



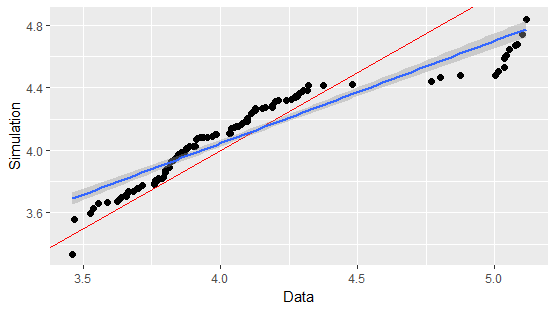
**Table 1: Summary of posterior distribution of all parameters.**



**Fig.2 Comparison of posteriors of S and φ**

The hypothesis behind this model was that the interactions (represented by the parameter, S) do affect the trait evolution of the species. Since, the parameter was constrained to be positive, we cannot reject the hypothesis that interactions have no effect. However relative values of S and φ may indicate which process has been dominant. But, looking at the plot above, there is almost no difference between these two values and absolute values do not inform anything important. Therefore, either both the environmental selection and interactions-driven selection are equally dominant or clearly, we need better method to differentiate between these two parameters.

Model fit:



**Fig.3 Fitting simulated data with actual trait values. The red line is a slope 1 line representing perfect match.**

The simulated values from the means of the posterior values matched quite well with the actual data. But this does not indicate the efficacy of the model, just that the tolerance values for the SMC ABC were narrow enough.

Exercise 2:

Testing the ABC model on simulated datasets:

Traits were simulated on predetermined simulated phylogenies with 20 or 50 or100 tips. The simulation was performed with two different models of competition:

Model 1: MC model

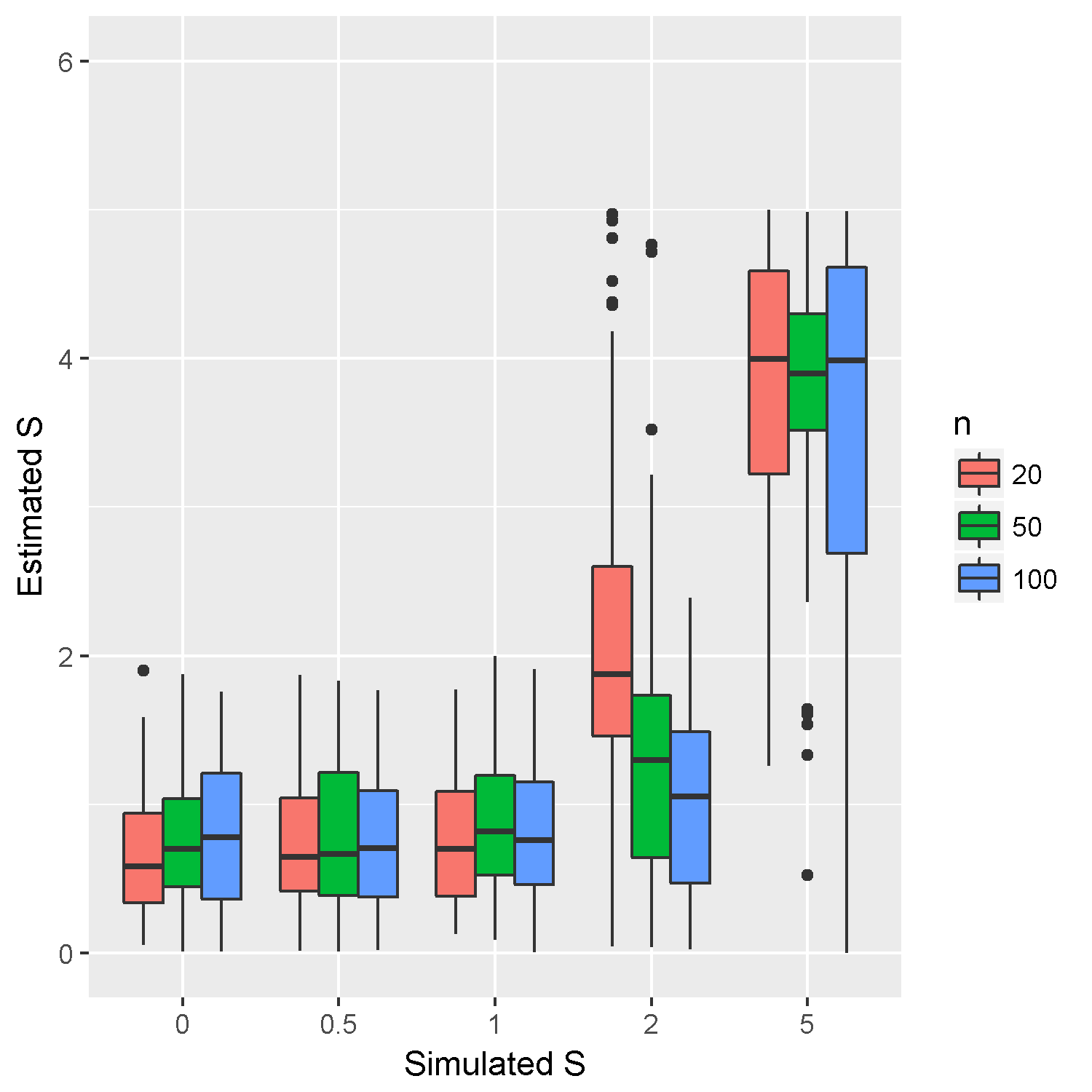
Model 2: LV model

Where,

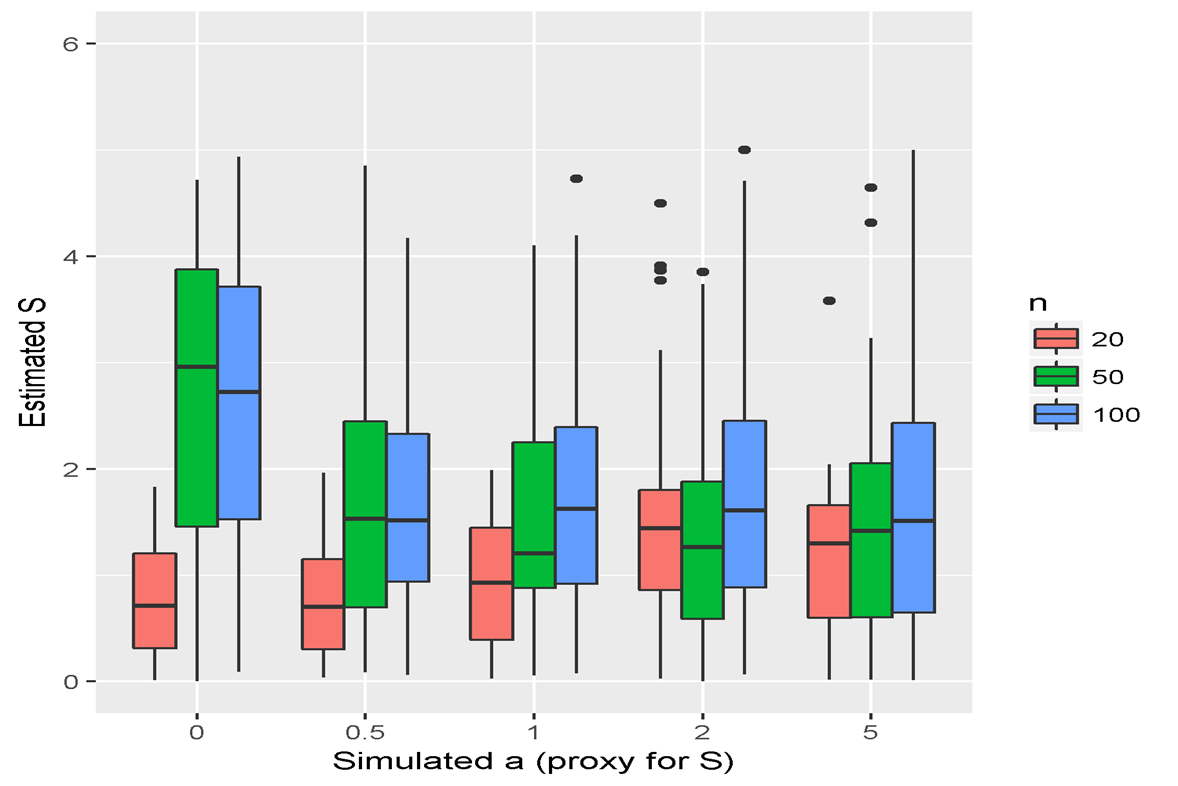
, if |zi-zj|<t\* for some arbitrary threshold t\*.

Parameters S and a should scale with each other.

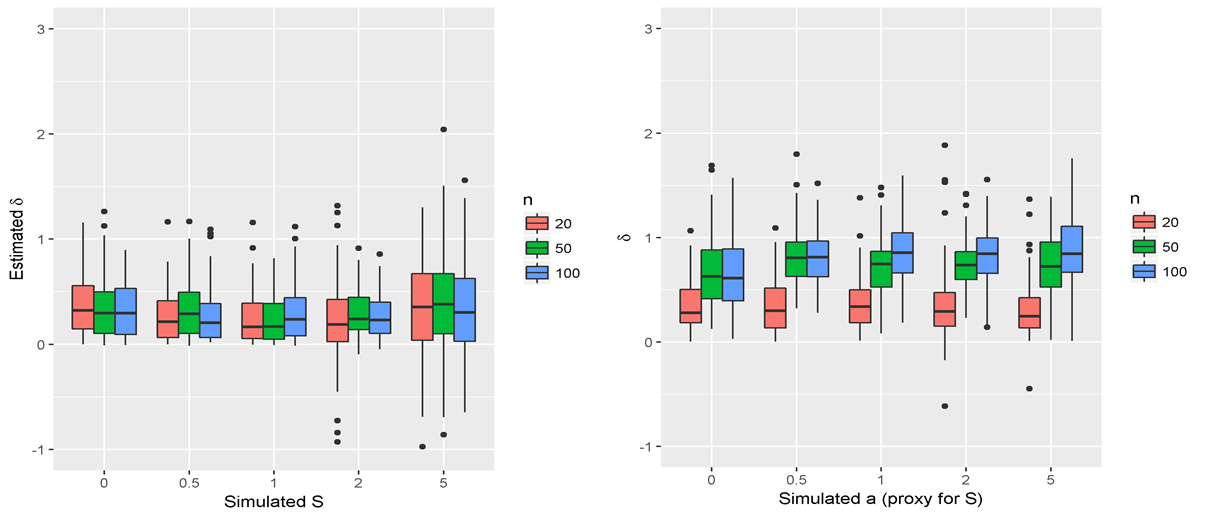
The ABC model shown in exercise 1 was tested on simulated datasets and the accuracy of estimations was observed.

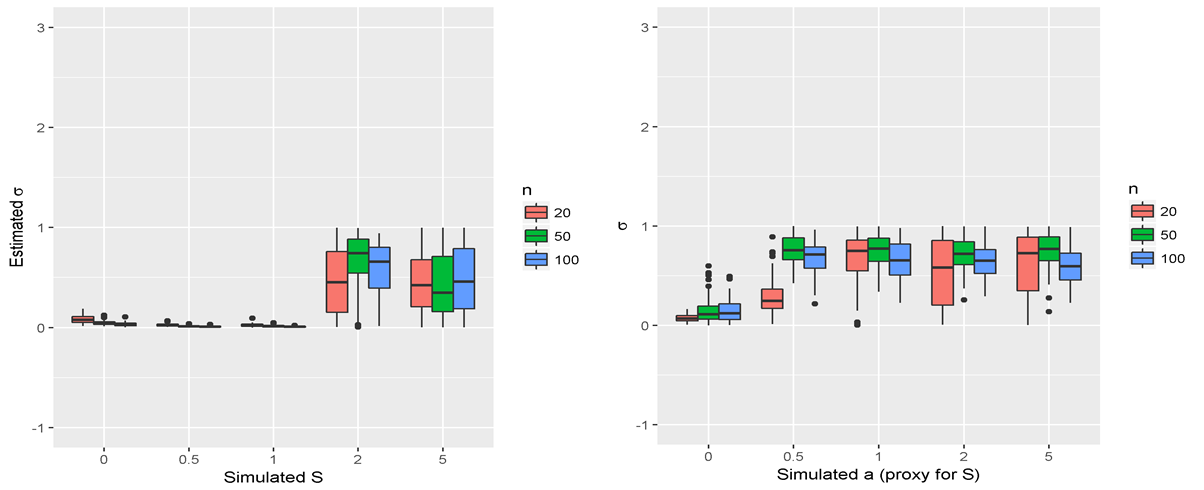


Match between estimated S and the S values assumed in simulation using the Matching Competition model.



Relationships between estimated S and assumed *a* when the MC model is fit to trait distributions produced by L-V model.

 Estimates of 𝛿 (the weight for OU process) for a) simulations using the MC process with different values of S, b) simulations using the L-V model with different values of a.



Parameter 𝜎 (BM variance for a) simulations using the MC process with different values of S, b) simulations using the L-V model with different values of *a*.

**References:**

1. Nuismer, S. L., & harmon, l. J. (2016). Predicting rates of interspecific interaction from phylogenetic trees. Ecology letters, 19(977–986), 17–27. Https://doi.org/10.1111/ele.12384
2. Butler, m. A., & king, a. A. (2004). Phylogenetic comparative analysis: a modeling approach for adaptive evolution. The american naturalist, 164(6), 683-695.
3. Beaumont, m. A. (2010). Approximate bayesian computation in evolution and ecology. Annual review of ecology, evolution, and systematics, 41, 379-406.
4. Sunnåker, m., busetto, a. G., numminen, e., corander, j., foll, m., & dessimoz, c. (2013). Approximate bayesian computation. Plos computational biology, 9(1), e1002803.
5. Losos, j. B., & ricklefs, r. E. (2009). Adaptation and diversification on islands. *Nature*, *457*(february). Https://doi.org/10.1038/nature07893
6. Mahler, d. L., revell, l. J., glor, r. E., & losos, j. B. (2010). Ecological opportunity and the rate of morphological evolution in the diversification of greater antillean anoles. *Evolution*, 2731–2745. Https://doi.org/10.1111/j.1558-5646.2010.01026.x
7. Mahler, d. L., ingram, t., revell, l. J., & losos, j. B. (2013). Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science*, *341*(6143), 292-295.

Appendix:

R code:

library(ape)  
library(phytools)  
library(geiger)  
library(ggplot2)  
library(cowplot)  
  
  
#Input phylogenetic tree from Mahler et.al.(2013)  
tree<-read.tree("GA\_Anolis\_MCC.tre")  
  
#Input trait data  
Data<-read.csv("GA\_Anolis\_traits.csv")  
  
#Trait to analyze is SVL, snout to vent length  
Data<-Data[,2]  
  
##Calculate speciation events  
  
  
#total time to evolve from the common ancestor (in terms of generations~5 years)  
  
t<-10^7  
brt<-branching.times(tree)  
  
#speciation events  
br<-floor(t\*(1-(brt/max(brt))))  
br<-sort(br,decreasing=FALSE)  
  
  
  
#Create a simulation for trait evolution  
#Inputs:S, a coefficient that represents strength of selection due to interactions  
# psi, a coefficient that represents strength of stabilizing abiotic selection  
# theta, an environmental optima   
# delta, brownian motion change (variance)  
# trait values of species  
# ancestral trait value  
  
  
evol<-function(ent,br){  
 ancestor<-ent[1]  
 S<-ent[2]  
 psi<-ent[3]  
 delta<-ent[5]  
 theta<-ent[4]  
 tr<-c(ancestor)  
for(i in 0:1000){  
   
 if(i%in%br){  
 l<-sum(i==br)  
 tr<-c(tr,tr[length(tr)]+rnorm(l,0,0.01))}  
 tr<-tr+S\*(mean(tr)-tr)+psi\*(theta-tr)+rnorm(length(tr),0,delta)  
 tr[which(1>tr)]<-1  
 tr[which(tr>8)]<-8  
}  
 return(tr)  
}  
  
# ABC model  
  
#Draw values for the parameters to know from their prior distributions  
#Set up a rejection threshold for euclidean distance  
#Model: tr.change=tr+S\*(mean(tr)-tr)+psi\*(theta-tr)+rnorm(length(tr),0,delta)  
#Parameters and their priors  
# Ancestral trait ancestor: dunif(1,8)  
# Coefficient for interaction term S:dlnorm(0,2)  
# Coefficient for stabilizing selection psi: dlnorm(0,2)  
# Selection optima 1 theta: dunif(1,8)  
# drift sd per time delta: dunif(0,10)  
  
final<-vector(length=5)  
pb<-txtProgressBar(min =1, max =50000, style = 3)  
for(i in 1:50000){  
 Sys.sleep(0.1)  
 # update progress bar  
 setTxtProgressBar(pb, i)  
 pars<-c(c(runif(1,1,8),rlnorm(2,0,2),runif(1,0,8),runif(1,0,10)))  
 res<-evol(pars,br)  
 if(sqrt(sum((Data-res)^2))<20){final<-rbind(final,pars)}  
}  
  
  
final1<-final[-1,]  
  
#Second filtering  
#create sets of parameter values based on first filtering  
#Deviate param values using rnorm function  
final2<-matrix(nrow=nrow(final1),ncol=ncol(final1))  
for(i in 1:length(final1)){  
 final2[i]<-rnorm(1,final1[i],0.1\*final1[i])  
}  
final3<-matrix(nrow=nrow(final1),ncol=ncol(final1))  
for(i in 1:length(final1)){  
 final3[i]<-rnorm(1,final1[i],0.1\*final1[i])  
}  
  
final.second<-rbind(final1,final2,final3)  
  
  
final.third<-vector(length=5)  
pb<-txtProgressBar(min = 1, nrow(final.second), style = 3)  
for(i in 1:nrow(final.second)){  
 Sys.sleep(0.1)  
 # update progress bar  
 setTxtProgressBar(pb, i)  
 pars<-c(final.second[i,])  
 res<-evol(pars,br)  
 if(sqrt(sum((Data-res)^2))<10){final.third<-rbind(final.third,pars)}  
}  
  
  
  
  
#Plot overlap of priors and posteriors  
  
par(mfrow=c(3,2))  
  
prior.A<-data.frame(points=runif(1000,1,8))  
post.A<-data.frame(points=final.third[,1])   
prior.A$type<-"prior for Ancestral trait"  
post.A$type<-"Posterior for Ancestral trait"  
A.dat<-rbind(prior.A,post.A)  
ggplot(A.dat,aes(points,fill=type))+geom\_density(alpha=0.2)+xlim(0,10)  
  
priorS<-data.frame(points=rlnorm(1000,0,2))  
postS<-data.frame(points=final.third[,2])   
priorS$type<-"prior for S"  
postS$type<-"Posterior for S"  
Sdat<-rbind(priorS,postS)  
ggplot(Sdat,aes(points,fill=type))+geom\_density(alpha=0.2)+xlim(0,20)  
  
prior.psi<-data.frame(points=rlnorm(1000,0,2))  
post.psi<-data.frame(points=final.third[,3])  
prior.psi$type<-"prior for psi"  
post.psi$type<-"Posterior for psi"  
psidat<-rbind(prior.psi,post.psi)  
ggplot(psidat,aes(points,fill=type))+geom\_density(alpha=0.2)+xlim(0,20)  
  
prior.delta<-data.frame(points=runif(1000,0,10))  
post.delta<-data.frame(points=final.third[,5])   
prior.delta$type<-"prior for delta"  
post.delta$type<-"Posterior for delta"  
delta.dat<-rbind(prior.delta,post.delta)  
ggplot(delta.dat,aes(points,fill=type))+geom\_density(alpha=0.2)  
  
prior.theta<-data.frame(points=runif(1000,0,8))  
post.theta<-data.frame(points=final.third[,4])   
prior.theta$type<-"prior for theta"  
post.theta$type<-"Posterior for theta"  
theta.dat<-rbind(prior.theta,post.theta)  
ggplot(theta.dat,aes(points,fill=type))+geom\_density(alpha=0.2)  
  
ggplot(postS,aes(points,fill=type))+geom\_density()+xlim(0,2)  
ggplot(post.psi,aes(points,fill=type))+geom\_density()+xlim(0,2)  
compare<-rbind(postS,post.psi)  
ggplot(compare,aes(points,fill=type))+geom\_density(alpha=0.2)+xlim(0,2.5)  
  
#Summary of posterior values  
means<-apply(final.third,2,mean)  
  
mostlike<-function(x){  
 a<-density(x)  
 return(a$x[which.max(a$y)])  
}  
modes<-apply(final.third,2,mostlike)  
sds<-apply(final.third,2,sd)  
  
Summary<-cbind(c("A","S","Psi","Theta","Delta"),means,modes,sds)  
Summary<-data.frame(Parameters=c("A","S","Psi","Theta","Delta"),Mean=means,Mode=modes,SD=sds)  
  
  
#Model fit  
  
simdat<-evol(means,br)  
  
simdat1<-sort(simdat,decreasing = F)  
Data1<-sort(Data,decreasing = F)  
result<-data.frame(Data=Data1,Simulation=simdat1)  
ggplot(result,aes(x=Data,y=Simulation))+geom\_point(size=2)+geom\_abline(slope=1,intercept = 0,color="red")+geom\_smooth(method="lm")