Reviewing the half-clean data with a Hardy-Weinberg and Structure Analysis

Grace Saville

16/05/2022

1. Prepping df for HWE Analysis

I would like to remove the SNPs not in Hardy-Weinberg equilibrium, therefore I need to reformat the data for input into HWE program.

```
# load(".RData") # if necessary
data <- read.csv("./data/RStudio/ratsSNPs_halfclean.csv")
```

1a. Reformatting the dataframe

```
copy <- data # making a copy
t(copy[1,1:20]) # checking column names
```

```
##                     1
## island              "Borneo_002"
## registration.number "NBC.LAB.1968"
## genus               "Rattus"
## species             "exulans"
## sex                 "female"
## country             "Indonesia"
## state_province      "Kalimantan Timur"
## island.1            "Borneo"
## locality            "Badang, Sungai Kajan"
## site                ""
## geo_lat             "-0.5102"
## geo_long            "117.0912"
## collector           "Victor von Plessen"
## collecting.date     "1935"
## field.number        "AMNH.103837"
## Populatie           "1"
## X299_CHR1_114679736 "?"
## X13_CHR1_116614092  "?"
## X14_CHR1_124857905  "?"
## X15_CHR1_134869867  "T:T"
```

```
copy <- copy[,-c(2:16)] # removing all but specimen names and SNPs
# copy[1,1:20] # checking

copy[copy == "?"] <- "?:?" # replacing single ? with double ? so alleles can be split

x <- data.frame(island = copy$island) # setting up new df for for loop
coln <- as.vector(colnames(copy)) # prepping to paste the column names into the for loop
dim(copy) # 379 rows 283 columns
```

```
## [1] 379 283
```

```
for (i in 2:283) {
  y <- colsplit(copy[,i], split = ":", names = c(coln[i], paste("blank", i, sep = "."))) # splitting ea
  x <- cbind(x, y) # combining output with current df
  rm(i, y) # removing temp objects
}

# Checking:
# dim(x3) # 379 rows 565 columns
# x2[1:5,1:5]
# x3[1:5,1:5] # comparing the 2 dfs to check the column naming worked correctly

copy <- x
rm(x, coln) # removing excess objects
```

1b. Producing the file necessary for PGDSpider program

```
copy <- copy[order(copy$island, decreasing = FALSE), ] # ordering df alphabetically by island
# as.matrix(copy[, 1]) # printing the island names and row numbers

# A=1, T=2, G=3, C=4
copy[copy == "A"] <- "1"
copy[copy == "T"] <- "2"
copy[copy == "G"] <- "3"
copy[copy == "C"] <- "4"

# row numbers in dataset df listed below for each popn.
popnames <- as.character(
  c(
    "pop = Aotea", # 1:10
    "pop = Borneo", # 11:28
    "pop = Doubtful_Sound", # 315
    "pop = Great_Mercury_Island", # 30
    "pop = Halmahera", # 31:42
    "pop = Hatutaa", # 43:63
    "pop = Honuea", # 64:83
    "pop = Kaikura_Island", # 84:103
    "pop = Kamaka", # 104:124
    "pop = Kayangel", # 125:145
    "pop = Late_Island", # 148:168
    "pop = Mainland", # 29, 146, 147, 169, 358, 359 (including Luzon here because
    # Luzon is part of the mainland cluster in the NeighborNet network)
    "pop = Malenge", # 170:181
    "pop = Mohotani", # 182:195
    "pop = Motukawanui", # 196:216
    "pop = New_Britain", # 217:226
    "pop = New_Guinea", # 227:229
    "pop = Normanby_Island", # 230
    "pop = Rakiura", # 231:251
    "pop = Reiono", # 252:272
    "pop = Rimatuu", # 273:293
    "pop = Slipper_Island", # 294:314
    "pop = Sulawesi", # 316:337
    "pop = Tahanea", # 338:357
    "pop = Wake_Island" # 360:379
  )
)

# Creating population dfs
a <- as.data.frame(copy[1:10,]) # Aotea
b <- as.data.frame(copy[11:28,]) # Borneo
c <- as.data.frame(copy[315,]) # Doubtful_Sound
d <- as.data.frame(copy[30,]) # Great_Mercury_Island
e <- as.data.frame(copy[31:42,]) # Halmahera
f <- as.data.frame(copy[43:63,]) # Hatutaa
g <- as.data.frame(copy[64:83,]) # Honuea
h <- as.data.frame(copy[84:103,]) # Kaikura_Island
i <- as.data.frame(copy[104:124,]) # Kamaka
j <- as.data.frame(copy[125:145,])  # Kayangel
k <- as.data.frame(copy[148:168,]) # Late_Island
l <- as.data.frame(copy[c(29, 146, 147, 169, 358, 359),]) # Mainland
m <- as.data.frame(copy[170:181,]) # Malenge
n <- as.data.frame(copy[182:195,]) # Mohotani
o <- as.data.frame(copy[196:216,]) #  Motukawanui
p <- as.data.frame(copy[217:226,]) #  New_Britain
q <- as.data.frame(copy[227:229,]) # New_Guinea
r <- as.data.frame(copy[230,]) #  Normanby_Island
s <- as.data.frame(copy[231:251,]) # Rakiura
t <- as.data.frame(copy[252:272,]) # Reiono
u <- as.data.frame(copy[273:293,]) # Rimatuu
v <- as.data.frame(copy[294:314,]) # Slipper_Island
w <- as.data.frame(copy[316:337,]) #  Sulawesi
x <- as.data.frame(copy[338:357,]) # Tahanea
y <- as.data.frame(copy[360:379,]) # Wake_Island

pops <- as.character(c(letters[seq(from = 1, to = 25)])) # list of popn object names
```

```
ncol(copy) #565
getwd()

sink("./data/ratsSNPs_PGDSpider_input.txt") # create empty file
cat("rats_SNPS", "npops = 25", "nloci = 282", fill = 1)
cat("\t", fill = FALSE)
cat(colnames(copy[,c(FALSE,TRUE)]), "\n", sep = "\t\t", fill = FALSE) # column/SNP
# names (even columns only)
for (i1 in 1:25) {
  cat(popnames[i1], fill = 1) # island name
  foo <- get(pops[i1]) # calling the island object based on the pops vector
  for (i2 in 1:nrow(foo)) {
    cat(as.character(foo[i2, ]), "\n", fill = FALSE, sep = "\t") # print SNP rows
  } # inner loop close
} # outer loop close
sink() # closing the sink connection (do not forget!)
```

```
rm(i1, i2, foo, popnames, pops)
rm(list = c(letters[seq(from = 1, to = 25)])) # removing excess objects
```

At this stage PGDSpider program and Arlequin were used to convert the file produced and run tests on the data. The resulting output is used here for analysis. Full method detailled in the README file.

2. HWE Analysis and removal

2a. Creating loop for reading the HWE files

```
filenames <- as.vector(list.files("./results/Arlequin_HardyWeinberg/hwe_results_by_island_14032022"))

# START OF MEGA FOR LOOP:
for (i in 1:length(filenames)) {
  df <- read.delim(paste0("./results/Arlequin_HardyWeinberg/hwe_results_by_island_14032022/", filenames
  m <- as.vector(grep("This locus is monomorphic", df[,1], value = FALSE, fixed = TRUE))
  # making list of rows that only say the above words
  df <- as.data.frame(df[-c(1,m),]) # removing the rows listed above, plus the dashed line

  df <- as.data.frame(gsub("  ", " ", df[,1], fixed = TRUE)) # removing spaces
  df <- as.data.frame(gsub("  ", " ", df[,1], fixed = TRUE)) # removing spaces
  df <- as.data.frame(gsub("  ", " ", df[,1], fixed = TRUE)) # removing spaces

  colnames(df) <- "Var1"
  df <- tidyr::separate(df, sep = " ", col = Var1, into = c("foo", "Locus", "Genot", "Obs.Het", "Exp.He
  df <- df[,-1] # removing extra row
  for (ii in 1:ncol(df)) {df[,ii] <- as.numeric(df[,ii])} # converting to numeric
  assign(paste(filenames[i]), df) # renaming object
  # write.table(df, paste("df", filenames[i], sep = "_"), row.names = FALSE, sep = "\t") # save to file
} # END OF FOR LOOP

rm(i, ii, m, df, filenames)
# setwd("C:/Users/airhe/OneDrive/Documents/Masters/Project 3/kiore-project")
```

2b. Checking the HWE P-values

```
objectnames <- as.vector(ls()) # should be islands only, otherwise remove extras from vector
objectnames
```

```
##  [1] "aotea.txt"                 "borneo.txt"
##  [3] "copy"                      "data"
##  [5] "doubtful_sound.txt"        "great_mercury_island.txt"
##  [7] "halmahera.txt"             "hatutaa.txt"
##  [9] "honuea.txt"                "kaikura_island.txt"
## [11] "kamaka.txt"                "kayangel.txt"
## [13] "late_island.txt"           "mainland.txt"
## [15] "malenge.txt"               "mohotani.txt"
## [17] "motukawanui.txt"           "new_britain.txt"
## [19] "new_guinea.txt"            "normanby_island.txt"
## [21] "rakiura.txt"               "reiono.txt"
## [23] "rimatuu.txt"               "slipper_island.txt"
## [25] "sulawesi.txt"              "tahanea.txt"
## [27] "wake_island.txt"
```

```
# If necessary:
objectnames <- objectnames[-c(3, 4)] # removing non-island objects, may not be the same numbers

# making df of all hwe results
hwe.all <- data.frame()
for (i in 1:length(objectnames)) {
  foo <- get(objectnames[i])
  islandpop <- c(rep(paste(objectnames[i]), paste(nrow(foo)))) # making a vector of the popn. name
  foo$islandpop <- islandpop # adding the column to the results df to identify popn.
  hwe.all <- rbind(hwe.all, foo) # adding the popn. df to the combined hwe results df
}

rm(islandpop, foo, i)
rm(list = objectnames) # removes all the island objects
```

2c. Running Holm's Sequential Bonferroni test to adjust p-values

```
nrow(hwe.all) # 2622
```

```
## [1] 2622
```

```
p.value.adjusted <- c(p.adjust(hwe.all$P.value, method = "holm")) # adjusting p-values
hwe.all$p.value.adjusted <- p.value.adjusted # making new column

rm(p.value.adjusted)
# getwd()
# write.csv(hwe.all, "./results/Arlequin_HardyWeinberg/HWEanalysis_allresults.csv", row.names = FALSE)
```

2d. Examining significant hwe p-values

```r
hwe.all <- read.csv("./results/Arlequin_HardyWeinberg/HWEanalysis_allresults.csv")
hwe.signif <- hwe.all[which(hwe.all$p.value.adjusted <= 0.05),]
kable(hwe.signif[,c(1,8,9)])
```

	Locus	islandpop	p.value.adjusted
598	127	honuea.txt	0.02613
608	182	honuea.txt	0.00000
637	41	kaikura_island.txt	0.02613
825	16	kayangel.txt	0.02613
836	37	kayangel.txt	0.02613
845	50	kayangel.txt	0.02613
847	54	kayangel.txt	0.02613
850	61	kayangel.txt	0.02613
851	62	kayangel.txt	0.00000
854	67	kayangel.txt	0.00000
862	84	kayangel.txt	0.00000
868	93	kayangel.txt	0.02613
881	122	kayangel.txt	0.00000
906	170	kayangel.txt	0.02613
907	171	kayangel.txt	0.02613
909	177	kayangel.txt	0.00000
924	211	kayangel.txt	0.02613
940	252	kayangel.txt	0.02613
946	266	kayangel.txt	0.00000
953	278	kayangel.txt	0.00000
1951	107	rakiura.txt	0.02613
1961	128	rakiura.txt	0.02613
2031	41	reiono.txt	0.00000

```r
kable(count(hwe.signif$Locus)) # 2 at locus 41 (Kaikura and Reiono), rest are singles
```

x	freq
16	1
37	1
41	2
50	1
54	1
61	1
62	1
67	1
84	1
93	1
107	1
122	1
127	1
128	1
170	1
171	1
177	1
182	1

x	freq
211	1
252	1
266	1
278	1

```
kable(count(hwe.signif$islandpop)) # concerning that 17 of 23 are Kayangel
```

x	freq
honuea.txt	2
kaikura_island.txt	1
kayangel.txt	17
rakiura.txt	2
reiono.txt	1

3. Removing samples/loci with issues identified in HWE and Structure analyses

```
# getwd()
halfclean <- read.csv("./data/RStudio/ratsSNPs_halfclean.csv")

# need to remove Kamaka_009
# not removing Rimatuu_19 and Rimatuu_20 because they are right next to Reiono
# and could have swam.
# both the (pre-cleanup) NeighborNet and Structure identify Kayangel17 as concerning,
# as well as Kayangel11, 13, 15, 19, and 21. HWE shows also several loci in Kayangel
# as problematic, but not in other popn.s (with 1 exception). Since the loci are
# only problematic in Kayangel popn., I believe the issue is in the specimens, not
# the loci themselves.

remove <- c(
    "Kamaka_009",
    "Rimatuu_19", # actually should re-do analysis keeping the rimatuu's!
    "Rimatuu_20",
    "Kayangel11",
    "Kayangel13",
    "Kayangel15",
    "Kayangel17",
    "Kayangel19",
    "Kayangel21"
  ) # names of specimens to remove (each name should have exactly 10 characters)
x <- sapply(remove, function(i) grep(i, x = halfclean$island, value = FALSE))
# finding the row numbers of the specimens to remove

halfclean[c(x),1] # checking the names match
```

```
## [1] "Kamaka_009" "Rimatuu_19" "Rimatuu_20" "Kayangel11" "Kayangel13"
## [6] "Kayangel15" "Kayangel17" "Kayangel19" "Kayangel21"
```

```
clean <- halfclean[-c(x),] # removing rows described above

# getwd()
# write.csv(clean, "./data/RStudio/ratsSNPs_clean.csv", row.names = FALSE)

rm(x, remove)
```

4. Double checking for monomorphic columns (SNP loci)

```
ncol(clean) #298
```

```
## [1] 298
```

```
monocols <- integer() # empty vector for the for loop
for (i in 17:298) {
  z <- length(unique(clean[,i])) # no. of unique values in the row (looking for 1, or 2 if there's "?")
    if (z <= 3)
      {monocols <- append(monocols, i) # if z is as so, add the column number to the vector
    }
  rm(z)
}
# tried with z <= 2 but no result, therefore tried z <= 3 and checked the results manually below.

monocols
```

```
##  [1]  29  30  42  43  52  54  76  84  86  89 105 113 115 123 127 136 149 154 155
## [20] 156 170 179 182 183 200 204 205 209 211 214 216 222 223 224 230 235 243 290
## [39] 295 296
```

```
for (i in monocols) {
  print(unique(clean[,i]))
}
```

```
## [1] "G:G" "?"   "C:G"
## [1] "T:T" "C:T" "C:C"
## [1] "C:C" "?"   "C:T"
## [1] "?"   "G:G" "A:G"
## [1] "A:G" "G:G" "?"
## [1] "C:C" "C:T" "T:T"
## [1] "?"   "A:G" "G:G"
## [1] "?"   "C:C" "C:T"
## [1] "?"   "G:G" "T:G"
## [1] "?"   "G:G" "A:G"
## [1] "G:G" "?"   "A:G"
## [1] "?"   "T:T" "A:T"
## [1] "T:T" "?"   "C:T"
## [1] "A:A" "?"   "G:G"
## [1] "C:C" "?"   "C:T"
## [1] "?"   "C:C" "C:T"
## [1] "A:G" "G:G" "?"
## [1] "C:C" "?"   "T:T"
## [1] "C:C" "?"   "A:C"
## [1] "?"   "C:C" "T:T"
## [1] "A:A" "A:G" "G:G"
## [1] "G:G" "?"   "A:G"
## [1] "T:T" "C:T" "C:C"
## [1] "G:G" "?"   "A:A"
## [1] "A:A" "A:G" "G:G"
## [1] "A:A" "A:G" "G:G"
## [1] "A:A" "?"   "A:C"
## [1] "?"   "C:C" "A:C"
## [1] "A:A" "?"   "A:C"
## [1] "?"   "T:T" "A:T"
## [1] "?"   "G:G" "T:G"
## [1] "?"   "G:G" "A:G"
## [1] "?"   "G:G" "C:G"
## [1] "?"   "G:G" "A:G"
## [1] "G:G" "A:G" "A:A"
## [1] "T:T" "C:T" "C:C"
## [1] "?"   "G:G" "A:G"
## [1] "C:C" "C:G" "G:G"
## [1] "A:A" "?"   "A:G"
## [1] "C:C" "?"   "C:T"
```

```
# none with only 1 unique SNP in each column. It's possible since the SNP loci were selected for their
```

```
# x <- x[,-c(monocols)] # for removal of monomorphic columns, but none found

rm(i, monocols)
```

5. Summary: specimens per Island after full data clean-up

```
kable(count(clean$island.1))
```

x	freq
Aotea (Great Barrier I)	10
Borneo	18
Doubtful Sound	1
Great Mercury Island	1
Halmahera	12
Hatutaa	21
Honuea	20
Kaikura Island	20
Kamaka	20
Kayangel	15
Late Island	21
Luzon	1

x	freq
Mainland	5
Malenge	12
Mohotani	14
Motukawanui	21
New Britain	10
New Guinea	3
Normanby Island	1
Rakiura (Stewart Isl)	21
Reiono	21
Rimatuu (Tetiaroa)	19
Slipper Island	21
Sulawesi	22
Tahanea	20
Wake Island	20

Island	freq before cleanup	freq after cleanup	difference
Aotea (Great Barrier I)	10	10	0
Borneo	25	18	7
Doubtful Sound	1	1	0
Great Mercury Island	1	1	0
Halmahera	25	12	13
Hatutaa	21	21	0
Honuea	21	20	1
Kaikura Island	20	20	0
Kamaka	21	20	1
Kayangel	21	15	6
Late Island	21	21	0
Luzon	1	1	0
Mainland	5	5	0
Malenge	25	12	13
Mohotani	14	14	0
Motukawanui	21	21	0
New Britain	26	10	16
New Guinea	25	3	22
Normanby Island	25	1	24
Rakiura (Stewart Isl)	21	21	0
Reiono	21	21	0
Rimatuu (Tetiaroa)	21	21	0
Slipper Island	21	21	0
Sulawesi	25	22	3
Tahanea	20	20	0
Wake Island	20	20	0

Islands represented by very few specimens (<= 3) are Doubtful Sound, Great Mercury Island, Luzon, New Guinea, and Normanby Island.