**SUPPLEMENTARY MATERIALS**

**Supplementary material 2: R scripts**

**A)**

**#BioGeoBEARS script 🡪 Biogeographic stochastic mapping (BSM)**

#Load required libraries

library(optimx)

library(FD)

library(snow)

library(parallel)

library(BioGeoBEARS)

source("http://phylo.wdfiles.com/local--files/biogeobears/cladoRcpp.R") # (needed now that traits model added; source FIRST!)

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_add\_fossils\_randomly\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_basics\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_calc\_transition\_matrices\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_classes\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_detection\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_DNA\_cladogenesis\_sim\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_extract\_Qmat\_COOmat\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_generics\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_models\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_on\_multiple\_trees\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_plots\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_readwrite\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_simulate\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_SSEsim\_makePlots\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_SSEsim\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_stochastic\_mapping\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_stratified\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_univ\_model\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/calc\_uppass\_probs\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/calc\_loglike\_sp\_v01.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/get\_stratified\_subbranch\_top\_downpass\_likelihoods\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/runBSM\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/stochastic\_map\_given\_inputs.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/summarize\_BSM\_tables\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_traits\_v1.R") # added traits model

calc\_loglike\_sp = compiler::cmpfun(calc\_loglike\_sp\_prebyte) # crucial to fix bug in uppass calculations

calc\_independent\_likelihoods\_on\_each\_branch = compiler::cmpfun(calc\_independent\_likelihoods\_on\_each\_branch\_prebyte)

##based on the BiogeoBEARS object that contain the best-supported BioGeoBEARS model

model\_name = "DEC"

# res = resDEC

clado\_events\_tables = NULL

ana\_events\_tables = NULL

lnum = 0

BSM\_inputs\_fn = "BSM\_inputs\_file.Rdata"

runInputsSlow = TRUE

if (runInputsSlow)

{

stochastic\_mapping\_inputs\_list = get\_inputs\_for\_stochastic\_mapping(res=res)

save(stochastic\_mapping\_inputs\_list, file=BSM\_inputs\_fn)

} else {

# Loads to "stochastic\_mapping\_inputs\_list"

load(BSM\_inputs\_fn)

} # END if (runInputsSlow)

names(stochastic\_mapping\_inputs\_list)

stochastic\_mapping\_inputs\_list$phy2

stochastic\_mapping\_inputs\_list$COO\_weights\_columnar

stochastic\_mapping\_inputs\_list$unconstr

set.seed(seed=as.numeric(Sys.time()))

runBSMslow = TRUE

if (runBSMslow == TRUE)

{

# Saves to: RES\_clado\_events\_tables.Rdata

# Saves to: RES\_ana\_events\_tables.Rdata

BSM\_output = runBSM(res, stochastic\_mapping\_inputs\_list=stochastic\_mapping\_inputs\_list, maxnum\_maps\_to\_try=100, nummaps\_goal=1000, maxtries\_per\_branch=40000, save\_after\_every\_try=TRUE, savedir=getwd(), seedval=12345, wait\_before\_save=0.01)

RES\_clado\_events\_tables = BSM\_output$RES\_clado\_events\_tables

RES\_ana\_events\_tables = BSM\_output$RES\_ana\_events\_tables

} else {

# Load previously saved...

# Loads to: RES\_clado\_events\_tables

load(file="RES\_clado\_events\_tables.Rdata")

# Loads to: RES\_ana\_events\_tables

load(file="RES\_ana\_events\_tables.Rdata")

BSM\_output = NULL

BSM\_output$RES\_clado\_events\_tables = RES\_clado\_events\_tables

BSM\_output$RES\_ana\_events\_tables = RES\_ana\_events\_tables

} # END if (runBSMslow == TRUE)

clado\_events\_tables = BSM\_output$RES\_clado\_events\_tables

ana\_events\_tables = BSM\_output$RES\_ana\_events\_tables

head(clado\_events\_tables[[1]])

head(ana\_events\_tables[[1]])

length(clado\_events\_tables)

length(ana\_events\_tables)

head(clado\_events\_tables[[1]])

head(ana\_events\_tables[[1]])

length(clado\_events\_tables)

length(ana\_events\_tables)

length(clado\_events\_tables)

length(ana\_events\_tables)

head(clado\_events\_tables[[1]][,-20])

tail(clado\_events\_tables[[1]][,-20])

head(ana\_events\_tables[[1]])

tail(ana\_events\_tables[[1]])

areanames = names(tipranges@df)

actual\_names = areanames

actual\_names

dmat\_times = get\_dmat\_times\_from\_res(res=res, numstates=NULL)

dmat\_times

clado\_events\_tables = BSM\_output$RES\_clado\_events\_tables

ana\_events\_tables = BSM\_output$RES\_ana\_events\_tables

BSMs\_w\_sourceAreas = simulate\_source\_areas\_ana\_clado(res, clado\_events\_tables, ana\_events\_tables, areanames)

clado\_events\_tables = BSMs\_w\_sourceAreas$clado\_events\_tables

ana\_events\_tables = BSMs\_w\_sourceAreas$ana\_events\_tables

counts\_list = count\_ana\_clado\_events(clado\_events\_tables, ana\_events\_tables, areanames, actual\_names)

hist\_event\_counts(counts\_list, pdffn=paste0(model\_name, "\_histograms\_of\_event\_counts.pdf"))

**B)**

**#BioGeoBEARS script🡪 Ancestral area reconstruction**

#Load required libraries

library(optimx)

library(FD)

library(snow)

library(parallel)

library(BioGeoBEARS)

source("http://phylo.wdfiles.com/local--files/biogeobears/cladoRcpp.R") # (needed now that traits model added; source FIRST!)

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_add\_fossils\_randomly\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_basics\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_calc\_transition\_matrices\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_classes\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_detection\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_DNA\_cladogenesis\_sim\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_extract\_Qmat\_COOmat\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_generics\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_models\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_on\_multiple\_trees\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_plots\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_readwrite\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_simulate\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_SSEsim\_makePlots\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_SSEsim\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_stochastic\_mapping\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_stratified\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_univ\_model\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/calc\_uppass\_probs\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/calc\_loglike\_sp\_v01.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/get\_stratified\_subbranch\_top\_downpass\_likelihoods\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/runBSM\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/stochastic\_map\_given\_inputs.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/summarize\_BSM\_tables\_v1.R")

source("http://phylo.wdfiles.com/local--files/biogeobears/BioGeoBEARS\_traits\_v1.R") # added traits model

calc\_loglike\_sp = compiler::cmpfun(calc\_loglike\_sp\_prebyte) # crucial to fix bug in uppass calculations

calc\_independent\_likelihoods\_on\_each\_branch = compiler::cmpfun(calc\_independent\_likelihoods\_on\_each\_branch\_prebyte)

#setting the working directory

extdata\_dir = np(system.file("extdata", package="BioGeoBEARS"))

extdata\_dir

list.files(extdata\_dir)

#read files (phylogenetic dated tree and area data)

trfn = np(paste(addslash(extdata\_dir), "Platyrrhini.nwk", sep=""))

moref(trfn)

tr = read.tree(trfn)

geogfn = np(paste(addslash(extdata\_dir), "Platyrrhini\_area.data", sep=""))

tipranges = getranges\_from\_LagrangePHYLIP(lgdata\_fn=geogfn)

#setting BiogeoBEARS object

max\_range\_size = 3

BioGeoBEARS\_run\_object = define\_BioGeoBEARS\_run()

BioGeoBEARS\_run\_object$trfn = trfn

BioGeoBEARS\_run\_object$geogfn = geogfn

BioGeoBEARS\_run\_object$max\_range\_size = max\_range\_size

BioGeoBEARS\_run\_object$min\_branchlength = 0.000001

BioGeoBEARS\_run\_object$include\_null\_range = TRUE

BioGeoBEARS\_run\_object$speedup = TRUE

BioGeoBEARS\_run\_object$use\_optimx = TRUE

BioGeoBEARS\_run\_object$num\_cores\_to\_use = 3

BioGeoBEARS\_run\_object$force\_sparse = FALSE

BioGeoBEARS\_run\_object$return\_condlikes\_table = TRUE

BioGeoBEARS\_run\_object$calc\_TTL\_loglike\_from\_condlikes\_table = TRUE

BioGeoBEARS\_run\_object$calc\_ancprobs = TRUE

BioGeoBEARS\_run\_object

check\_BioGeoBEARS\_run(BioGeoBEARS\_run\_object)

# run DEC model

runslow = TRUE

resfn = "Platyrrhini\_DEC.Rdata"

if (runslow)

{

res = bears\_optim\_run(BioGeoBEARS\_run\_object)

res

save(res, file=resfn)

resDEC = res

} else {

# Loads to "res"

load(resfn)

resDEC = res

}

### Run DEC+J

# setting BiogeoBEARS object

BioGeoBEARS\_run\_object = define\_BioGeoBEARS\_run()

BioGeoBEARS\_run\_object$trfn = trfn

BioGeoBEARS\_run\_object$geogfn = geogfn

BioGeoBEARS\_run\_object$max\_range\_size = max\_range\_size

BioGeoBEARS\_run\_object$min\_branchlength = 0.000001

BioGeoBEARS\_run\_object$include\_null\_range = TRUE

BioGeoBEARS\_run\_object$speedup = TRUE

BioGeoBEARS\_run\_object$use\_optimx = TRUE

BioGeoBEARS\_run\_object$num\_cores\_to\_use = 1

BioGeoBEARS\_run\_object$force\_sparse = FALSE

BioGeoBEARS\_run\_object = readfiles\_BioGeoBEARS\_run(BioGeoBEARS\_run\_object)

# Good default settings to get ancestral states

BioGeoBEARS\_run\_object$return\_condlikes\_table = TRUE

BioGeoBEARS\_run\_object$calc\_TTL\_loglike\_from\_condlikes\_table = TRUE

BioGeoBEARS\_run\_object$calc\_ancprobs = TRUE

# Set up DEC+J model (Adding the parameter "j")

dstart = resDEC$outputs@params\_table["d","est"]

estart = resDEC$outputs@params\_table["e","est"]

jstart = 0.0001

# Input starting values for d, e

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["d","init"] = dstart

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["d","est"] = dstart

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["e","init"] = estart

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["e","est"] = estart

# Add j as a free parameter

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["j","type"] = "free"

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["j","init"] = jstart

BioGeoBEARS\_run\_object$BioGeoBEARS\_model\_object@params\_table["j","est"] = jstart

check\_BioGeoBEARS\_run(BioGeoBEARS\_run\_object)

#run DEC+j model

resfn = "Platyrrhini\_DEC+J.Rdata"

runslow = TRUE

if (runslow)

{

 #sourceall("/Dropbox/\_njm/\_\_packages/BioGeoBEARS\_setup/")

 res = bears\_optim\_run(BioGeoBEARS\_run\_object)

 res

 save(res, file=resfn)

 resDECj = res

} else {

 # Loads to "res"

 load(resfn)

 resDECj = res

}

## Model comparison

LnL\_2 = get\_LnL\_from\_BioGeoBEARS\_results\_object(resDEC)

LnL\_1 = get\_LnL\_from\_BioGeoBEARS\_results\_object(resDECj)

numparams1 = 3

numparams2 = 2

stats = AICstats\_2models(LnL\_1, LnL\_2, numparams1, numparams2)

stats

# DEC, null model for Likelihood Ratio Test (LRT)

res2 = extract\_params\_from\_BioGeoBEARS\_results\_object(results\_object=resDEC, returnwhat="table", addl\_params=c("j"), paramsstr\_digits=4)

# DEC+J, alternative model for Likelihood Ratio Test (LRT)

res1 = extract\_params\_from\_BioGeoBEARS\_results\_object(results\_object=resDECj, returnwhat="table", addl\_params=c("j"), paramsstr\_digits=4)

rbind(res2, res1)

tmp\_tests = conditional\_format\_table(stats)

restable = rbind(restable, res2, res1)

teststable = rbind(teststable, tmp\_tests)

##plot\_DEC (best supported model)

analysis\_titletxt ="BioGeoBEARS DEC on Platyrrhini"

# Setup

results\_object = resDEC

scriptdir = np(system.file("extdata/a\_scripts", package="BioGeoBEARS"))

# States

res2 = plot\_BioGeoBEARS\_results(results\_object, analysis\_titletxt, addl\_params=list("j"), plotwhat="text", label.offset=0.45, tipcex=0.7, statecex=0.7, splitcex=0.6, titlecex=0.8, plotsplits=FALSE, cornercoords\_loc=scriptdir, include\_null\_range=TRUE, tr=tr, tipranges=tipranges)

# Pie chart

plot\_BioGeoBEARS\_results(results\_object, analysis\_titletxt, addl\_params=list("j"), plotwhat="pie", label.offset=0.45, tipcex=0.7, statecex=0.7, splitcex=0.6, titlecex=0.8, plotsplits=FALSE, cornercoords\_loc=scriptdir, include\_null\_range=TRUE, tr=tr, tipranges=tipranges)

**C)**

**Quantile regression script**

#load package quantreg: quantile regression#

library(quantreg)

data<-read.csv("data.csv", header=T,sep=";", dec=".")

attach(data)

#variables

y <- cbind(richness)

x <- cbind(aet, bio2, bio3, bio5, bio7, bio11, bio12, bio13, bio16, bio17, bio19, pet, densidad)

summary(x)

summary(y)

#OLS

regression<- lm(y ~ x, data= data)

summary(regression)

#quantile regression

quantreg25<- rq(y ~ x, data= data, tau=0.25)

summary(quantreg25)

quantreg50<- rq(y ~ x, data= data, tau=0.5)

summary(quantreg50)

quantreg75<- rq(y ~ x, data= data, tau=0.75)

summary(quantreg75)

#Regression by quantiles simultaneously###

quantreg2575<- rq(y ~ x, data= data, tau=c(0.25, 0.75))

summary(quantreg2575)

#Anova###

anova(quantreg25, quantreg75)

#plotting data

quantreg.all<- rq(y ~ x, tau=seq(0.05, 0.95, by= 0.05))

quantreg.plot<-summary(quantreg.all)

plot(quantreg.plot)

quantreg.all<- rq(y ~ x, tau=seq(0.01, 0.99, by= 0.01))

summary(quantreg.all)

qt0.01<-rq(y~x,tau=0.01)

summary(qt0.01,se="boot",R=1000)

qt0.99<-rq(y~x,tau=0.99)

summary(qt0.99,se="boot",R=1000)

plot((y~x),pch=16,cex=0.7, ylab="Richness",cex.lab=1.5,cex.axis=1,xlab="AET")

abline(qt0.01,lwd=2, col=2)

abline(qt0.99,lwd=2, col=2)