R-code for ‘Effects of hunting on black grouse inbreeding and dispersal’

compiled by Rebecca Shuhua Chen

03/05/2022

Contents

This document contains all R-code used in the workflow for the manuscript *“Sex-specific fine-scale population structure and effects of hunting on inbreeding and dispersal in Finnish black grouse (Lyrurus tetrix)”* by Rebecca Shuhua Chen, Carl Soulsbury, Christophe Lebigre, Kees van Oers and Joseph Hoffman (in prep). The raw data can be found on Zenodo as well as in the public GitHub repository together with full R scripts and processed datafiles. Please contact me at rebecca.chen@uni-bielefeld.de for any questions.

Within this markdown file, we follow the same order of analyses as described in the Materials and Methods. However, not all analyses are executed through R, and other softwares were used in combination with our workflow in R to collect all results as presented in the manuscript.

Libraries

The following packages were used in the analyses:

```
library(tidyverse);library(adegenet); library(pegas)
library(data.table);library(hierfstat); library(plot.matrix); library(lme4)
library(forcats); library(ape); library(ParallelStructure)
library(pophelper); library(inbreedR); library(lmerTest); library(DHARMa)
library(performance);library(MuMIn); library(readxl); library(glmmTMB)
library(RColorBrewer); library(extrafont); library(devtools);library(gridExtra)
```

Data

Adults

Raw datasheets are provided in three different formats: 1) unsplit genotypes with one row per individual, 2) STRUCTURE (.stru) datafiles with two consecutive rows per individual, one row per allele. In both these files, populations are coded as integers. Thirdly, a full, easy to read .RData can be found including all details on the sites and hunted status, as well as sex.

```
base::load("data/rawdata/Fulldata_adults.RData")
hunted.ad$site <- as.factor(hunted.ad$site)
summary(hunted.ad[,c(3,7,8,9)])
```

```
##    year     sex            hunt             site
##  2001: 83   F: 813   hunted  : 544   Kummunsuo:308
##  2002:240   M:1065   unhunted:1334   Teerisuo :307
##  2003:214                            Nyrölä   :290
##  2004:234                            Koskenpää:248
##  2005:252                            Lehtusuo :194
##  2006:498                            Saarisuo :161
##  2007:357                            (Other)  :370
```

```
#structure file to summarise genotypes
adults.stru.raw <- read.structure("data/rawdata/Microsat.adults.forstructure.stru",
                        n.ind = 1878, n.loc = 14, onerowperind = F,
                        col.lab = 1, col.pop = 2, col.others = NULL,
                        row.marknames = 0, NA.char = "-9", pop = NULL,
                        sep = NULL,ask = F, quiet = TRUE)

summary(adults.stru.raw)
```

```
##
## // Number of individuals: 1878
## // Group sizes: 248 308 46 194 290 59 62 40 91 161 307 72
## // Number of alleles per locus: 8 11 14 13 9 20 11 9 23 9 10 11 21 5
## // Number of alleles per group: 130 130 99 136 129 96 101 103 102 125 138 118
## // Percentage of missing data: 9.18 %
## // Observed heterozygosity: 0.45 0.81 0.83 0.72 0.78 0.8 0.81 0.18 0.87 0.72 0.8 0.78 0.72 0.19
## // Expected heterozygosity: 0.8 0.8 0.83 0.75 0.8 0.8 0.82 0.18 0.87 0.72 0.8 0.79 0.82 0.19
```

Chicks

```r
base::load("data/rawdata/Fulldata_chicks.RData")
hunted.chick$Site <- as.factor(hunted.chick$Site)
hunted.chick$hunt <- as.factor(hunted.chick$hunt)
hunted.chick$sex <- droplevels(hunted.chick$sex)
summary(hunted.chick[,c(2,9,3,4)])
```

```
##    Year       sex            hunt              Site    
##  2001: 64   F   :370   hunted  : 236   Koskenpää:187  
##  2002:203   M   :325   unhunted:1134   Kummunsuo:409  
##  2003:245   NA's:675                   Lehtusuo :111  
##  2004:202                              Nyrölä   :210  
##  2005:241                              Saarisuo :103  
##  2006:415                              Teerisuo :301  
##                                        Utusuo   : 49
```

```r
#structure file to summarise genotypes
chicks.stru.raw <- read.structure("data/rawdata/Microsat.chicks.forstructure.stru",
                         n.ind = 1370, n.loc = 12, onerowperind = F,
                         col.lab = 1, col.pop = 2, col.others = NULL,
                         row.marknames = 0, NA.char = "-9", pop = NULL,
                         sep = NULL, ask = F, quiet = TRUE)

summary(chicks.stru.raw)
```

```
## 
## // Number of individuals: 1370
## // Group sizes: 187 409 111 210 103 301 49
## // Number of alleles per locus: 9 7 8 17 10 5 19 8 8 10 13
## // Number of alleles per group: 83 95 78 85 72 97 64
## // Percentage of missing data: 5.37 %
## // Observed heterozygosity: 0.82 0.75 0.77 0.78 0.81 0.18 0.88 0.67 0.81 0.79 0.71
## // Expected heterozygosity: 0.83 0.74 0.8 0.8 0.82 0.2 0.89 0.7 0.81 0.79 0.8
```

Test for Hardy-Weinberg equilibrium

We tested for Hardy-Weinberg only in the adult data, as the chick data contains closely related individuals sampled from the same broods.

```r
#### Testing for Hardy-Weinberg equilibrium ####

# First all together
adultHWE.all <- pegas::hw.test(adults.stru.raw, B = 1000)
#B = 1000 for 1000 Monte Carlo permutations

summary(adultHWE.all )
```

```
##      chi^2               df          Pr(chi^2 >)         Pr.exact    
##  Min.   :   7.96   Min.   : 10.00   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:  56.49   1st Qu.: 36.00   1st Qu.:0.00000   1st Qu.:0.0130
##  Median : 122.36   Median : 55.00   Median :0.09383   Median :0.0625
##  Mean   : 541.21   Mean   : 84.14   Mean   :0.33231   Mean   :0.1897
##  3rd Qu.: 451.50   3rd Qu.: 87.75   3rd Qu.:0.66410   3rd Qu.:0.1767
##  Max.   :2340.11   Max.   :253.00   Max.   :0.98916   Max.   :0.8790
```

```
# Then per population using a for loop
adultpop <- seppop(adults.stru.raw)
# Run loop
adultHWE = NULL
for(i in 1:length(adultpop)) {
  hwt <- pegas::hw.test(adultpop[[i]], B=1000)
  smry <- summary(adultpop[[i]])

  Hobs <- smry[[6]]
  Hexp <- smry[[7]]
  pexact <- hwt[,4] #hw.test does chi2 test and exact test.
  #We use p-values of exact test which are given in 4th col
  qval.FDR <- p.adjust(pexact, method = "fdr")
  qval.bon <- p.adjust(pexact, method = "bonferroni")
  adultHWE <- as.data.frame(cbind(adultHWE, Hobs, Hexp, pexact, qval.FDR, qval.bon))

}

sites<-rep(names(adultpop[1:length(adultpop)]),each=5)
adultHWE <- rbind(adultHWE, sites)
adultHWE <- adultHWE[c(nrow(adultHWE),1:(nrow(adultHWE)-1)),]
rownames(adultHWE)[1] <- "Site"

adultHWE.t <- as.data.frame(t(adultHWE))
nums <- c(2:15)
adultHWE.t[nums] <- lapply(adultHWE.t[nums], as.numeric)

adultHWE.t[,c(2:15)]<- round(adultHWE.t[,c(2:15)], 2)
head(adultHWE.t)
```

```
##          Site  L01  L02  L03  L04  L05  L06  L07  L08  L09  L10  L11  L12  L13
## Hobs        1 0.35 0.83 0.81 0.73 0.80 0.84 0.85 0.23 0.83 0.70 0.82 0.77 0.79
## Hexp        1 0.80 0.82 0.81 0.76 0.79 0.81 0.81 0.24 0.89 0.71 0.81 0.77 0.83
## pexact      1 0.00 0.04 0.72 0.36 0.92 0.11 0.59 0.47 0.40 0.11 0.57 0.56 0.00
## qval.FDR    1 0.00 0.21 0.77 0.70 0.92 0.32 0.70 0.70 0.70 0.32 0.70 0.70 0.01
## qval.bon    1 0.00 0.62 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.03
## Hobs.1      2 0.37 0.81 0.85 0.70 0.76 0.79 0.79 0.25 0.89 0.69 0.82 0.73 0.71
##           L14
## Hobs     0.19
## Hexp     0.19
## pexact   0.60
## qval.FDR 0.70
## qval.bon 1.00
## Hobs.1   0.28
```

This table includes Hobs, Hexp, pexact, qval.FDR and qval.bon for each locus in (columns) for each site. The full output of the table can be found in Supplementary Table 3. We included a threshold in which a locus was excluded if the FDR-corrected value was lower than 0.05 in over 70% of the sites. Subsequently, we exclude locus 1 and 13 in both adult and chick data.

Analysing population structure

To investigate patterns of genetic differentiation, we calculated pairwise F_{ST} values in R, conducted an AMOVA (in GUI based software Arlequin, not R), and executed Mantel tests and spatial auto-correlation in GenAlEx (Excel add-in, not R).

Here, we go through calculating summary statistics, constructing a PCA to get a grasp of the distribution of our data and identify potential outliers, and calculating the pairwise F_{ST} values for adult males, adult females and chicks separately.

```
#using filtered structure files excluding loci out of HWE
males.stru <- read.structure("data/cleandata/Microsat.males.noLOCUS1+13.forstructure.stru",
                             n.ind = 1065, n.loc = 12, onerowperind = F,
                             col.lab = 1, col.pop = 2, col.others = NULL,
                             row.marknames = 0, NA.char = "-9", pop = NULL,
                             sep = NULL, ask = F, quiet = T)

females.stru <- read.structure("data/cleandata/Microsat.females.noLOCUS1+13.forstructure.stru",
                               n.ind = 813, n.loc = 12, onerowperind = F,
                               col.lab = 1, col.pop = 2, col.others = NULL,
                               row.marknames = 0, NA.char = "-9", pop = NULL,
                               sep = NULL, ask = F, quiet = T)

chicks.stru <- read.structure("data/cleandata/Microsat.chicks.noLOCUS1+13+14.forstructure.stru",
                              n.ind = 1370, n.loc = 11, onerowperind = F,
                              col.lab = 1, col.pop = 2, col.others = NULL,
                              row.marknames = 0, NA.char = "-9", pop = NULL,
                              sep = NULL, ask = F, quiet = T)

all <- read.structure("data/rawdata/Microsat.all.stru", n.ind = 3248, n.loc = 14,
                      onerowperind = F, col.lab = 1, col.pop = 2, col.others = NULL,
                      row.marknames = 0, NA.char = "-9", pop = NULL, sep = NULL,
                      ask = F, quiet = T)

#### Summary statistics ####

# This is based on all individuals and all loci
basicstat.all <- basic.stats(all, diploid = TRUE, digits = 2)
allelic.richness.all <- allelic.richness(all, diploid = TRUE)

#Ar
allelic.richness.all.df <- as.data.frame(allelic.richness.all$Ar)
head(allelic.richness.all.df)
```

```
##            1        2        4        5       10       11       12        3
## L01 1.787061 1.812942 1.768182 1.772418 1.791748 1.767878 1.824423 1.785237
## L02 1.824248 1.821935 1.769314 1.788199 1.770299 1.800419 1.750351 1.795509
## L03 1.807640 1.819242 1.829571 1.821905 1.840162 1.820058 1.817599 1.803870
## L04 1.747673 1.707360 1.732144 1.773017 1.722796 1.741969 1.771407 1.723841
## L05 1.785305 1.779625 1.798952 1.795847 1.798366 1.786829 1.778060 1.787625
## L06 1.814539 1.800359 1.809411 1.784179 1.802405 1.773021 1.780494 1.778548
##            6        7        8        9
## L01 1.794292 1.780488 1.776582 1.768381
## L02 1.808096 1.789011 1.775000 1.806083
## L03 1.840070 1.834645 1.841139 1.832251
## L04 1.767348 1.778652 1.700000 1.748892
## L05 1.774880 1.800813 1.807278 1.811973
## L06 1.814429 1.850643 1.809177 1.763402
```

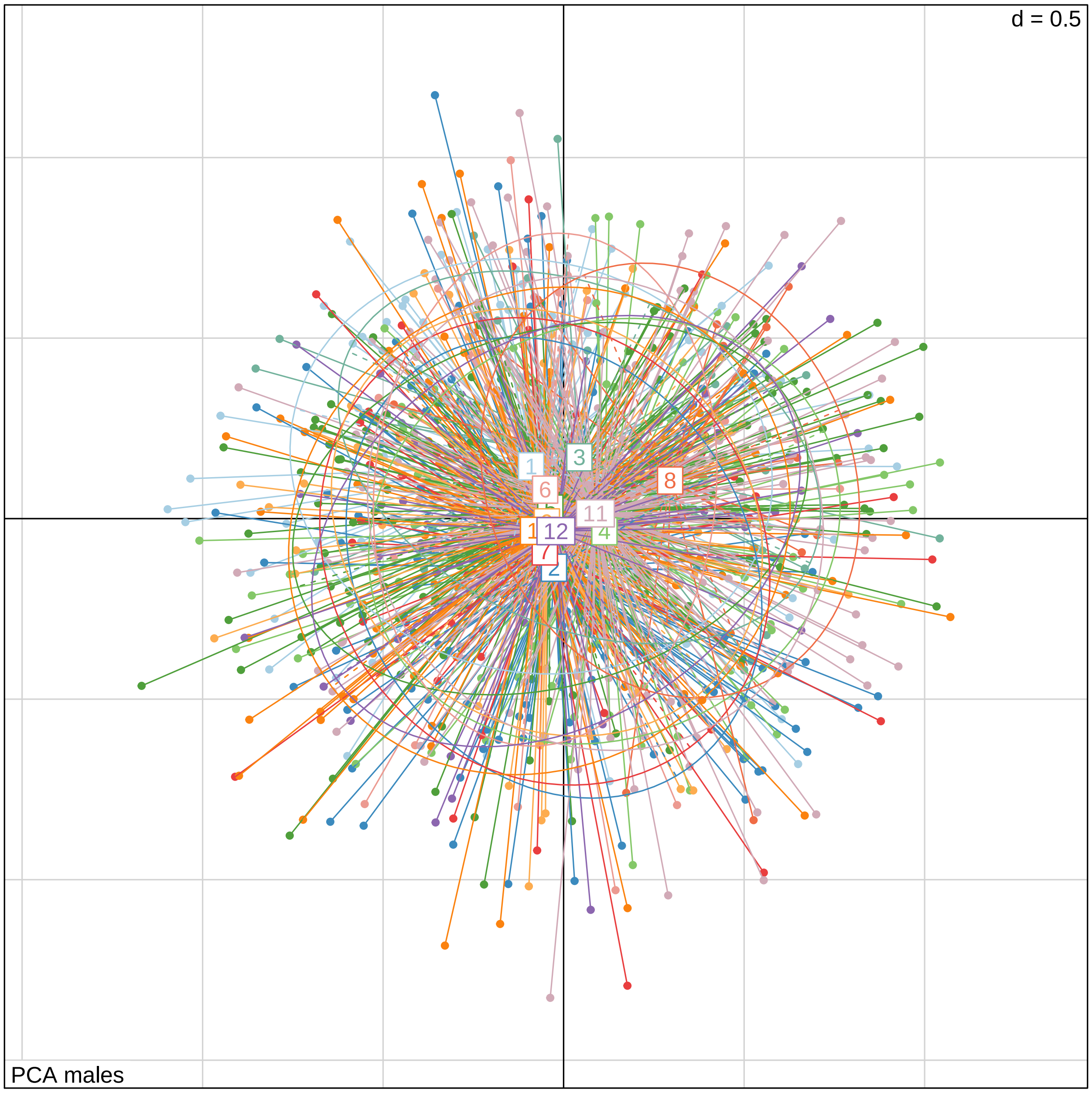
```
# full table of allelic richness can be found in Supplementary Table 1

#get mean Ho and Hx per pop
x.pop = seppop(all)
summary.by.pop = lapply(x.pop, summary)
Hobs.ls = rep(NA, length(summary.by.pop))
for (i in 1:length(summary.by.pop)){
  Hobs.ls[i] = mean(summary.by.pop[[i]]$Hobs, na.rm=TRUE)
}
Hobs.ls
```

```
##  [1] 0.6704879 0.6806497 0.6661082 0.6679969 0.6722625 0.6575876 0.6811236
##  [8] 0.6599379 0.7108386 0.7634409 0.6739695 0.6796281
```

```
Hexp.ls = rep(NA, length(summary.by.pop))
for (i in 1:length(summary.by.pop)){
  Hexp.ls[i] = mean(summary.by.pop[[i]]$Hexp, na.rm=TRUE)
}
Hexp.ls
```

```
##  [1] 0.7148786 0.7176830 0.7006664 0.6974417 0.7076302 0.6891500 0.7063049
##  [8] 0.6849261 0.6815566 0.7829920 0.6970019 0.7022487
```

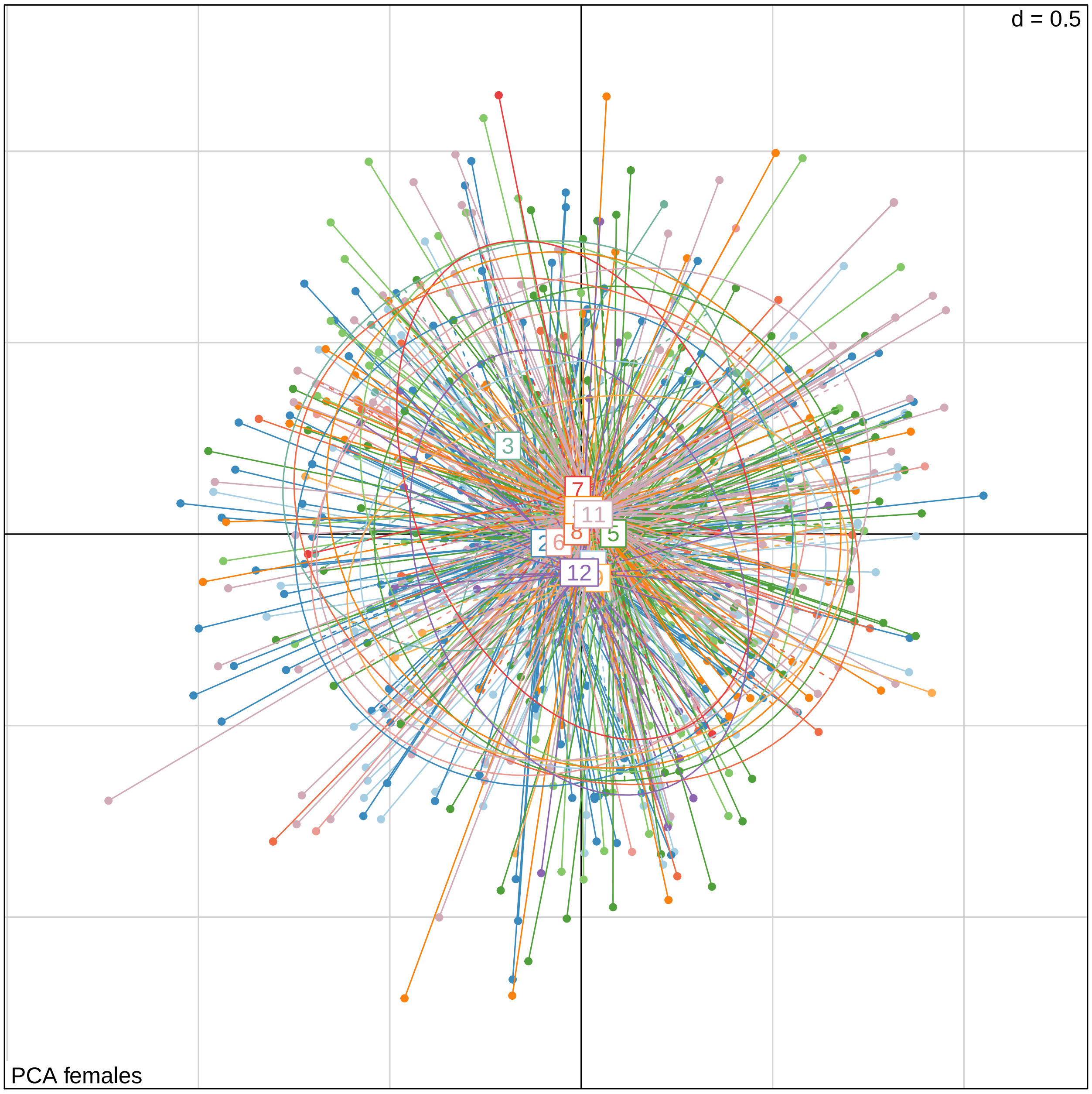
```
#### PCA ####

## males
x.males <- tab(males.stru, freq=TRUE, NA.method="mean")
pca.males <- dudi.pca(x.males, center=TRUE, scale=FALSE, scannf=F, nf=3)

s.class(pca.males$li, fac=pop(males.stru), col=funky(15), sub = "PCA males")
```



```
# percentages of variation explained
eig.perc.males <- 100*pca.males$eig/sum(pca.males$eig)
head(eig.perc.males)
```

```
## [1] 4.556447 4.117341 3.890883 3.740533 3.480553 3.296404
```

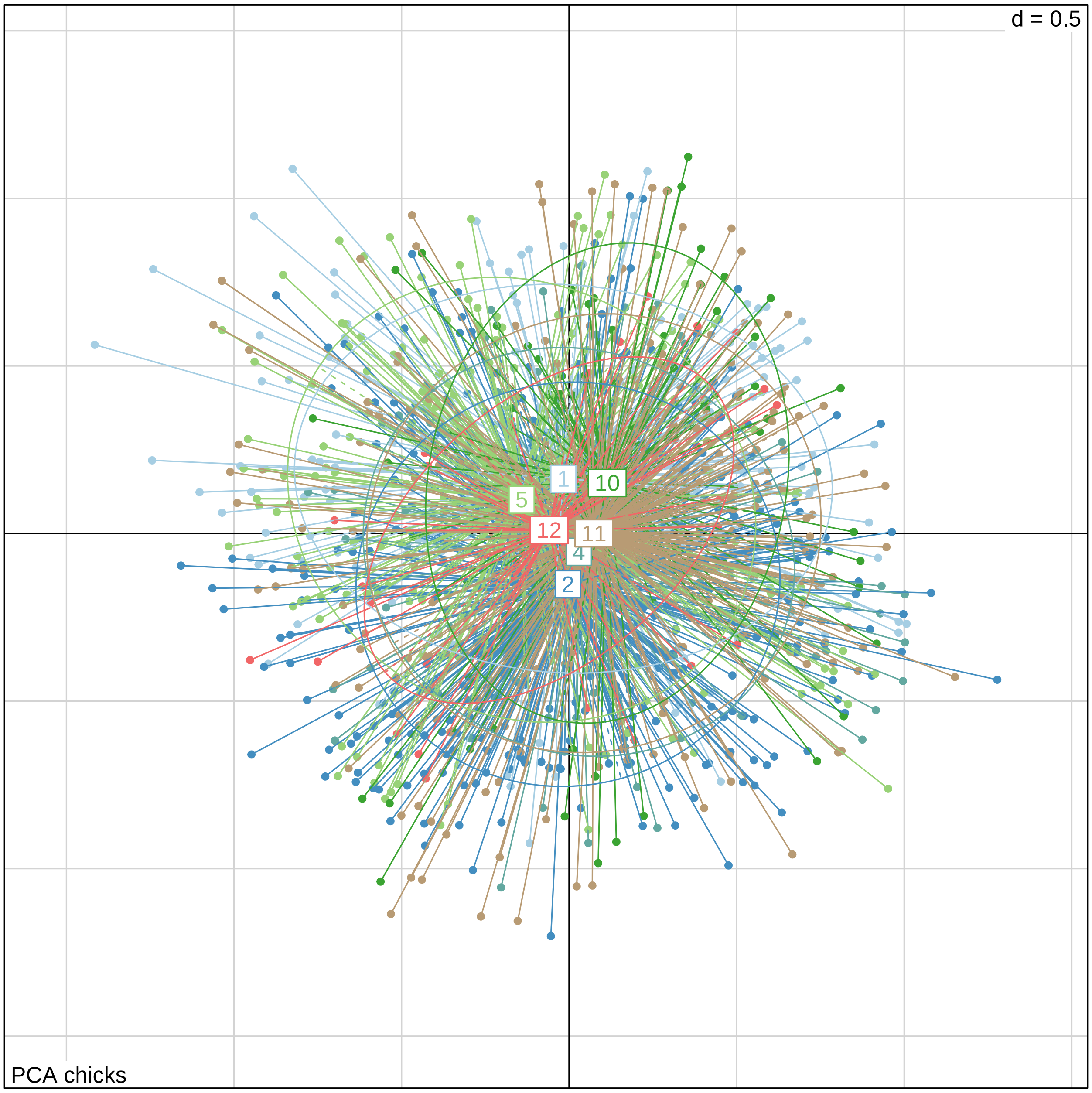
```
## females
x.females <- tab(females.stru, freq=TRUE, NA.method="mean")
pca.females <- dudi.pca(x.females, center=TRUE, scale=FALSE, scannf=F, nf=3)
s.class(pca.females$li, fac=pop(females.stru), col=funky(15), sub = "PCA females")
```



```
# percentages of variation explained
eig.perc.females <- 100*pca.females$eig/sum(pca.females$eig)
head(eig.perc.females)
```

```
## [1] 4.534886 4.343551 3.946859 3.731425 3.487025 3.297834
```

```
## chicks
x.chicks <- tab(chicks.stru, freq=TRUE, NA.method="mean")
pca.chicks <- dudi.pca(x.chicks, center=TRUE, scale=FALSE, scannf=F, nf=3)
s.class(pca.chicks$li, fac=pop(chicks.stru), col=funky(16), sub = "PCA chicks")
```



```r
# percentages of variation explained
eig.perc.chicks <- 100*pca.chicks$eig/sum(pca.chicks$eig)
head(eig.perc.chicks)
```

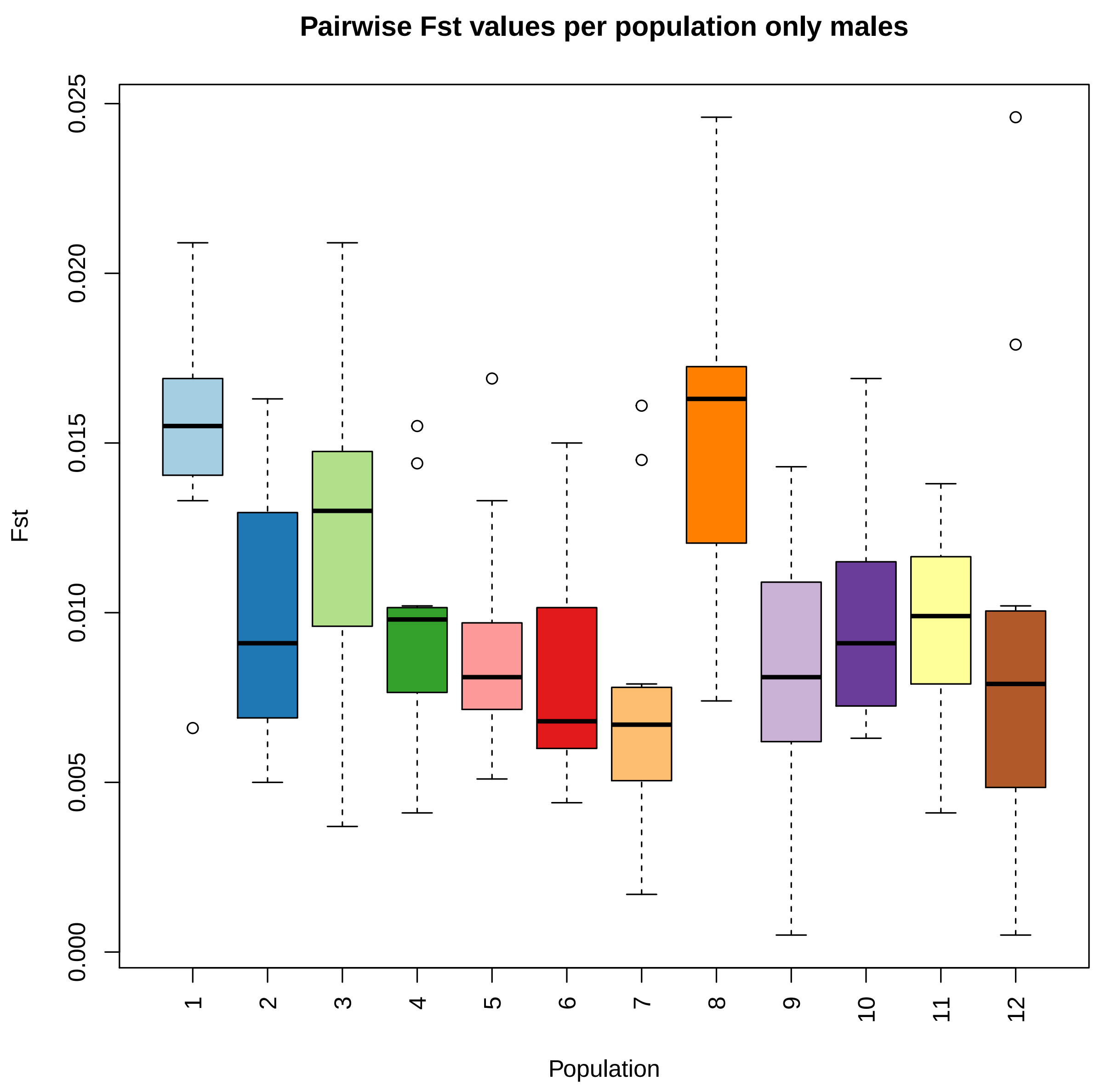
```
## [1] 4.972361 4.584830 4.151872 4.001866 3.879195 3.584750
```

```r
#### Calculate Fst ####

## Males
#convert to hfstat object
males.hfstat <- genind2hierfstat(males.stru)
#calculate stats
basicstat.males <- basic.stats(males.stru, diploid = TRUE, digits = 2)
```

```r
# per locus
fst.males.perlocus <- basicstat.males$perloc$Fst
fst.males.perlocus <- data.frame(Locus = seq(from = 1, to = 12),
                                 Fst = fst.males.perlocus)

# Pairwise Fst
fst.males <- pairwise.neifst(males.hfstat)
head(fst.males)
```

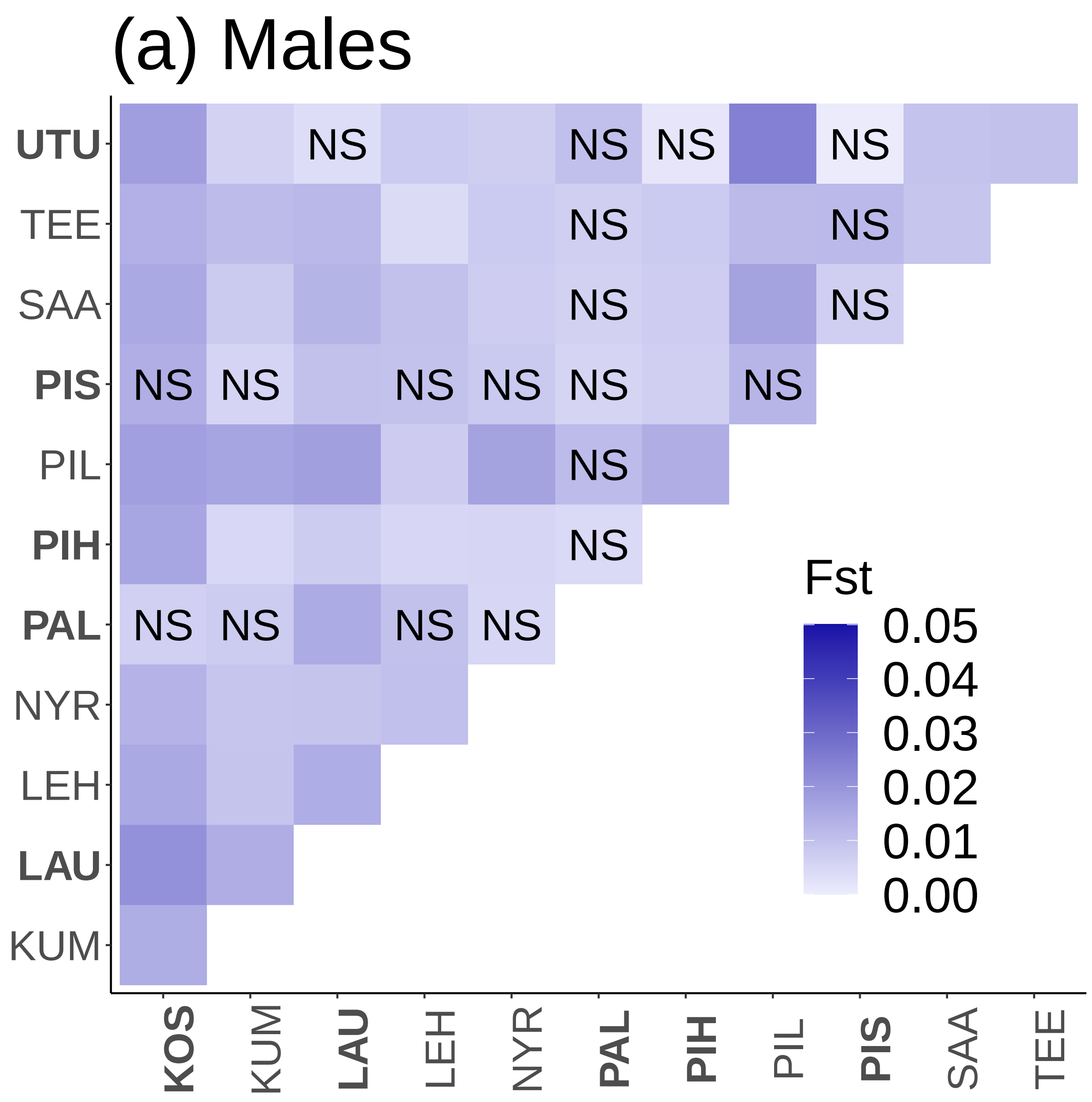
```
##        1      2      3      4      5      6      7      8      9     10     11
## 1     NA 0.0146 0.0209 0.0155 0.0133 0.0066 0.0161 0.0177 0.0143 0.0155 0.0138
## 2 0.0146     NA 0.0145 0.0093 0.0091 0.0078 0.0050 0.0163 0.0056 0.0080 0.0114
## 3 0.0209 0.0145     NA 0.0144 0.0092 0.0150 0.0077 0.0176 0.0100 0.0130 0.0119
## 4 0.0155 0.0093 0.0144     NA 0.0102 0.0101 0.0051 0.0074 0.0098 0.0100 0.0041
## 5 0.0133 0.0091 0.0092 0.0102     NA 0.0051 0.0054 0.0169 0.0081 0.0073 0.0079
## 6 0.0066 0.0078 0.0150 0.0101 0.0051     NA 0.0044 0.0113 0.0057 0.0063 0.0068
##       12
## 1 0.0179
## 2 0.0060
## 3 0.0037
## 4 0.0079
## 5 0.0070
## 6 0.0102
```

```r
# Fst per population
boxplot(fst.males, col=funky(nPop(males.stru)), las=3,
        xlab="Population", ylab="Fst",
        main = "Pairwise Fst values per population only males")
```



```r
#pop 6 only has 1 sample

#Bootstrap
boot.fst.males <- boot.ppfst(males.hfstat, nboot = 1000)

#create a long dataframe for pairwise Fst
boot.fst.males.UL <- boot.fst.males$ul
boot.fst.males.LL <- boot.fst.males$ll

flat.matrix <- function(d){
  data.frame(i=rep(row.names(d),ncol(d)),
             j=rep(colnames(d),each=nrow(d)),
             score=as.vector(d))
}
```

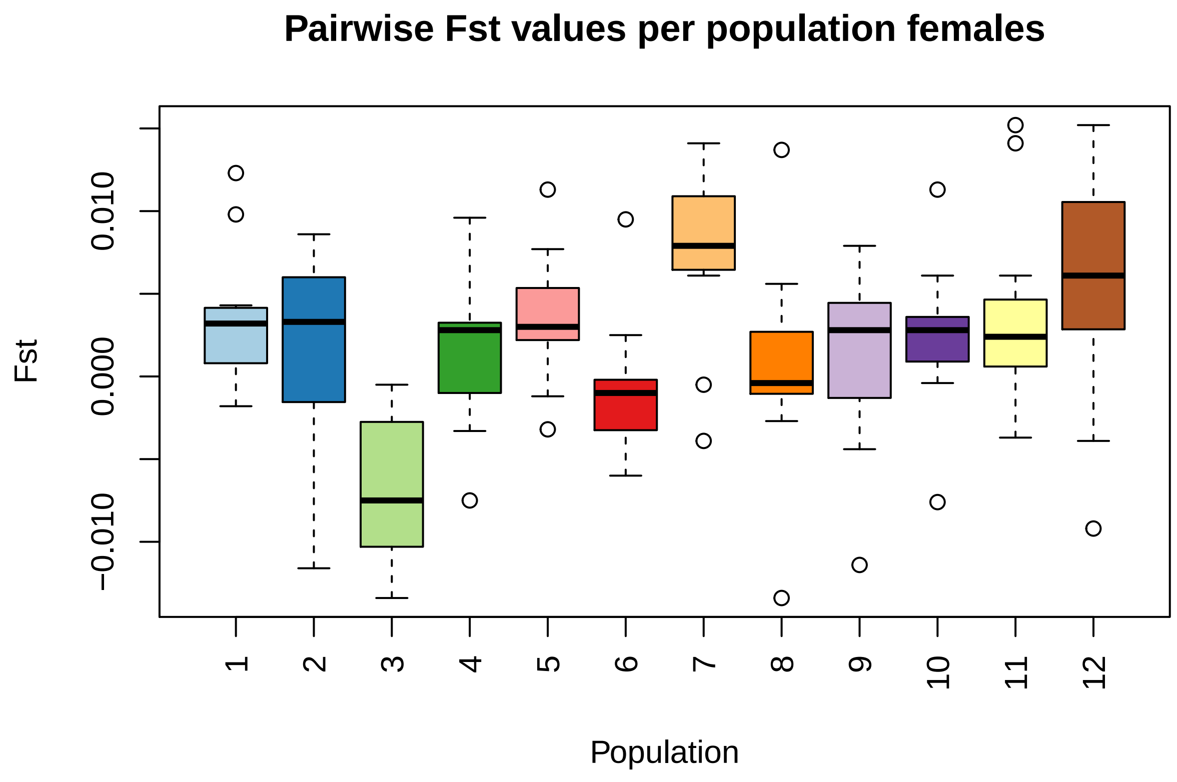
```r
fst.males.flat <- flat.matrix(fst.males)
names(fst.males.flat) <- c("site.x", "site.y", "Fst")

boot.fst.males.LL.flat <- flat.matrix(boot.fst.males.LL)
names(boot.fst.males.LL.flat) <- c("site.x", "site.y", "LL")

boot.fst.males.UL.flat <- flat.matrix(boot.fst.males.UL)
names(boot.fst.males.UL.flat) <- c("site.x", "site.y", "UL")

pairwise.fst.males <- left_join(fst.males.flat, boot.fst.males.LL.flat,
                                by = c("site.x", "site.y"))
pairwise.fst.males <- left_join(pairwise.fst.males, boot.fst.males.UL.flat,
                                by = c("site.x", "site.y"))

pairwise.fst.males <- subset(pairwise.fst.males, site.x != "-9" & site.y != "-9")
pairwise.fst.males <- subset(pairwise.fst.males, !is.na(Fst) & !is.na(UL))
pairwise.fst.males <- pairwise.fst.males %>% mutate(Significance = case_when(
  UL > 0 & LL > 0 ~ "significant",
  UL > 0 & LL < 0 ~ "insignificant",
  UL < 0 & LL < 0 ~ "significant" ))

mypalette3 <- c("#EDEDFD","#C2C1EC","#9795DB","#6C69C9","#413DB8","#1611A7")
theme_set(theme_classic())

pairwise.fst.males <- read.csv("data/tables/Pairwise_Fst_males.csv")
# contains abbreviations

ggplot(pairwise.fst.males, aes(abb.x, abb.y, fill = Fst)) + geom_tile() + theme_classic() +
  scale_fill_gradientn(colors = mypalette3, limits = c(0,0.05)) +
  geom_text(aes(label = Sig), size = 8)+
  theme(text = element_text(size = 22),
        axis.text.x = element_text(angle = 90, size = 22,
                                   face = c("bold", "plain", "bold", "plain", "plain",
                                            "bold", "bold", "plain", "bold",
                                            "plain", "plain")),
        axis.text.y = element_text(size = 22,
                                   face = c("plain", "bold", "plain", "plain",
                                            "bold", "bold", "plain", "bold",
                                            "plain", "plain", "bold")),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.text = element_text(size = 26),
        legend.title = element_text(size = 26),
        legend.key.size = unit(1, 'cm'),
        plot.title = element_text(size = 38),
        legend.position = c(0.8, 0.3)) +
  ggtitle('(a) Males')
```



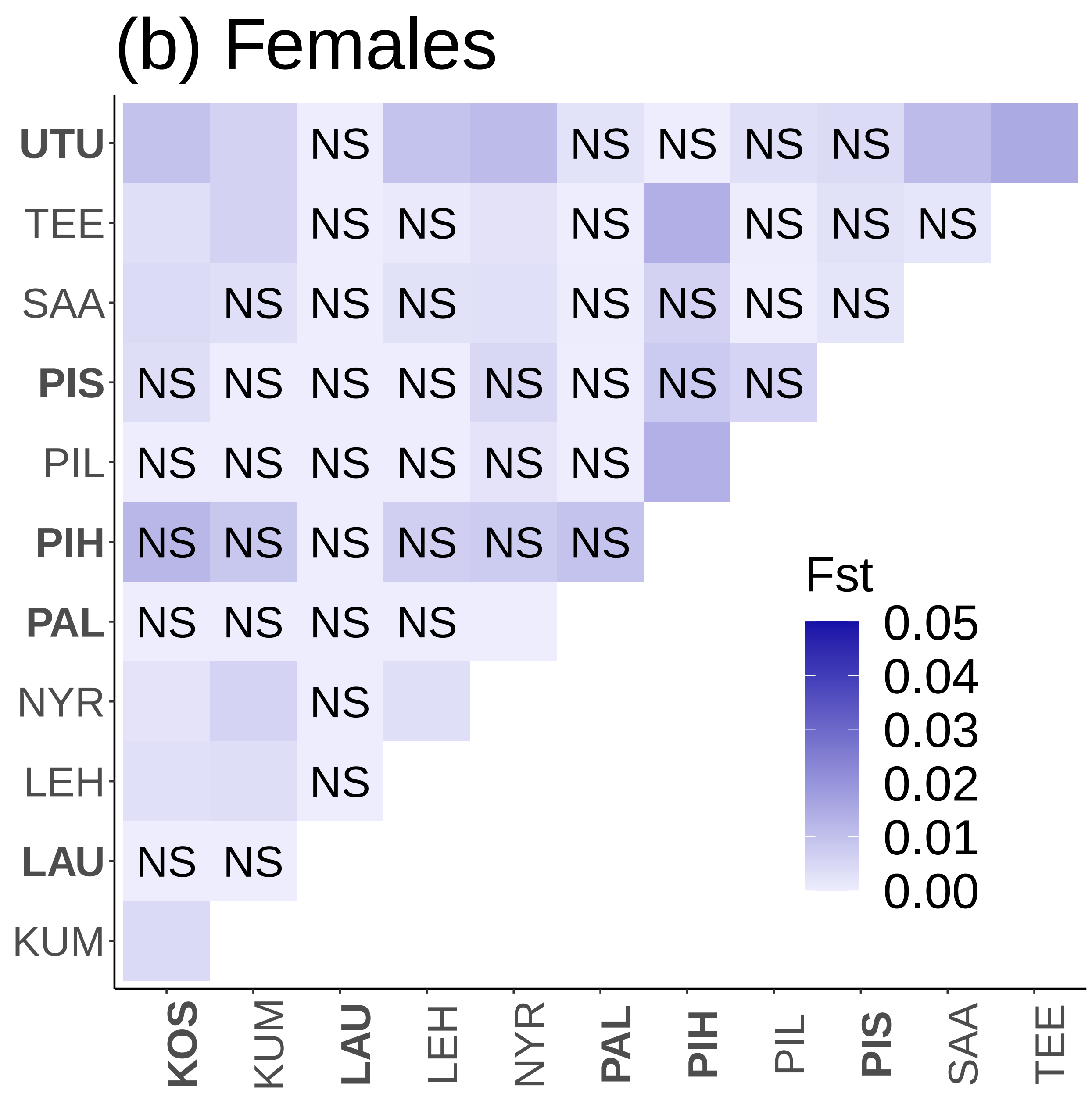
```r
## Females
#convert to hfstat object
females.hfstat <- genind2hierfstat(females.stru)
#calculate stats
basicstat.females <- basic.stats(females.stru, diploid = TRUE, digits = 2)

# per locus
fst.females.perlocus <- basicstat.females$perloc$Fst
fst.females.perlocus <- data.frame(Locus = seq(from = 1, to = 12),
                                   Fst = fst.females.perlocus)

# Pairwise Fst
fst.females <- pairwise.neifst(females.hfstat)
head(fst.females)
```

```
##          1        2        3        4        5        6        7        8        9
## 1       NA  0.0043 -0.0018  0.0030  0.0022 -0.0007  0.0123 -0.0006  0.0034
## 2  0.0043       NA -0.0116  0.0033  0.0059 -0.0010  0.0086 -0.0027 -0.0021
## 3 -0.0018 -0.0116       NA -0.0075 -0.0012 -0.0060 -0.0005 -0.0134 -0.0114
## 4  0.0030  0.0033 -0.0075       NA  0.0032 -0.0033  0.0068 -0.0015 -0.0005
## 5  0.0022  0.0059 -0.0012  0.0032       NA -0.0032  0.0077  0.0022  0.0048
## 6 -0.0007 -0.0010 -0.0060 -0.0033 -0.0032       NA  0.0095 -0.0006 -0.0044
##         10       11       12
## 1  0.0040  0.0032  0.0098
## 2  0.0032  0.0061  0.0061
## 3 -0.0076 -0.0037 -0.0092
## 4  0.0028  0.0010  0.0096
## 5  0.0030  0.0024  0.0113
## 6  0.0002 -0.0023  0.0025
```

```r
# Fst per population
boxplot(fst.females, col=funky(nPop(females.stru)), las=3,
        xlab="Population", ylab="Fst",
        main = "Pairwise Fst values per population females")
```



```r
#Bootstrap
boot.fst.females <- boot.ppfst(females.hfstat, nboot = 1000)

#create long dataframe
boot.fst.females.UL <- boot.fst.females$ul
boot.fst.females.LL <- boot.fst.females$ll

fst.females.flat <- flat.matrix(fst.females)
names(fst.females.flat) <- c("site.x", "site.y", "Fst")

boot.fst.females.LL.flat <- flat.matrix(boot.fst.females.LL)
names(boot.fst.females.LL.flat) <- c("site.x", "site.y", "LL")

boot.fst.females.UL.flat <- flat.matrix(boot.fst.females.UL)
names(boot.fst.females.UL.flat) <- c("site.x", "site.y", "UL")

pairwise.fst.females <- left_join(fst.females.flat, boot.fst.females.LL.flat,
                                  by = c("site.x", "site.y"))
pairwise.fst.females <- left_join(pairwise.fst.females, boot.fst.females.UL.flat,
                                  by = c("site.x", "site.y"))

pairwise.fst.females <- subset(pairwise.fst.females, site.x != "-9" & site.y != "-9")
pairwise.fst.females <- subset(pairwise.fst.females, !is.na(Fst) & !is.na(UL))
pairwise.fst.females <- pairwise.fst.females %>% mutate(Significance = case_when(
  UL > 0 & LL > 0 ~ "significant",
  UL > 0 & LL < 0 ~ "insignificant",
  UL < 0 & LL < 0 ~ "significant" ))
```



```
## Chicks
chicks.hfstat <- genind2hierfstat(chicks.stru)
#calculate stats
basicstat.chicks <- basic.stats(chicks.stru, diploid = TRUE, digits = 2)
# per locus
fst.chicks.perlocus <- basicstat.chicks$perloc$Fst
fst.chicks.perlocus <- data.frame(Locus = seq(from = 1, to = 11),
                                  Fst = fst.chicks.perlocus)

## Pairwise Fst
fst.chicks <- pairwise.neifst(chicks.hfstat)

#Fst per population for chicks
boxplot(fst.chicks, col=funky(nPop(chicks.stru)), las=3,
        xlab="Population", ylab="Fst",
        main = "Pairwise Fst values per population chicks")

# Bootstrap
boot.fst.chicks <- boot.ppfst(chicks.hfstat, nboot = 1000)
boot.fst.chicks.UL <- boot.fst.chicks$ul
boot.fst.chicks.LL <- boot.fst.chicks$ll

#create long df
fst.chicks.flat <- flat.matrix(fst.chicks)
names(fst.chicks.flat) <- c("site.x", "site.y", "Fst")

boot.fst.chicks.LL.flat <- flat.matrix(boot.fst.chicks.LL)
names(boot.fst.chicks.LL.flat) <- c("site.x", "site.y", "LL")

boot.fst.chicks.UL.flat <- flat.matrix(boot.fst.chicks.UL)
names(boot.fst.chicks.UL.flat) <- c("site.x", "site.y", "UL")

pairwise.fst.chicks <- left_join(fst.chicks.flat, boot.fst.chicks.LL.flat,
                                 by = c("site.x", "site.y"))
pairwise.fst.chicks <- left_join(pairwise.fst.chicks, boot.fst.chicks.UL.flat,
                                 by = c("site.x", "site.y"))

pairwise.fst.chicks <- subset(pairwise.fst.chicks, site.x != "-9" & site.y != "-9")
pairwise.fst.chicks <- subset(pairwise.fst.chicks, !is.na(Fst) & !is.na(UL))
pairwise.fst.chicks <- pairwise.fst.chicks %>% mutate(Significance = case_when(
  UL > 0 & LL > 0 ~ "significant",
  UL > 0 & LL < 0 ~ "insignificant",
  UL < 0 & LL < 0 ~ "significant" ))
```

```
chicks.fst <- ggplot(pairwise.fst.chicks, aes(abb.x, abb.y, fill = Fst)) + geom_tile() + theme_classic(
  scale_fill_gradientn(colors = mypalette3, limits = c(0,0.05)) +
  geom_text(aes(label = Sig), size = 8)+
  theme(text = element_text(size = 22),
        axis.text.x = element_text(angle = 90, size = 22,
                                   face = c("bold", "plain", "plain",
                                            "plain", "plain", "plain")),
        axis.text.y = element_text(size = 20,
                                   face = c("plain", "plain", "plain", "plain",
                                            "plain", "bold")),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.text = element_text(size = 26),
        legend.title = element_text(size = 26),
        legend.key.size = unit(1, 'cm'),
        plot.title = element_text(size = 38),
        legend.position = c(0.8, 0.3)) +
  ggtitle("(c) Chicks ")
```

Structure analysis

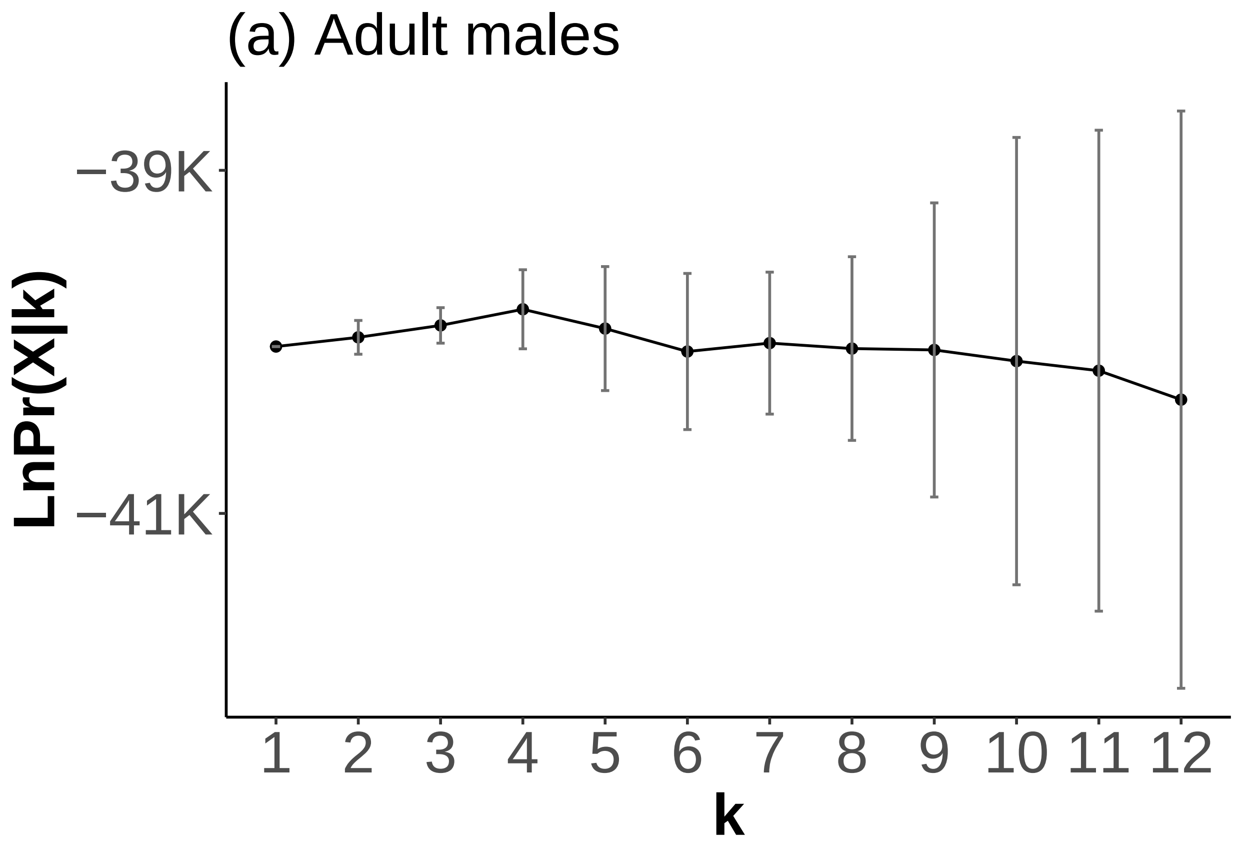
We used ParallelStructure to run STRUCTURE on multiple cores. Details on inferring the highest likelihood K and plotting the barcharts can be found in the full script within the GitHub directory called 3.STRUCTUREanalysis.R.

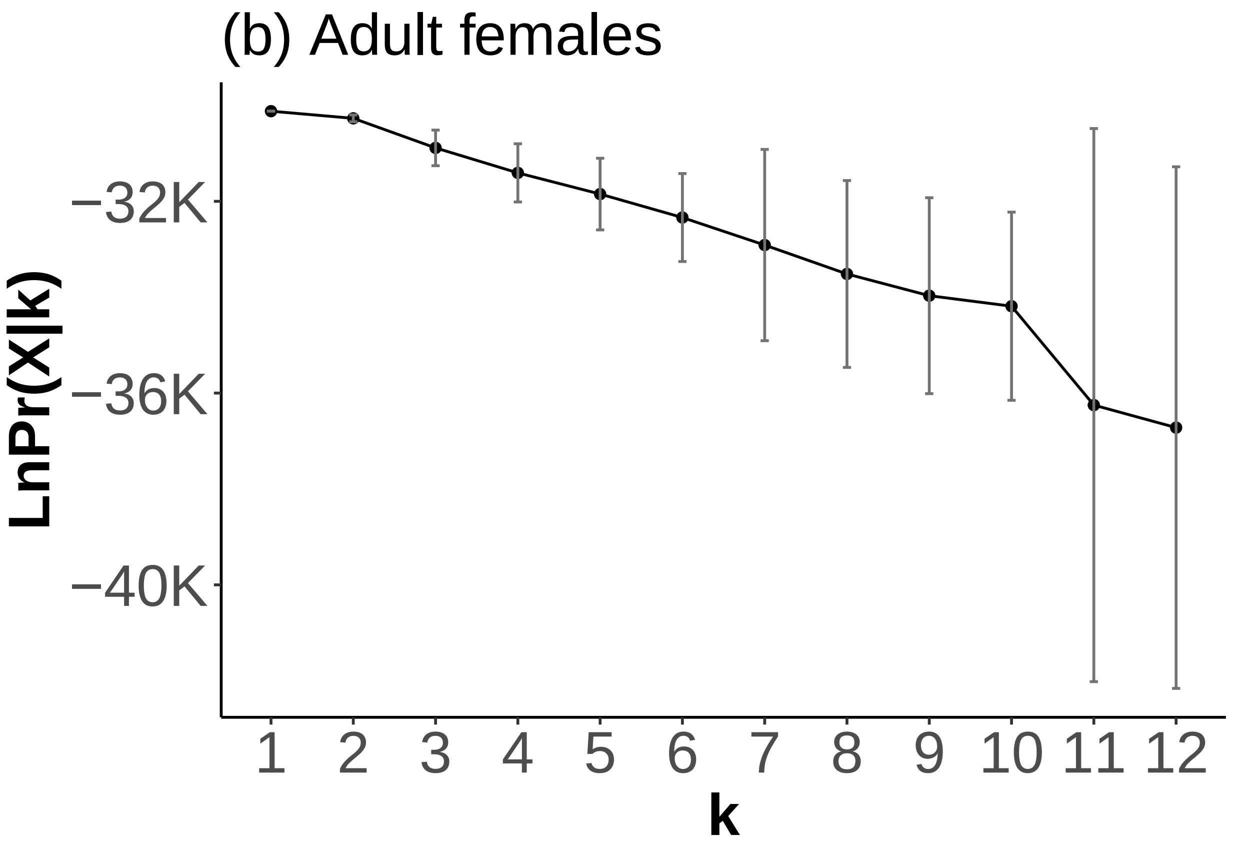
Ensure you have enough computational power to conduct this analysis.

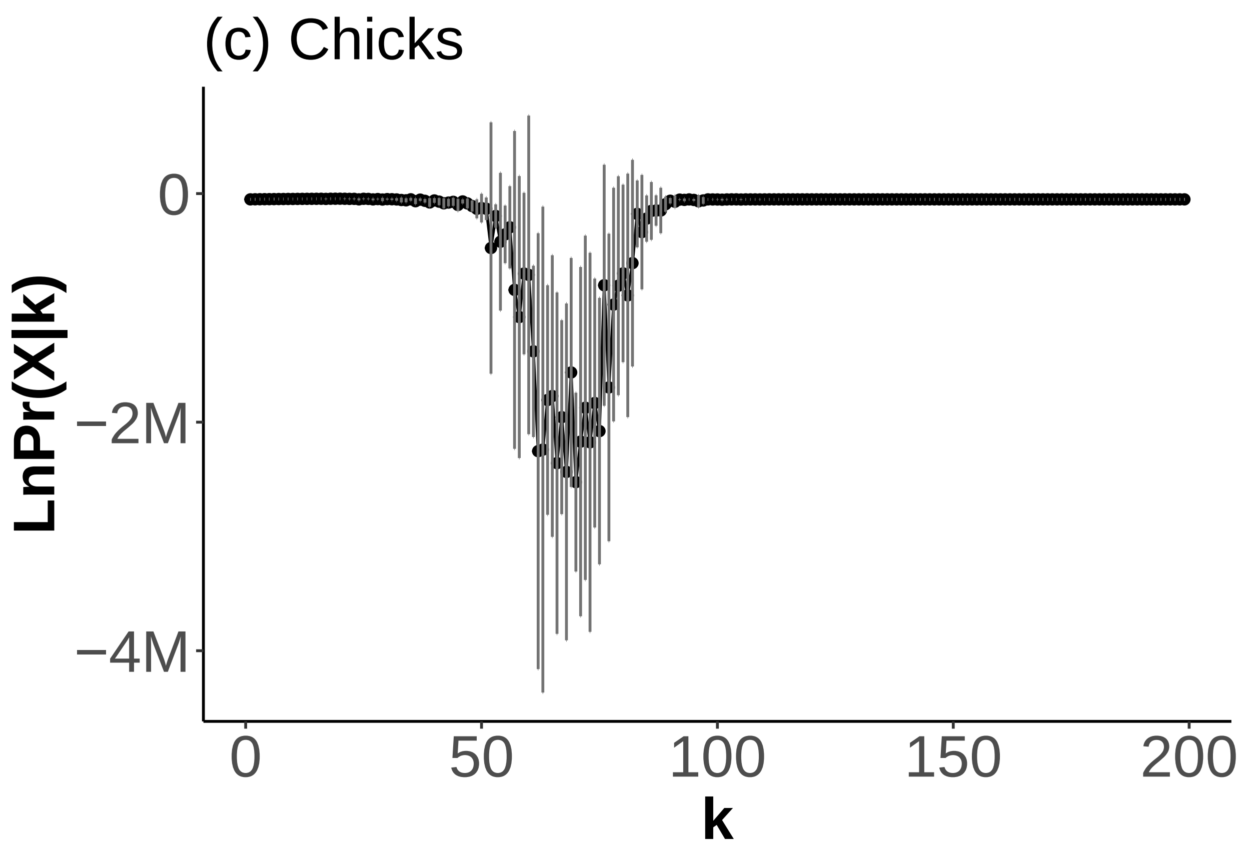
```
##### Running structure for males #####

males <- fread("data/cleandata/Microsat.adults.noLOCUS1+13.forstructure.stru")
infile <- "data/cleandata/Microsat.males.noLOCUS1+13.forstructure.stru"
# system("mkdir data/Results_stru_males")
outpath <- "data/structure/Results_stru_males/"

# job matrix and write to job file
nrep <- 10
burnin <- 10000
niter <- 10000
up_to_k <- 12

# job matrix
k_var <- rep(1:up_to_k, each = nrep)
ID_var <- as.character(sapply(c(1:up_to_k), function(k)
  sapply(c(1:nrep), function(x) paste0("T",k, "_", x))))

# make the job matrix
pop <- "1,2,3,4,5,6,7,8,9,10,11,12" #number of pops in the file

hunt_jobs <- matrix(c(ID_var, rep(pop, nrep * up_to_k), k_var,
                      rep(burnin, nrep * up_to_k),
                      rep(niter, nrep * up_to_k)), nrow = nrep * up_to_k)

write(t(hunt_jobs), ncol = length(hunt_jobs[1,]),
      file = "data/structure/hunt_jobs_adults.txt")

# file path to structure

STR_path='/usr/local/bin/'

# Run Parallel Structure

# Run structure (from terminal, do not run this last part in Rstudio)

ParallelStructure::
  parallel_structure(structure_path=STR_path,
                     joblist='data/structure/hunt_jobs_adults.txt',
                     n_cpu=45, infile=infile,outpath=outpath,
                     numinds = nrow(males)/2,numloci=ncol(males)-2,noadmix = 0,
                     alpha = 1.0,freqscorr=1,lambda = 1,printqhat=1,
                     plot_output=0,onerowperind=0, locprior = 0)

##### Running structure for females #####
females <- fread("data/cleandata/Microsat.females.noLOCUS1+13.forstructure.stru")
```

```r
infile <- "data/cleandata/Microsat.females.noLOCUS1+13.forstructure.stru"
# system("mkdir data/Results_stru_females")
outpath <- "data/structure/Results_stru_females/"

ParallelStructure::
  parallel_structure(structure_path=STR_path,
                     joblist='data/structure/hunt_jobs_adults.txt',
                     n_cpu=45, infile=infile,outpath=outpath,
                     numinds = nrow(females)/2,numloci=ncol(females)-2,
                     noadmix = 0, alpha = 1.0,freqscorr=1,lambda = 1,
                     printqhat=1,plot_output=0,onerowperind=0, locprior = 0)

##### Running structure for chicks #####

chicks <- fread("data/cleandata/Microsat.chicks.noLOCUS1+13+14.forstructure.stru")
infile <- "data/cleandata/Microsat.chicks.noLOCUS1+13+14.forstructure.stru"
# system("mkdir data/Results_stru_chicks")
outpath <- "data/structure/Results_stru_chicks/"

# job matrix and write to job file
nrep <- 10
burnin <- 10000
niter <- 10000
up_to_k <- 199 #number of broods

# job matrix
k_var <- rep(1:up_to_k, each = nrep)
ID_var <- as.character(sapply(c(1:up_to_k), function(k)
  sapply(c(1:nrep), function(x) paste0("T",k, "_", x))))

# make the job matrix
pop <- "1,2,4,5,10,11,12" #number of pops in the file

hunt_jobs <- matrix(c(ID_var, rep(pop, nrep * up_to_k), k_var,
                      rep(burnin, nrep * up_to_k),
                      rep(niter, nrep * up_to_k)), nrow = nrep * up_to_k)

write(t(hunt_jobs), ncol = length(hunt_jobs[1,]),
      file = "data/structure/hunt_jobs_chicks.txt")

# file path to structure

STR_path='/usr/local/bin/'


# Run Parallel Structure

# Run structure (from terminal, do not run this last part in Rstudio)


ParallelStructure::
  parallel_structure(structure_path=STR_path,
                     joblist='/data/structure/hunt_jobs_chicks.txt',
                     n_cpu=45, infile=infile,outpath=outpath,
                     numinds = nrow(chicks)/2,numloci=ncol(chicks)-2,
                     noadmix = 0, alpha = 1.0,freqscorr=1,lambda = 1,
                     printqhat=1,plot_output=0,onerowperind=0, locprior = 0)
```

These sex- and age-specific STRUCTURE analyses gave us the following the results:




(b) Adult females
LnPr(X|k)
−32K
−36K
−40K
1
2
3
4
5
6
7
8
9
10
11
12
k



Calculating and modelling sMLH

To quantify inbreeding levels, we calculated sMLH using the inbreedR package. Next, to understand the effects of hunting on inbreeding, we built a mixed-model and investigated the fit of the models using various packages.

```
## Change formats to be loaded into inbreedR
males.inb <- males.inb %>% remove_rownames %>% column_to_rownames(var="id")
males.inb <- males.inb[,-1]

females.inb <- females.inb %>% remove_rownames %>% column_to_rownames(var="id")
females.inb <- females.inb[,-1]

chicks.inb <- chicks.inb %>% remove_rownames %>% column_to_rownames(var="id")
chicks.inb <- chicks.inb[,-1]

# convert to inbreedR
males.inb <- convert_raw(males.inb)
females.inb <- convert_raw(females.inb)
chicks.inb <- convert_raw(chicks.inb)

#### Calculate sMLH ####

sMLH_females <- sMLH(females.inb) #sMLH
```

```r
het_var_females <- var(sMLH_females, na.rm=TRUE) # variance in sMLH
summary(sMLH_females)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3593  0.8571  0.9796  1.0001  1.1020  1.4371
```

```r
het_var_females
```

```
## [1] 0.02861244
```

```r
sMLH_males <- sMLH(males.inb) #sMLH
het_var_males <- var(sMLH_males, na.rm=TRUE) # variance in sMLH
summary(sMLH_males)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3713  0.8664  0.9902  1.0000  1.1140  1.4532
```

```r
het_var_males
```

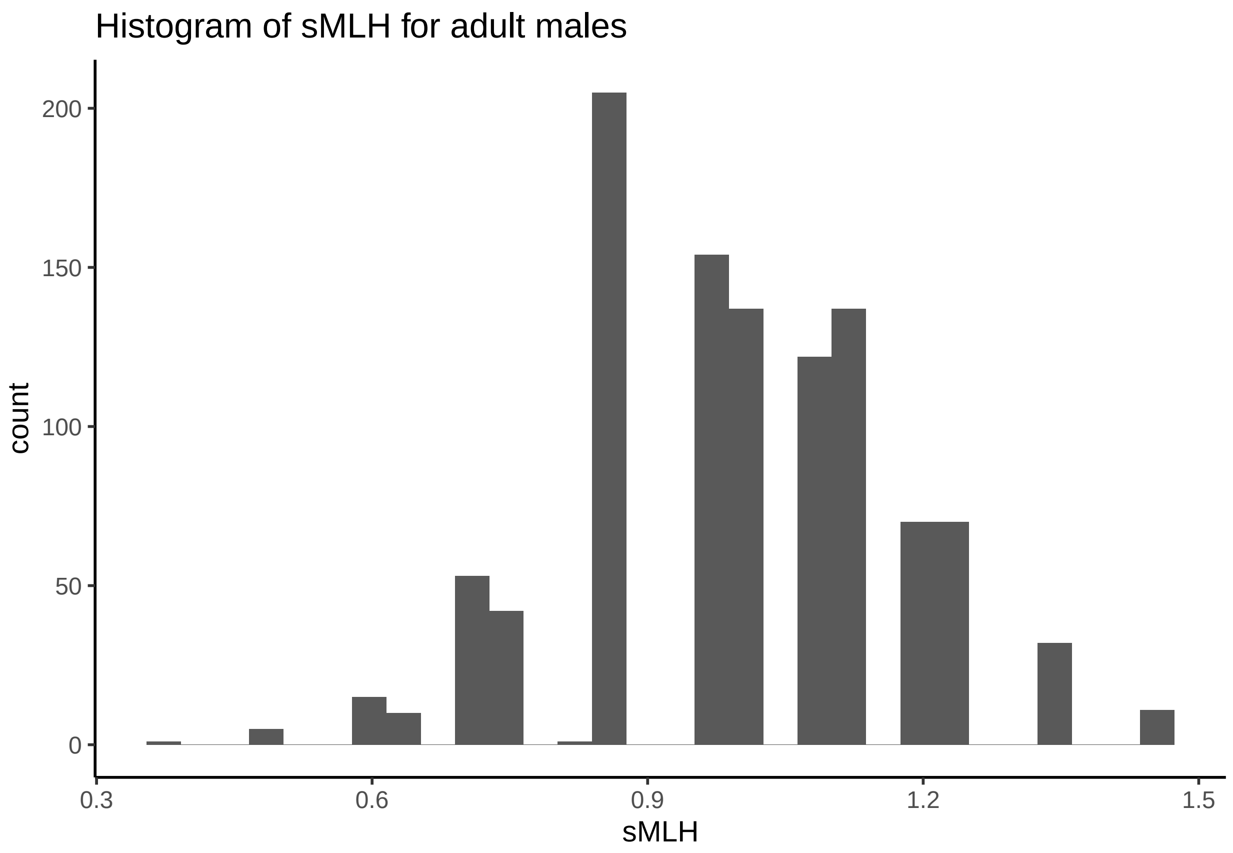
```
## [1] 0.03047389
```

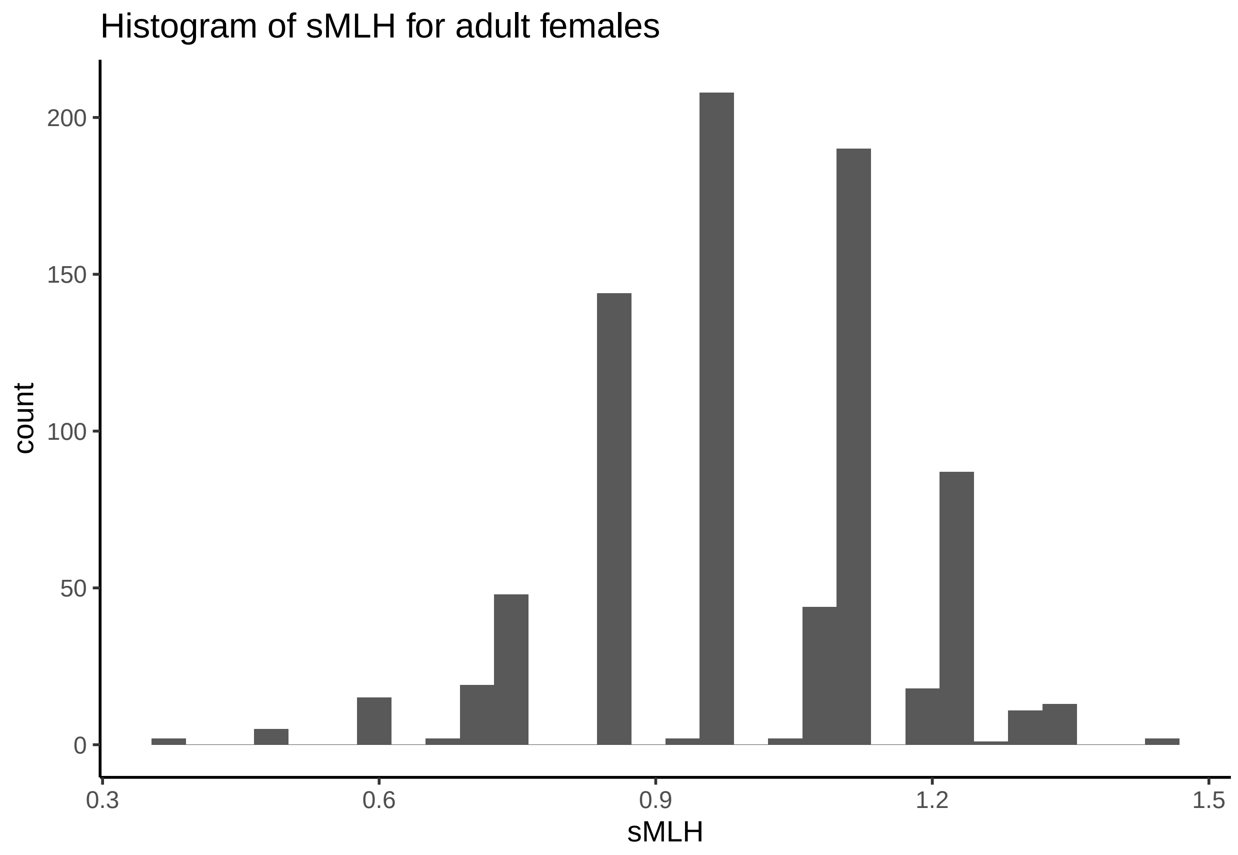
```r
sMLH_chicks <- sMLH(chicks.inb)
het_var_chicks <- var(sMLH_chicks, na.rm=TRUE) # variance in sMLH
summary(sMLH_chicks)
```

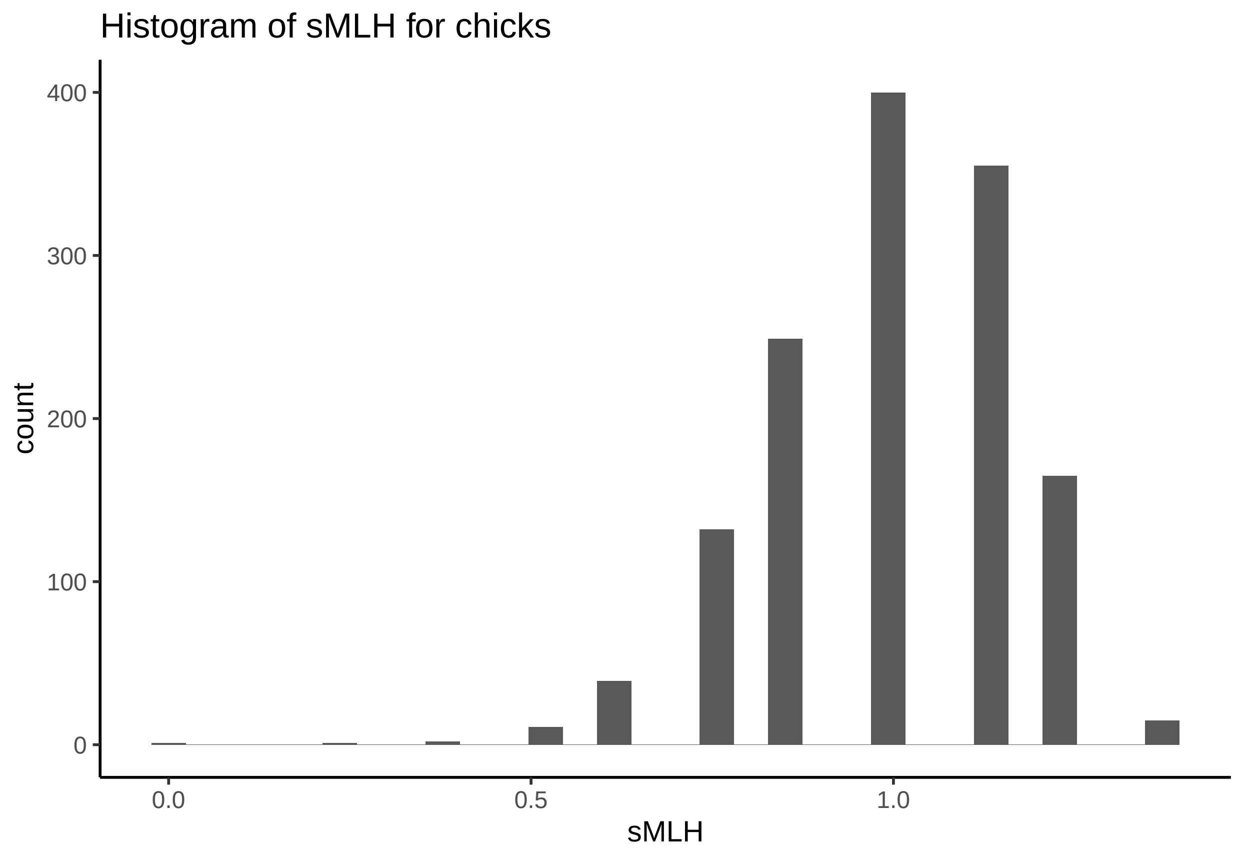
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.8725  0.9971  1.0000  1.1217  1.3710
```

```r
het_var_chicks
```

```
## [1] 0.02953932
```


Histogram of sMLH for adult males
count
200
150
100
50
0
0.3
0.6
0.9
1.2
1.5
sMLH


Histogram of sMLH for adult females
count
sMLH
0
50
100
150
200
0.3
0.6
0.9
1.2
1.5



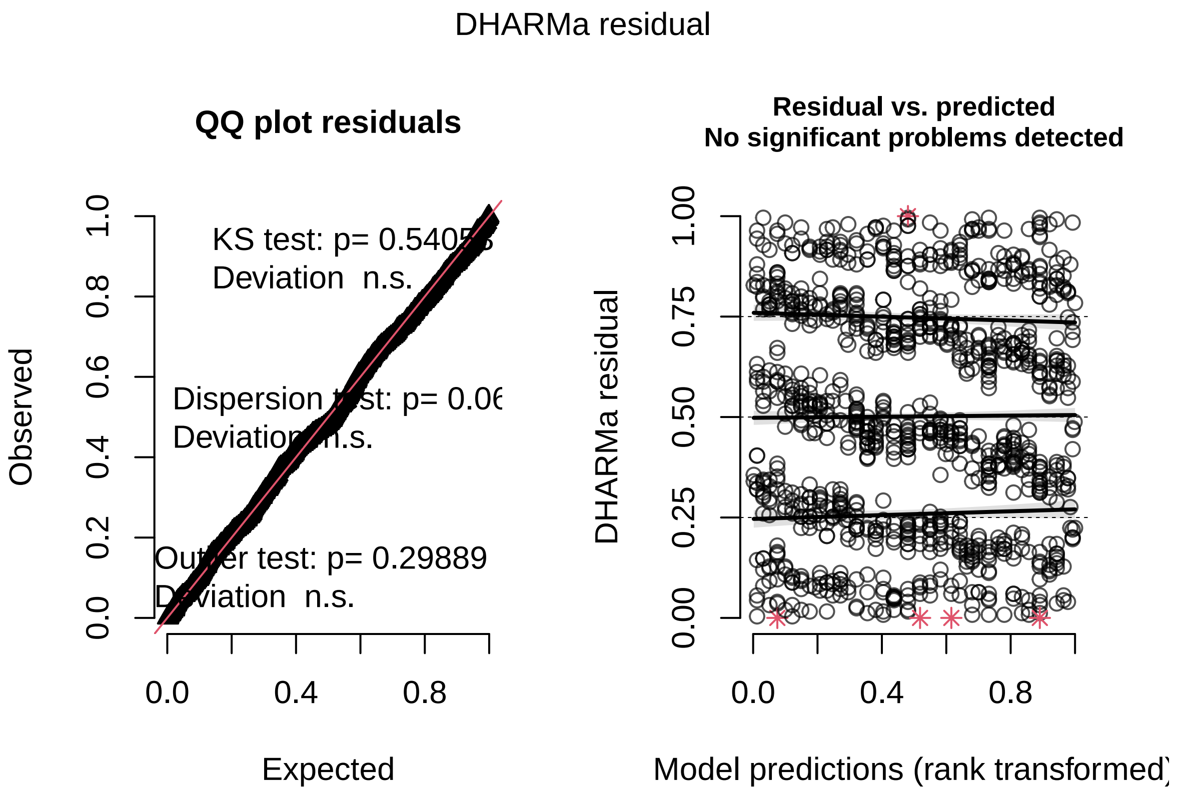
```r
### Setup models ####
#male model
sMLH.model.males.lmer <- lmerTest::lmer(sMLH ~ hunt + pop*year + hunt*year +
                                        (1|pop)+ (1|year) , data = sMLH.males)

#failed to converge
#Rescale and center continuous parameters
numcols <- grep("^c\\.",names(sMLH.males))
sMLH.males_rescale <- sMLH.males
sMLH.males_rescale[,numcols] <- scale(sMLH.males_rescale[,numcols])
sMLH.model.males.lmer_rescale <- update(sMLH.model.males.lmer,
                                       data=sMLH.males_rescale)

# restart and bump up max interations
ss <- getME(sMLH.model.males.lmer_rescale,c("theta","fixef"))
sMLH.model.males.lmer_noerror <- update(sMLH.model.males.lmer_rescale,
                                       start=ss,
                                       control=lmerControl(optCtrl=list(maxfun=2e4)))

#fixed!
head(coef(summary(sMLH.model.males.lmer_noerror)))
```

```
##                   Estimate Std. Error           df     t value  Pr(>|t|)
## (Intercept)    1.055203275 0.07168685 1.012184e-07 14.71962143 0.9999988
## huntunhunted -0.002040534 0.06667593 1.017000e+03 -0.03060376 0.9755915
## popKummunsuo   0.012378497 0.07603285 4.553306e-08  0.16280459 0.9999997
## popLauttasuo   0.029534840 0.07824069 5.105668e-08  0.37748695 0.9999996
## popLehtusuo    0.007552887 0.07906346 5.323841e-08  0.09552943 0.9999996
## popNyrölä     -0.090560527 0.07468617 4.239187e-08 -1.21254752 0.9999996
```

```
VarCorr(sMLH.model.males.lmer_noerror)
```

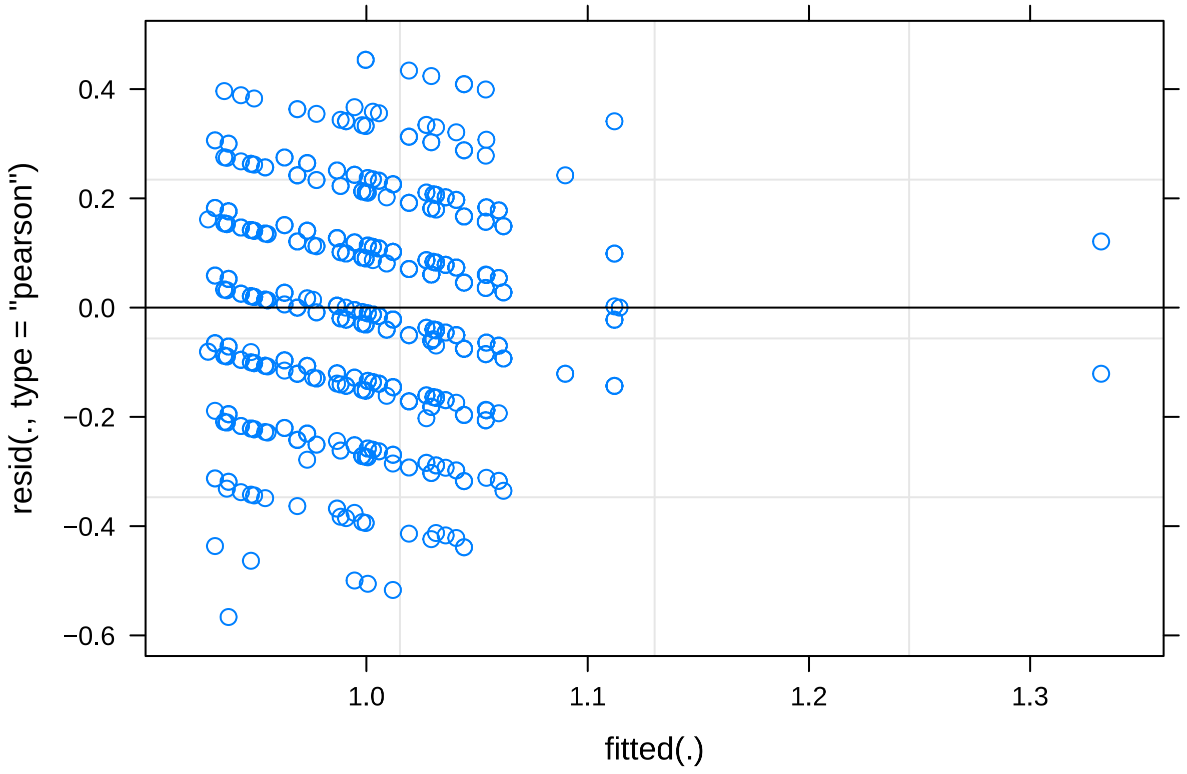
```
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 0.033380
##  year     (Intercept) 0.018864
##  Residual             0.173612
```

```
simulateResiduals(fittedModel = sMLH.model.males.lmer_noerror, plot = T)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.132 0.708 0.588 0.488 0.064 0.428 0.064 0.608 0.748 0.92 0.468 0.692 0.276
```

```
plot(sMLH.model.males.lmer_noerror)
```



```
r.squaredGLMM(sMLH.model.males.lmer_noerror)
```

```
##             R2m        R2c
## [1,] 0.0500138 0.09419199
```

```
icc(model = sMLH.model.males.lmer_noerror, by_group = TRUE)
```

```
## # ICC by Group
##
## Group |   ICC
## --------------
## pop   | 0.035
## year  | 0.011
```

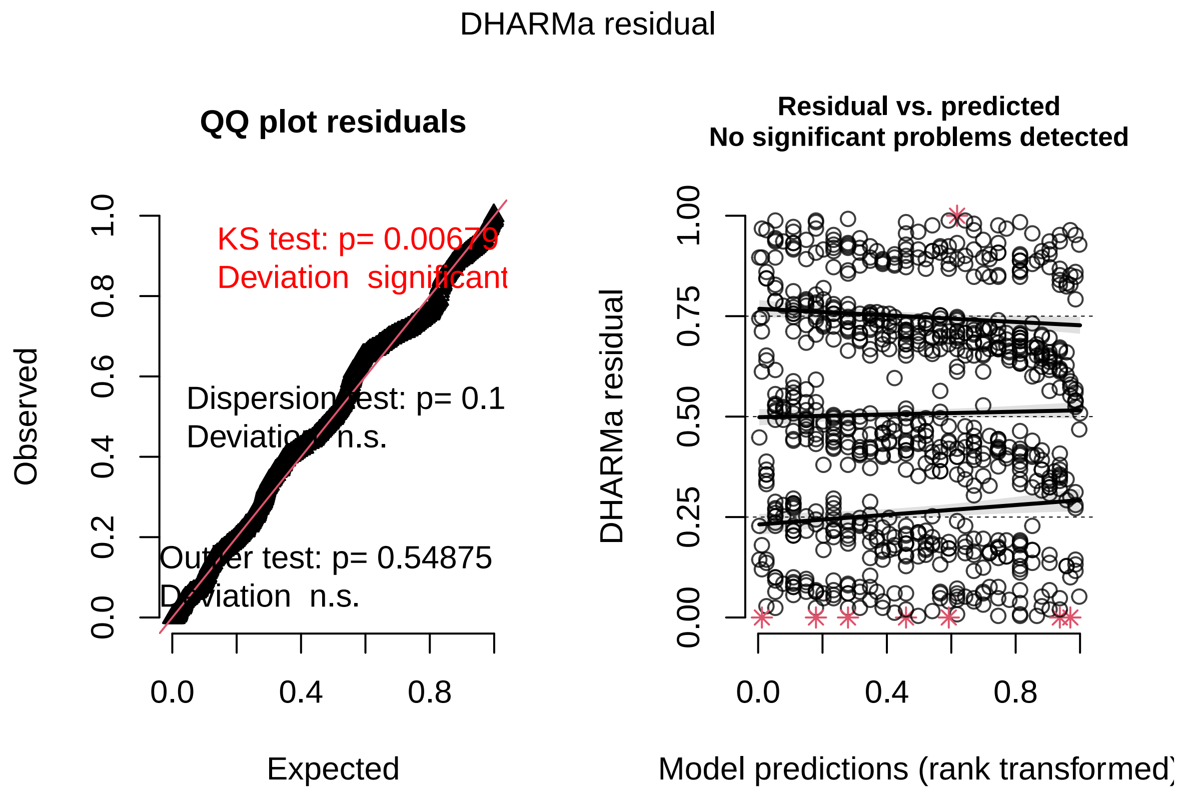
```
#females
sMLH.model.females.lmer <- lmerTest::lmer(sMLH ~ hunt + pop*year + hunt*year+(1|pop)+ (1|year) , data =
head(coef(summary(sMLH.model.females.lmer)))
```

```
##                  Estimate Std. Error           df    t value  Pr(>|t|)
## (Intercept)    1.07612581 0.05970276 2.321401e-07 18.0247256 0.9999974
## huntunhunted -0.10867322 0.05615415 4.730745e-08 -1.9352659 0.9999995
## popKummunsuo  0.10121876 0.05878428 5.681287e-08  1.7218679 0.9999995
## popLauttasuo -0.01908828 0.07638497 1.619696e-07 -0.2498958 0.9999988
## popLehtusuo   0.13333302 0.07598186 1.585776e-07  1.7548007 0.9999986
## popNyrölä     0.10903765 0.06798246 1.016224e-07  1.6039085 0.9999991
```

```r
VarCorr(sMLH.model.females.lmer)
```

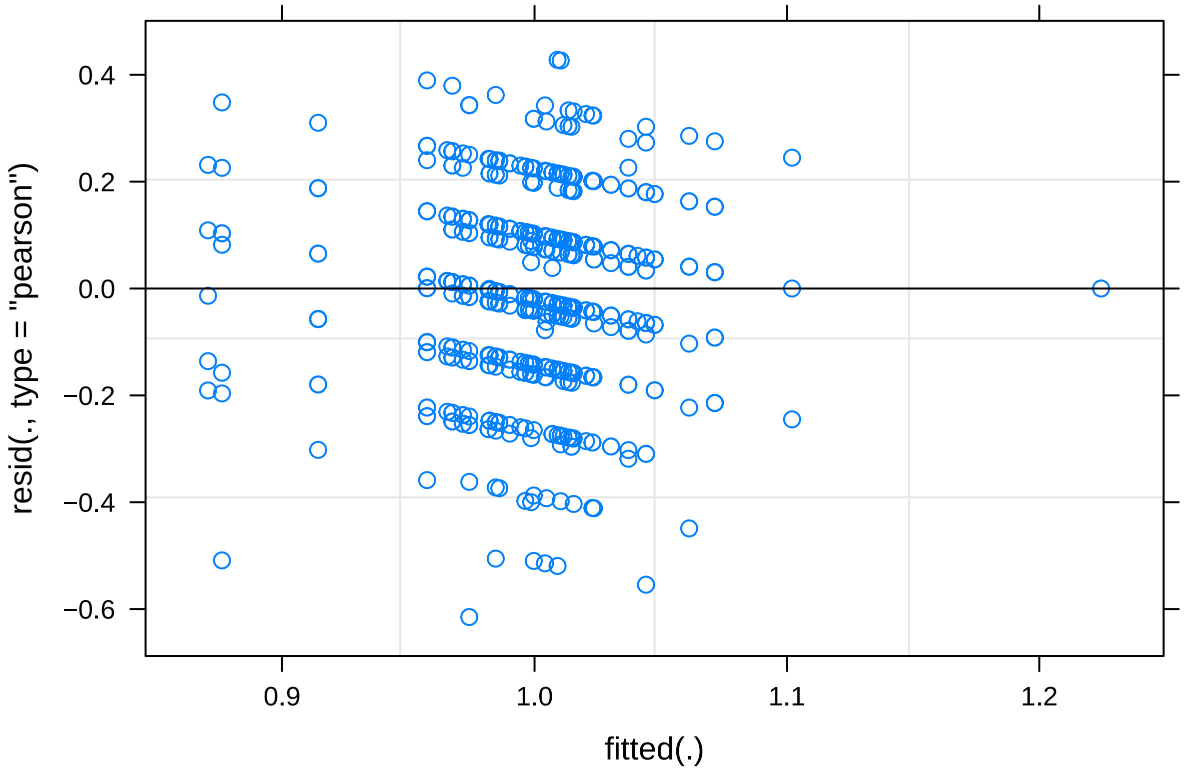
```
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 0.0175207
##  year     (Intercept) 0.0032054
##  Residual             0.1703556
```

```r
simulateResiduals(fittedModel = sMLH.model.females.lmer, plot = T)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.168 0.668 0.18 0.412 0.156 0.432 0.732 0.216 0.156 0.4 0.912 0.2 0.696 0.18
```

```r
plot(sMLH.model.females.lmer)
```



```
r.squaredGLMM(sMLH.model.females.lmer)
```

```
##             R2m        R2c
## [1,] 0.03590093 0.04632618
```

```
icc(model = sMLH.model.females.lmer, by_group = TRUE)
```

```
## # ICC by Group
##
## Group |       ICC
## -----------------
## pop   |     0.010
## year  | 3.502e-04
```

```
## chicks
sMLH.model.chicks.lmer <- lmerTest::lmer(sMLH ~ hunt + pop*year + hunt*year+
                                             (1|pop)+ (1|year) , data = sMLH.chicks)

# model failed to converge, still fails when taking out pop*year or hunt*year

#Rescale and center continuous parameters
numcols <- grep("^c\\.",names(sMLH.chicks))
sMLH.chicks_rescale <- sMLH.chicks
sMLH.chicks_rescale[,numcols] <- scale(sMLH.chicks_rescale[,numcols])
```

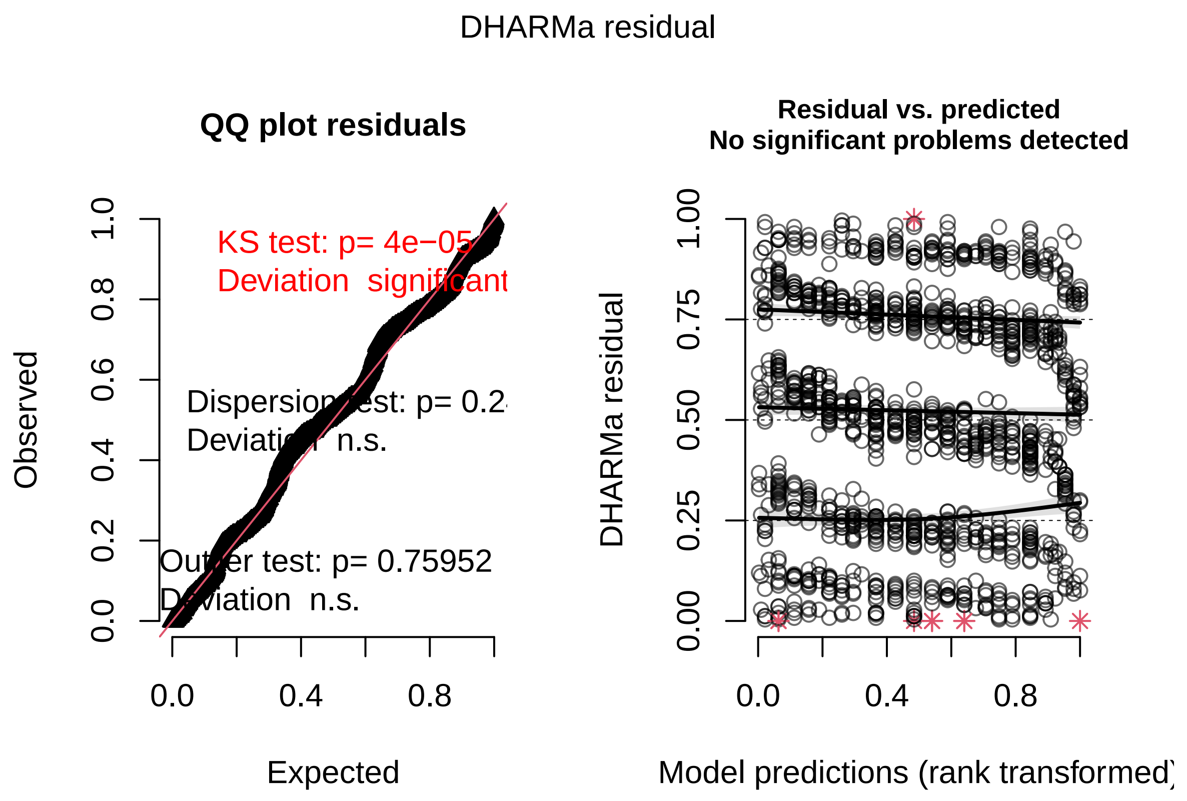
```r
sMLH.model.chicks.lmer_rescale <- update(sMLH.model.chicks.lmer,
                                         data=sMLH.chicks_rescale)

# restart and bump up max interations
ss <- getME(sMLH.model.chicks.lmer_rescale,c("theta","fixef"))
sMLH.model.chicks.lmer_noerror <- update(sMLH.model.chicks.lmer_rescale,
                                         start=ss,
                                         control=lmerControl(optCtrl=list(maxfun=2e4)))

# fixed
head(coef(summary(sMLH.model.chicks.lmer_noerror)))
```

```
##                  Estimate Std. Error           df    t value  Pr(>|t|)
## (Intercept)   0.936402882 0.04599114 2.096418e-07 20.3605074 0.9999976
## huntunhunted  0.045106398 0.03566295 1.801724e-07  1.2647973 0.9999984
## popKummunsuo  0.014044332 0.03284825 1.296786e-07  0.4275519 0.9999990
## popLehtusuo  -0.005071772 0.03626336 1.926157e-07 -0.1398594 0.9999988
## popNyrölä    -0.063090419 0.03445649 1.570013e-07 -1.8310172 0.9999986
## popSaarisuo  -0.023410484 0.03639544 1.954372e-07 -0.6432258 0.9999984
```

```r
VarCorr(sMLH.model.chicks.lmer_noerror)
```

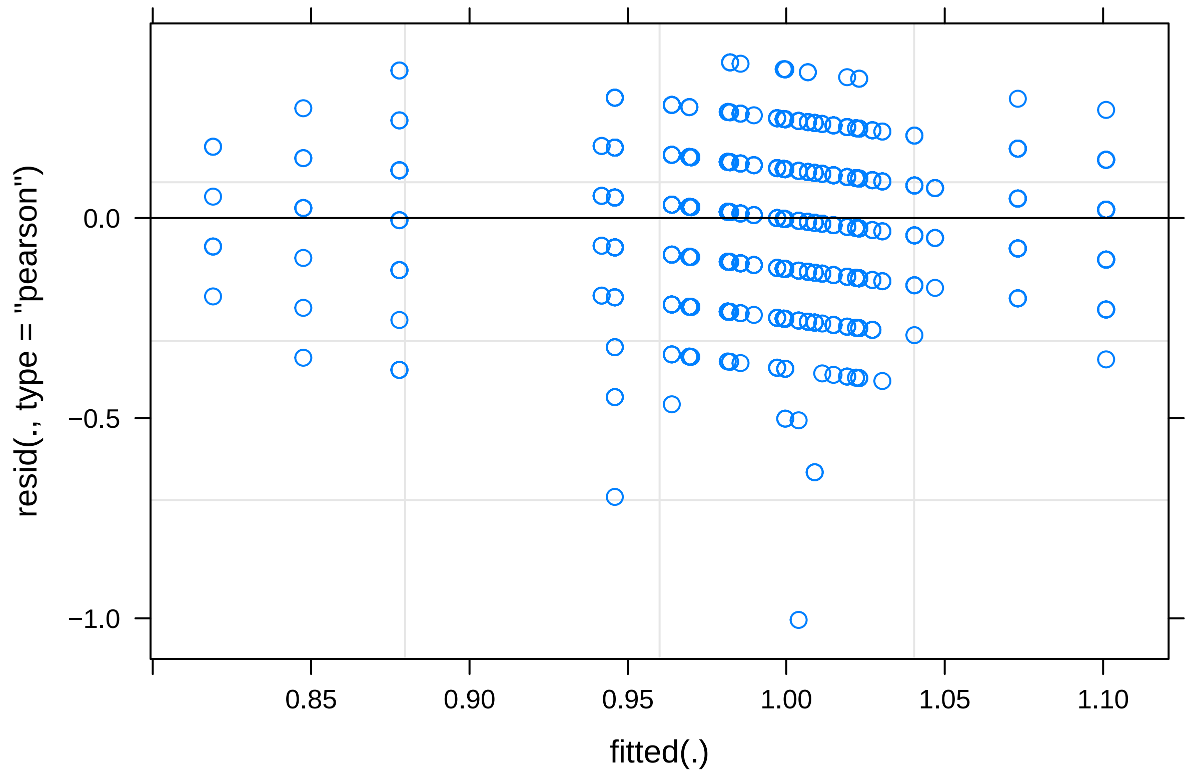
```
##  Groups   Name        Std.Dev.
##  pop      (Intercept) 0.013122
##  year     (Intercept) 0.023878
##  Residual             0.168827
```

```r
simulateResiduals(fittedModel = sMLH.model.chicks.lmer_noerror, plot = T)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.556 0.776 0.792 0.524 0.508 0.312 0.528 0.256 0.284 0.512 0.248 0.008 0.55
```

```
plot(sMLH.model.chicks.lmer_noerror)
```



```r
r.squaredGLMM(sMLH.model.chicks.lmer_noerror)
```

```
##             R2m        R2c
## [1,] 0.05312068 0.07715702
```

```r
icc(model = sMLH.model.chicks.lmer_noerror, by_group = TRUE)
```

```
## # ICC by Group
##
## Group |   ICC
## -------------
## pop   | 0.006
## year  | 0.019
```

Migration models

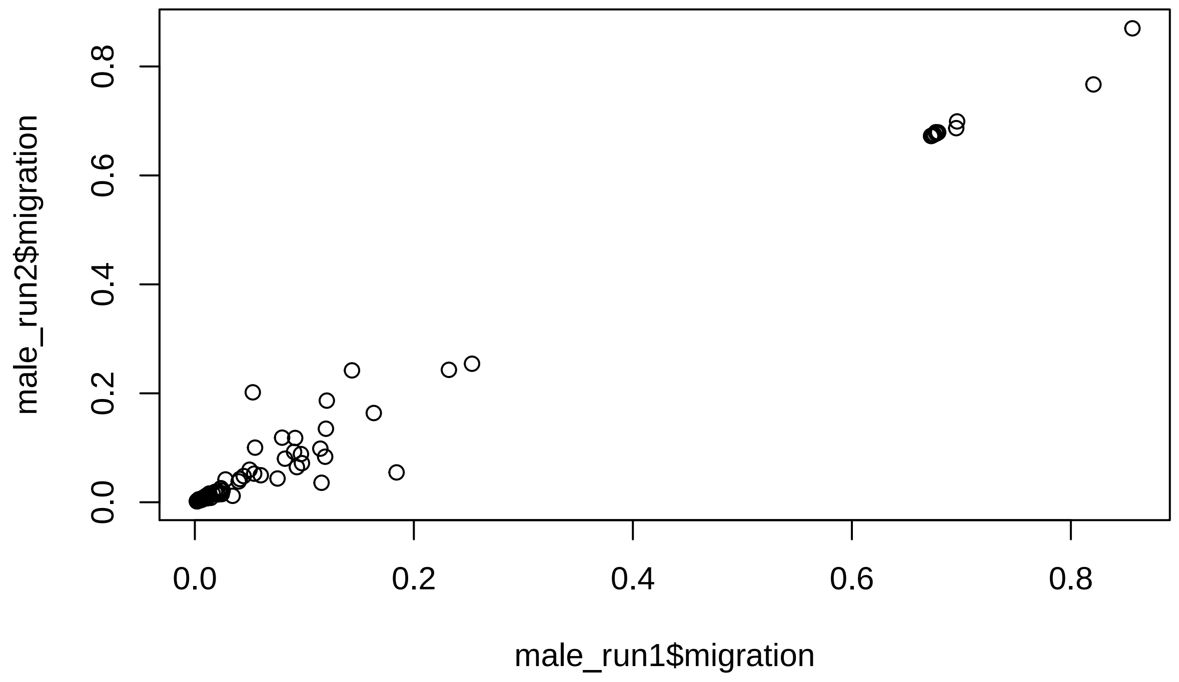
Next, we investigated patterns of dispersion and how these are affected by hunting. We used BA3 to calculate migration directions and rates, followed by building mixed models to estimate the effect of hunting on migration.

```r
# load in pop data
pops <- read.csv("data/details/Codes.pops.both.filtered_withcoord.csv")
pops$pop_num <- as.factor(pops$pop_num)
```

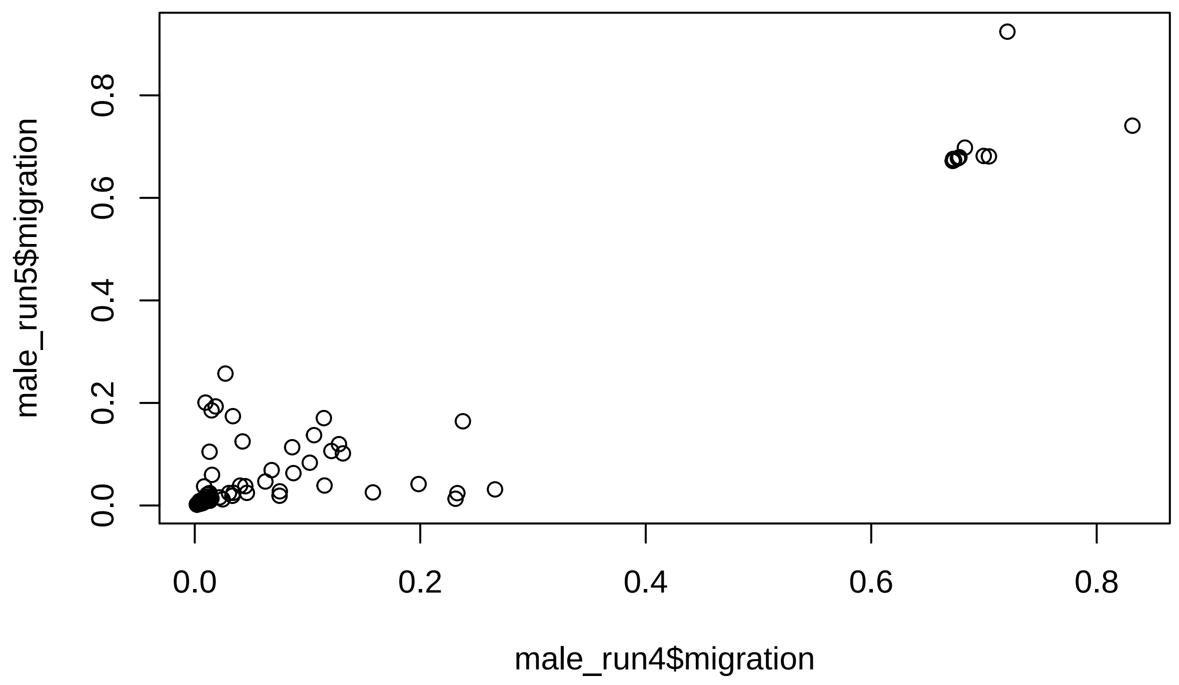
```
## load in Matrix distances between sites as calculated with GenAlEx ##
distance <- read_excel("data/details/CalculateDistanceSitesGenAlEx.xlsx",
                       sheet = "MatrixForR")
names(distance)[1] <- "Site_A"
distance_long <- melt(distance)
names(distance_long) <- c("Site_A", "Site_B", "Distance")
distance_long <- subset(distance_long, distance_long$Site_A != distance_long$Site_B)

# files were reformatted to fit BA3, see the script in the github directory
# called 5.migrationmodels.R for more details

#### Running BA3 ####

## males
#install BA3, run through command line/terminal
#made a directory per run using the command:
system("mkdir /data/migrationanalysis/BA3runs/males_run1") #repeat for run1-run5

#5 runs with 5 different random seeds
system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 65323 -i 10000000
       -b 1000000 -n 1000 -o males_run1.txt
       /data/migrationanalysis/data_males_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 76553 -i 10000000
       -b 1000000 -n 1000 -o males_run2.txt
       /data/migrationanalysis/data_males_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 124643 -i 10000000
       -b 1000000 -n 1000 -o males_run3.txt
       /data/migrationanalysis/data_males_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 885256 -i 10000000
       -b 1000000 -n 1000 -o males_run4txt
       /data/migrationanalysis/data_males_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 235776 -i 10000000
       -b 1000000 -n 1000 -o males_run5.txt
       /data/migrationanalysis/data_males_ba3.txt")

## females

system("mkdir /data/migrationanalysis/BA3runs/females_run1") #repeat for run1-run5

#5 runs with 5 different random seeds
system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 65323 -i 10000000
       -b 1000000 -n 1000 -o females_run1.txt
       /data/migrationanalysis/data_females_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 76553 -i 10000000
       -b 1000000 -n 1000 -o females_run2.txt
       /data/migrationanalysis/data_females_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 124643 -i 10000000
       -b 1000000 -n 1000 -o females_run3.txt
       /data/migrationanalysis/data_females_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 885256 -i 10000000
       -b 1000000 -n 1000 -o females_run4txt
       /data/migrationanalysis/data_females_ba3.txt")

system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 235776 -i 10000000
       -b 1000000 -n 1000 -o females_run5.txt
       /data/migrationanalysis/data_females_ba3.txt")
```

```
#### Compare all 10 runs ####
temp <- list.files(path = "data/migrationanalysis/BA3runs/",
                   pattern = ".txt", full.names=T)
myfiles = lapply(temp, fread, skip = 18, nrows = 12, header = F)

# formula for reshaping the dataframes

reshape_ba3 <- function(m) {
  m1 <- m[,c(1,2)]
  names(m1) <- c("pops", "migration")
  m2 <- m[,c(3,4)]
  names(m2) <- c("pops", "migration")
  m3 <- m[,c(5,6)]
  names(m3) <- c("pops", "migration")
  m4 <- m[,c(7,8)]
  names(m4) <- c("pops", "migration")
  m5 <- m[,c(9,10)]
  names(m5) <- c("pops", "migration")
  m6 <- m[,c(11,12)]
  names(m6) <- c("pops", "migration")
  m7 <- m[,c(13,14)]
  names(m7) <- c("pops", "migration")
  m8 <- m[,c(15,16)]
  names(m8) <- c("pops", "migration")
  m9 <- m[,c(17,18)]
  names(m9) <- c("pops", "migration")
  m10 <- m[,c(19,20)]
  names(m10) <- c("pops", "migration")
  m11 <- m[,c(21,22)]
  names(m11) <- c("pops", "migration")
  m12 <- m[,c(23,24)]
  names(m12) <- c("pops", "migration")
  mnew<-rbind(m1, m2, m3, m4, m5,m6,m7,m8,m9,m10,m11,m12)
  mnew$m_in <- c(rep(c(1:12), times = 12, each = 1))
  mnew$m_out <- c(rep(c(1:12), times = 1, each = 12))
  mnew <- separate(data = mnew, col = "migration", into = c("migration", "migration_SE"),
                   sep = "[(]") #seperate migration and its SE
  mnew$migration_SE <- gsub(mnew$migration_SE, pattern = "[)]", replacement = "")
  mnew <- mnew[,c(4,5,2,3)]
  return(mnew)
}
```

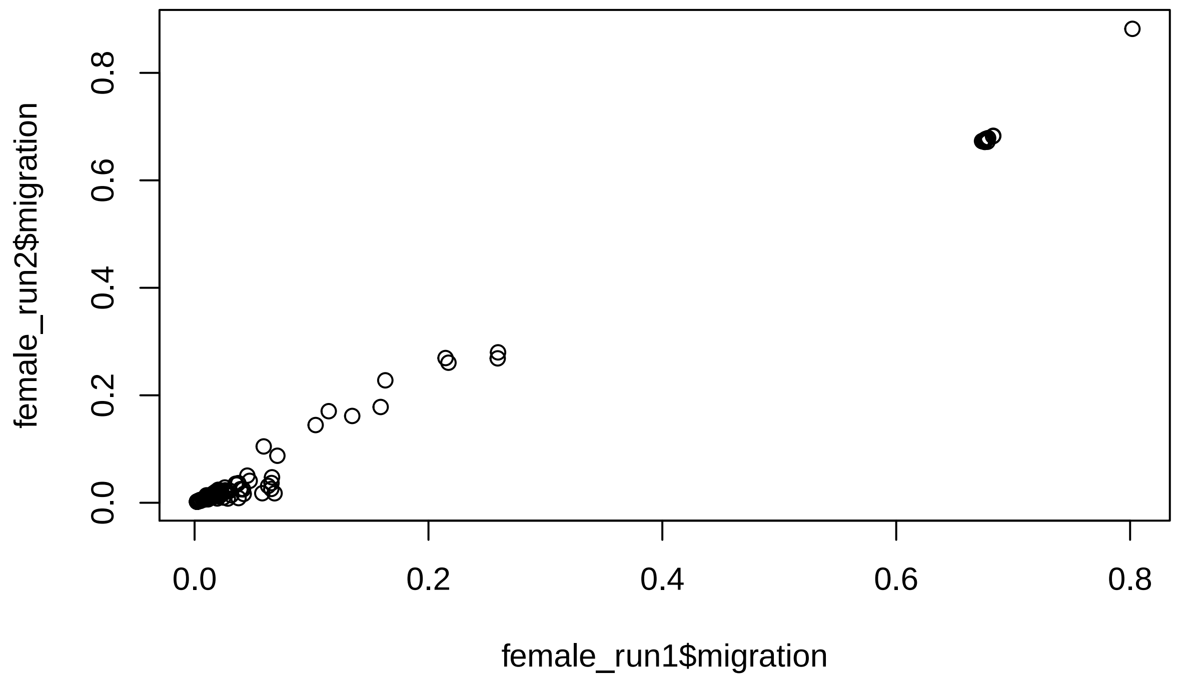
```r
# run it for all files
for (i in 1:length(myfiles)) {
  myfiles[[i]]<-reshape_ba3(myfiles[[i]])
}

#separate for males and females
maleruns <- myfiles[c(6:10)]
femaleruns <- myfiles[c(1:5)]

#separate per run to compare
male_run1 <- maleruns[[1]]
male_run2 <- maleruns[[2]]
male_run3 <- maleruns[[3]]
male_run4 <- maleruns[[4]]
male_run5 <- maleruns[[5]]

female_run1 <- femaleruns[[1]]
female_run2 <- femaleruns[[2]]
female_run3 <- femaleruns[[3]]
female_run4 <- femaleruns[[4]]
female_run5 <- femaleruns[[5]]

#### Compare runs ####
plot(male_run1$migration, male_run2$migration)
```



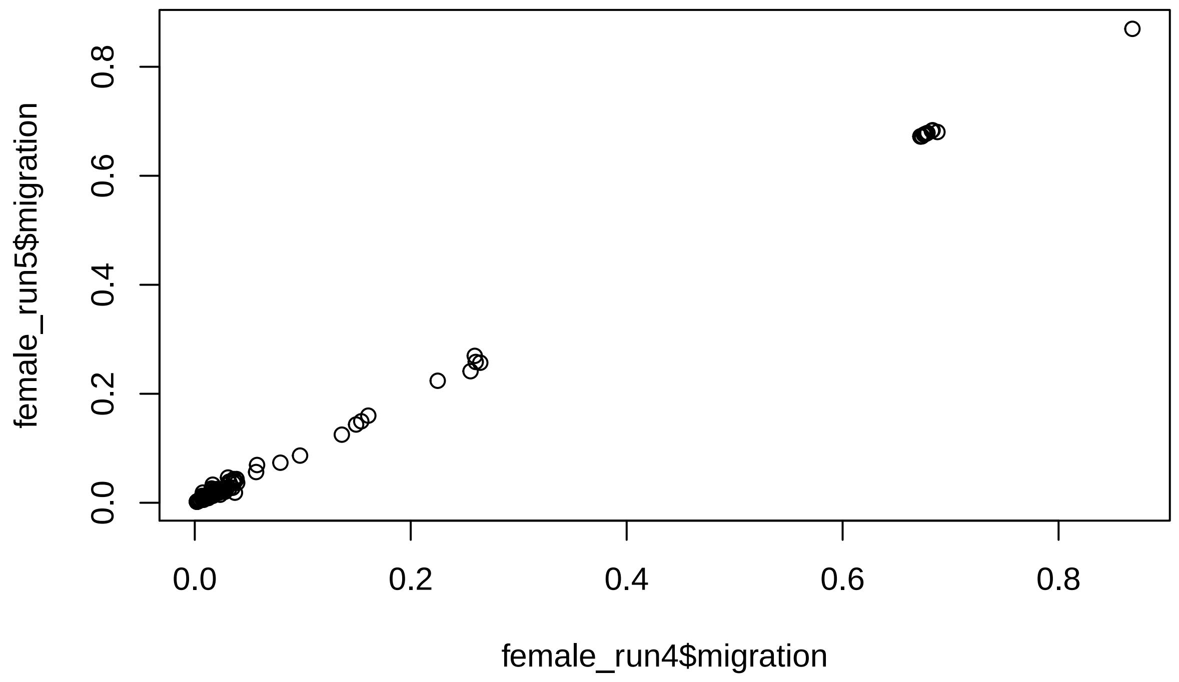
```
#plot(male_run1$migration, male_run3$migration)
# plot(male_run1$migration, male_run4$migration)
# plot(male_run1$migration, male_run5$migration)
# plot(male_run2$migration, male_run3$migration)
# plot(male_run2$migration, male_run4$migration)
# plot(male_run2$migration, male_run5$migration)
# plot(male_run3$migration, male_run4$migration)
# plot(male_run3$migration, male_run5$migration)
plot(male_run4$migration, male_run5$migration)
```



```
# all runs correspond

plot(female_run1$migration, female_run2$migration)
```



```
# plot(female_run1$migration, female_run3$migration)
# plot(female_run1$migration, female_run4$migration)
# plot(female_run1$migration, female_run5$migration)
# plot(female_run2$migration, female_run3$migration)
# plot(female_run2$migration, female_run4$migration)
# plot(female_run2$migration, female_run5$migration)
# plot(female_run3$migration, female_run4$migration)
# plot(female_run3$migration, female_run5$migration)
plot(female_run4$migration, female_run5$migration)
```



```
# all runs correspond

## Going to pick run 5 for both
```

```
#### Load in cleaned migration files #### again, see R script 5.migrationmodels.R
#### for details on how to merge raw BA3 files with ESS, how the migration values
#### were corrected (set to NA when ESS < 200), added hunted status, added distance
#### between sites

male_run5_clean <- read.csv("data/migrationanalysis/run5_males_clean.csv")
female_run5_clean <- read.csv("data/migrationanalysis/run5_females_clean.csv")

### Plotting migration rates from run 5 #### first, exclude the 'non-migration
### rates' which are those where pop in = pop out
male_run5_clean <- subset(male_run5_clean, m_in != m_out)
female_run5_clean <- subset(female_run5_clean, m_in != m_out)

#### Modelling migration ####

female_run5_clean$sex <- "Female"
male_run5_clean$sex <- "Male"

# change levels hunted/unhunted
male_run5_clean$hunt_in <- relevel(as.factor(male_run5_clean$hunt_in), ref = "unhunted")
male_run5_clean$hunt_out <- relevel(as.factor(male_run5_clean$hunt_out), ref = "unhunted")
```

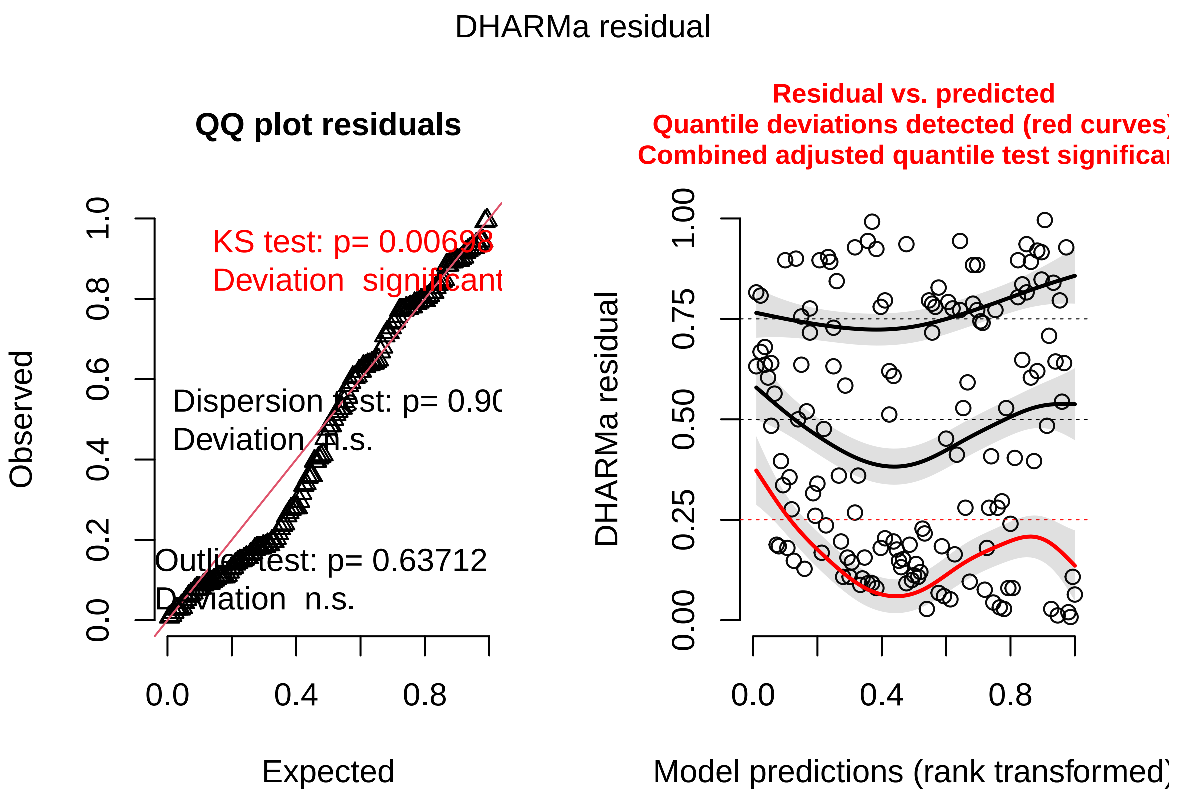
```
female_run5_clean$hunt_in <- relevel(as.factor(female_run5_clean$hunt_in), ref = "unhunted")
female_run5_clean$hunt_out <- relevel(as.factor(female_run5_clean$hunt_out), ref = "unhunted")

# combine in one df
migration_both <- rbind(male_run5_clean, female_run5_clean)

# out
model.both.out <- glmer(migration_ESSc ~ hunt_out + Distance + sex + (1 | pop_out) +
    (1 | pop_in), data = migration_both, family = Gamma(link = "log"))

model.both.out.TMB <- glmmTMB(migration_ESSc ~ hunt_out + Distance + sex + (1 | pop_out) +
    (1 | pop_in), data = migration_both, family = Gamma(link = "log"))

model.both.out.TMB.inter <- glmmTMB(migration_ESSc ~ hunt_out + Distance * sex +
    (1 | pop_out) + (1 | pop_in), data = migration_both, family = Gamma(link = "log"))

compare_performance(model.both.out, model.both.out.TMB, model.both.out.TMB.inter,
    rank = T)
```

```
## # Comparison of Model Performance Indices
##
## Name                     |    Model | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma | AIC weights
## ---------------------------------------------------------------------------------------------------
## model.both.out.TMB.inter |  glmmTMB |      0.885 |      0.094 | 0.873 | 0.002 | 0.244 |     < 0.001
## model.both.out.TMB       |  glmmTMB |      0.869 |      0.075 | 0.858 | 0.002 | 0.259 |     < 0.001
## model.both.out           | glmerMod |      0.654 |      0.187 | 0.574 | 0.002 | 0.304 |       1.000
```

```
# TMB with interaction has a higher performance score
summary(model.both.out.TMB.inter)
```

```
##  Family: Gamma  ( log )
## Formula:          migration_ESSc ~ hunt_out + Distance * sex + (1 | pop_out) +
##     (1 | pop_in)
## Data: migration_both
##
##      AIC      BIC   logLik deviance df.resid
##  -1480.7  -1456.6    748.4  -1496.7      142
##
## Random effects:
##
## Conditional model:
##  Groups  Name        Variance Std.Dev.
##  pop_out (Intercept) 0.05152  0.2270
##  pop_in  (Intercept) 0.35738  0.5978
## Number of obs: 150, groups:  pop_out, 10; pop_in, 12
##
## Dispersion estimate for Gamma family (sigma^2): 0.0594
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -4.567953   0.232963 -19.608  < 2e-16 ***
## hunt_outhunted   -0.355191   0.165590  -2.145   0.0320 *
## Distance         -0.005827   0.002613  -2.230   0.0257 *
## sexMale           -0.602723   0.085969  -7.011 2.37e-12 ***
## Distance:sexMale   0.011732   0.002910   4.032 5.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simulateResiduals(fittedModel = model.both.out.TMB.inter, plot = T)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.204 0.268 0.18 0.08 0.512 0.396 0.196 0.18 0.168 0.184 0.068 0.34 0.148 0.
```

```
# immigration model
model.both.in <- lme4::glmer(migration_ESSc ~ hunt_in + Distance + sex + (1 | pop_out) +
    (1 | pop_in), data = migration_both, family = Gamma(link = "log"))


model.both.in.TMB <- glmmTMB(migration_ESSc ~ hunt_in + Distance + sex + (1 | pop_out) +
    (1 | pop_in), data = migration_both, family = Gamma(link = "log"))

model.both.in.TMB.inter <- glmmTMB(migration_ESSc ~ hunt_in + Distance * sex + (1 |
    pop_out) + (1 | pop_in), data = migration_both, family = Gamma(link = "log"))

compare_performance(model.both.in, model.both.in.TMB, model.both.in.TMB.inter, rank = T)
```

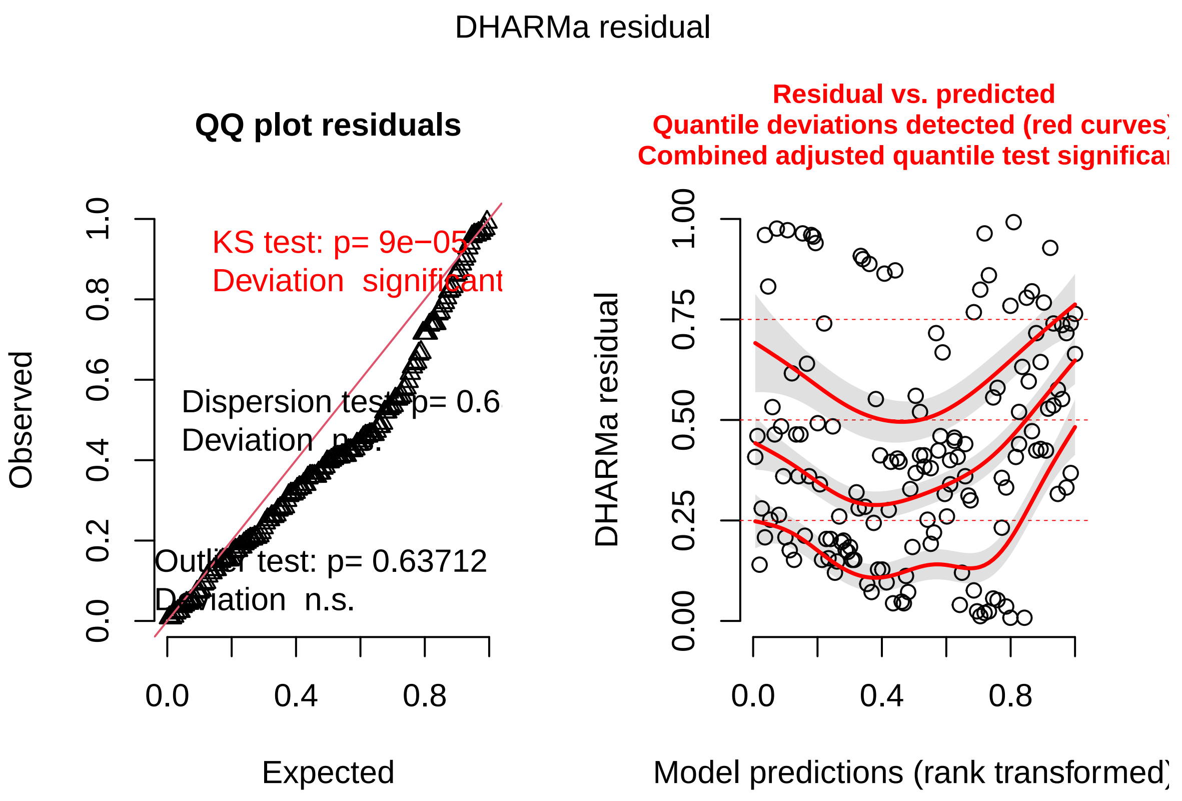
```
## # Comparison of Model Performance Indices
```

```
##
## Name                       |    Model | R2 (cond.) | R2 (marg.) |   ICC |  RMSE | Sigma | AIC weights |
## -----------------------------------------------------------------------------------------------------
## model.both.in.TMB.inter |  glmmTMB |      0.888 |      0.314 | 0.837 | 0.002 | 0.244 |     < 0.001 |
## model.both.in.TMB       |  glmmTMB |      0.870 |      0.321 | 0.809 | 0.002 | 0.259 |     < 0.001 |
## model.both.in           | glmerMod |      0.747 |      0.440 | 0.548 | 0.002 | 0.296 |       1.000 |
```

```
# TMB and lmer model perform the same, but TMB captures more variation, just keep
# TMB for consistency with emigration model

summary(model.both.in.TMB.inter)
```

```
##  Family: Gamma  ( log )
## Formula:
## migration_ESSc ~ hunt_in + Distance * sex + (1 | pop_out) + (1 |      pop_in)
## Data: migration_both
##
##      AIC      BIC   logLik deviance df.resid
##  -1482.0  -1457.9    749.0  -1498.0      142
##
## Random effects:
##
## Conditional model:
##  Groups  Name        Variance Std.Dev.
##  pop_out (Intercept) 0.08214  0.2866
##  pop_in  (Intercept) 0.22338  0.4726
## Number of obs: 150, groups:  pop_out, 10; pop_in, 12
##
## Dispersion estimate for Gamma family (sigma^2): 0.0595
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.127254   0.231367 -22.161  < 2e-16 ***
## hunt_inhunted     0.724004   0.276100   2.622  0.00874 **
## Distance         -0.006588   0.002598  -2.535  0.01123 *
## sexMale          -0.595888   0.086419  -6.895 5.37e-12 ***
## Distance:sexMale  0.011490   0.002929   3.923 8.74e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simulateResiduals(fittedModel = model.both.in.TMB.inter, plot = T)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DH
##
## Scaled residual values: 0.052 0.076 0.056 0.04 0.232 0.3 0.036 0.264 0.212 0.252 0.208 0.464 0.208 0
```

```
## model performance
icc(model = model.both.out, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ----------------
## pop_in  | 0.445
## pop_out | 0.129
```

```
icc(model = model.both.out.TMB, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ----------------
## pop_out | 0.096
## pop_in  | 0.762
```

```r
icc(model = model.both.out.TMB.inter, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ---------------
## pop_out | 0.110
## pop_in  | 0.763
```

```r
icc(model = model.both.in, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ---------------
## pop_in  | 0.344
## pop_out | 0.205
```

```r
icc(model = model.both.in.TMB, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ---------------
## pop_out | 0.191
## pop_in  | 0.618
```

```r
icc(model = model.both.in.TMB.inter, by_group = TRUE)
```

```
## # ICC by Group
##
## Group   |   ICC
## ---------------
## pop_out | 0.225
## pop_in  | 0.612
```

```r
r.squaredGLMM(model.both.out)
```

```
##                 R2m       R2c
## delta     0.1866620 0.6535519
## lognormal 0.1895129 0.6635336
## trigamma  0.1836411 0.6429749
```

```r
r.squaredGLMM(model.both.out.TMB)
```

```
##                  R2m       R2c
## delta     0.07528584 0.8688960
## lognormal 0.07560350 0.8725622
## trigamma  0.07494962 0.8650156
```

```
r.squaredGLMM(model.both.out.TMB.inter)
```

```
##                   R2m       R2c
## delta     0.09401487 0.8851110
## lognormal 0.09432444 0.8880254
## trigamma  0.09368896 0.8820426
```

```
r.squaredGLMM(model.both.in)
```

```
##                  R2m       R2c
## delta     0.4398634 0.7469891
## lognormal 0.4445053 0.7548722
## trigamma  0.4349193 0.7385930
```

```
r.squaredGLMM(model.both.in.TMB)
```

```
##                  R2m       R2c
## delta     0.3213707 0.8703384
## lognormal 0.3227171 0.8739846
## trigamma  0.3199453 0.8664780
```

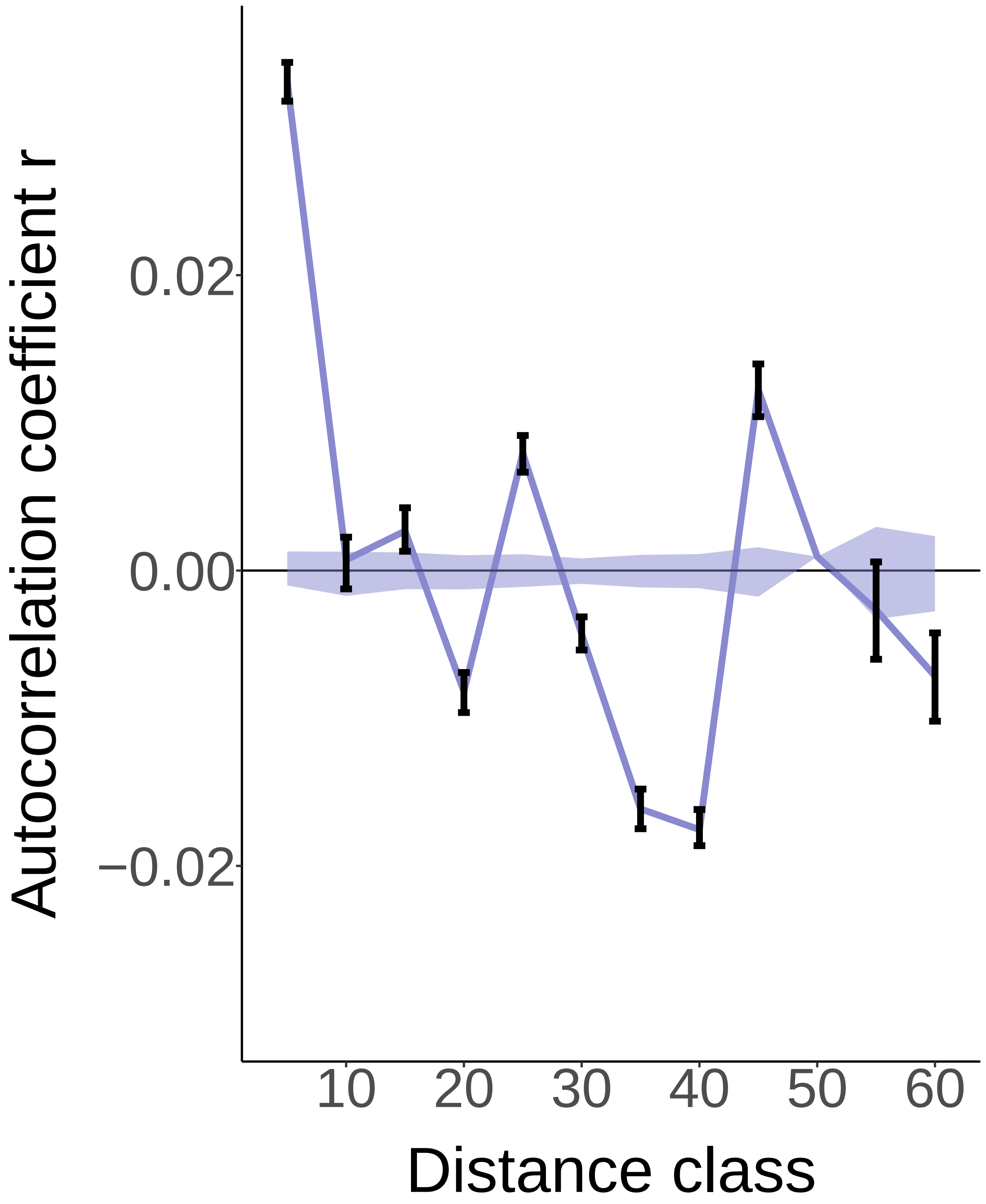
```
r.squaredGLMM(model.both.in.TMB.inter)
```

```
##                  R2m       R2c
## delta     0.3141518 0.8881711
## lognormal 0.3151608 0.8910238
## trigamma  0.3130891 0.8851667
```

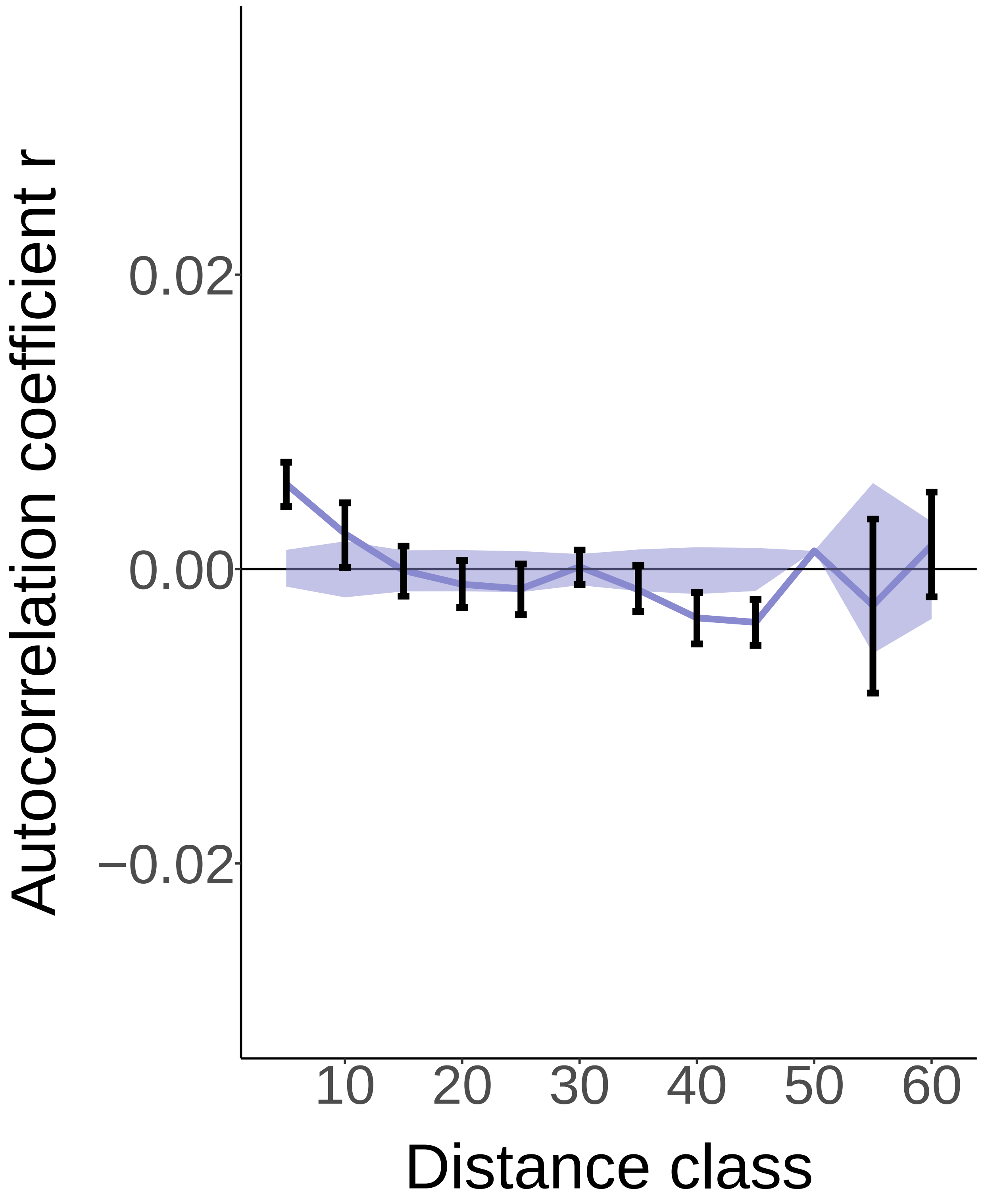
Other plots included in the manuscript

Figure 1 was created with qGIS, for barcharts of STRUCTURE output as in Supplementary Figure 2, please refer to R script 6.CreatingPlots.R

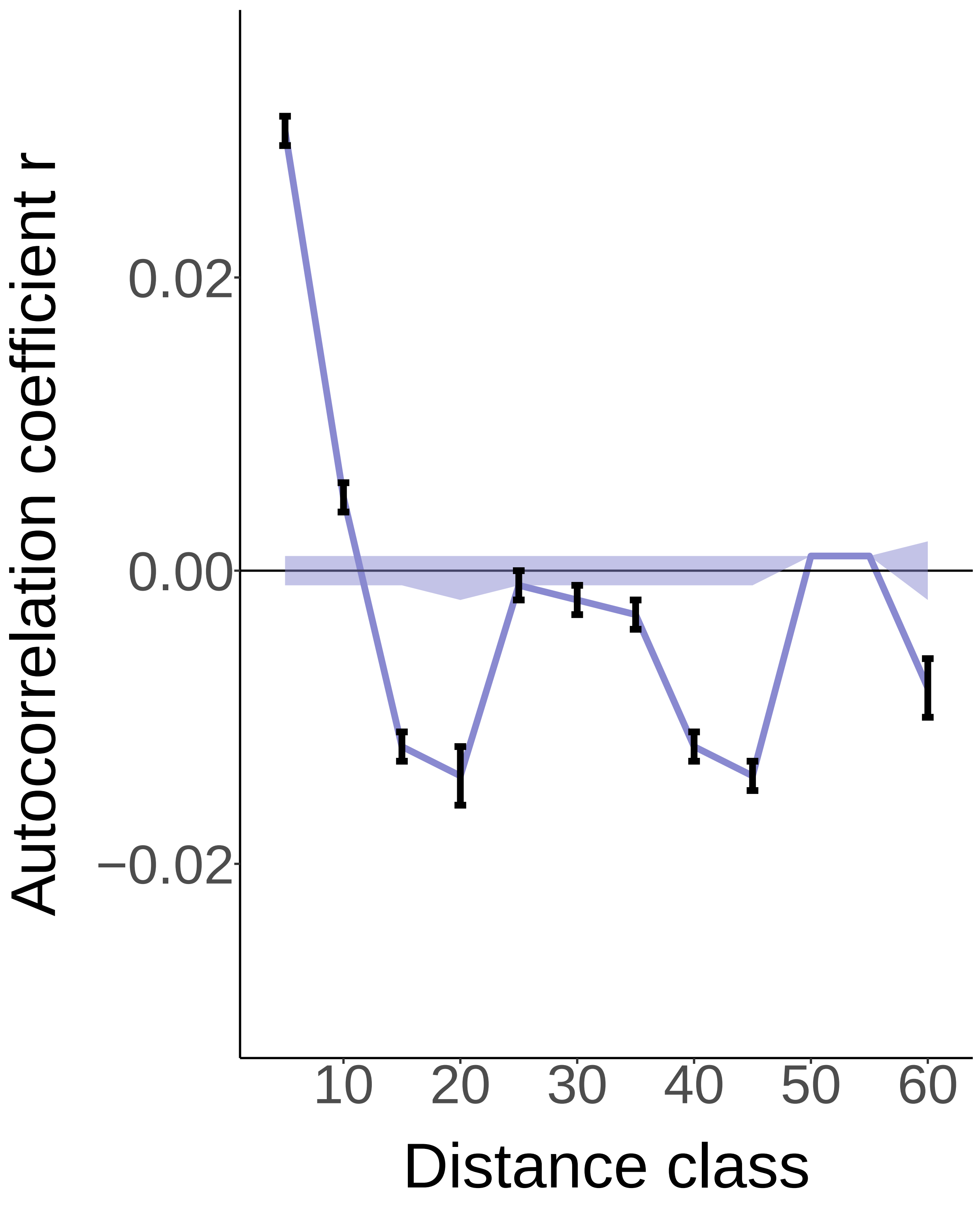
```
### Figure 2 - correlograms
spatial <- read_excel("data/tables/SpatialAutocor_4.4.22.xlsx", sheet = "ForR")

### Males - spatial
spatial %>% filter(Who == "Male") %>% ggplot(aes(x = What)) +
  geom_line(aes(y = r), col = "#8989D0", size = 1.5) + theme_classic() +
  geom_hline(yintercept = 0, col = "black")+
  geom_ribbon(aes(ymax = U, ymin = L), fill = "#8989D0", alpha = 0.5)+
  geom_errorbar(aes(y = r, ymin = r-Le, ymax = r+Ue), width=1,
                size=1.5, color="black", stat = "identity")+
  ylim(-0.03, 0.035) +
  xlab("Distance class") + ylab("Autocorrelation coefficient r")+
  scale_x_continuous(breaks = c(seq(0, 60, by = 10)), limits=c(4,61))+
  theme(text = element_text(size = 14),
        plot.title = element_text(size = 38),
        axis.text.x = element_text(size = 26, margin = margin(b = 10)),
        axis.text.y = element_text(size = 26, margin = margin(l = 10)),
        axis.title.x = element_text(size = 30),
        axis.title.y = element_text(size = 30))
```



```r
### Females - spatial

spatial %>% filter(Who == "Females") %>% ggplot(aes(x = What)) +
  geom_line(aes(y = r), col = "#8989D0", size = 1.5) + theme_classic() +
  geom_hline(yintercept = 0, col = "black")+
  geom_ribbon(aes(ymax = U, ymin = L), fill = "#8989D0", alpha = 0.5)+
  geom_errorbar(aes(y = r, ymin = r-Le, ymax = r+Ue), width=1, size=1.5,
                color="black", stat = "identity")+
  ylim(-0.03, 0.035) +
  xlab("Distance class") + ylab("Autocorrelation coefficient r")+
  scale_x_continuous(breaks = c(seq(0, 60, by = 10)), limits=c(4,61))+
  theme(text = element_text(size = 14),
        plot.title = element_text(size = 38),
        axis.text.x = element_text(size = 26, margin = margin(b = 10)),
        axis.text.y = element_text(size = 26, margin = margin(l = 10)),
        axis.title.x = element_text(size = 30),
        axis.title.y = element_text(size = 30))
```

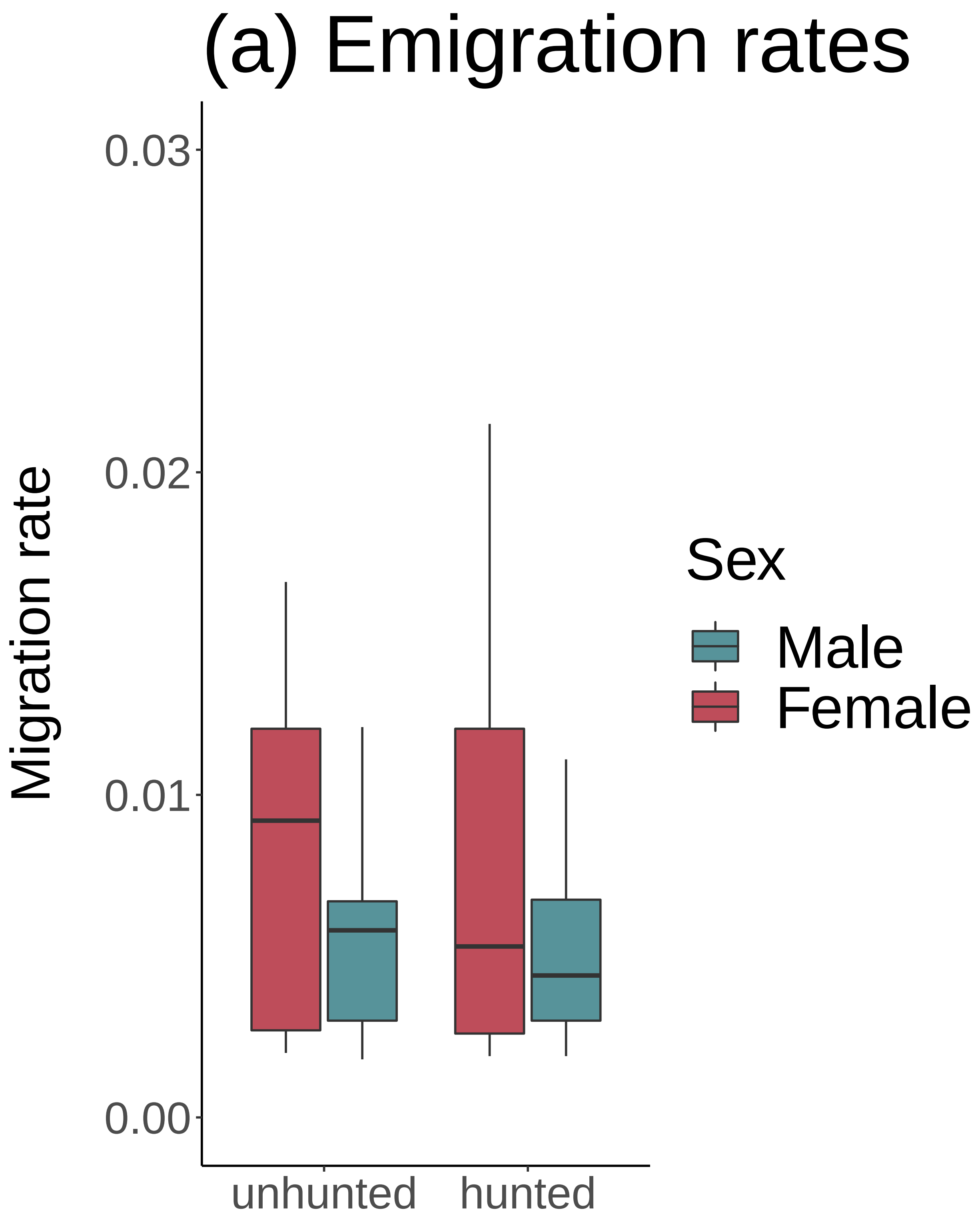


```r
### Chicks - spatial
spatial %>% filter(Who == "Chicks") %>% ggplot(aes(x = What)) +
  geom_line(aes(y = r), col = "#8989D0", size = 1.5) + theme_classic() +
  geom_hline(yintercept = 0, col = "black")+
  geom_ribbon(aes(ymax = U, ymin = L), fill = "#8989D0", alpha = 0.5)+
  geom_errorbar(aes(y = r, ymin = r-Le, ymax = r+Ue), width=1, size=1.5,
                color="black", stat = "identity")+
  ylim(-0.03, 0.035) +
  xlab("Distance class") + ylab("Autocorrelation coefficient r")+
  scale_x_continuous(breaks = c(seq(0, 60, by = 10)), limits=c(4,61))+
  theme(text = element_text(size = 14),
        plot.title = element_text(size = 38),
        axis.text.x = element_text(size = 26, margin = margin(b = 10)),
        axis.text.y = element_text(size = 26, margin = margin(l = 10)),
        axis.title.x = element_text(size = 30),
        axis.title.y = element_text(size = 30))
```



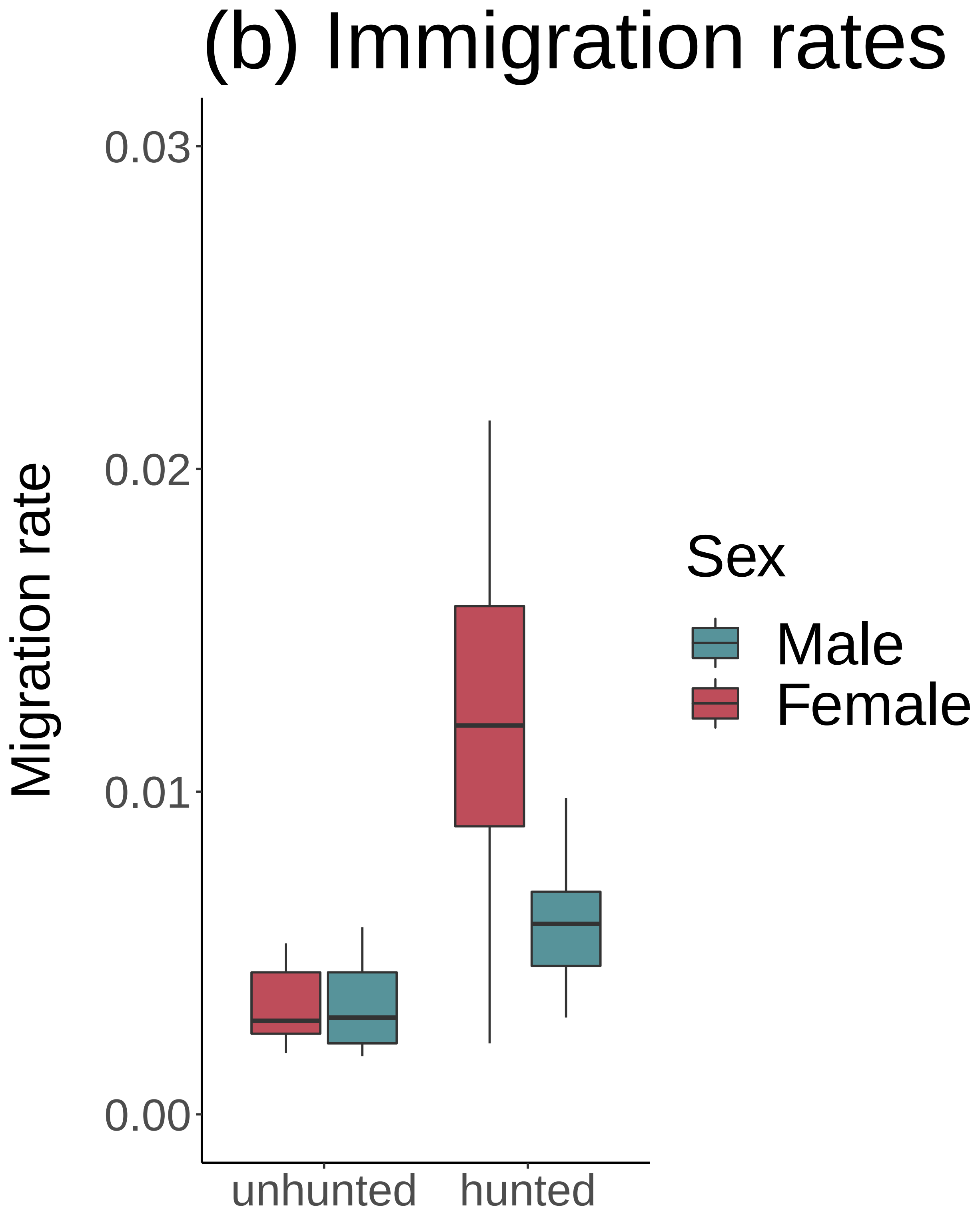
```
### Figure 3: boxplots migration rates ####
male_run5_clean <- read.csv("data/migrationanalysis/run5_males_clean.csv")
female_run5_clean <- read.csv("data/migrationanalysis/run5_females_clean.csv")

female_run5_clean$sex <- "Female"
```

```r
male_run5_clean$sex <- "Male"

#combine in one df
migration_both <- rbind(male_run5_clean, female_run5_clean)

#change levels
migration_both$hunt_in <- relevel(as.factor(migration_both$hunt_in),
                                  ref = "unhunted")
migration_both$hunt_out <- relevel(as.factor(migration_both$hunt_out),
                                   ref = "unhunted")

#first, exclude the 'non-migration rates' which are those where pop in = pop out
migration_both <- subset(migration_both, m_in != m_out)

# plot emigration rates
ggplot(migration_both, aes(x = hunt_out, y = migration_ESSc, fill = sex)) +
  geom_boxplot(outlier.shape = NA, aes(middle = mean(migration_ESSc))) +
  ylim(0, 0.03)+
  labs(title = "(a) Emigration rates") +
  ylab("Migration rate")+
  theme(text = element_text(size = 26),
        legend.text = element_text(size = 28),
        legend.title = element_text(size = 28),
        legend.key.size = unit(1, 'cm'),
        plot.title = element_text(size = 38),
        axis.title.x = element_blank(),
        axis.title.y = element_text(margin = margin
                                    (t = 0, r = 20, b = 0, l = 0)))+
  scale_fill_manual(values = c("Male" = "#57939a",
                               "Female" = "#be4d5a"),
                    labels=c("Male", "Female"))+
  guides(fill = guide_legend("Sex"))
```



```
#plot immigration rates

ggplot(migration_both, aes(x = hunt_in, y = migration_ESSc, fill = sex)) +
  geom_boxplot(outlier.shape = NA, aes(middle = mean(migration_ESSc))) +
  ylim(0, 0.03)+
```

```r
  labs(title = "(b) Immigration rates") +
  ylab("Migration rate")+
  theme(text = element_text(size = 26),
        legend.text = element_text(size = 28),
        legend.title = element_text(size = 28),
        legend.key.size = unit(1, 'cm'),
        plot.title = element_text(size = 38),
        axis.title.x = element_blank(),
        axis.title.y = element_text(margin = margin
                                    (t = 0, r = 20, b = 0, l = 0)))+
  scale_fill_manual(values = c("Male" = "#57939a",
                               "Female" = "#be4d5a"),
                    labels=c("Male", "Female"))+
  guides(fill = guide_legend("Sex"))
```



See below session information including package versions.

```
session_info()
```

```
## - Session info --------------------------------------------------------------
##  setting  value
##  version  R version 4.0.2 (2020-06-22)
##  os       macOS  10.16
##  system   x86_64, darwin17.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Europe/Amsterdam
##  date     2022-05-03
##
## - Packages -------------------------------------------------------------------
##  package           * version    date       lib
##  ade4              * 1.7-18     2021-09-16 [1]
##  adegenet          * 2.1.5      2021-10-09 [1]
##  ape               * 5.5        2021-04-25 [1]
##  assertthat          0.2.1      2019-03-21 [1]
##  backports           1.2.1      2020-12-09 [1]
##  boot                1.3-27     2021-02-12 [1]
##  broom               0.7.5      2021-02-19 [1]
##  cachem              1.0.4      2021-02-13 [1]
##  callr               3.5.1      2020-10-13 [1]
##  cellranger          1.1.0      2016-07-27 [1]
##  cli                 3.0.1      2021-07-17 [1]
##  cluster             2.1.1      2021-02-14 [1]
##  coda                0.19-4     2020-09-30 [1]
##  codetools           0.2-18     2020-11-04 [1]
##  colorspace          2.0-2      2021-06-24 [1]
##  combinat            0.0-8      2012-10-29 [1]
##  crayon              1.4.1      2021-02-08 [1]
##  data.table        * 1.14.0     2021-02-21 [1]
##  DBI                 1.1.1      2021-01-15 [1]
##  dbplyr              2.1.0      2021-02-03 [1]
##  desc                1.2.0      2018-05-01 [1]
##  devtools          * 2.3.2      2020-09-18 [1]
##  DHARMa            * 0.4.5      2022-01-16 [1]
##  digest              0.6.27     2020-10-24 [1]
##  doParallel          1.0.16     2020-10-16 [1]
##  dplyr             * 1.0.7      2021-06-18 [1]
##  ellipsis            0.3.2      2021-04-29 [1]
##  emmeans             1.7.3      2022-03-27 [1]
##  estimability        1.3        2018-02-11 [1]
##  evaluate            0.14       2019-05-28 [1]
##  extrafont         * 0.17       2014-12-08 [1]
##  extrafontdb         1.0        2012-06-11 [1]
##  fansi               0.5.0      2021-05-25 [1]
##  farver              2.1.0      2021-02-28 [1]
##  fastmap             1.1.0      2021-01-25 [1]
##  forcats           * 0.5.1      2021-01-27 [1]
##  foreach             1.5.1      2020-10-15 [1]
##  formatR             1.7        2019-06-11 [1]
##  fs                  1.5.0      2020-07-31 [1]
##  gap                 1.2.3-1    2021-04-21 [1]
##  generics            0.1.0      2020-10-31 [1]
##  ggplot2           * 3.3.5      2021-06-25 [1]
##  glmmTMB           * 1.1.3      2022-03-13 [1]
##  glue                1.4.2      2020-08-27 [1]
##  gridExtra         * 2.3        2017-09-09 [1]
##  gtable              0.3.0      2019-03-25 [1]
##  haven               2.3.1      2020-06-01 [1]
##  hierfstat         * 0.5-7      2020-07-20 [1]
##  highr               0.8        2019-03-20 [1]
##  hms                 1.0.0      2021-01-13 [1]
##  htmltools           0.5.1.1    2021-01-22 [1]
##  httpuv              1.5.5      2021-01-13 [1]
##  httr                1.4.2      2020-07-20 [1]
##  igraph              1.2.6      2020-10-06 [1]
##  inbreedR          * 0.3.2      2016-09-09 [1]
##  insight             0.16.0     2022-02-17 [1]
##  iterators           1.0.13     2020-10-15 [1]
##  jsonlite            1.7.2      2020-12-09 [1]
##  knitr               1.31       2021-01-27 [1]
##  label.switching     1.8        2019-07-01 [1]
##  labeling            0.4.2      2020-10-20 [1]
##  later               1.1.0.1    2020-06-05 [1]
##  lattice             0.20-41    2020-04-02 [1]
##  lifecycle           1.0.0      2021-02-15 [1]
##  lme4              * 1.1-26     2020-12-01 [1]
##  lmerTest          * 3.1-3      2020-10-23 [1]
##  lpSolve             5.6.15     2020-01-24 [1]
##  lubridate           1.7.10     2021-02-26 [1]
##  magrittr            2.0.1      2020-11-17 [1]
##  MASS                7.3-53.1   2021-02-12 [1]
##  Matrix            * 1.3-2      2021-01-06 [1]
##  memoise             2.0.0      2021-01-26 [1]
##  mgcv                1.8-34     2021-02-16 [1]
##  mime                0.11       2021-06-23 [1]
##  minqa               1.2.4      2014-10-09 [1]
##  modelr              0.1.8      2020-05-19 [1]
##  multcomp            1.4-19     2022-04-26 [1]
##  MuMIn             * 1.46.0     2022-02-24 [1]
##  munsell             0.5.0      2018-06-12 [1]
##  mvtnorm             1.1-1      2020-06-09 [1]
##  nlme                3.1-152    2021-02-04 [1]
##  nloptr              1.2.2.2    2020-07-02 [1]
##  numDeriv            2016.8-1.1 2019-06-06 [1]
##  ParallelStructure * 1.0        2018-05-11 [1]
##  pegas             * 1.0-1      2021-05-17 [1]
##  performance       * 0.8.0      2021-10-01 [1]
##  permute             0.9-5      2019-03-12 [1]
##  pillar              1.6.1      2021-05-16 [1]
##  pkgbuild            1.2.0      2020-12-15 [1]
##  pkgconfig           2.0.3      2019-09-22 [1]
##  pkgload             1.2.0      2021-02-23 [1]
##  plot.matrix       * 1.6        2021-04-26 [1]
##  plyr                1.8.6      2020-03-03 [1]
##  pophelper         * 2.3.1      2022-01-24 [1]
##  prettyunits         1.1.1      2020-01-24 [1]
##   processx            3.4.5      2020-11-30 [1]
##   promises            1.2.0.1    2021-02-11 [1]
##   ps                  1.6.0      2021-02-28 [1]
##   purrr             * 0.3.4      2020-04-17 [1]
##   qgam                1.3.4      2021-11-22 [1]
##   R6                  2.5.0      2020-10-28 [1]
##   RColorBrewer      * 1.1-2      2014-12-07 [1]
##   Rcpp                1.0.7      2021-07-07 [1]
##   readr             * 1.4.0      2020-10-05 [1]
##   readxl            * 1.3.1      2019-03-13 [1]
##   remotes             2.2.0      2020-07-21 [1]
##   reprex              1.0.0      2021-01-27 [1]
##   reshape2            1.4.4      2020-04-09 [1]
##   rlang               0.4.11     2021-04-30 [1]
##   rmarkdown           2.7        2021-02-19 [1]
##   rprojroot           2.0.2      2020-11-15 [1]
##   rstudioapi          0.13       2020-11-12 [1]
##   Rttf2pt1            1.3.9      2021-07-22 [1]
##   rvest               0.3.6      2020-07-25 [1]
##   sandwich            3.0-1      2021-05-18 [1]
##   scales              1.1.1      2020-05-11 [1]
##   seqinr              4.2-8      2021-06-09 [1]
##   sessioninfo         1.1.1      2018-11-05 [1]
##   shiny               1.6.0      2021-01-25 [1]
##   statmod             1.4.35     2020-10-19 [1]
##   stringi             1.7.3      2021-07-16 [1]
##   stringr           * 1.4.0      2019-02-10 [1]
##   survival            3.2-7      2020-09-28 [1]
##   testthat            3.0.2      2021-02-14 [1]
##   TH.data             1.1-1      2022-04-26 [1]
##   tibble            * 3.1.2      2021-05-16 [1]
##   tidyr             * 1.1.3      2021-03-03 [1]
##   tidyselect          1.1.1      2021-04-30 [1]
##   tidyverse         * 1.3.0      2019-11-21 [1]
##   TMB                 1.7.19     2021-02-05 [1]
##   usethis           * 2.0.1      2021-02-10 [1]
##   utf8                1.2.1      2021-03-12 [1]
##   vctrs               0.3.8      2021-04-29 [1]
##   vegan               2.5-7      2020-11-28 [1]
##   withr               2.4.2      2021-04-18 [1]
##   xfun                0.21       2021-02-10 [1]
##   xml2                1.3.2      2020-04-23 [1]
##   xtable              1.8-4      2019-04-21 [1]
##   yaml                2.2.1      2020-02-01 [1]
##   zoo                 1.8-8      2020-05-02 [1]
##   source
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##   CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.5)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.1)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.5)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.5)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  R-Forge (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  Github (royfrancis/pophelper@3807852)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##  CRAN (R 4.0.2)
##
## [1] /Library/Frameworks/R.framework/Versions/4.0/Resources/library
```