

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

Liam J. Revell et al.

2023-11-29

This code reiterates the analysis of *Anolis dewlap* color 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 & data from files
treed <- read.tree("dewlap_tree_22feb2018.phy")
dat <- read.csv("dewlap_colors_22feb2018.csv", row.names = 1)

## extract dominant color & substitute summary colors
dat$domcol <- as.character(dat$dominant.color)
dat$domcol[dat$domcol %in% c("black", "blue", "purple")] <- "black"
dat$domcol[dat$domcol %in%
  c("green", "yellow-green", "yellow", "brown/yellow")] <- "yellow"
dat$domcol[dat$domcol %in% c("red", "")] <- "red" #"" is wermuthi, clearly red
dat$domcol[dat$domcol %in% c("orange", "brown")] <- "orange"
dat$domcol[dat$domcol %in% c("peach", "pink")] <- "pink"
dat$domcol[dat$domcol %in% c("white", "grey")] <- "white"

## dominant color - this has been aggregated to group (somewhat)
## similar colors and get it down to a manageable number with decent
```

```

## sample sizes
summary(as.factor(dat$domcol))

## black orange pink red white yellow
## 9 39 37 22 38 39

## check names
all(treed$tip.label == rownames(dat))

## [1] TRUE

## paint the tree with (predominantly) mainland vs island clades.
## There are a few minor exceptions, since I wanted to have reasonably
## large clades for a split QuaSSE analysis. Could also use the "Island"
## variable to split off the mainland species.
nodeA <- getMRCA(treed, c("auratus", "occultus"))
treedA <- paintSubTree(treed, nodeA, "I", "M", stem=0.5)
nodeB <- getMRCA(treed, c("auratus", "sericeus"))
treedB <- paintSubTree(treedA, nodeB, "M", stem=0.5)
nodeC1 <- getMRCA(treed, c("grahami", "reconditus"))
treedC1 <- paintSubTree(treedB, nodeC1, "I", "M", stem=0.5)
nodeC <- getMRCA(treed, c("luciae", "roquet"))
treedC <- paintSubTree(treedC1, nodeC, "I", stem=0.5)

cols<-setNames(c("brown", "blue"), c("M", "I"))
tmp<-treedC
tmp$tip.label<-paste(" ", tmp$tip.label, " ", sep="")
plot(tmp, cols, fsize=0.7, type="arc", arc_height=0.1, lwd=3, ftype="i",
mar=rep(0.5, 4), part=1)
DomCol <- as.factor(setNames(as.character(dat$domcol), rownames(dat)))
tiplabels(pie=to.matrix(DomCol[treedC$tip.label],
levels(DomCol)), piecol=levels(DomCol), cex=0.3)
obj<-legend(x="topleft", legend=c("mainland", "islands"), pch=22, pt.cex=2,
pt.bg=c("brown", "blue"), bty="n", title="Geographic context", title.adj=0,
plot=FALSE)
leg.cols<-c("black, blue, & purple",
"orange & brown",
"pink & peach",
"red",
"white & grey",
"yellow & green")
text.x<-legend(x=obj$rect$left, y=obj$rect$top-10, legend=leg.cols, title.adj=0,
pch=21, pt.cex=2, pt.bg=levels(DomCol), bty="n", title="Dewlap color")$text$x[1]
foo<-function(n, text.x){
leg.text<-paste(c("mainland", "islands"), paste(rep(" ", n), collapse=""), sep="")
curr.x<-legend(x="topleft", legend=leg.text, pch=22, pt.cex=2,
pt.bg=c("brown", "blue"), bty="n", title="Geographic context",
title.adj=0, plot=FALSE)$text$x[1]
abs(curr.x-text.x)
}
fit<-optimize(foo, c(0, 10), text.x=text.x)
legend(x="topleft",
legend=paste(c("mainland", "islands"), paste(rep(" ", round(fit$minimum)),
collapse=""), sep=""),
pch=22, pt.cex=2,

```

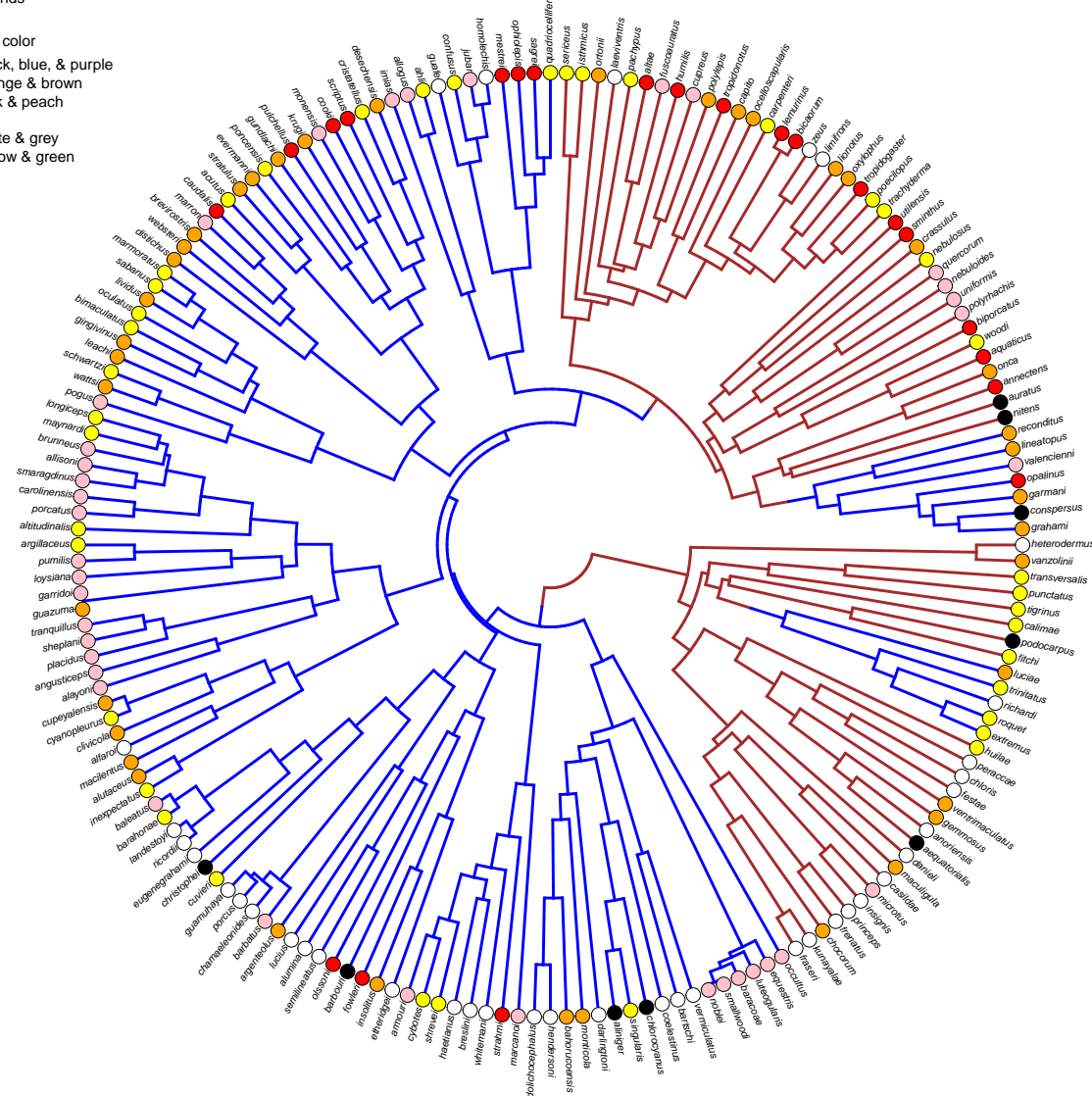
```
pt.bg=c("brown","blue"),bty="n",title="Geographic context",
title.adj=0)
```

Geographic context

■ mainland
■ islands

Dewlap color

● black, blue, & purple
● orange & brown
● pink & peach
● red
● white & grey
● yellow & green



```
pdf(file="Figure_3.pdf",width=13.33,height=13.33)
cols<-setNames(c("brown","blue"),c("M","I"))
tmp<-treedC
tmp$tip.label<-paste(" ",tmp$tip.label," ",sep="")
plot(tmp,cols,fs=0.7,type="arc",arc_height=0.1,lwd=3,ftype="i",
mar=rep(0.5,4),part=1)
DomCol <- as.factor(setNames(as.character(dat$domcol), rownames(dat)))
tiplabels(pie=to.matrix(DomCol[treedC$tip.label],
levels(DomCol)),piecol=levels(DomCol),cex=0.3)
obj<-legend(x="topleft",legend=c("mainland","islands"),pch=22,pt.cex=2,
pt.bg=c("brown","blue"),bty="n",title="Geographic context",title.adj=0,
```

```

    plot=FALSE)
leg.cols<-c("black, blue, & purple",
"orange & brown",
"pink & peach",
"red",
"white & grey",
"yellow & green")
text.x<-legend(x=obj$rect$left,y=obj$rect$top-10,legend=leg.cols,title.adj=0,
pch=21,pt.cex=2,pt.bg=levels(DomCol),bty="n",title="Dewlap color")$text$x[1]
foo<-function(n,text.x){
  leg.text<-paste(c("mainland","islands"),paste(rep(" ",n),collapse=""),sep="")
  curr.x<-legend(x="topleft",legend=leg.text,pch=22,pt.cex=2,
pt.bg=c("brown","blue"),bty="n",title="Geographic context",
title.adj=0,plot=FALSE)$text$x[1]
  abs(curr.x-text.x)
}
fit<-optimize(foo,c(0,10),text.x=text.x)
legend(x="topleft",
legend=paste(c("mainland","islands"),paste(rep(" ",round(fit$minimum)),
collapse=""),sep=""),
pch=22,pt.cex=2,
pt.bg=c("brown","blue"),bty="n",title="Geographic context",
title.adj=0)
dev.off()

```

```
## pdf
```

```
## 2
```

```
## set number of optimization iterations per model
```

```
niter<-50
```

```
## open cluster for parallelized optimization
```

```
ncores<-min(niter,parallel::detectCores()-2)
```

```
mc<-makeCluster(ncores,type="PSOCK")
```

```
registerDoParallel(cl=mc)
```

```
## single-rate ER model
```

```
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitMk(treedC, DomCol, model="ER",
rand_start=TRUE)
}
```

```
logL<-sapply(fits,logLik)
```

```
print(logL)
```

```
## [1] -329.6837 -329.6837 -329.7925 -329.6837 -329.6837 -329.6837 -329.6837
```

```
## [8] -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837
```

```
## [15] -329.7925 -329.6837 -329.6837 -329.6837 -329.6837 -329.7925 -329.6837
```

```
## [22] -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837
```

```
## [29] -329.6837 -329.6837 -308.6466 -329.6837 -329.6837 -329.6837 -329.6837
```

```
## [36] -329.6837 -329.6837 -308.6466 -329.6837 -329.6837 -329.6837 -308.6466
```

```
## [43] -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837 -329.6837
```

```
## [50] -329.6837
```

```
fitER1<-fits[[which(logL==max(logL))[1]]]
```

```
print(fitER1,digits=3)
```

```
## Object of class "fitMk".
```

```

##
## Fitted (or set) value of Q:
##      black orange  pink    red  white yellow
## black -0.047  0.009  0.009  0.009  0.009  0.009
## orange 0.009 -0.047  0.009  0.009  0.009  0.009
## pink   0.009  0.009 -0.047  0.009  0.009  0.009
## red    0.009  0.009  0.009 -0.047  0.009  0.009
## white  0.009  0.009  0.009  0.009 -0.047  0.009
## yellow 0.009  0.009  0.009  0.009  0.009 -0.047
##
## Fitted (or set) value of pi:
##  black orange  pink    red  white yellow
##  0.167  0.167  0.167  0.167  0.167  0.167
## due to treating the root prior as (a) flat.
##
## Log-likelihood: -308.647
##
## Optimization method used was "nlminb"
##
## R thinks it has found the ML solution.
## single-rate SYM model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitMk(treedC, DomCol, model="SYM",
    rand_start=TRUE)
}
logL<-sapply(fits,logLik)
print(logL)

## [1] -294.5123 -302.3786 -290.0942 -294.0484 -299.2240 -304.2455 -307.8959
## [8] -302.5765 -289.8421 -289.9577 -296.4326 -294.2063 -306.6359 -306.9420
## [15] -295.9510 -314.9154 -291.7308 -301.4078 -289.8064 -303.9513 -301.2521
## [22] -307.1975 -301.7130 -294.1298 -311.1527 -294.2544 -290.2203 -305.6770
## [29] -306.8875 -296.7208 -289.9012 -290.2563 -310.4354 -294.5202 -304.7176
## [36] -294.1298 -289.8914 -294.2291 -296.6076 -295.5401 -294.9699 -300.1345
## [43] -289.8196 -307.0777 -293.3803 -294.7337 -289.8064 -292.6309 -294.5755
## [50] -314.3250

fitSYM1<-fits[[which(logL==max(logL))[1]]]
print(fitSYM1,digits=3)

## Object of class "fitMk".
##
## Fitted (or set) value of Q:
##      black orange  pink    red  white yellow
## black -0.012  0.000  0.000  0.000  0.005  0.007
## orange 0.000 -1.034  0.027  0.000  0.023  0.984
## pink   0.000  0.027 -0.036  0.004  0.005  0.000
## red    0.000  0.000  0.004 -0.046  0.000  0.042
## white  0.005  0.023  0.005  0.000 -0.032  0.000
## yellow 0.007  0.984  0.000  0.042  0.000 -1.033
##
## Fitted (or set) value of pi:
##  black orange  pink    red  white yellow
##  0.167  0.167  0.167  0.167  0.167  0.167

```

```

## due to treating the root prior as (a) flat.
##
## Log-likelihood: -289.806
##
## Optimization method used was "nlminb"
##
## R thinks optimization may not have converged.
## single-rate ARD model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitMk(treedC, DomCol, model="ARD",
    rand_start=TRUE)
}
logL<-sapply(fits,logLik)
print(logL)

## [1] -301.0275 -294.1004 -299.2395 -296.9226 -282.9429 -282.6902 -304.9173
## [8] -282.6909 -282.6902 -292.7863 -300.3837 -282.9734 -282.8866 -282.6902
## [15] -285.6397 -309.1580 -294.1305 -282.6902 -310.6702 -282.6902 -293.7822
## [22] -282.6902 -282.8033 -301.1140 -289.1051 -289.5726 -284.3075 -284.8165
## [29] -303.8689 -282.8033 -288.9210 -305.8427 -285.0999 -282.8228 -307.1712
## [36] -295.0381 -282.8033 -284.2045 -307.9174 -282.6909 -296.6556 -282.9429
## [43] -307.3684 -285.3845 -298.2665 -305.3033 -299.9302 -304.3063 -295.3983
## [50] -284.4858

fitARD1<-fits[[which(logL==max(logL))[1]]]
print(fitARD1,digits=3)

## Object of class "fitMk".
##
## Fitted (or set) value of Q:
##      black orange  pink    red  white yellow
## black -0.043  0.000  0.000  0.000  0.000  0.043
## orange  0.005 -0.063  0.001  0.035  0.004  0.018
## pink    0.000  0.004 -0.033  0.000  0.000  0.028
## red     0.000  0.000  0.040 -0.057  0.000  0.017
## white   0.007  0.016  0.011  0.000 -0.033  0.000
## yellow  0.000  0.043  0.000  0.000  0.012 -0.055
##
## Fitted (or set) value of pi:
##      black orange  pink    red  white yellow
##      0.167  0.167  0.167  0.167  0.167  0.167
## due to treating the root prior as (a) flat.
##
## Log-likelihood: -282.69
##
## Optimization method used was "nlminb"
##
## R thinks it has found the ML solution.
## multi-rate ER model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitmultiMk(treedC, DomCol, model="ER",
    rand_start=TRUE)
}
logL<-sapply(fits,logLik)

```

```

print(logL)

## [1] -322.9163 -322.9163 -315.8149 -322.9163 -322.9163 -322.9163 -322.9163
## [8] -322.9163 -322.9163 -322.9163 -322.9163 -322.9163 -322.9163 -322.9163
## [15] -315.8149 -322.9163 -322.9163 -308.6463 -322.9163 -322.9163 -315.8149
## [22] -322.9163 -322.9163 -322.9163 -322.9163 -322.9163 -315.8149 -322.9163
## [29] -315.8149 -322.9163 -315.8149 -322.9163 -315.8149 -315.8149 -322.9163
## [36] -322.9163 -308.6463 -322.9163 -322.9163 -322.9163 -322.9163 -322.9163
## [43] -322.9163 -322.9163 -322.9163 -322.9163 -322.9163 -315.8149 -322.9163
## [50] -322.9163

fitERmulti1<-fits[[which(logL==max(logL))[1]]]
print(fitERmulti1,digits=3)

## Object of class "fitmultiMk".
##
## Fitted value of Q[I]:
##      black orange  pink    red  white yellow
## black -0.047  0.009  0.009  0.009  0.009  0.009
## orange 0.009 -0.047  0.009  0.009  0.009  0.009
## pink   0.009  0.009 -0.047  0.009  0.009  0.009
## red    0.009  0.009  0.009 -0.047  0.009  0.009
## white  0.009  0.009  0.009  0.009 -0.047  0.009
## yellow 0.009  0.009  0.009  0.009  0.009 -0.047
##
## Fitted value of Q[M]:
##      black orange  pink    red  white yellow
## black -0.047  0.009  0.009  0.009  0.009  0.009
## orange 0.009 -0.047  0.009  0.009  0.009  0.009
## pink   0.009  0.009 -0.047  0.009  0.009  0.009
## red    0.009  0.009  0.009 -0.047  0.009  0.009
## white  0.009  0.009  0.009  0.009 -0.047  0.009
## yellow 0.009  0.009  0.009  0.009  0.009 -0.047
##
## Fitted (or set) value of pi:
##      black  orange  pink    red  white  yellow
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
##
## Log-likelihood: -308.646
##
## Optimization method used was "nlminb"

## multi-rate SYM model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitmultiMk(treedC, DomCol, model="SYM",
    rand_start=TRUE)
}
logL<-sapply(fits,logLik)
print(logL)

## [1] -297.4591 -285.6250 -301.8123 -297.6323 -290.3716 -309.3595 -299.1796
## [8] -296.6057 -305.8030 -303.7292 -301.4877 -303.2345 -298.6021 -292.8361
## [15] -290.7471 -291.6367 -297.2147 -297.3854 -301.2607 -291.8893 -301.1555
## [22] -305.5389 -301.2071 -306.2246 -290.1360 -305.6466 -306.3225 -292.9118
## [29] -299.1138 -298.0740 -302.7806 -291.2260 -303.6982 -296.5576 -294.8068

```

```
## [36] -312.7621 -305.6971 -301.7048 -289.0033 -301.2496 -305.5603 -311.8261
## [43] -292.8024 -302.1161 -292.7356 -301.9105 -302.1070 -308.9463 -306.5689
## [50] -299.2918
```

```
fitSYMmulti1<-fits[[which(logL==max(logL))[1]]]
print(fitSYMmulti1,digits=3)
```

```
## Object of class "fitmultiMk".
##
## Fitted value of Q[I]:
##      black orange  pink    red  white yellow
## black -0.010  0.000  0.000  0.000  0.005  0.005
## orange 0.000 -1.030  0.033  0.017  0.006  0.975
## pink   0.000  0.033 -0.041  0.000  0.008  0.000
## red    0.000  0.017  0.000 -0.029  0.004  0.008
## white  0.005  0.006  0.008  0.004 -0.033  0.010
## yellow 0.005  0.975  0.000  0.008  0.010 -0.998
##
## Fitted value of Q[M]:
##      black orange  pink    red  white yellow
## black -0.016  0.015  0.000  0.000  0.001  0.000
## orange 0.015 -0.104  0.000  0.005  0.027  0.057
## pink   0.000  0.000 -0.027  0.024  0.003  0.000
## red    0.000  0.005  0.024 -0.077  0.000  0.048
## white  0.001  0.027  0.003  0.000 -0.031  0.000
## yellow 0.000  0.057  0.000  0.048  0.000 -0.105
##
## Fitted (or set) value of pi:
##      black  orange  pink    red  white  yellow
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
##
## Log-likelihood: -285.625
##
## Optimization method used was "nlminb"
```

```
## multi-rate ARD model
fits<-foreach(i=1:niter)%dopar%{
  phytools::fitmultiMk(treedC, DomCol, model="ARD",
    rand_start=TRUE)
}
logL<-sapply(fits,logLik)
print(logL)
```

```
## [1] -272.6277 -274.0446 -273.7548 -298.6813 -274.3388 -282.7084 -276.2509
## [8] -274.5008 -274.0380 -278.2453 -298.3574 -278.4477 -283.0903 -275.2531
## [15] -283.6556 -275.2619 -290.2587 -298.2107 -289.6032 -273.1384 -277.8577
## [22] -297.3659 -275.1407 -299.2080 -273.4300 -296.1167 -273.5429 -273.6368
## [29] -275.8367 -280.4391 -280.1030 -277.4754 -280.7186 -274.6254 -277.3239
## [36] -278.3200 -274.3503 -275.4784 -275.9924 -273.5168 -274.7683 -281.4010
## [43] -274.7877 -272.6306 -272.6457 -281.6546 -287.5743 -274.7625 -287.6086
## [50] -272.9211
```

```
fitARDmulti1<-fits[[which(logL==max(logL))[1]]]
print(fitARDmulti1)
```

```
## Object of class "fitmultiMk".
```



```

##
## Fitted value of Q[I]:
##      black      orange      pink      red      white      yellow
## black -0.056781  0.000000  0.000000  0.000000  0.011254  0.045527
## orange 0.004233 -0.092104  0.008465  0.036470  0.000000  0.042937
## pink   0.000000  0.004021 -0.037726  0.000000  0.000000  0.033705
## red    0.000000  0.066798  0.015977 -0.082775  0.000000  0.000000
## white  0.007634  0.001727  0.017626  0.006556 -0.034939  0.001396
## yellow 0.000000  0.077687  0.000000  0.000000  0.014991 -0.092678
##
## Fitted value of Q[M]:
##      black      orange      pink      red      white      yellow
## black -0.070347  0.000000  0.000000  0.000000  0.000000  0.070347
## orange 0.027987 -0.081588  0.000000  0.000000  0.053602  0.000000
## pink   0.000000  0.000000 -0.056090  0.056090  0.000000  0.000000
## red    0.000000  0.046782  0.000000 -0.071719  0.000000  0.024937
## white  0.000231  0.033781  0.004552  0.000000 -0.038565  0.000000
## yellow 0.008885  0.000000  0.000000  0.000000  0.000000 -0.008885
##
## Fitted (or set) value of pi:
##      black      orange      pink      red      white      yellow
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
##
## Log-likelihood: -272.627691
##
## Optimization method used was "nlminb"
stopCluster(cl=mc)

## compare models
anova(fitER1,fitERmulti1,fitSYM1,fitSYMmulti1,
      fitARD1,fitARDmulti1)

##      log(L) d.f.      AIC      weight
## fitER1      -308.6466    1 619.2931 7.818179e-03
## fitERmulti1 -308.6463    2 621.2926 2.876935e-03
## fitSYM1      -289.8064   15 609.6128 9.889125e-01
## fitSYMmulti1 -285.6250   30 631.2501 1.980075e-05
## fitARD1      -282.6902   30 625.3804 3.726146e-04
## fitARDmulti1 -272.6277   60 665.2554 8.175727e-13

## compute the mean rate by regime for each model
## ER-S model
fitER1$rates

## [1] 0.009327247

## ER-M model (island, mainland)
fitERmulti1$rates

## [1] 0.009302223 0.009371025

## SYM-S model
mean(fitSYM1$rates)

## [1] 0.07306317

```

```

## SYM-M model (island, mainland)
mean(fitSYMmulti1$rates[1:15])

## [1] 0.07140516
mean(fitSYMmulti1$rates[16:30])

## [1] 0.01199107
## ARD-S model
mean(fitARD1$rates)

## [1] 0.009442434
## ARD-M model (island, mainland)
mean(fitARDmulti1$rates[1:30])

## [1] 0.01323342
mean(fitARDmulti1$rates[31:60])

## [1] 0.01090646
save.image(file="dewlap-analysis.Rda")

```