Testing the Heterozygosity per Island

Grace Saville

01/02/2022

Preamble

```
library(stringr)
library(dplyr)
getwd()
setwd("C:/Users/airhe/OneDrive/Documents/Masters/Project 3/kiore-project")
```

Dataset loading

```
data <- read.csv("./data/ratsSNPs_clean.csv")

copy <- data # making a copy of the data
colnames(copy)
copy <- copy[,-c(2:16)] # removing unnecessary columns for this analysis
```

Replaces SNP's with symbols for heterozygous and homozygous

```
dim(copy) # 370 rows 283 columns
str(copy)
unique(unlist(copy[,17:283])) # checking what SNP combinations are present
# "?"   "A:A" "G:G" "A:G" "T:T" "A:T" "C:C" "C:T" "T:G" "C:G" "A:C"

het <- c("A:G", "A:T", "C:T", "T:G", "C:G", "A:C") # vector of heterozygous combinations
hom <- c("G:G", "A:A", "C:C", "T:T") # vector of homozygous combinations

# Replacing specific combos
for (i in 1:nrow(copy)){
  copy[i,][copy[i,] %in% het] <- "E"
  copy[i,][copy[i,] %in% hom] <- "O"
}

rm(i)

str(copy)
copy[copy == "?"] <- NA # replacing ? with NA's
```

Calculating heterozygous and homozygous totals per specimen

```
o.freq <- vector()
e.freq <- vector()
na.freq <- vector()
for (i in 1:370) {
  x <- sum(grepl("O", copy[i,], fixed = TRUE)) # counting row O's
  o.freq <- append(o.freq, x) # adding sum to vector
  x <- sum(grepl("E", copy[i,], fixed = TRUE)) # counting row E's
  e.freq <- append(e.freq, x)
  x <- sum(is.na(copy[i,])) # counting row NA's
  na.freq <- append(na.freq, x)
  rm(x)
}

rm(i)

per.specimen <- data.frame(copy$island, o.freq, e.freq, na.freq) # making a df with the freq totals
perc.o <- round(as.vector((per.specimen$o.freq / (per.specimen$o.freq + per.specimen$e.freq)) * 100), di
per.specimen$perc.o <- perc.o
perc.missing <- round(as.vector((per.specimen$na.freq / (per.specimen$o.freq + per.specimen$e.freq + per
per.specimen$perc.missing <- perc.missing

rm(o.freq, e.freq, na.freq, perc.o, perc.missing)

str(per.specimen)
```

Calculating heterozygous and homozygous totals per island

```
data <- data[order(data$island, decreasing = FALSE),] # ordering df alphabetically by island
names(data)
data[c(grep("Mainland", data$island.1)),c(1, 8)] # checking which populations fall in the mainland cate
unique(data$island)
shortpopnames <- as.character(
  c(
    "aotea",
    "borneo",
    "cambodia",
    "grtmercury",
    "halmaher",
    "hatutaa",
    "honuea",
    "kaikura",
    "kamaka",
    "kayangel",
    "laos",
    "late",
    "luzon",
    "malenge",
    "mohotani",
```

```
    "motukawa",
    "newbrita",
    "newguine",
    "normanby",
    "rakiura",
    "reiono",
    "rimatuu",
    "slipper",
    "southland",
    "sulawesi",
    "tahanea",
    "thailand",
    "wake"
  )
) # writing shortened names as is in the data df so I can use the character strings with grep()

# splitting the data df into df objects by island:
for (i in 1:length(shortpopnames)) {
  y <- as.vector(grep(shortpopnames[i], copy[,1], ignore.case = TRUE, value = FALSE))
  assign(paste(shortpopnames[i]), copy[y,])
}
rm(y, i)

# making a "mainland" df:
mainland <- rbind(cambodia, thailand, laos)
rm(cambodia, thailand, laos)
shortpopnames
shortpopnames <- shortpopnames[-c(3, 11, 27)] # removing the names now in mainland category
shortpopnames <- append(shortpopnames, "mainland")

# double checking the dfs have the correct specimens per island:
length(shortpopnames) #26
for (i in 1:26) {
  print(shortpopnames[i])
  y <- get(shortpopnames[i])
  print(y[,1])
}
rm(y)
```

```
# counting values:
per.island <- data.frame()
for (i in 1:length(shortpopnames)) {
  o <- sum(grepl("O", unlist(get(shortpopnames[i])), fixed = TRUE)) # counting row O's
  e <- sum(grepl("E", unlist(get(shortpopnames[i])), fixed = TRUE)) # counting row E's
  na <- sum(is.na(get(shortpopnames[i]))) # counting row NA's
  vec <- as.vector(c(shortpopnames[i], nrow(get(shortpopnames[i])), o, e, na))
  per.island <- rbind(per.island, vec)
}
rm(o, e, na, vec, i)

colnames(per.island) <- c("island", "specimens.based.on", "o.freq", "e.freq", "na.freq") # renaming col
str(per.island)
per.island$o.freq <- as.integer(per.island$o.freq) # changing the numbers from characters to integers
```

```
per.island$e.freq <- as.integer(per.island$e.freq)
per.island$na.freq <- as.integer(per.island$na.freq)

# calculating percentages:
perc.o <- round(as.vector((per.island$o.freq / (per.island$o.freq + per.island$e.freq)) * 100), digits =
per.island$perc.o <- perc.o
perc.missing <- round(as.vector((per.island$na.freq / (per.island$o.freq + per.island$e.freq + per.islar
per.island$perc.missing <- perc.missing

rm(perc.o, perc.missing, het, hom)
rm(list = shortpopnames)
```

```
getwd()
write.csv(per.island, "./results/heterozygosity_testing_results_table.csv", row.names = FALSE)
```

Considering island size and distance from mainland

```
island.km <- read.csv("./data/island_size_data.csv", header = TRUE)
head(island.km)
island.km <- island.km[,c(2, 4)]
island.km <- island.km[order(island.km$ISLAND, decreasing = FALSE),] # sorting alphabetically
```

```
distance.from.ML <- data[,c(1,8,11,12)]
distance.from.ML <- distance.from.ML[!duplicated(distance.from.ML$island.1),] # keeping only 1 coordina
distance.from.ML <- distance.from.ML[order(distance.from.ML$island.1, decreasing = FALSE),] # sorting a
row.names(distance.from.ML) <- seq(nrow(distance.from.ML)) # renaming row numbers to be sequential
distance.from.ML # checking

#shortening some of the long island names to match what is already in the island.km df:
distance.from.ML[1,2] <- "Aotea"
distance.from.ML[20,2] <- "Rakiura"
distance.from.ML[22,2] <- "Rimatuu"

#decided to keep Cambodia as the mainland coordinates, will use this as the base for distance from main
distance.from.ML <- distance.from.ML[,-1] # removing specimen ID column since no longer necessary

library(geosphere)
km.from.ML <- vector()
for (i in 1:nrow(distance.from.ML)) {
  x <- distGeo(as.vector(distance.from.ML[13,c(3,2)]), as.vector(distance.from.ML[i,c(3,2)]))
  km.from.ML <- append(km.from.ML, x)
}

km.from.ML <- km.from.ML / 1000 # converting from metres to kilometres
distance.from.ML$km.from.ML <- km.from.ML # adding the kms to the distance df

rm(i, x, km.from.ML)
```

```r
names(island.km)
names(distance.from.ML)
island.km <- merge(island.km, distance.from.ML, by.x = "ISLAND", by.y = "island.1", all = TRUE)

rm(distance.from.ML)

island.km$ISLAND
per.island$island
per.island$island <- c(
  "Aotea",
  "Borneo",
  "Great Mercury Island",
  "Halmahera",
  "Hatutaa",
  "Honuea",
  "Kaikura Island",
  "Kamaka",
  "Kayangel",
  "Late Island",
  "Luzon",
  "Malenge",
  "Mohotani",
  "Motukawanui",
  "New Britain",
  "New Guinea",
  "Normanby Island",
  "Rakiura",
  "Reiono",
  "Rimatuu",
  "Slipper Island",
  "Doubtful Sound",
  "Sulawesi",
  "Tahanea",
  "Wake Island",
  "Mainland"
) # editing the names to match those in the other df so I can merge them

per.island <- merge(per.island, island.km, by.x = "island", by.y = "ISLAND", all = TRUE)
```

```r
getwd()
write.csv(per.island, "./results/heterozygosity_testing_results_table.csv", row.names = FALSE)
```

Plots

```r
names(per.island)

library(ggplot2)

ggplot(data = per.island, aes(x = km.from.ML, y = perc.o)) + geom_point(shape = 1, colour = "red") + ge
```

Linear regression

```
LM <- lm(perc.o ~ km.from.ML, data = per.island)
summary(LM)
```

Results:

Residuals:

Min	1Q	Median	3Q	Max
-14.4165	-1.6867	0.6933	2.6053	9.1871

Coefficients:

	Estimate	Std. Error	t value	p-value
(Intercept)	8.249e+01	2.156e+00	38.262	< 2e-16 ***
km.from.ML	8.927e-04 (coefficient)	2.470e-04	3.614	0.00139 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.149 on 24 degrees of freedom
Multiple R-squared: 0.3525, Adjusted R-squared: 0.3255 F-statistic: 13.06 on 1 and 24 DF, p-value: 0.001387

- Higher t-value is generally more significant, related to p-value.
- Significant p-value, can reject H0 where x and y are not correlated.