R-code for ‘Mushroom Genotyping Project’

J.I. Hoffman, R. Nagel, V. Litzke, D. A. Wells, W. Amos

Contents

Download packages and libraries

Exploring the data ................................................................. 2
Calculating the number of unique genets per population ......................... 4
Plotting genet frequency over time ............................................. 4
Clone Correction ......................................................................... 5
Test correlation between genet number and number of sporocarps ................ 8
Plotting spatial autocorrelation obtained using GenAlEx .......................... 8
Plotting the relatedness matrix obtained using GenAlEx ............................ 9
Calculating sMLH values using ‘inbreedR’ ...................................... 11
Mean relatedness per individual & sMLH values .................................. 12
Mean tree ages ......................................................................... 14

This document provides all the ‘R code’ for the manuscript titled Genetic analysis of Boletus edulis uncovers a transition from sexual to vegetative reproduction with increasing woodland age. Both the R Markdown file and the data is available via DRYAD (https://datadryad.org/stash/share/t2AOYp64yC-4JOP3cfSZSwR AQRAR49UBCOjZt8wbmH8). If you have any questions, do not hesitate to contact Joseph Hoffman.

The data originates from samples of the common Steinpilz (Boletus edulis) collected from Bielefeld, Germany.

Download packages and libraries

In order to repeat analyses presented in this manuscript a number of packages that extend the functionalities of base ‘R’ are required. These can be installed using the code ‘install.packages(“xxPACKAGENAMExx”, dependencies = TRUE)’

```r
library(poppr)
library(dplyr)
library(tidyrl)
library(ggplot2)
library(ggrepel)
library(viridis)
library(scales)
library(ggpubr)
library(corrrplot)
library(inbreedR)
library(readxl)
library(magrittr)
library(grid)
library(gridExtra)
library(AICcmodavg)
library(Matrix)
```
Exploring the data

The datasheet used for the initial set of analyses is a GenAlEx file including individual id, site id, and genotype information at all seven loci.

```r
mush <- read.genalex("German cep genotypes_Bielefeld.csv")

# Genotype accumulation curve
# determining the minimum number of loci necessary to discriminate between individuals
# in a population
genotype_curve(mush, sample = 1000, quiet = TRUE)
```
# look at missing data & remove individuals missing more than 3 loci
mush %>% missingno("geno", cutoff = 4/nLoc(mush)) %>% info_table(plot = FALSE)

<table>
<thead>
<tr>
<th>Locus</th>
<th>Population</th>
<th>AAC92</th>
<th>AAC71</th>
<th>AC8</th>
<th>AT102</th>
<th>AC111</th>
<th>AC101</th>
<th>ACC81</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>0.0556</td>
<td>.</td>
<td>.</td>
<td>0.0079</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>.</td>
<td>.</td>
<td>0.4000</td>
<td>.</td>
<td>.</td>
<td>0.0571</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.1000</td>
<td>.</td>
<td>.</td>
<td>0.2000</td>
<td>0.0571</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>.</td>
<td>0.2500</td>
<td>0.2500</td>
<td>0.2500</td>
<td>0.2500</td>
<td>0.2500</td>
<td>0.1429</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>.</td>
<td>.</td>
<td>0.3333</td>
<td>.</td>
<td>.</td>
<td>0.3333</td>
<td>0.0952</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>.</td>
<td>0.7500</td>
<td>0.7500</td>
<td>0.2500</td>
<td>1.0000</td>
<td>0.3929</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>0.0263</td>
<td>0.0263</td>
<td>0.0263</td>
<td>0.0263</td>
<td>0.0263</td>
<td>0.0150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
</tbody>
</table>
Calculating the number of unique genets per population

A genet is a group of genetically identical individuals, whereby an individual refers to the visible fruiting body or sporocarp that develops above ground. Given that fungi can reproduce sexually or clonally, it is important to distinguish between the total number of sporocarps sampled at a given site and the number of unique genets present at that site.

```
loci <- cbind(data.frame(mush@strata), data.frame(mlg.vector(mush)))
loci.count <- data.frame(table(loci$mlg.vector.mush.))
loci$names <- loci$names
loci.count$Var1 <- as.numeric(loci.count$Var1)
loci <- left_join(loci, loci.count, by="Var1")
loci <- loci[!duplicated(loci$Var1),]
# renumber populations by number of unique genets
loci.sum <- as.data.frame(cbind(Sum=rownsum(loci$Freq, loci$Pop, na.rm = TRUE), Pop=c(1,2,3,4,5,6,7,9,10,11,12,13,14,15)))
loci.sum$Pop <- as.factor(loci.sum$Pop)
loci <- inner_join(loci, loci.sum, by="Pop")
loci.reorder <- loci[order(loci$Sum, loci$Freq, decreasing = T),]
loci.reorder$renr <- 1:52

# plot
figure3 <- ggplot(aes(x = reorder(Pop, -Freq, sum), y = Freq, fill = renr), data = loci.reorder) +
  geom_bar(stat = "identity", color = "black") +
  labs(x = "Sample site", y = "Number of sporocarps") +
  scale_y_continuous(expand = c(0, 0), limits = c(0,40)) +
  scale_x_discrete(labels = c(1:14)) +
  scale_fill_viridis(name = "Genotype", option="viridis", direction = -1, begin = 0.13) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        text = element_text(size = 15),
        axis.text.x = element_text(vjust=0.5, hjust=1))
```

Plotting genet frequency over time

Sites at which sporocarps were collected at multiple time points are shown here. This gives an indication of how genet frequency varies per site over time.

```
added <- read.table("German cep genotypes 28_08_19_MoreInfo.csv", header=TRUE, sep="")
added$DateSampled <- as.Date(added$DateSampled, "%d-%m-%y")
head(added)
```

```
## Indiv Pop_old Pop_new Pop_name DateSampled Genotype Genotype1 Frequency AAC92
## 1 Bi70 12 1 hallowed 2015-09-20 8 1 10 3
## 2 Bi71 12 1 hallowed 2015-09-20 8 1 10 3
## 3 Bi72 12 1 hallowed 2015-09-20 8 1 10 3
## 4 Bi73 12 1 hallowed 2015-09-20 8 1 10 3
## 5 Bi74 12 1 hallowed 2015-09-20 8 1 10 3
## 6 Bi75 12 1 hallowed 2015-09-20 8 1 10 3
```
Clone Correction

The 'poppr' package is useful for the analysis of genetic data originating from systems with mixed modes of reproduction (i.e. sexual and clonal reproduction)\(^1\,^2\). We strongly recommend the poppr primer for an introduction to the package.

---


index of association = clonal populations are identified by an IA value that differs significantly from zero.

```r
ia(mush, sample = 999)
```

Population: Total
N: 134
Data: mush
Permutations: 999

```
## Ia     p.Ia  rbarD  p.rD
## 1.0621056 0.0010000 0.1818518 0.0010000
```

clone correction = remove potential bias caused by cloned genotypes

```r
mcc <- clonecorrect(mush)
ia(mcc, sample = 999)
```
Population: Total
N: 54
Data: mcc
Permutations: 999

![Graph showing distribution of a dataset with population counts and genotypic diversity calculations.]

## Ia  p.Ia  rbarD  p.rD
## 0.51683488 0.00100000 0.08922348 0.00100000

# locus table
locus_table(mcc, info = FALSE)

## summary
## locus  allele  1-D  Hexp  Evenness
## AAC92  6.00  0.75  0.76  0.92
## AAC71  7.00  0.69  0.70  0.70
## ACS   6.00  0.52  0.52  0.63
## AT102  5.00  0.61  0.61  0.67
## AC111 12.00  0.75  0.76  0.56
## AC101 11.00  0.77  0.78  0.63
## ACC81  5.00  0.19  0.19  0.40
## mean  7.43  0.61  0.62  0.64

# Calculating genotypic diversity
poppr(mcc)
Test correlation between genet number and number of sporocarps

```r
genets <- data.frame(  
  Pop = c(1:14),  
  NrSporocarps = c(38,20,18,14,8,7,6,5,4,4,3,2,1),  
  NrGenets = c(6,14,3,4,1,1,4,5,4,4,3,3,1,1))

cor.test(genets$NrGenets, genets$NrSporocarps)
```

```
## Pearson's product-moment correlation
##
data: genets$NrGenets and genets$NrSporocarps
## t = 2.0443, df = 12, p-value = 0.06351
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.03060276 0.81818112
## sample estimates:
## cor
## 0.508229
```

```r
figure5 <- ggscatter(genets, x = "NrGenets", y = "NrSporocarps",  
  add = "reg.line", conf.int = TRUE,  
  cor.coef = TRUE, cor.method = "pearson",  
  xlab = "Number of Genets", ylab = "Number of Sporocarps")
```

Plotting spatial autocorrelation obtained using GenAlEx

Spatial autocorrelation analysis was done at a distance class of 250m with the number of permutations and bootstraps set to 1000 using GenAlEx version 6.5. The software is freely available and helpful tutorials can be found online.

```r
twofifty <- read.table("SpatialAutocorrelation_250m.csv",  
  header = TRUE, sep = ",")

ggplot(data = twofifty[1:27],  
  aes(x=DistanceClass, y=r, ymin=Ur, ymax=Lr)) +  
  geom_point_range(color="black", fill="black", shape=19, size=0.5) +  
  geom_line(aes(y=U), color = "#55C667FF", linetype="dashed", size = 1) +  
  geom_line(aes(y=L), color = "#55C667FF", linetype="dashed", size = 1) +  
  geom_line(aes(y=0))
```

---

Plotting the relatedness matrix obtained using GenAlEx

Pairwise relatedness was estimated according to Lynch & Ritland (1999)\(^4\) using GenAlEx version 6.5 (please see above). Results are visualized here using ‘corrplot’\(^5\).

# population means
relate.pop <- read.table("Lynch_Ritland_1999_Related_Poprenamed_PopMean_Bielefeld.csv",
header=TRUE, sep="","

relate.pop$Pop <- factor(relate.pop$Pop, levels = relate.pop$Pop)

ggplot(data = relate.pop,
       aes(x=as.factor(Pop), y=Mean, ymin=Uv, ymax=Lv)) +
geom_pointrange(color="black", fill="white", shape=22) +
labs(x = "Population", y = "Mean Relatedness",
     title = "Within Population Pairwise Values",
     subtitle = "Lynch & Ritland (1999)") +
theme_classic() +
theme(axis.text.x = element_text(angle = 45, hjust = 1),

corrplot(as.matrix(relate), type="upper", method="color", is.corr = FALSE, tl.pos = "td",
     tl.col="black", col = viridis(100, direction = -1, begin = 0.13))
Calculating sMLH values using ‘inbreedR’

In order to use ‘inbreedR’\(^6\), the working format is typically an *individual* x *loci* matrix, where rows represent individuals and every two columns represent a single locus. If an individual is heterozygous at a given locus, it is coded as 1, whereas a homozygote is coded as 0, and missing data are coded as NA.

```r
# read data
mushroom <- readxl::read_xlsx("data.xlsx")[1:55, ]
# express alleles as numerals
mushroom[3:ncol(mushroom)] <- lapply(mushroom[3:ncol(mushroom)], as.numeric)
head(mushroom)
```

```
## # A tibble: 6 x 16
##   ID Population AAC92.a AAC92.b AAC71.a AAC71.b AC8.a AC8.b AT102.a AT102.b
## 1 1    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
## 2 2    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
## 3 3    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
## 4 4    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
## 5 5    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
## 6 6    0.0158 0.0158 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747 0.0747
```

Since demographic data is present in the beginning of our data frame, we will start our new genotype file from the 3rd column onwards. The function `convert_raw` converts a common format for genetic markers (two columns per locus) into the `inbreedR` working format. Afterwards, `check_data` allows us to test whether the genotype data frame has the correct format for subsequent analyses that use `inbreedR` functions.

```r
mushroom_geno <- convert_raw(mushroom[3:ncol(mushroom)])

invisible(lapply(mushroom_geno, table, useNA = "always"))
check_data(mushroom_geno, num_ind = 55, num_loci = 7)
```

### Create Dataframe & Estimate heterozygosity

```r
# estimate heterozygosity
het <- sMLH(mushroom_geno)
```

### Mean relatedness per individual & sMLH values

```r
mlh <- read.table("sMLH_r_230919.csv", header = TRUE, sep = ",")
mlh$Population <- as.factor(mlh$Population)
head(mlh)
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Population</th>
<th>het</th>
<th>mean.r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi70</td>
<td>1</td>
<td>0.97</td>
<td>-0.04732684</td>
</tr>
<tr>
<td>Bi85</td>
<td>1</td>
<td>1.94</td>
<td>-0.03014350</td>
</tr>
<tr>
<td>Bi86</td>
<td>1</td>
<td>1.29</td>
<td>-0.04564737</td>
</tr>
<tr>
<td>Bi93</td>
<td>1</td>
<td>1.29</td>
<td>-0.04317690</td>
</tr>
<tr>
<td>Bi97</td>
<td>1</td>
<td>0.97</td>
<td>-0.02722012</td>
</tr>
<tr>
<td>Bi113</td>
<td></td>
<td>1.94</td>
<td>-0.0200333</td>
</tr>
</tbody>
</table>

### significant differences in relatedness values among the study sites?

```r
shapiro.test(mlh$mean.r)
```

```r
# Shapiro-Wilk normality test
#
# data:  mlh$mean.r
# W = 0.93566, p-value = 0.006171
```

#data is not normally distributed -> Kruskal-Wallis Test

```r
kruskal.test(mean.r ~ Population, data = mlh)
```

```r
#
```
## Kruskal-Wallis rank sum test

## data: mean.r by Population

Kruskal-Wallis chi-squared = 41.352, df = 13, p-value = 8.354e-05

significant differences in individual sMLH among the study sites?

```r
shapiro.test(mlh$het)
```

## Shapiro-Wilk normality test

## data: mlh$het

W = 0.9341, p-value = 0.005329

#data is not normally distributed -> Kruskal-Wallis Test

```r
kruskal.test(het ~ Population, data = mlh)
```

## Kruskal-Wallis rank sum test

## data: het by Population

Kruskal-Wallis chi-squared = 23.024, df = 13, p-value = 0.04139

r and sMLH correlated?

```r
cor.test(mlh$het, mlh$mean.r)
```

## Pearson's product-moment correlation

## data: mlh$het and mlh$mean.r

t = -2.5274, df = 52, p-value = 0.01457

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:
-0.54982112 -0.06911481

sample estimates:
cor
-0.3307544
ggscatter(mlh, x = "mean.r", y = "het",
add = "reg.line", conf.int = TRUE,
cor.coef = TRUE, cor.method = "pearson",
clab = expression(italic("r")), ylab = "sMLH")

Mean tree ages

The diameter of all trees within a given site area was used to assess the age of the trees. The average age of all trees found at a given site is then used as a proxy for site/woodland age.

```r
tmeans <- read.table("TreeDiameters_MeanPerSite.csv", header = TRUE, sep = ",")
head(tmeans)
```

<table>
<thead>
<tr>
<th></th>
<th>MeanAge</th>
<th>GenetsNr</th>
<th>SporocarpNr</th>
<th>Site</th>
<th>Shannon</th>
<th>sMLH</th>
<th>r</th>
<th>He</th>
<th>NrTrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76.07143</td>
<td>6</td>
<td>38</td>
<td>1</td>
<td>1.79</td>
<td>1.40</td>
<td>0.077</td>
<td>0.72</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>23.33333</td>
<td>14</td>
<td>20</td>
<td>2</td>
<td>2.64</td>
<td>0.83</td>
<td>0.142</td>
<td>0.42</td>
<td>108</td>
</tr>
<tr>
<td>3</td>
<td>50.04545</td>
<td>3</td>
<td>18</td>
<td>3</td>
<td>1.10</td>
<td>0.97</td>
<td>0.165</td>
<td>0.47</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>72.62500</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>1.39</td>
<td>0.97</td>
<td>0.106</td>
<td>0.52</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>70.60000</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>0.00</td>
<td>1.29</td>
<td>0.000</td>
<td>0.57</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>189.14286</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>0.00</td>
<td>0.97</td>
<td>0.000</td>
<td>0.43</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Simpson</th>
<th>MainSpecies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.83</td>
<td>Beech</td>
</tr>
<tr>
<td>2</td>
<td>0.93</td>
<td>Beech</td>
</tr>
<tr>
<td>3</td>
<td>0.67</td>
<td>Beech</td>
</tr>
<tr>
<td>4</td>
<td>0.75</td>
<td>Beech</td>
</tr>
</tbody>
</table>
# build model for relationship between age of sites and weighted number of genets

```
hist(cbind(tmeans$GenetsNr, tmeans$SporocarpsNr-tmeans$GenetsNr),
      main = "", xlab = "Number of genets weighted by number of sporocarps")
```

```
m1 <- glm(formula = GenetsNr/SporocarpsNr ~ MeanAge, family = poisson(),
          data = tmeans, weights = SporocarpsNr)
summary(m1)
```

```
## Call:
## glm(formula = GenetsNr/SporocarpsNr ~ MeanAge, family = poisson(),
##     data = tmeans, weights = SporocarpsNr)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0856  -0.1703   0.7267   1.2364   1.7705
## Coefficients:
##                Estimate Std. Error   z value Pr(>|z|)
## (Intercept)    -0.062510   0.322836  -0.1940   0.8464
## MeanAge        -0.014495   0.005548  -2.6124   0.0089 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 30.840  on 13 degrees of freedom
## Residual deviance: 22.033  on 12 degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 5

```r
ggplot(data = tmeans, aes(x=MeanAge, y=GenetsNr/SporocarpsNr, weight = SporocarpsNr)) +
  geom_point(aes(color= factor(Site), size= SporocarpsNr)) +
  scale_size_continuous(range = c(2, 8)) +
  geom_text_repel(aes(label=factor(Site)), segment.color = NA, box.padding = 0.01) +
  scale_color_viridis(discrete=TRUE, direction = -1, begin = 0.13) +
  theme_classic() +
  theme(text = element_text(size = 15),
         legend.position = "none") +
  labs(y = "Genets / sporocarps", x = "Mean age (years)") +
  geom_smooth(method = "glm", se = TRUE,
               method.args = list(family = "poisson"), color = "black")
```
# build model for relationship between age of sites and weighted number of genets, excluding non-beech sites (6 & 13)

```r
m2 <- glm(formula = GenetsNr/SporocarpsNr ~ MeanAge,
          family = poisson(),
          data = tmeans[tmeans$Site[-c(6, 13)],],
          weights = SporocarpsNr)
suppressWarnings(tab_model(m2))
```

<table>
<thead>
<tr>
<th>GenetsNr/SporocarpsNr</th>
<th>Predictors</th>
<th>Incidence Rate Ratios</th>
<th>CI</th>
<th>p</th>
<th>(Intercept)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.42</td>
</tr>
</tbody>
</table>
# build model for relationship between age of sites and weighted number of genets, excluding oak sites (6, 7 & 13)

```r
m3 <- glm(formula = GenetsNr/SporocarpsNr ~ MeanAge,
          family = poisson(),
          data = tmeans[tmeans$Site[-c(6,7,13)],],
          weights = SporocarpsNr)
suppressWarnings(tab_model(m3))
```

GenetsNr/SporocarpsNr

Predictors

Incidence Rate Ratios

CI

p

(Intercept)

1.40

0.66 – 2.79

0.353

MeanAge

0.98

0.96 – 0.99

0.001

Observations

12

R2 Nagelkerke

0.687

# build model for relationship between age of sites and weighted number of genets, including main tree species as factor

```r
m4 <- glm(formula = GenetsNr/SporocarpsNr ~ MeanAge + MainSpecies,
          family = poisson(),
          data = tmeans,
          weights = SporocarpsNr)
suppressWarnings(tab_model(m4))
```

Observations

11

R2 Nagelkerke

0.685
weights = SporocarpsNr
suppressWarnings(tab_model(m4))

GenetsNr/SporocarpsNr
Predictors
Inci dence Rate Ratios
CI
p
(Intercept)
1.41
0.66 – 2.80
0.344
MeanAge
0.98
0.96 – 0.99
0.001
MainSpecies [Birch]
1.07
0.32 – 2.69
0.898
MainSpecies [Oak]
12.78
1.05 – 128.58
0.034
Observations
14
R2 Nagelkerke
0.676

# build model for relationship between age of sites and weighted number of genets, including "main tree

m5 <- lmer(formula = GenetsNr/SporocarpsNr ~ MeanAge + (1|MainSpecies),
            data = tmeans,
weights = as.numeric(SporocarpsNr))
suppressWarnings(tab_model(m5))

GenetsNr/SporocarpsNr
Predictors
Estimates
CI
p
(Intercept)
0.78
0.28 – 1.28
0.002
MeanAge
-0.00
-0.01 – 0.00
0.060
Random Effects
2
0.85
0.00 MainSpecies
0.03
ICC
0.04
N MainSpecies
3
Observations
14
Marginal R2 / Conditional R2
0.058 / 0.092

# build model for relationship between age of sites and r, sMLH, Simpson's index, 
# Shannon's H

m2 <- glm(formula = MeanAge ~ Shannon, data = tmeans)
m3 <- glm(formula = MeanAge ~ sMLH, data = tmeans)
m4 <- glm(formula = MeanAge ~ r, data = tmeans)
m5 <- glm(formula = MeanAge ~ Simpson, data = tmeans)

Shannon <- ggplot(data = tmeans, aes(x=Shannon, y=MeanAge)) +
  geom_point(aes(color= factor(Site), size= SporocarpsNr)) +
  scale_size_continuous(range = c(2, 8)) +
  geom_text_repel(aes(label=factor(Site)), segment.color = NA) +
  scale_color_viridis(discrete=TRUE, direction = -1, begin = 0.13) +
  theme_classic() +
  theme(text = element_text(size = 15),
        legend.position = "none") +
  labs(y = "", x = "Shannon's H") +
  geom_smooth(method = "glm", se = TRUE, color = "black") +
  scale_y_continuous(limits = c(-50,200)) +
  scale_x_continuous(breaks = pretty_breaks(5))
Simpson <- ggplot(data = tmeans, aes(x = Simpson, y = MeanAge)) +
  geom_point(aes(color = factor(Site), size = SporocarpsNr)) +
  scale_size_continuous(range = c(2, 8)) +
  geom_text_repel(aes(label = factor(Site)), segment.color = NA) +
  scale_color_viridis(discrete = TRUE, direction = -1, begin = 0.13) +
  theme_classic() +
  theme(text = element_text(size = 15),
        legend.position = "none") +
  labs(y = "", x = "Simpson's index") +
  geom_smooth(method = "glm", se = TRUE, color = "black") +
  scale_y_continuous(limits = c(-50, 200)) +
  scale_x_continuous(breaks = pretty_breaks(5))

sMLH <- ggplot(data = tmeans, aes(x = sMLH, y = MeanAge)) +
  geom_point(aes(color = factor(Site), size = SporocarpsNr)) +
  scale_size_continuous(range = c(2, 8)) +
  geom_text_repel(aes(label = factor(Site)), segment.color = NA) +
  scale_color_viridis(discrete = TRUE, direction = -1, begin = 0.13) +
  theme_classic() +
  theme(text = element_text(size = 15),
        legend.position = "none") +
  labs(y = "", x = "sMLH") +
  geom_smooth(method = "glm", se = TRUE, color = "black") +
  scale_y_continuous(limits = c(-50, 200)) +
  scale_x_continuous(breaks = pretty_breaks(5))

r <- ggplot(data = tmeans, aes(x = r, y = MeanAge)) +
  geom_point(aes(color = factor(Site), size = SporocarpsNr)) +
  scale_size_continuous(range = c(2, 8)) +
  geom_text_repel(aes(label = factor(Site)), segment.color = NA) +
  scale_color_viridis(discrete = TRUE, direction = -1, begin = 0.13) +
  theme_classic() +
  theme(text = element_text(size = 15),
        legend.position = "none") +
  labs(y = "", x = "Relatedness") +
  geom_smooth(method = "glm", se = TRUE, color = "black") +
  scale_y_continuous(limits = c(-50, 200)) +
  scale_x_continuous(breaks = pretty_breaks(5))

fig <- ggarrange(Shannon, Simpson, r, sMLH, ncol = 2, nrow = 2)
annotate_figure(fig,
               left = text_grob("Mean age (years)", rot = 90, vjust = 1.5, size = 15))
### - Session info

---

### - Packages

---

### - Session info

---

### - Packages
<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Date</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ape</td>
<td>5.3</td>
<td>2019-03-17</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>assertthat</td>
<td>0.2.1</td>
<td>2019-03-21</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>backports</td>
<td>1.1.7</td>
<td>2020-05-13</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>bayestestR</td>
<td>0.6.0</td>
<td>2020-04-20</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>boot</td>
<td>1.3-25</td>
<td>2020-04-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>broom</td>
<td>0.5.6</td>
<td>2020-04-20</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>car</td>
<td>* 3.0-8</td>
<td>2020-05-21</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>carData</td>
<td>* 3.0-4</td>
<td>2020-05-22</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>cellranger</td>
<td>1.1.0</td>
<td>2016-07-27</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>class</td>
<td>7.3-17</td>
<td>2020-04-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>classInt</td>
<td>0.4-3</td>
<td>2020-04-07</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>cli</td>
<td>2.0.2</td>
<td>2020-02-28</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>cluster</td>
<td>2.1.0</td>
<td>2019-06-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>coda</td>
<td>0.19-3</td>
<td>2019-07-05</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>codetools</td>
<td>0.2-16</td>
<td>2018-12-24</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>colorspace</td>
<td>1.4-1</td>
<td>2019-03-18</td>
<td>CRAN (R 3.6.1)</td>
</tr>
<tr>
<td>corrplot</td>
<td>* 0.84</td>
<td>2017-10-16</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>cowplot</td>
<td>1.0.0</td>
<td>2019-07-11</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>crayon</td>
<td>1.3.4</td>
<td>2017-09-16</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>curl</td>
<td>4.3</td>
<td>2019-12-02</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>data.table</td>
<td>1.12.8</td>
<td>2019-12-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>DBI</td>
<td>1.1.0</td>
<td>2019-12-15</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>deldir</td>
<td>0.1-25</td>
<td>2020-02-03</td>
<td>CRAN (R 3.6.2)</td>
</tr>
<tr>
<td>digest</td>
<td>0.6.25</td>
<td>2020-02-23</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>dplyr</td>
<td>* 0.8.5</td>
<td>2020-03-07</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>e1071</td>
<td>1.7-3</td>
<td>2019-11-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>effectsize</td>
<td>0.3.1</td>
<td>2020-05-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ellipsis</td>
<td>0.3.1</td>
<td>2020-05-15</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>emmeans</td>
<td>1.4.6</td>
<td>2020-04-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>estimability</td>
<td>1.3</td>
<td>2018-02-11</td>
<td>CRAN (R 3.6.0)</td>
</tr>
<tr>
<td>evaluate</td>
<td>0.14</td>
<td>2019-05-28</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>expm</td>
<td>0.999-4</td>
<td>2019-03-21</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>fansi</td>
<td>0.4.1</td>
<td>2020-01-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>farver</td>
<td>2.0.3</td>
<td>2020-01-16</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>fastmap</td>
<td>1.0.1</td>
<td>2019-10-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>fastmatch</td>
<td>1.1-0</td>
<td>2017-01-28</td>
<td>CRAN (R 3.6.0)</td>
</tr>
<tr>
<td>forcats</td>
<td>0.5.0</td>
<td>2020-03-01</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>foreign</td>
<td>0.8-75</td>
<td>2020-01-20</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>gdata</td>
<td>2.18.0</td>
<td>2017-06-06</td>
<td>CRAN (R 3.6.2)</td>
</tr>
<tr>
<td>generics</td>
<td>0.0.2</td>
<td>2018-11-29</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ggeffects</td>
<td>0.14.3</td>
<td>2020-04-20</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ggplot2</td>
<td>* 3.3.0</td>
<td>2020-03-05</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ggpubr</td>
<td>* 0.3.0</td>
<td>2020-05-04</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ggrepel</td>
<td>* 0.8.2</td>
<td>2020-03-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>ggsignif</td>
<td>0.6.0</td>
<td>2019-08-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>glue</td>
<td>1.4.1</td>
<td>2020-05-13</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>gmodels</td>
<td>2.18.1</td>
<td>2018-06-25</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>gridExtra</td>
<td>* 2.3</td>
<td>2017-09-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>gtable</td>
<td>0.3.0</td>
<td>2019-03-25</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>gtools</td>
<td>3.8.2</td>
<td>2020-03-31</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>haven</td>
<td>2.3.0</td>
<td>2020-05-24</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>hms</td>
<td>0.5.3</td>
<td>2020-01-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>htmltools</td>
<td>0.4.0</td>
<td>2019-10-04</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>httpuv</td>
<td>1.5.2</td>
<td>2019-09-11</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>Package</td>
<td>Version</td>
<td>Date</td>
<td>Repository</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>httr</td>
<td>1.4.1</td>
<td>2019-08-05</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>igraph</td>
<td>1.2.5</td>
<td>2020-03-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>inbreedR</td>
<td>0.3.2</td>
<td>2016-09-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>insight</td>
<td>0.8.4</td>
<td>2020-05-13</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>kableExtra</td>
<td>1.1.0</td>
<td>2019-03-16</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>KernSmooth</td>
<td>2.23-17</td>
<td>2020-04-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>knitr</td>
<td>1.28</td>
<td>2020-02-06</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>labeling</td>
<td>0.3</td>
<td>2014-08-23</td>
<td>CRAN (R 3.6.0)</td>
</tr>
<tr>
<td>later</td>
<td>1.0.0</td>
<td>2019-10-04</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>lattice</td>
<td>0.20-41</td>
<td>2020-04-02</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>LearnBayes</td>
<td>2.15.1</td>
<td>2018-03-18</td>
<td>CRAN (R 3.6.0)</td>
</tr>
<tr>
<td>lifecycle</td>
<td>0.2.0</td>
<td>2020-03-06</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>lme4</td>
<td>1.1-23</td>
<td>2020-04-07</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>magrittr</td>
<td>1.5</td>
<td>2014-11-22</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>MASS</td>
<td>7.3-51.6</td>
<td>2020-04-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>Matrix</td>
<td>1.2-18</td>
<td>2019-11-27</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>mgcv</td>
<td>1.8-31</td>
<td>2019-11-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>mime</td>
<td>0.9</td>
<td>2020-02-04</td>
<td>CRAN (R 3.6.2)</td>
</tr>
<tr>
<td>minqa</td>
<td>1.2.4</td>
<td>2014-10-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>modelr</td>
<td>0.1.8</td>
<td>2020-05-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>munsell</td>
<td>0.5.0</td>
<td>2018-06-12</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>mtvnorm</td>
<td>1.1-0</td>
<td>2020-02-24</td>
<td>CRAN (R 3.6.2)</td>
</tr>
<tr>
<td>nlme</td>
<td>3.1-148</td>
<td>2020-05-24</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>nloptr</td>
<td>1.2.2.1</td>
<td>2020-03-11</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>openxlsx</td>
<td>4.1.5</td>
<td>2020-05-06</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>parameters</td>
<td>0.7.0</td>
<td>2020-05-18</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>pegas</td>
<td>0.13</td>
<td>2020-03-10</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>performance</td>
<td>0.4.6</td>
<td>2020-05-03</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>permute</td>
<td>0.9-5</td>
<td>2019-03-12</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>phangorn</td>
<td>2.5.5</td>
<td>2019-06-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>pillar</td>
<td>1.4.4</td>
<td>2020-05-05</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>pkgconfig</td>
<td>2.0.3</td>
<td>2019-09-22</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>plyr</td>
<td>1.8.6</td>
<td>2020-03-03</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>polysat</td>
<td>1.7-4</td>
<td>2019-03-06</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>poppr</td>
<td>2.8.5</td>
<td>2020-02-25</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>promises</td>
<td>1.1.0</td>
<td>2019-10-04</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>purrr</td>
<td>0.3.4</td>
<td>2020-04-17</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>quadprog</td>
<td>1.5-8</td>
<td>2019-11-20</td>
<td>CRAN (R 3.6.1)</td>
</tr>
<tr>
<td>qvalue</td>
<td>2.18.0</td>
<td>2019-10-29</td>
<td>Bioconductor</td>
</tr>
<tr>
<td>R6</td>
<td>2.4.1</td>
<td>2019-11-12</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>raster</td>
<td>3.1-5</td>
<td>2020-04-19</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>Rcpp</td>
<td>1.0.3</td>
<td>2019-11-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>readr</td>
<td>1.3.1</td>
<td>2018-12-21</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>readxl</td>
<td>1.3.1</td>
<td>2019-03-13</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>reshape2</td>
<td>1.4.4</td>
<td>2020-04-09</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rio</td>
<td>0.5.16</td>
<td>2018-11-26</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rlang</td>
<td>0.4.6</td>
<td>2020-05-02</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rmarkdown</td>
<td>2.1</td>
<td>2020-01-20</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rstatix</td>
<td>0.5.0</td>
<td>2020-04-28</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rstudioapi</td>
<td>0.11</td>
<td>2020-02-07</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>rvtest</td>
<td>0.3.5</td>
<td>2019-11-08</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>scales</td>
<td>1.1.1</td>
<td>2020-05-11</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>seqinr</td>
<td>3.6-1</td>
<td>2019-09-07</td>
<td>CRAN (R 3.6.3)</td>
</tr>
<tr>
<td>sessioninfo</td>
<td>1.1.1</td>
<td>2018-11-05</td>
<td>CRAN (R 3.6.3)</td>
</tr>
</tbody>
</table>
## sf 0.9-3 2020-05-04 [1] CRAN (R 3.6.3)
## shiny 1.4.0.2 2020-03-13 [1] CRAN (R 3.6.3)
## sjlabelled 1.1.5 2020-05-25 [1] CRAN (R 3.6.3)
## sjmisc 2.8.4 2020-04-03 [1] CRAN (R 3.6.3)
## sjPlot * 2.8.4 2020-05-24 [1] CRAN (R 3.6.3)
## sjstats 0.18.0 2020-05-06 [1] CRAN (R 3.6.3)
## sp 1.4-2 2020-05-20 [1] CRAN (R 3.6.3)
## spData 0.3.5 2020-04-06 [1] CRAN (R 3.6.3)
## spdep 1.1-3 2019-09-18 [1] CRAN (R 3.6.3)
## statmod 1.4.34 2020-02-17 [1] CRAN (R 3.6.3)
## stringi 1.4.6 2020-02-17 [1] CRAN (R 3.6.2)
## stringr 1.4.0 2019-02-10 [1] CRAN (R 3.6.3)
## survival 3.1-12 2020-04-10 [1] CRAN (R 3.6.3)
## tibble 3.0.1 2020-04-20 [1] CRAN (R 3.6.3)
## tidyselect 1.1.0 2020-05-20 [1] CRAN (R 3.6.3)
## units 0.6-6 2020-03-16 [1] CRAN (R 3.6.3)
## unmarked 1.0.0 2020-05-04 [1] CRAN (R 3.6.3)
## utf8 1.4.0 2019-02-10 [1] CRAN (R 3.6.3)
## vctrs 0.3.0 2020-05-11 [1] CRAN (R 3.6.3)
## vegan 2.5-6 2019-09-01 [1] CRAN (R 3.6.3)
## VGAM 1.1-3 2020-04-28 [1] CRAN (R 3.6.3)
## viridis * 0.5.1 2018-03-29 [1] CRAN (R 3.6.3)
## viridisLite * 0.3.0 2018-02-01 [1] CRAN (R 3.6.3)
## webshot 0.5.3 2019-11-22 [1] CRAN (R 3.6.3)
## withr 2.2.0 2020-04-20 [1] CRAN (R 3.6.3)
## xfun 0.14 2020-05-20 [1] CRAN (R 3.6.3)
## xml2 1.3.2 2020-04-23 [1] CRAN (R 3.6.3)
## xtable 1.8-4 2019-04-21 [1] CRAN (R 3.6.3)
## yaml 2.2.1 2020-02-01 [1] CRAN (R 3.6.2)
## zip 2.0.4 2019-09-01 [1] CRAN (R 3.6.3)
##