

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

compiled by Rebecca Shuhua Chen

04/10/2022

Contents

Libraries	2
Data	2
Test for Hardy-Weinberg equilibrium	2
Analysing population structure	4
Structure analysis	8
Calculating and modelling sMLH	12
Migration models	19
Other plots included in the manuscript	23

This document contains all R-code used in the workflow for the manuscript “*Effects of hunting on genetic diversity, inbreeding and dispersal in Finnish black grouse (*Lyrurus tetrix*)*” by Rebecca Shuhua Chen, Carl Soulsbury, Christophe Lebigre, Kees van Oers and Joseph Hoffman (in review). The raw data can be found 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 in R, and other softwares were used in combination with our workflow in R to collect all results presented in the manuscript.

Libraries

The following packages were used in the analyses:

```
pacman::p_load(tidyverse, adegenet, pegas, data.table, hierfstat,
               plot.matrix, lme4, forcats, ape, ParallelStructure, pophelper,
               inbreedR, lmerTest, DHARMA, performance, MuMIn, readxl, glmmTMB,
               RColorBrewer, extrafont, devtools, gridExtra)
```

Data

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  un hunted: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
all.raw <- read.structure("data/rawdata/Microsat.adults.plus.unrelated.chicks.forstructure.stru",
                          n.ind = 2078, 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(all.raw)
```

```
##
## // Number of individuals: 2078
## // Group sizes: 274 363 46 211 320 59 62 40 91 184 350 78
## // Number of alleles per locus: 8 11 14 13 9 23 11 9 23 9 10 11 21 5
## // Number of alleles per group: 132 131 99 136 130 96 101 103 102 125 142 119
## // Percentage of missing data: 9.4 %
## // Observed heterozygosity: 0.44 0.81 0.83 0.73 0.78 0.8 0.81 0.17 0.86 0.71 0.81 0.78 0.71 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
```

Test for Hardy-Weinberg equilibrium

We tested for Hardy-Weinberg equilibrium in the combined dataset consisting of adults and unrelated chicks (one chick per brood).

```
#### Calculate Hardy-Weinberg equilibrium ####
```

```
# First all together
```

```
allHWE.all <- pegas::hw.test(all.raw, B = 1000)
```

```
#B = 1000 for 1000 Monte Carlo permutations
```

```
summary(allHWE.all )
```

```
##      chi^2      df      Pr(chi^2 >)      Pr.exact
## Min.   : 7.96   Min.   : 10.00   Min.   :0.00000   Min.   :0.0000
## 1st Qu.: 65.45   1st Qu.: 36.00   1st Qu.:0.00000   1st Qu.:0.0115
## Median : 121.30   Median : 55.00   Median :0.08816   Median :0.0610
## Mean   : 631.64   Mean    : 88.64   Mean    :0.33660   Mean    :0.1764
## 3rd Qu.: 490.24   3rd Qu.: 87.75   3rd Qu.:0.69941   3rd Qu.:0.1220
## Max.   :2771.65   Max.    :253.00   Max.    :0.99999   Max.    :0.8460
```

```
# Then per population using a for loop
```

```
allpop <- seppop(all.raw)
```

```
# Run loop
```

```
allHWE = NULL
```

```
for(i in 1:length(allpop)) {
```

```
  hwt <- pegas::hw.test(allpop[[i]], B=1000)
```

```
  smry <- summary(allpop[[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")
```

```
  allHWE <- as.data.frame(cbind(allHWE, Hobs, Hexp, pexact, qval.FDR, qval.bon))
```

```
}
```

```
#then we transform the resulting allpop dataframe into something readable (hidden)
```

```
#### Results Hardy-Weinberg ####
```

```
head(allHWE.t) #HWE per locus per site
```

```
##      Site L01 L02 L03 L04 L05 L06 L07 L08 L09 L10 L11 L12 L13
## Hobs      1 0.34 0.83 0.82 0.73 0.79 0.84 0.84 0.21 0.83 0.67 0.81 0.75 0.78
## Hexp      1 0.80 0.82 0.81 0.76 0.79 0.81 0.81 0.21 0.89 0.70 0.81 0.77 0.83
## pexact     1 0.00 0.08 0.62 0.61 0.97 0.17 0.54 0.46 0.45 0.12 0.41 0.42 0.01
## qval.FDR    1 0.00 0.37 0.67 0.67 0.97 0.47 0.67 0.67 0.67 0.41 0.67 0.67 0.04
## qval.bon    1 0.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.08
## Hobs.1     2 0.36 0.82 0.84 0.70 0.74 0.79 0.80 0.27 0.88 0.69 0.81 0.74 0.71
##      L14
## Hobs    0.19
## Hexp    0.19
## pexact  0.59
## qval.FDR 0.67
## qval.bon 1.00
## Hobs.1  0.28
```

This table includes observed heterozygosity (Hobs), expected heterozygosity (Hexp), pexact, qval.FDR and qval.bon for each locus in (columns) for each site (1-12). 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 (locus 1 is not included in the manuscript as it resulted in PCA plots for females with very unexpected patterns (erogenous clustering)), most likely due to some genotyping and/or scoring error.

Analysing population structure

To investigate patterns of genetic differentiation, we calculated pairwise F_{ST} values and conducted a STRUC-
TURE analysis. 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.

Moreover, we compare observed and expected heterozygosity as well as allelic richness between hunted and
unhunted sites, through modelling the effect of hunting on these three genetic diversity measures.

```
#using filtered structure files excluding loci out of HWE
all <- read.structure("data/cleandata/Microsat.adults.plus.unrelated.chicks.noLOCUS1+13.forstructure.st.
                      n.ind = 2078, 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 = FALSE)
```

```
##
## Converting data from a STRUCTURE .stru file to a genind object...
```

```
#### Summary statistics ####

# This is based on adults and unrelated chicks across 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           3           4           5           6           7           8
## L01 1.824264 1.814686 1.795509 1.769247 1.793099 1.808096 1.789011 1.775000
## L02 1.812080 1.821476 1.803870 1.832592 1.821816 1.840070 1.834645 1.841139
## L03 1.759738 1.721630 1.723841 1.728597 1.762764 1.767348 1.778652 1.700000
## L04 1.791671 1.793210 1.787625 1.803537 1.799412 1.774880 1.800813 1.807278
## L05 1.810506 1.800365 1.778548 1.806768 1.790072 1.814429 1.850643 1.809177
## L06 1.811080 1.808226 1.833731 1.794407 1.820189 1.839345 1.840808 1.795538
##           9           10          11          12
## L01 1.806083 1.773232 1.801402 1.788668
## L02 1.832251 1.844913 1.826493 1.816873
## L03 1.748892 1.737102 1.739391 1.773284
## L04 1.811973 1.799742 1.786466 1.781969
```

```
## L05 1.763402 1.802837 1.773211 1.784864
## L06 1.846761 1.830337 1.807819 1.837800
```

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] 0.6923074 0.6972209 0.6702899 0.6904110 0.6842245 0.7007807 0.7653959
## [8] 0.6875874 0.6822344 0.6964826 0.6758077 0.6983662
```

```
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
```

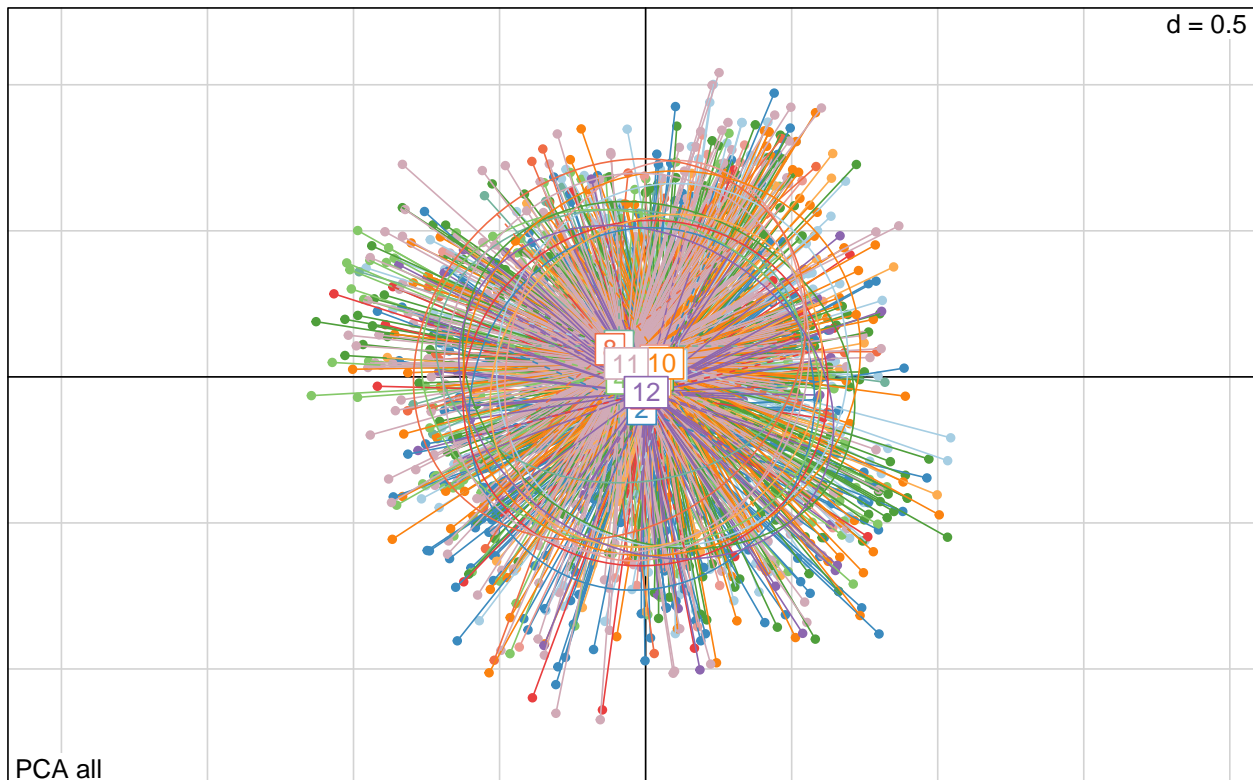
```
## [1] 0.6979300 0.7025220 0.6666470 0.6883917 0.6843798 0.6878527 0.7656413
## [8] 0.6808575 0.6722769 0.6940767 0.6779083 0.6976739
```

PCA

```
basic.stats(all)
```

```
## $perloc
##      Ho      Hs      Ht      Dst      Htp      Dstp      Fst      Fstp      Fis      Dest
## L01 0.8094 0.7948 0.8000 0.0052 0.8005 0.0057 0.0065 0.0071 -0.0184 0.0278
## L02 0.8432 0.8273 0.8321 0.0048 0.8325 0.0053 0.0058 0.0063 -0.0193 0.0305
## L03 0.7240 0.7452 0.7500 0.0047 0.7504 0.0052 0.0063 0.0069 0.0285 0.0203
## L04 0.7796 0.7950 0.8037 0.0087 0.8045 0.0095 0.0108 0.0118 0.0193 0.0463
## L05 0.8069 0.7987 0.8039 0.0052 0.8044 0.0057 0.0065 0.0071 -0.0103 0.0282
## L06 0.8060 0.8222 0.8259 0.0037 0.8263 0.0040 0.0045 0.0049 0.0197 0.0228
## L07 0.1461 0.1575 0.1508 -0.0067 0.1502 -0.0073 -0.0441 -0.0488 0.0721 -0.0087
## L08 0.8773 0.8669 0.8711 0.0042 0.8715 0.0046 0.0048 0.0052 -0.0119 0.0343
## L09 0.7338 0.7206 0.7256 0.0050 0.7261 0.0055 0.0069 0.0076 -0.0183 0.0197
## L10 0.7940 0.7926 0.7966 0.0040 0.7970 0.0044 0.0051 0.0055 -0.0018 0.0213
## L11 0.7853 0.7931 0.7951 0.0020 0.7953 0.0022 0.0025 0.0027 0.0098 0.0105
## L12 0.1839 0.1856 0.1859 0.0003 0.1860 0.0004 0.0019 0.0020 0.0090 0.0005
##
## $overall
##      Ho      Hs      Ht      Dst      Htp      Dstp      Fst      Fstp      Fis      Dest
## 0.6908 0.6916 0.6951 0.0034 0.6954 0.0038 0.0050 0.0054 0.0012 0.0122
```

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



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

```
#convert to hfst object
```

```
all.hfst <- genind2hierfstat(all)
```

```
#calculate statsf
```

```
basicstat.all <- basic.stats(all, diploid = TRUE, digits = 2)
```

```
# per locus
```

```
fst.all.perlocus <- basicstat.all$perloc$Fst
```

```
fst.all.perlocus <- data.frame(Locus = seq(from = 1, to = 12), Fst = fst.all.perlocus)
```

```
# Pairwise Fst
```

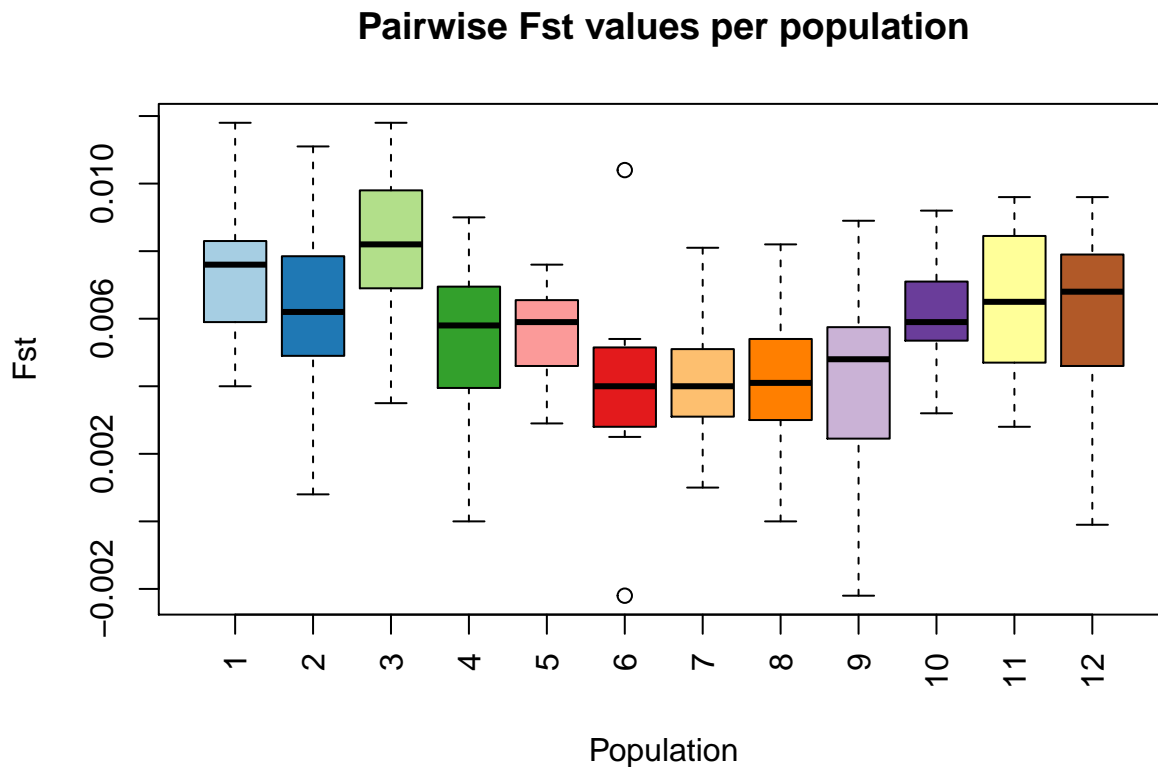
```
fst.all <- pairwise.neifst(all.hfst)
```

```
head(fst.all)
```

```
##      1      2      3      4      5      6      7      8      9     10     11
## 1    NA 0.0081 0.0118 0.0073 0.0059 0.0040 0.0081 0.0056 0.0059 0.0085 0.0076
## 2 0.0081    NA 0.0111 0.0066 0.0076 0.0054 0.0046 0.0052 0.0008 0.0062 0.0088
## 3 0.0118 0.0111    NA 0.0090 0.0064 0.0104 0.0035 0.0082 0.0074 0.0092 0.0081
## 4 0.0073 0.0066 0.0090    NA 0.0067 0.0052 0.0027 0.0000 0.0048 0.0058 0.0031
## 5 0.0059 0.0076 0.0064 0.0067    NA 0.0029 0.0040 0.0047 0.0045 0.0059 0.0054
## 6 0.0040 0.0054 0.0104 0.0052 0.0029    NA 0.0025 0.0027 -0.0022 0.0051 0.0040
##      12
## 1 0.0089
## 2 0.0041
## 3 0.0055
## 4 0.0072
## 5 0.0068
```

```
## 6 0.0051
```

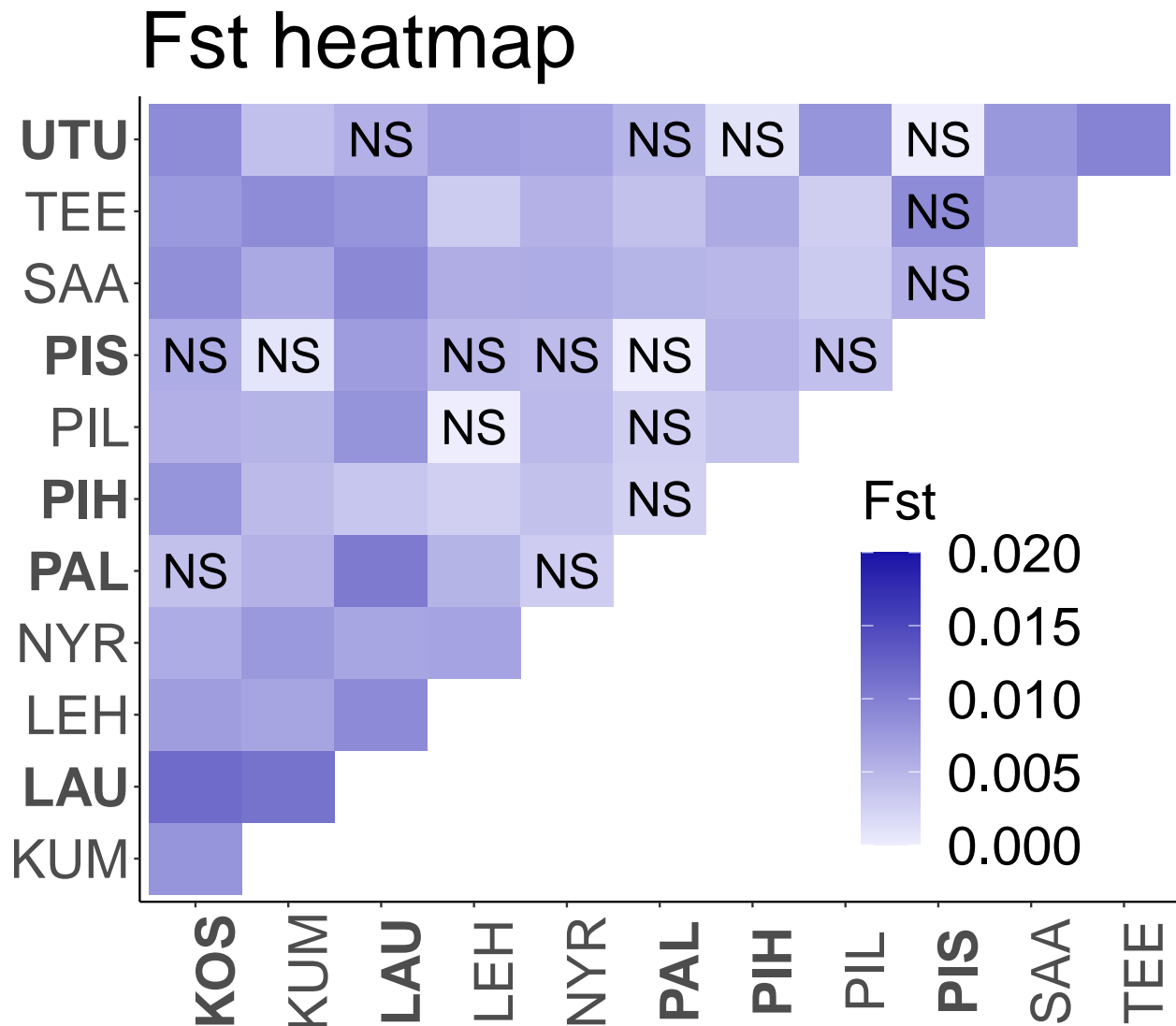
```
# Fst per population
boxplot(fst.all, col=funky(nPop(all)), las=3,
        xlab="Population", ylab="Fst", main = "Pairwise Fst values per population")
```



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

```
#purples from structure
ggplot(pairwise.fst, aes(abb.x, abb.y, fill = Fst)) + geom_tile() + theme_classic() +
  scale_fill_gradientn(colors = mypalette3, limits = c(0,0.02)) +
  geom_text(aes(label = Sig), size = 8)+
  theme(text = element_text(size = 22),
        axis.text.x = element_text(angle = 90, size = 26,
                                     face = c("bold", "plain", "bold", "plain", "plain",
                                               "bold", "bold", "plain", "bold",
                                               "plain", "plain")),
        axis.text.y = element_text(size = 26,
                                     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('Fst heatmap')
```



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 #####
```

```
all <- fread("data/cleandata/Microsat.adults.plus.unrelated.
             chicks.noLOCUS1+13.forstructure.stru")
```

```
infile <- "/data/home/rchen/Hunting/blackgrouse-hunting/data/"
```



```

cleandata/Microsat.adults.plus.unrelated.chicks.noLOCUS1+13.forstructure.stru"

outpath <- "/data/home/rchen/Hunting/blackgrouse-hunting/analyses/
structure/results/"

# 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 = "analyses/structure/structure_jobs.txt")

# file path to structure
STR_path='/usr/local/bin/'

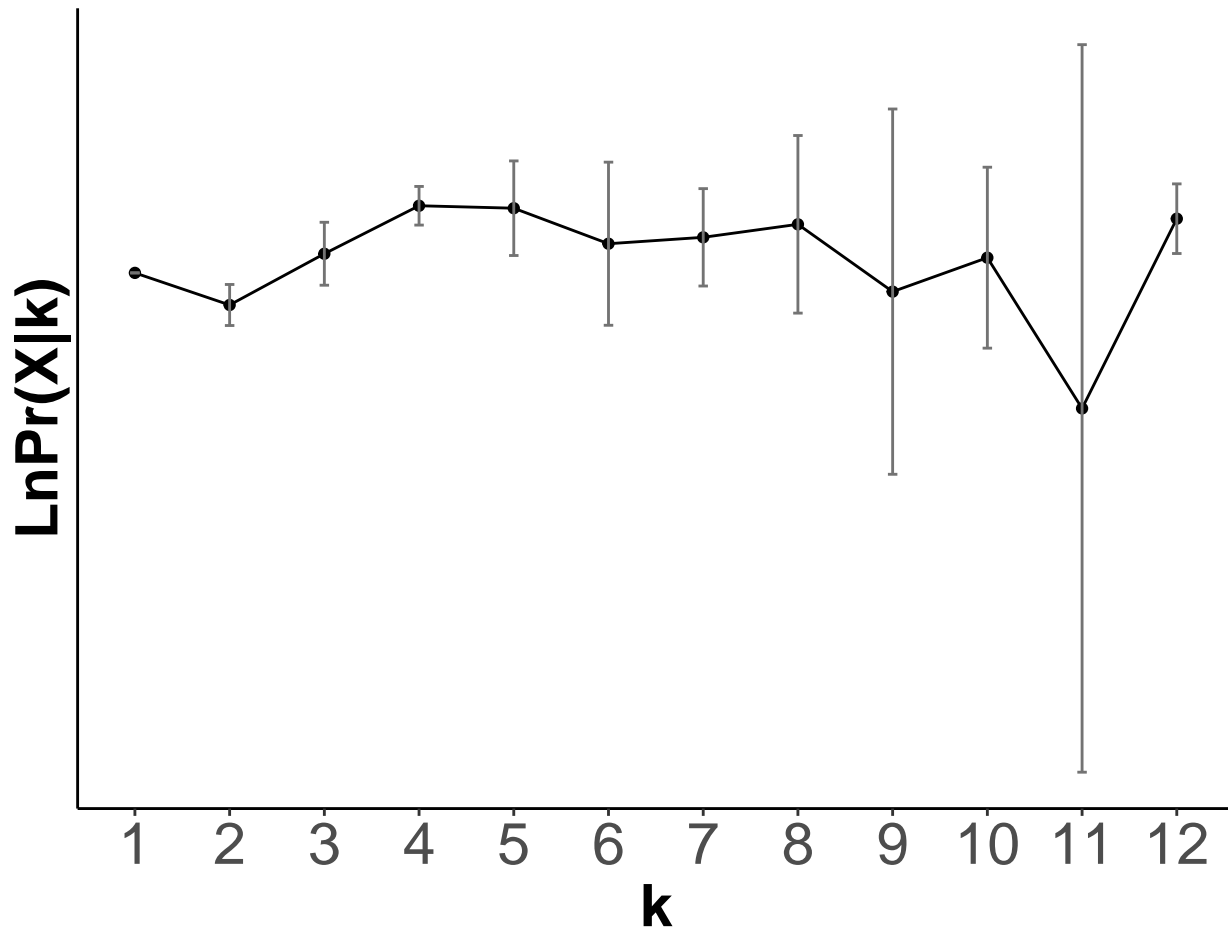
# Run Parallel Structure

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

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

```

These STRUCTURE analyses gave us the following results:



Effect of hunting on genetic diversity

To understand the effect of hunting on genetic diversity, we first conduct a t-test comparing observed heterozygosity, expected heterozygosity and allelic richness between hunted and unhunted populations. Next, we construct a lm to include random effects of locus and population on the genetic diversity measures.

```
## First: t-test to compare differences in genetic diversity between
## hunted vs unhunted sites (where we have the mean value of all loci per site)
```

```
head(measures)
```

```
##      pop      Hobs      Hexp      Ar      hunt
## 1     1 0.6923074 0.6979300 1.699265  hunted
## 2     2 0.6972209 0.7025220 1.703523 unhunted
## 3     3 0.6702899 0.6666470 1.673973  hunted
## 4     4 0.6904110 0.6883917 1.690081 unhunted
## 5     5 0.6842245 0.6843798 1.685471 unhunted
## 6     6 0.7007807 0.6878527 1.694508  hunted
```

```
t.test(measures$Hobs[which(measures$hunt == "hunted")],
       measures$Hobs[which(measures$hunt == "unhunted")])
```

```
##
## Welch Two Sample t-test
##
```

```
## data: measures$Hobs[which(measures$hunt == "hunted")] and measures$Hobs[which(measures$hunt == "unhunted")]
## t = 0.92695, df = 5.5838, p-value = 0.3923
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.02184505 0.04772519
## sample estimates:
## mean of x mean of y
## 0.7015624 0.6886223
```

```
t.test(measures$Hexp[which(measures$hunt == "hunted")],
       measures$Hexp[which(measures$hunt == "unhunted")])
```

```
##
## Welch Two Sample t-test
##
## data: measures$Hexp[which(measures$hunt == "hunted")] and measures$Hexp[which(measures$hunt == "unhunted")]
## t = 0.66604, df = 5.6518, p-value = 0.5316
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.02724204 0.04720395
## sample estimates:
## mean of x mean of y
## 0.6980036 0.6880227
```

```
t.test(measures$Ar[which(measures$hunt == "hunted")],
       measures$Ar[which(measures$hunt == "unhunted")])
```

```
##
## Welch Two Sample t-test
##
## data: measures$Ar[which(measures$hunt == "hunted")] and measures$Ar[which(measures$hunt == "unhunted")]
## t = 0.74388, df = 5.9353, p-value = 0.4853
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.02019666 0.03777352
## sample estimates:
## mean of x mean of y
## 1.699395 1.690606
```

```
## Then, we model the effect of hunting on the three genetic diversity measures,
## where we take each measure per locus per site and perform a likelihood
## ratio test to assess the difference between the null and alternative model
```

```
head(measures.all)
```

```
##   locus pop      Hobs      Hexp      Ar  hunt
## 1  L01   1 0.8284672 0.8227596 1.824264 hunted
## 2  L02   1 0.8175182 0.8105986 1.812080 hunted
## 3  L03   1 0.7335766 0.7583515 1.759738 hunted
## 4  L04   1 0.7919708 0.7902259 1.791671 hunted
## 5  L05   1 0.8357664 0.8090269 1.810506 hunted
## 6  L06   1 0.8357664 0.8095996 1.811080 hunted
```

```

# excluding BG20 (out of HWE)
Ho_model <- lmerTest::lmer(Hobs ~ hunt + (1|locus) + (1|pop),
                          data = subset(measures.all, locus != "L13"))

Ho_model_null <- lmerTest::lmer(Hobs ~ (1|locus) + (1|pop),
                              data = subset(measures.all, locus != "L13"))

He_model <- lmerTest::lmer(Hexp ~ hunt + (1|locus) + (1|pop),
                          data = subset(measures.all, locus != "L13"))
He_model_null <- lmerTest::lmer(Hexp ~ (1|locus) + (1|pop),
                              data = subset(measures.all, locus != "L13"))

Ar_model <- lmerTest::lmer(Ar ~ hunt + (1|locus) + (1|pop),
                          data = subset(measures.all, locus != "L13"))
Ar_model_null <- lmerTest::lmer(Ar ~ (1|locus) + (1|pop),
                              data = subset(measures.all, locus != "L13"))

#LRT
anova(Ho_model, Ho_model_null) # no significant difference

## Data: subset(measures.all, locus != "L13")
## Models:
## Ho_model_null: Hobs ~ (1 | locus) + (1 | pop)
## Ho_model: Hobs ~ hunt + (1 | locus) + (1 | pop)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## Ho_model_null    4 -392.20 -380.35 200.10  -400.20
## Ho_model         5 -390.51 -375.70 200.26  -400.51 0.314  1    0.5753

anova(He_model, He_model_null) # no significant difference

## Data: subset(measures.all, locus != "L13")
## Models:
## He_model_null: Hexp ~ (1 | locus) + (1 | pop)
## He_model: Hexp ~ hunt + (1 | locus) + (1 | pop)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## He_model_null    4 -263.00 -251.12 135.50  -271.00
## He_model         5 -261.48 -246.63 135.74  -271.48 0.478  1    0.4893

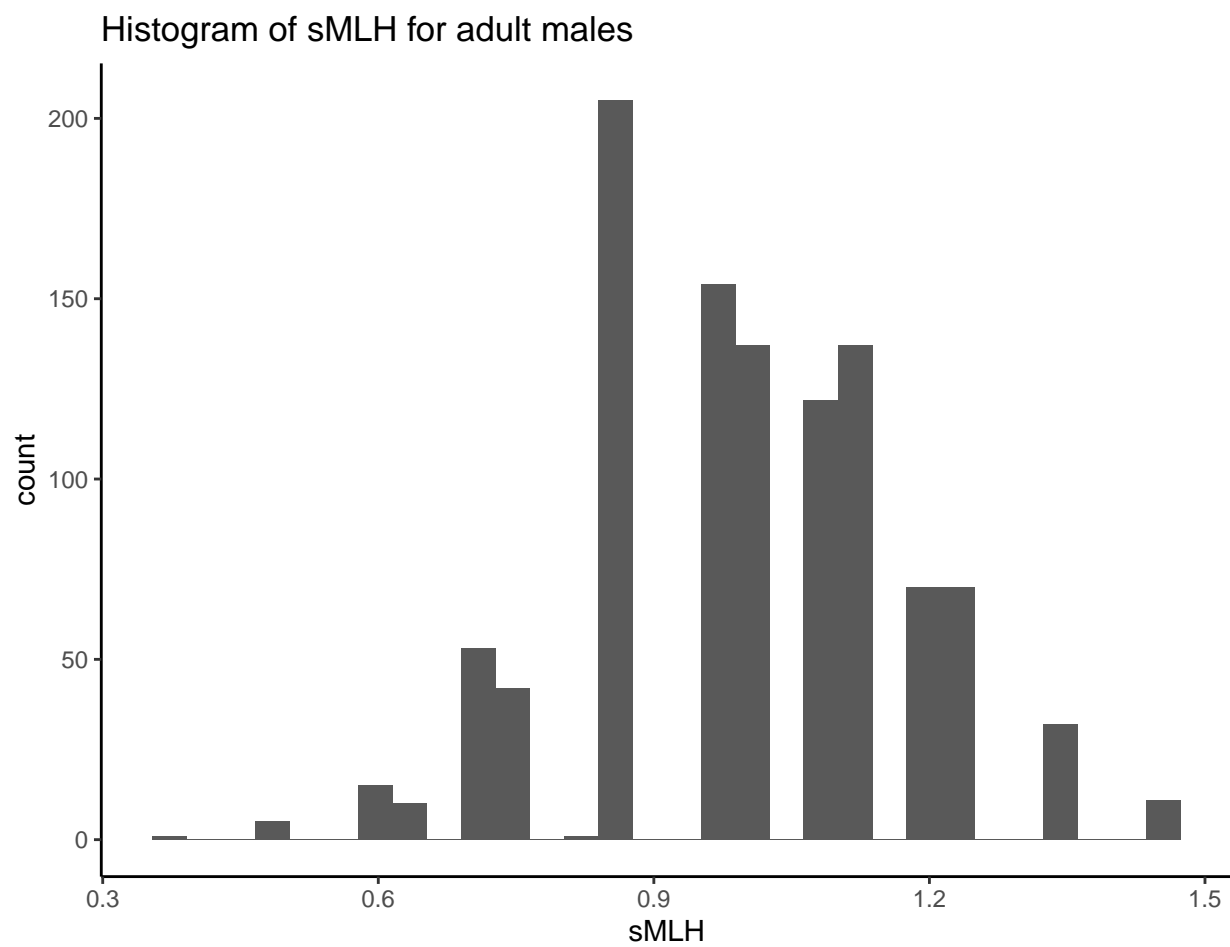
anova(Ar_model, Ar_model_null) # no significant difference

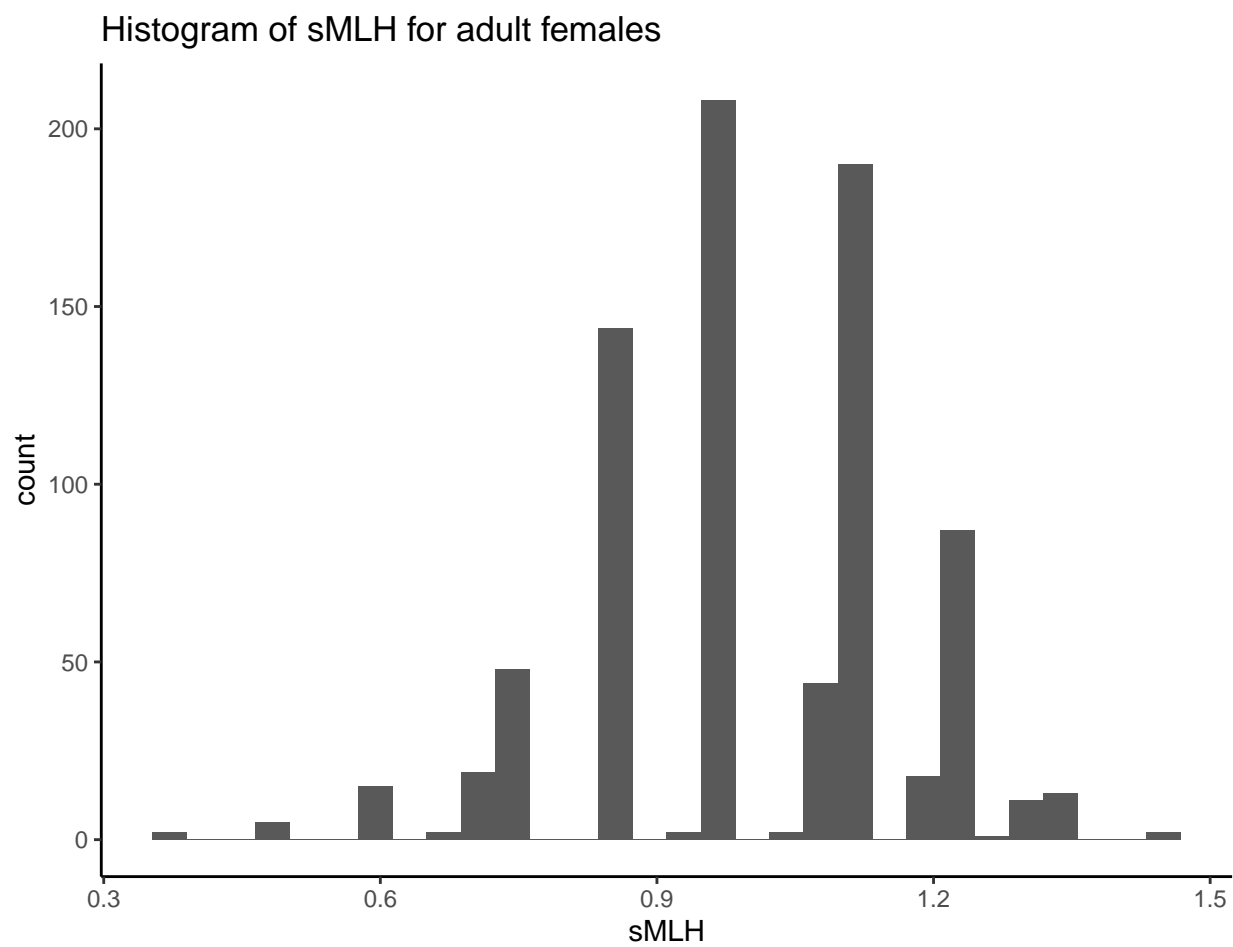
## Data: subset(measures.all, locus != "L13")
## Models:
## Ar_model_null: Ar ~ (1 | locus) + (1 | pop)
## Ar_model: Ar ~ hunt + (1 | locus) + (1 | pop)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## Ar_model_null    4 -489.23 -477.38 248.61  -497.23
## Ar_model         5 -487.23 -472.42 248.62  -497.23 0.006  1    0.9384

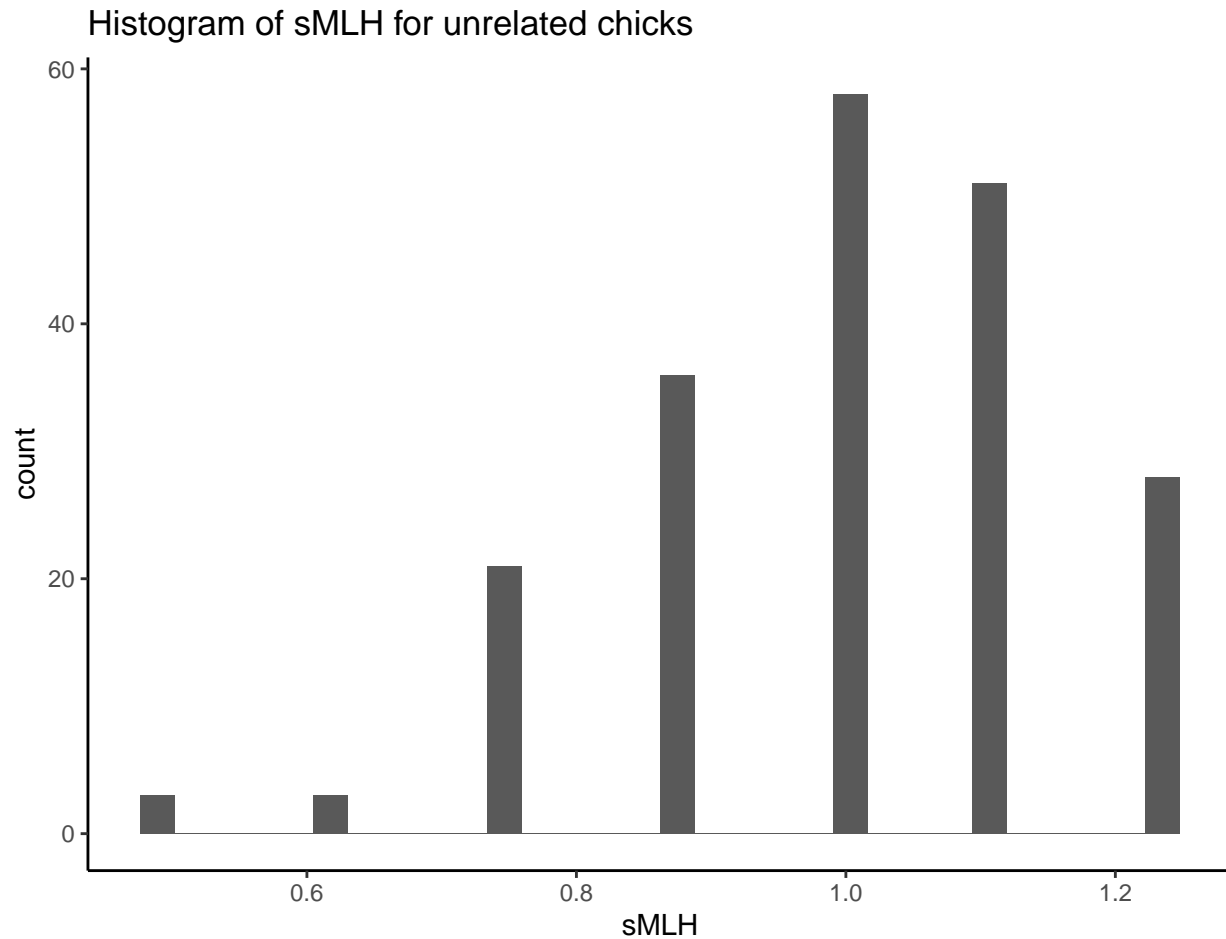
```

Calculating and modelling sMLH

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







```
### Setup models ###
sMLH.all.no.na <- subset(sMLH.all, !is.na(density))
#exclude those without density otherwise lrt won't work

sMLH.model.all.lmer <- lmerTest::lmer(sMLH ~
                                     hunt*density + sex + age+ (1|pop),
                                     data = sMLH.all.no.na)
sMLH.model.all.lmer.noint <- lmerTest::lmer(sMLH ~
                                             hunt + sex + age+ density + (1|pop),
                                             data = sMLH.all.no.na)
sMLH.model.all.lmer.null <- lmerTest::lmer(sMLH ~
                                           density+ sex + age+ (1|pop),
                                           data = sMLH.all.no.na)

anova(sMLH.model.all.lmer.null, sMLH.model.all.lmer)
```

```
## Data: sMLH.all.no.na
## Models:
## sMLH.model.all.lmer.null: sMLH ~ density + sex + age + (1 | pop)
## sMLH.model.all.lmer: sMLH ~ hunt * density + sex + age + (1 | pop)
##               npar      AIC      BIC logLik deviance Chisq Df
## sMLH.model.all.lmer.null    6 -1321.8 -1288.4 666.91 -1333.8
## sMLH.model.all.lmer         8 -1320.0 -1275.4 667.99 -1336.0 2.159 2
##               Pr(>Chisq)
```

```
## sMLH.model.all.lmer.null
## sMLH.model.all.lmer          0.3398
```

```
anova(sMLH.model.all.lmer.null, sMLH.model.all.lmer.noint)
```

```
## Data: sMLH.all.no.na
## Models:
## sMLH.model.all.lmer.null: sMLH ~ density + sex + age + (1 | pop)
## sMLH.model.all.lmer.noint: sMLH ~ hunt + sex + age + density + (1 | pop)
##           npar      AIC      BIC logLik deviance Chisq Df
## sMLH.model.all.lmer.null      6 -1321.8 -1288.4 666.91 -1333.8
## sMLH.model.all.lmer.noint      7 -1321.7 -1282.6 667.83 -1335.7 1.8443 1
##           Pr(>Chisq)
## sMLH.model.all.lmer.null
## sMLH.model.all.lmer.noint      0.1744
```

```
coef(summary(sMLH.model.all.lmer))
```

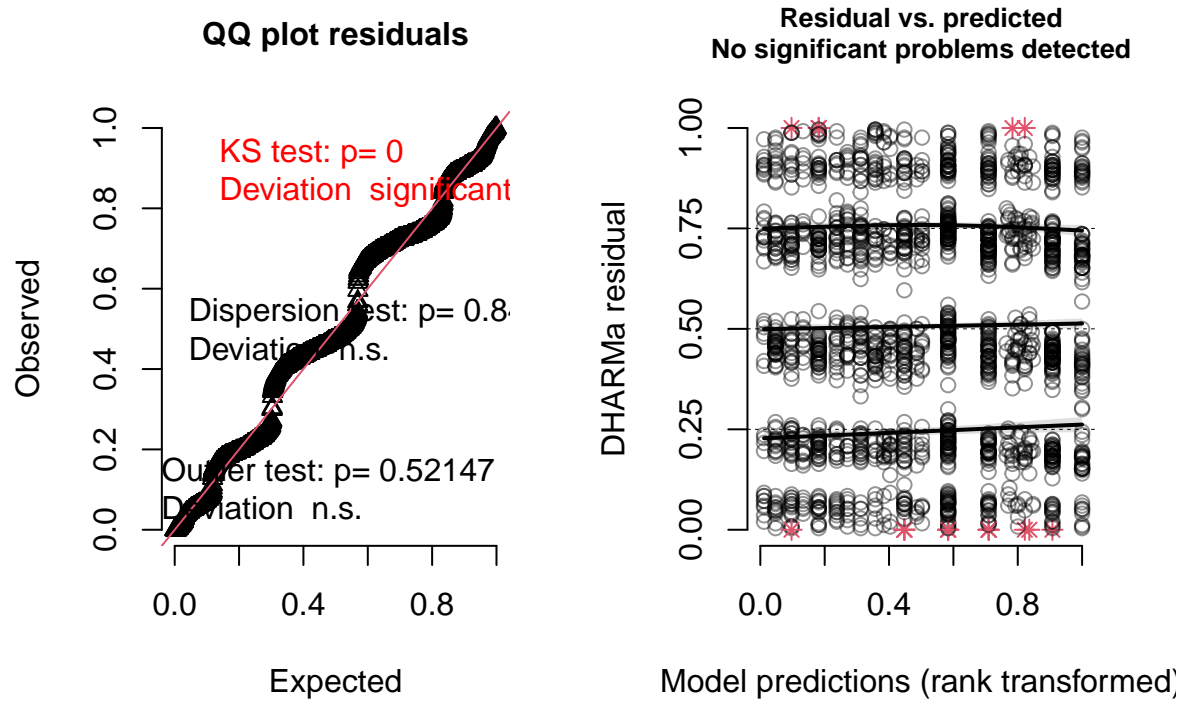
	Estimate	Std. Error	df	t value	Pr(> t)
## (Intercept)	0.980353176	0.015753847	198.1223	62.2294471	5.288112e-132
## hunthunted	-0.005409208	0.031355077	125.7485	-0.1725146	8.633104e-01
## density	0.001509275	0.001381680	967.5943	1.0923474	2.749524e-01
## sexm	-0.001392771	0.007890709	1771.8702	-0.1765077	8.599153e-01
## agechick	0.014933141	0.018091870	1930.0150	0.8254061	4.092431e-01
## hunthunted:density	0.001516963	0.002583694	415.7640	0.5871295	5.574354e-01

```
VarCorr(sMLH.model.all.lmer)
```

## Groups	Name	Std.Dev.
## pop	(Intercept)	0.0039595
## Residual		0.1718672

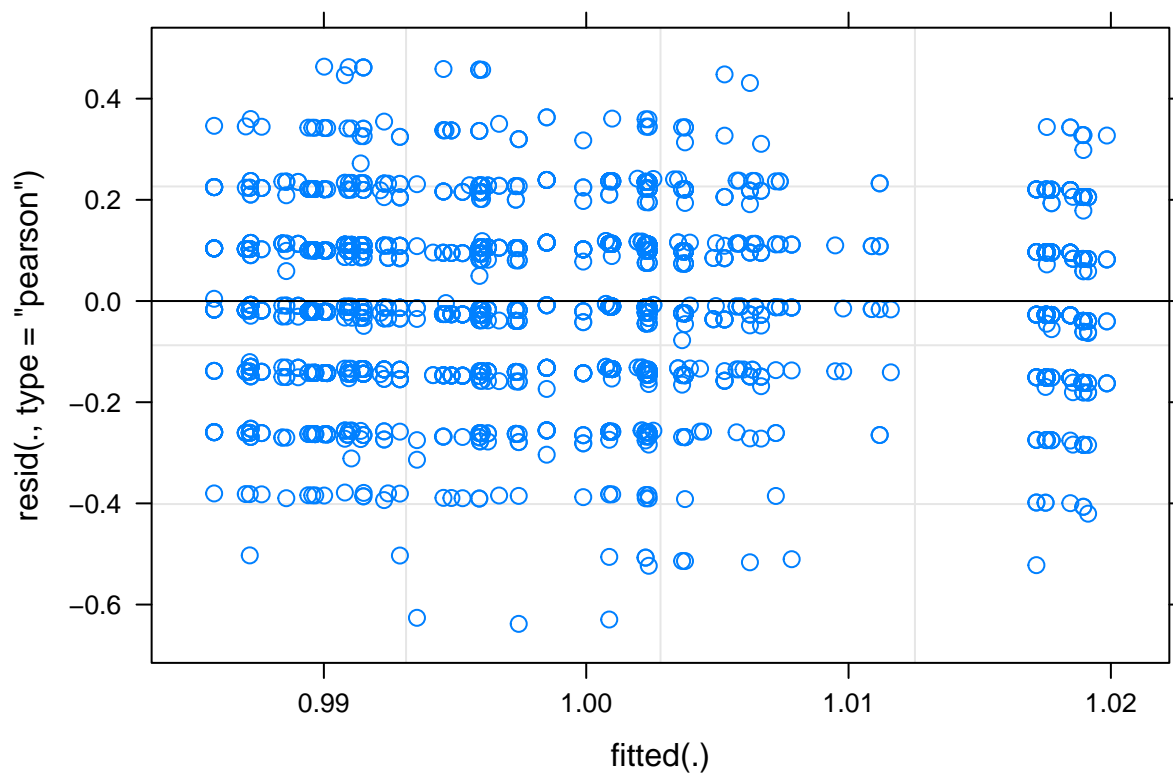
```
simulateResiduals(fittedModel = sMLH.model.all.lmer, plot = T)
```


DHARMa residual



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DHARMa
##
## Scaled residual values: 0.188 0.76 0.756 0.048 0.468 0.076 0.748 0.732 0.884 0.744 0.884 0.436 0.988
```

```
plot(sMLH.model.all.lmer)
```



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

```
##           R2m           R2c
## [1,] 0.003327348 0.003856053
```

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

```
## # ICC by Group
##
## Group |      ICC
## -----
## pop   | 5.305e-04
```

```
compare_performance(sMLH.model.all.lmer.null, sMLH.model.all.lmer, rank = T)
```

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

```
##
## Name | Model | R2 (cond.) | R2 (marg.) | ICC | RMSE | Sigma | A
## -----
## sMLH.model.all.lmer | lmerModLmerTest | 0.004 | 0.003 | 5.305e-04 | 0.172 | 0.172 |
## sMLH.model.all.lmer.null | lmerModLmerTest | 0.003 | 0.002 | 5.137e-04 | 0.172 | 0.172 |
```

```
compare_performance(sMLH.model.all.lmer.null, sMLH.model.all.lmer.noint, rank = T)
```

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

## Name	Model	R2 (cond.)	R2 (marg.)	ICC	RMSE	Sigma
## sMLH.model.all.lmer.noint	lmerModLmerTest	0.004	0.003	3.198e-04	0.172	0.172
## sMLH.model.all.lmer.null	lmerModLmerTest	0.003	0.002	5.137e-04	0.172	0.172

Migration models

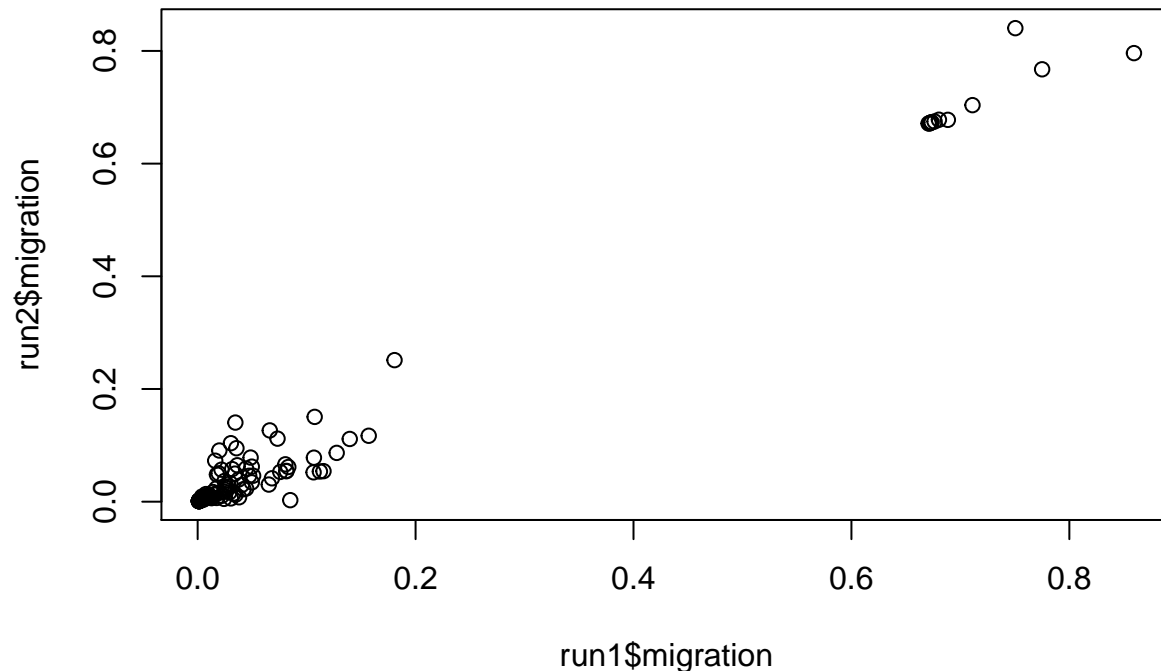
Next, we investigated patterns of dispersion and how these are affected by hunting. We used BA3 to calculate migration directions and rates, followed by building mixed models to estimate the effect of hunting on migration.

```
#### Running BA3 ####

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

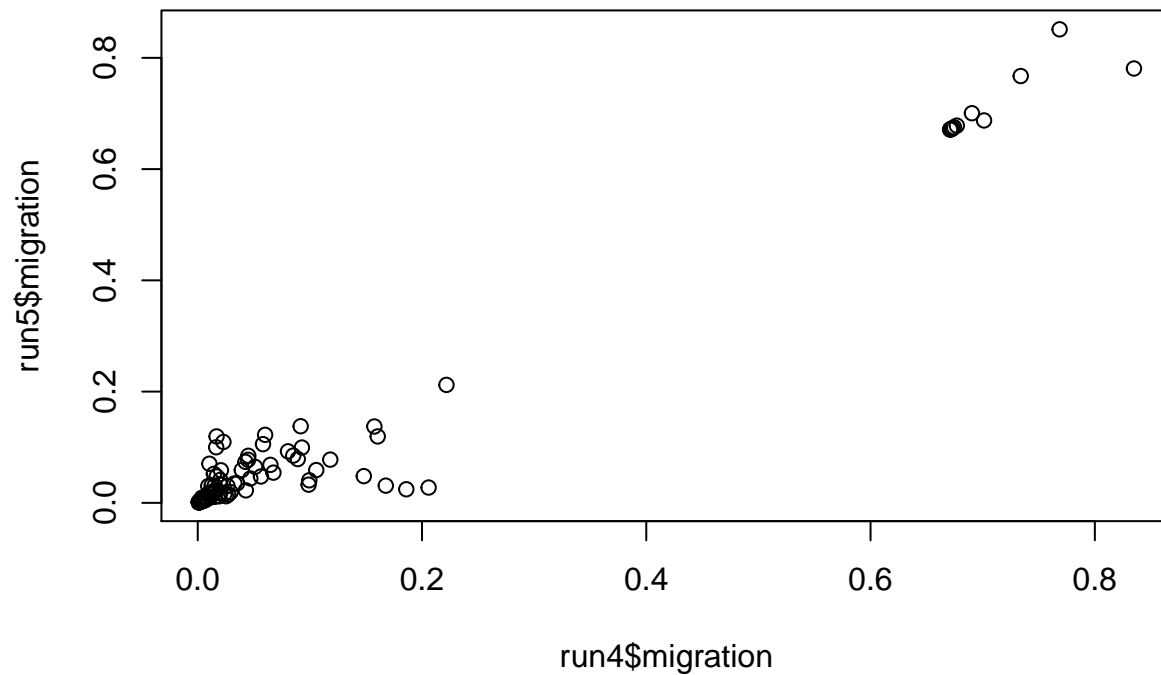
#5 runs with 5 different random seeds
system("~/bin/BA3/BA3MSAT -v -t -g -u -a 0.30 -f 0.40 -s 65323 -i 10000000
      -b 1000000 -n 1000 -o run1.txt
      analyses/migrationanalysis/data_all_ba3.txt") #repeat for another 3 runs

#### Compare runs ####
plot(run1$migration, run2$migration)
```



```
# plot(run1$migration, run3$migration) #hide these in RMarkdown
# plot(run1$migration, run4$migration) plot(run1$migration, run5$migration)
# plot(run2$migration, run3$migration) plot(run2$migration, run4$migration)
# plot(run2$migration, run5$migration) plot(run3$migration, run4$migration)
```

```
# plot(run3$migration, run5$migration)
plot(run4$migration, run5$migration)
```



```
# all runs correspond
```

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

##	m_in	m_out	migration	migration_SE	ESS	migration_ESSc	pop_in
## 1	1	1	0.6856	0.0122	9.2	NA	Koskenp\xe4\xe4
## 2	1	2	0.0097	0.0080	38.6	NA	Koskenp\xe4\xe4
## 3	1	3	0.0013	0.0013	1765.1	0.0013	Koskenp\xe4\xe4
## 4	1	4	0.0169	0.0152	17.5	NA	Koskenp\xe4\xe4
## 5	1	5	0.2073	0.0457	6.1	NA	Koskenp\xe4\xe4
## 6	1	6	0.0020	0.0025	19.3	NA	Koskenp\xe4\xe4

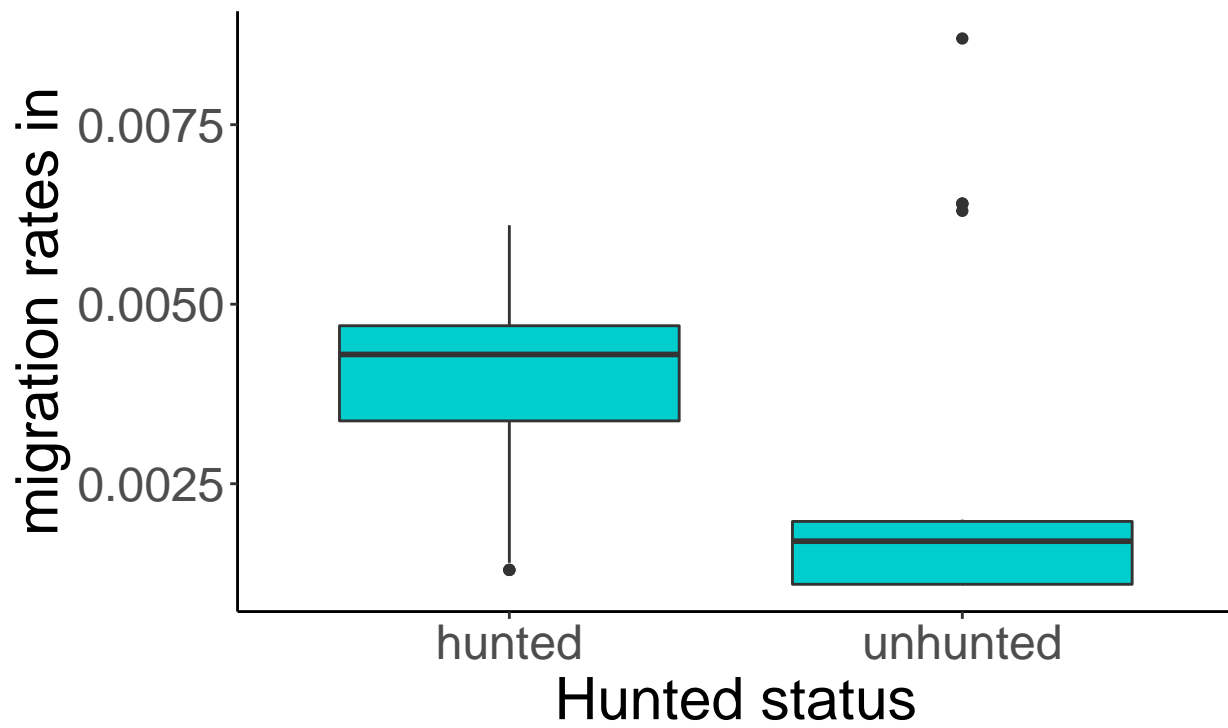
##	hunt_in	pop_out	hunt_out	Distance
## 1	hunted	Koskenp\xe4\xe4	hunted	NA
## 2	hunted	Kummunsuo	unhunted	41.01050
## 3	hunted	Lauttasuo	hunted	54.91027
## 4	hunted	Lehtusuo	unhunted	27.19093
## 5	hunted	Nyr\xf61\xe4	unhunted	34.75899
## 6	hunted	Palosuo	hunted	38.39597

Migration rates IN

Excluding values with ESS < 200

Student t-test comparing hunted/unhunted site filtered migrat

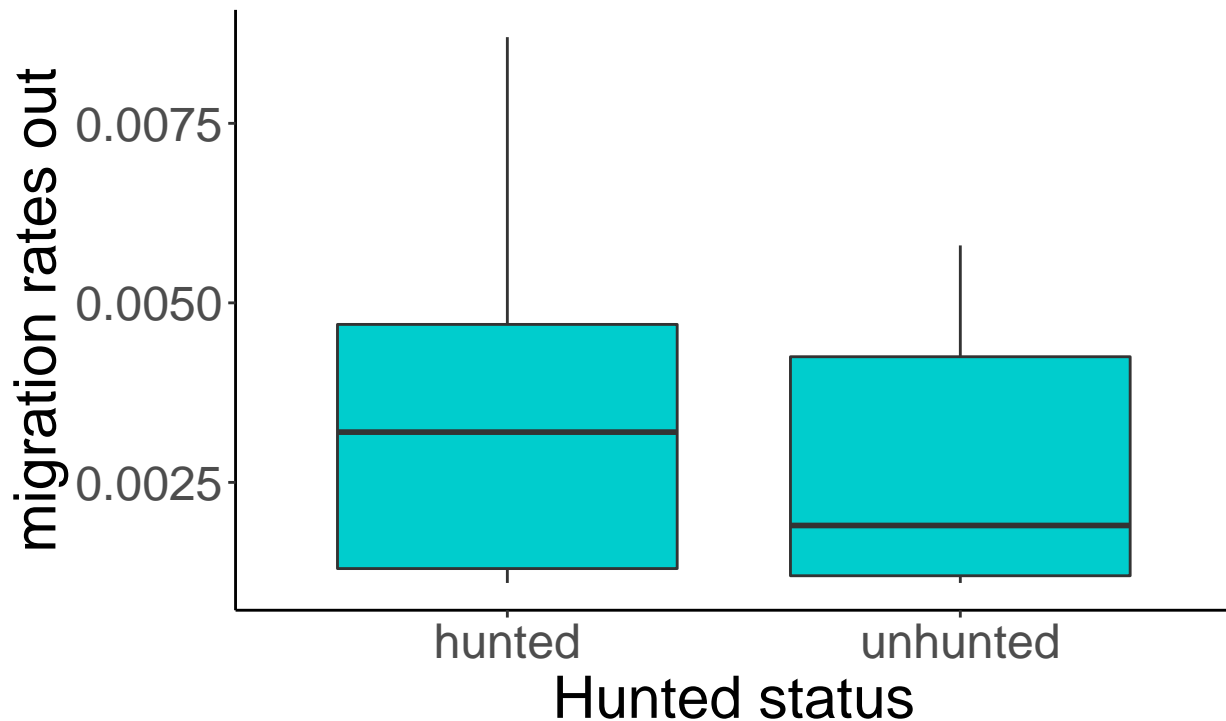
t = 2.85 and p-value = 0.01



Migration rates OUT

Excluding values with ESS < 200

Student t-test comparing hunted/unhunted site corrected mig
 $t = 0.86$ and $p\text{-value} = 0.399$



```
#### Modelling migration ####

#change levels hunted/unhunted
run5_clean$hunt_in <- relevel(as.factor(run5_clean$hunt_in), ref = "unhunted")
run5_clean$hunt_out <- relevel(as.factor(run5_clean$hunt_out), ref = "unhunted")

#immigration

model.in <- glmmTMB(migration_ESSc~ hunt_in + Distance + (1|pop_out) + (1|pop_in),
                    data = run5_clean,
                    family = Gamma(link = "log"))

model.in.null <- glmmTMB(migration_ESSc~ Distance + (1|pop_out) + (1|pop_in),
                        data = run5_clean,
                        family = Gamma(link = "log"))

anova(model.in, model.in.null)

summary(model.in)
simulateResiduals(fittedModel = model.in, plot = T)

#out

model.out <- glmmTMB(migration_ESSc~ hunt_out + Distance + (1|pop_out) + (1|pop_in),
```

```

        data = run5_clean,
        family = Gamma(link = "log"))
model.out.null <- glmmTMB(migration_ESSc~ Distance + (1|pop_out) + (1|pop_in),
        data = run5_clean,
        family = Gamma(link = "log"))

anova(model.out, model.out.null)

summary(model.out)
simulateResiduals(fittedModel = model.out, plot = T)

compare_performance(model.in, model.in.null, rank=T)
compare_performance(model.out, model.out.null, rank=T)

```

Other plots included in the manuscript

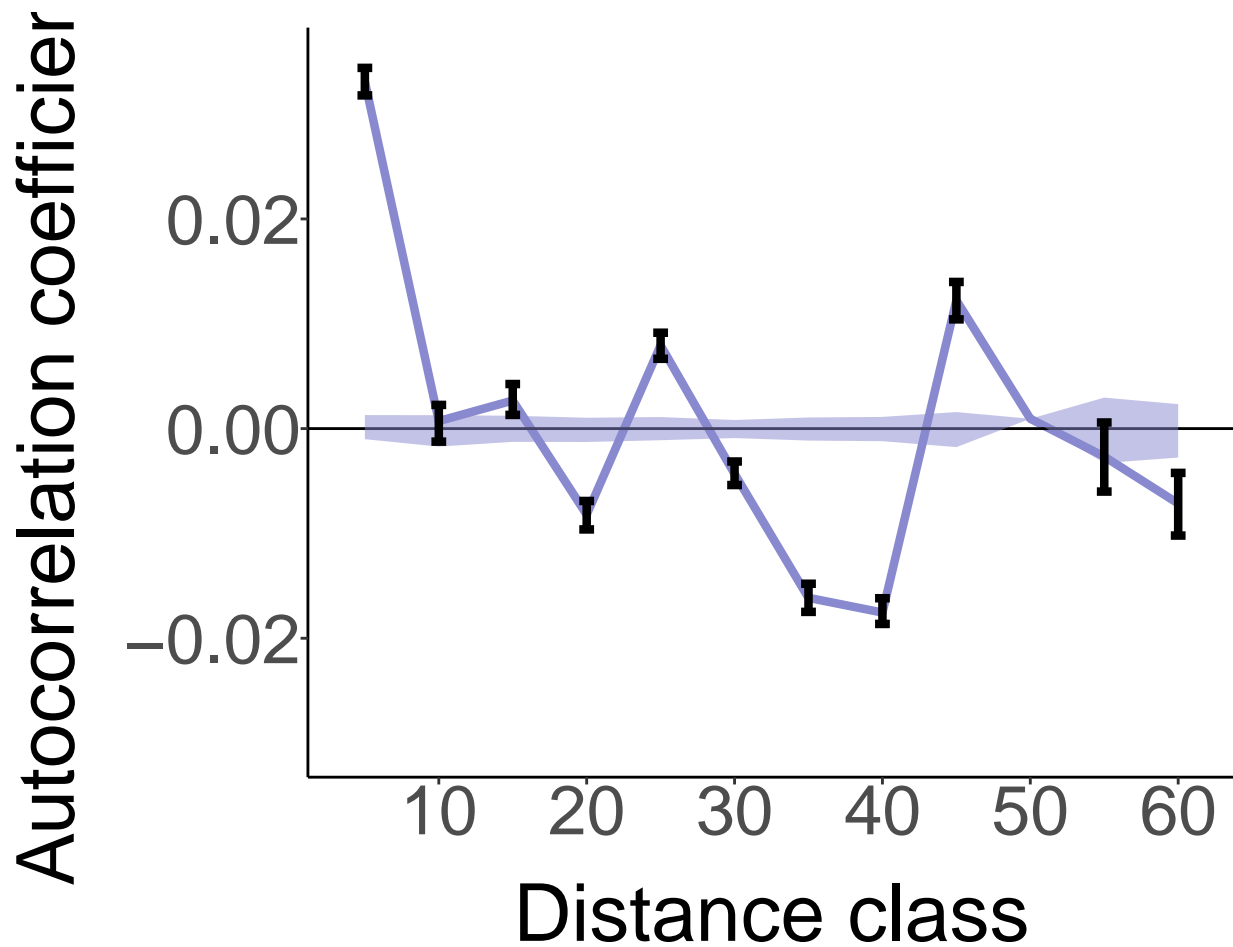
Figure 1 was created with qGIS, for barcharts of STRUCTURE output as in Supplementary Figure 2, please refer to R script 6.CreatingPlots.R

```

### Figure 2 - correlograms
spatial <- read_excel("tables/SpatialAutocor_04.10.22.xlsx", sheet = "ForR")

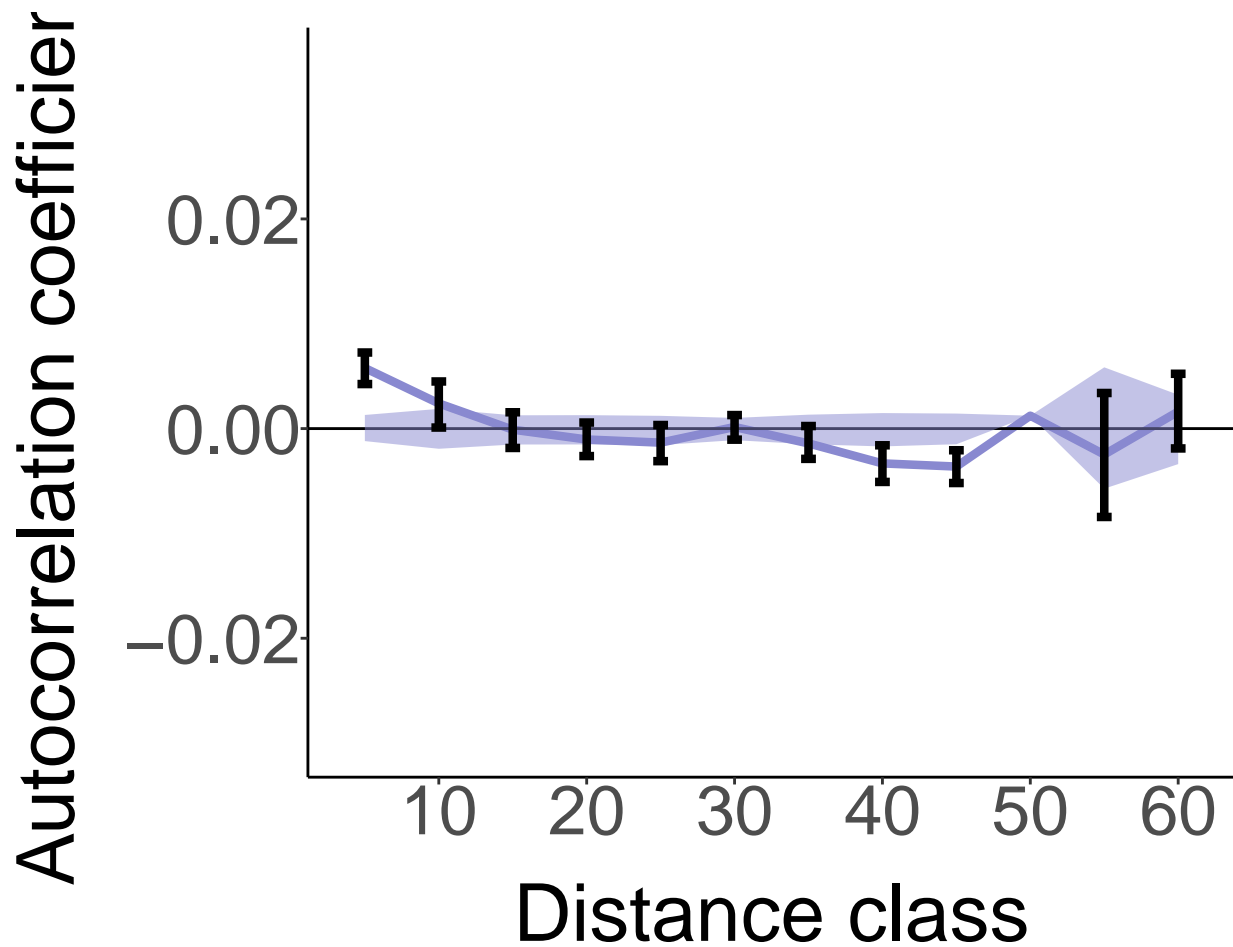
### Males - spatial
spatial %>% filter(Who == "Male") %>% ggplot(aes(x = What)) +
  geom_line(aes(y = r), col = "#8989D0", size = 1.5) + theme_classic() +
  geom_hline(yintercept = 0, col = "black")+
  geom_ribbon(aes(ymax = U, ymin = L), fill = "#8989D0", alpha = 0.5)+
  geom_errorbar(aes(y = r, ymin = 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))

```

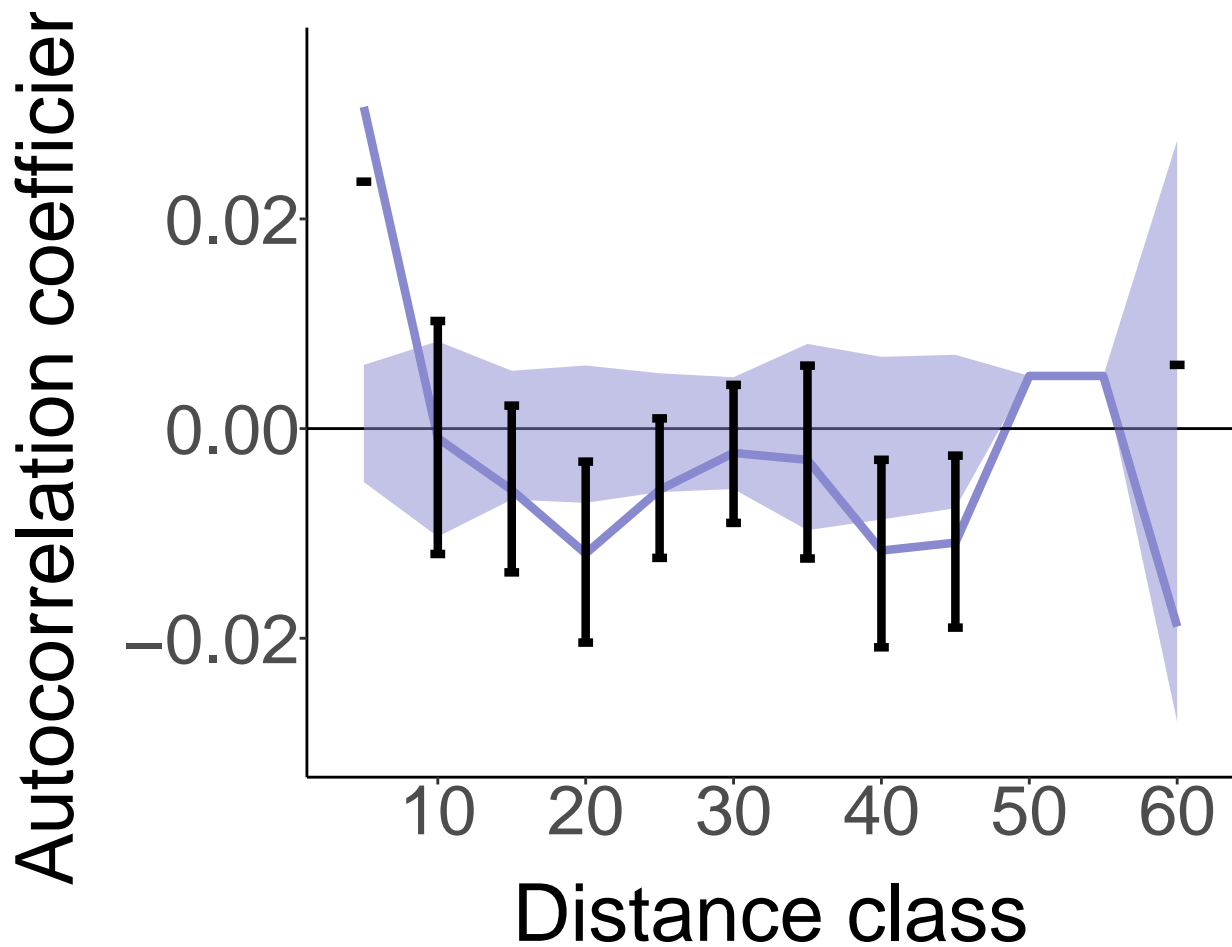


```
### Females - spatial

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))
```

```
### Chicks - spatial
spatial %>% filter(Who == "Unrelated_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))
```



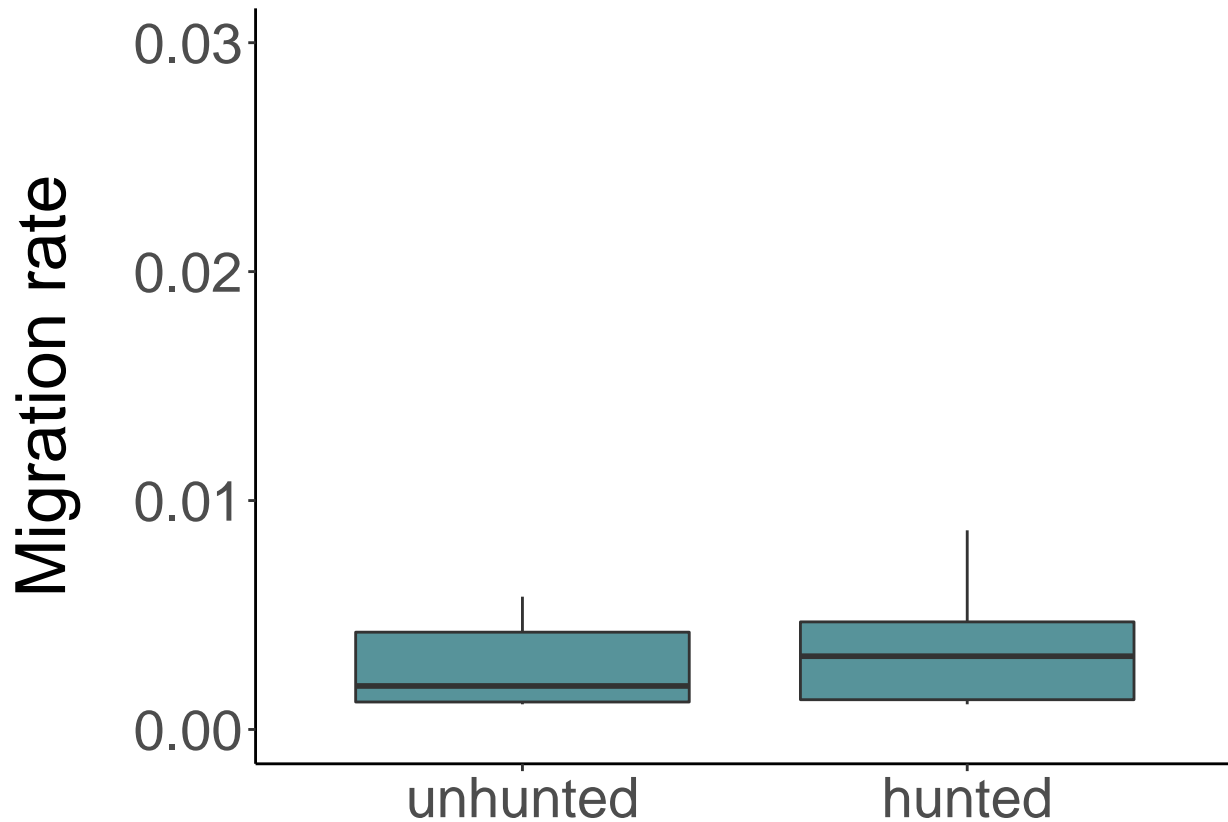
```
### Figure 3: boxplots migration rates ###
run5_clean <- read.csv("analyses/migrationanalysis/run5_clean.csv")

#change levels
run5_clean$hunt_in <- relevel(as.factor(run5_clean$hunt_in), ref = "unhunted")
run5_clean$hunt_out <- relevel(as.factor(run5_clean$hunt_out), ref = "unhunted")

#first, exclude the 'non-migration rates' which are those where pop in = pop out
run5_clean <- subset(run5_clean, m_in != m_out)

# plot emigration rates
ggplot(run5_clean, aes(x = hunt_out, y = migration_ESSc, fill = "#57939a")) +
  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)),
        legend.position = "none")+
  scale_fill_manual(values=c("#57939a"))
```

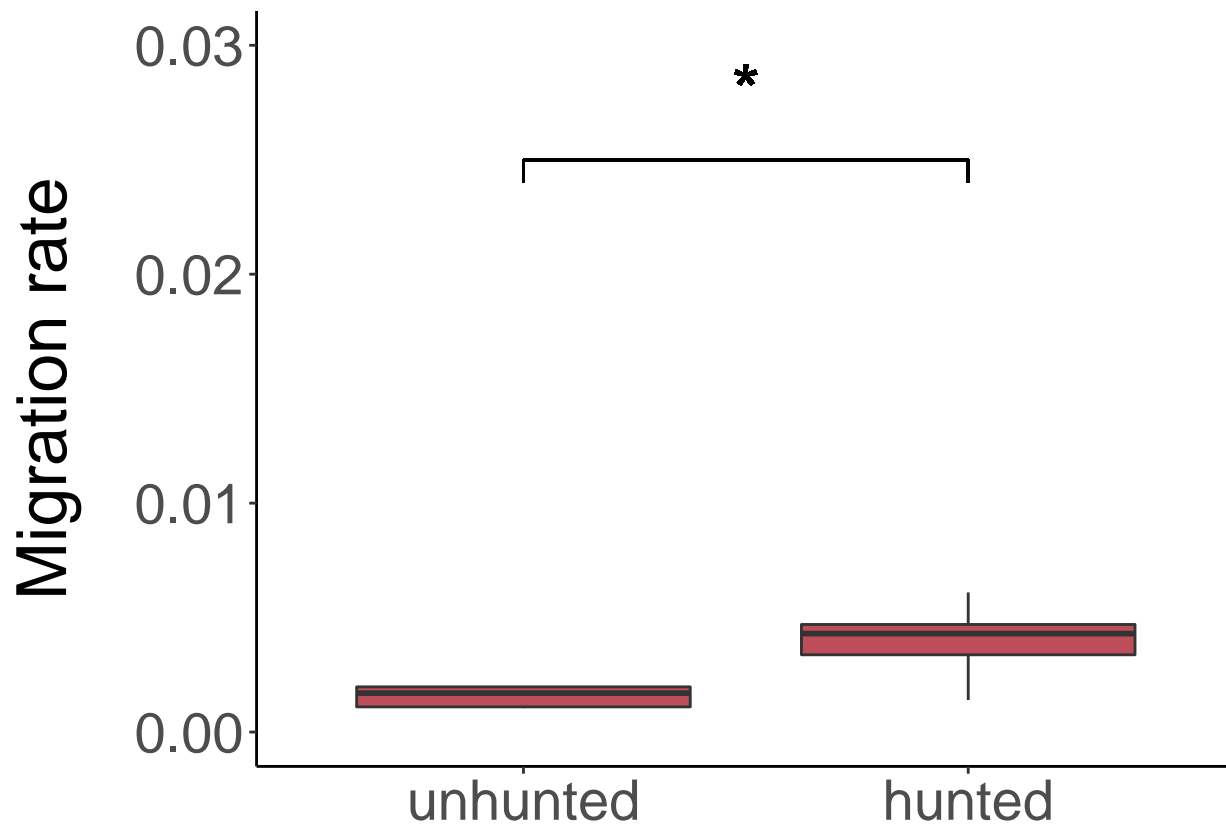
(a) Emigration rates



#plot immigration rates

```
ggplot(run5_clean, aes(x = hunt_in, y = migration_ESSc, fill = "#be4d5a")) +
  geom_boxplot(outlier.shape = NA, aes(middle = mean(migration_ESSc))) + ylim(0, 0.03)+
  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)),
        legend.position = "none")+
  scale_fill_manual(values = c("#be4d5a"))+geom_text(x = 1.5, y = 0.028, label = "*", size = 10) + geom.
  geom_segment(x=1, xend = 1, y = 0.025, yend = 0.024)+geom_segment(x=2, xend = 2, y = 0.025, yend = 0.024)
```

(b) Immigration rates



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/Berlin
## date         2022-12-17
##
## - Packages -----
## package      * version      date      lib
## ade4          * 1.7-18       2021-09-16 [1]
## adegenet      * 2.1.5        2021-10-09 [1]
## ape           * 5.6-2        2022-03-02 [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	3.3.0	2022-04-25	[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	1.5.1	2022-03-26	[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	0.6.29	2021-12-01	[1]
## doParallel	1.0.16	2020-10-16	[1]
## dplyr	* 1.0.9	2022-04-28	[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	0.16	2022-08-09	[1]
## extrafont	* 0.17	2014-12-08	[1]
## extrafontdb	1.0	2012-06-11	[1]
## fansi	1.0.3	2022-03-24	[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	0.1.3	2022-07-05	[1]
## ggplot2	* 3.3.5	2021-06-25	[1]
## glmmTMB	* 1.1.3	2022-03-13	[1]
## glue	1.6.2	2022-02-24	[1]
## gridExtra	* 2.3	2017-09-09	[1]
## gtable	0.3.0	2019-03-25	[1]
## haven	2.4.3	2021-08-04	[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	0.5.2	2021-08-25	[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	1.40	2022-08-24	[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	1.0.1	2021-09-24	[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	2.0.3	2022-03-30	[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	1.8-40	2022-03-29	[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]
##	pacman	0.5.1	2019-03-11	[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	1.7.0	2022-02-01	[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	2.5.1	2021-08-19	[1]
##	RColorBrewer	* 1.1-3	2022-04-03	[1]
##	Rcpp	1.0.9	2022-07-08	[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	1.0.5	2022-08-31	[1]
##	rmarkdown	2.14	2022-04-25	[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	1.7.8	2022-07-11	[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	* 3.1.7	2022-05-03	[1]
##	tidyr	* 1.1.3	2021-03-03	[1]
##	tidyselect	1.1.2	2022-02-21	[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	1.2.2	2021-07-24	[1]
##	vctrs	0.4.1	2022-04-13	[1]
##	vegan	2.5-7	2020-11-28	[1]
##	withr	2.4.2	2021-04-18	[1]
##	xfun	0.31	2022-05-10	[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 4.0.5)			
##	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 4.0.5)			
##	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 4.0.5)			
##	CRAN (R 4.0.2)			
##	CRAN (R 4.0.2)			
##	CRAN (R 4.0.2)			
##	CRAN (R 4.0.2)			

[illegible]

[illegible]

```
## 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
```