Cleaning the Polynesian Rat SNP raw data file

Grace Saville

11/02/2022

Preamble

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

Loading the data

```
data <- read.delim("./data/Genotyping-007.010-01_SNP_Raw_data.tsv")
dim(data)  #478 rows (specimens), 333 columns (SNP loci)
data[1, 1:17]  # SNP data in columns 17 to 333
```

```
class(data[5, 17])  # character
count(data$island.1)  #  how many samples from each island there are
data[data$island.1 == "", ]  # checking why 2 'island.1' cells are blank
data[c(471, 473), 1:10]  # the blanks are from Laos and Cambodia, therefore replacing the blanks with '
data[471, "island.1"] <- "Mainland"
data[473, "island.1"] <- "Mainland"

x <- data  # keeping 'data' as backup original
```

Tidying SNP order

I'm doing this to make R evaluation easier (e.g when checking for counts it does not count A:G and G:A separately)

```
dim(x)  # 333 cols
count(unlist(x[, 17:333]))
x[x == "T:A"] <- "A:T"
x[x == "C:A"] <- "A:C"
x[x == "G:A"] <- "A:G"
x[x == "T:C"] <- "C:T"
x[x == "G:C"] <- "C:G"
x[x == "G:T"] <- "T:G"
count(unlist(x[, 17:333]))  # checking success
```

Specimens per Island before data clean-up

```
count(data$island.1)
```

Island	freq
Aotea (Great Barrier I)	10
Borneo	25
Doubtful Sound	1
Great Mercury Island	1
Halmahera	25
Hatutaa	21
Honuea	21
Kaikura Island	20
Kamaka	21
Kayangel	21
Late Island	21
Luzon	1
Mainland	5
Malenge	25
Mohotani	14
Motukawanui	21
New Britain	26
New Guinea	25
Normanby Island	25
Rakiura (Stewart Isl)	21
Reiono	21
Rimatuu (Tetiaroa)	21
Slipper Island	21
Sulawesi	25
Tahanea	20
Wake Island	20

Removing SNP columns with no variation (invariant/monomorphic)

```
ncol(x)  #333
monocols <- integer()  # empty vector for the for loop
for (i in 17:333) {
    z <- length(unique(x[, 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  # 17  34  73  80  88  95  98 101 102 108 119 129 139 154 156 171 176 177 178 179 194 203 207 
for (i in monocols) {
```

```
    print(count(x[, i]))
}
# none with only 1 unique SNP in each column ...? It's
# possible since the SNP loci were selected for their
# differences, but double check this
```

```
# x <- x[,-c(monocols)] # for removal of monomorphic
# columns

rm(i, monocols)
```

Removing columns (SNPs) with few samples

```
ncol(x)  #333
percblank <- integer()  # empty df for the for loop
for (i in 17:333) {
    y <- count(grepl("?", x[, i], fixed = TRUE))  # finds and counts freq of ?
    z <- signif((nrow(x) - y[1, 2])/nrow(x) * 100, 4)  # percentage of ? in the column, to 4 signif dig
    if (z > 60) {
        percblank <- append(percblank, i)
    }
    rm(z)
    rm(y)
}

percblank  # 17  18  19  25  48  65  69  73  80  88  89  96 102 108 131 133 146 147 156 159 162 165 179
# checking: count(x[,212]) 320/478
```

```
x <- x[, -c(percblank)]  # removing columns listed above, with more than 60% missing data
ncol(x)  #298

rm(i, percblank)
```

Removing rows (specimens) with few samples

```
x2 <- data.table::transpose(x)  # transposing the df temporarily since count() doesn't work well on row

ncol(x2)  #478 specimens
percblank <- integer()  # empty df for the for loop
for (i in 1:478) {
    y <- count(grepl("?", x2[, i], fixed = TRUE))  # finds and counts freq of '?'
    z <- signif((nrow(x2) - y[1, 2])/nrow(x2) * 100, 4)  # percentage of ? in the specimen, to 4 signif
    # I used the no. rows-false outcomes instead of the
    # true outcomes because some rows have no '?' and
    # result in errors.
    if (z > 56)
        # 56 percent missing allowed because it gives 90%
```

```
        # completeness (see below)
    {
        percblank <- append(percblank, i)
    }
    rm(z)
    rm(y)
}

percblank  # 1   7   9  10  11  18  25  48  49  50  51  52  53  55  56  57  58  59  60  62  63  65  66
# checking work: count(x2[17:298,171]) 185/298
```

```
x <- x[-c(percblank), ]  # removing the rows listed (percblank) that have too many '?' from the df
nrow(x)  #379
```

```
# checking the % of all '?'s in the df:
z <- count(grepl("?", unlist(x), fixed = TRUE))
signif(z[2, 2]/(z[1, 2] + z[2, 2]) * 100, 4)  # 9.723% '?'
100 - 9.723  # 90.277% complete df, ideal point where there is more than 90% completeness but not too m

rm(i, percblank, x2, z)
getwd()
save(list = ls(all = TRUE), file = ".RData")  # save RDATA for later use if necessary
write.csv(x, "./data/ratsSNPs_halfclean.csv", row.names = FALSE)
```

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 function.

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

```
x2 <- x  # making a copy
x2[1, 1:20]  # checking column names
x2 <- x2[, -c(2:16)]  # removing all but specimen names and SNPs
x2[1, 1:20]  # checking

x2[x2 == "?"] <- "?:?"  # replacing single ? with double ? so alleles can be split

x3 <- data.frame(island = x2$island)  # setting up new df for for loop
coln <- as.vector(colnames(x2))  # prepping to paste the column names into the for loop
dim(x2)  # 379 rows 283 columns
for (i in 2:283) {
    y <- colsplit(x2[, i], split = ":", names = c(coln[i], paste("blank",
        i, sep = ".")))  # splitting each i column and renaming them
    x3 <- cbind(x3, 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

x2 <- x3
rm(x3, coln)  # removing excess objects
```

Producing the file necessary for PGDSpider program

```
x2 <- x2[order(x2$island, decreasing = FALSE), ] # ordering df alphabetically by island
print(as.matrix(x2[, 1])) # printing the island names and row numbers

# A=1, T=2, G=3, C=4
x2[x2 == "A"] <- "1"
x2[x2 == "T"] <- "2"
x2[x2 == "G"] <- "3"
x2[x2 == "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
    "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(x2[1:10,]) # Aotea
b <- as.data.frame(x2[11:28,]) # Borneo
c <- as.data.frame(x2[315,]) # Doubtful_Sound
d <- as.data.frame(x2[30,]) # Great_Mercury_Island
e <- as.data.frame(x2[31:42,]) # Halmahera
f <- as.data.frame(x2[43:63,]) # Hatutaa
g <- as.data.frame(x2[64:83,]) # Honuea
h <- as.data.frame(x2[84:103,]) # Kaikura_Island
i <- as.data.frame(x2[104:124,]) # Kamaka
j <- as.data.frame(x2[125:145,])  # Kayangel
k <- as.data.frame(x2[148:168,]) # Late_Island
l <- as.data.frame(x2[c(29, 146, 147, 169, 358, 359),]) # Mainland
m <- as.data.frame(x2[170:181,]) # Malenge
n <- as.data.frame(x2[182:195,]) # Mohotani
o <- as.data.frame(x2[196:216,]) #  Motukawanui
p <- as.data.frame(x2[217:226,]) #  New_Britain
q <- as.data.frame(x2[227:229,]) # New_Guinea
r <- as.data.frame(x2[230,]) #  Normanby_Island
s <- as.data.frame(x2[231:251,]) # Rakiura
t <- as.data.frame(x2[252:272,]) # Reiono
u <- as.data.frame(x2[273:293,]) # Rimatuu
v <- as.data.frame(x2[294:314,]) # Slipper_Island
w <- as.data.frame(x2[316:337,]) #  Sulawesi
# x already in use
y <- as.data.frame(x2[338:357,]) # Tahanea
z <- as.data.frame(x2[360:379,]) # Wake_Island

pops <- as.character(c(letters[seq(from = 1, to = 23)], "y", "z")) # list of popn object names
```

```
ncol(x2)  #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(x2[, c(FALSE, TRUE)]), "\n", sep = "\t\t", fill = FALSE)  # column/SNP names (even columns
for (i1 in 1:25) {
    # outer loop
    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)) {
        # inner loop
        cat(as.character(foo[i2, ]), "\n", fill = FALSE, sep = "\t")  # printing the 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 = 23)], "y", "z"))  # removing excess objects
```

- check which specimens will be in the mainland popn. (e.g. luzon, also Borneo and Sulawesi), and if populations w 1 specimen are viable (e.g. doubtful sound and great mercury isl.)

HWE Analysis and removal

Creating loop for reading the HWE files

```
getwd()
setwd("./results/arlequin_results/hwe_results_by_island_14032022")
filenames <- as.vector(list.files())

for (i in 1:length(filenames)) {
    df <- read.delim(filenames[i])
    m <- as.vector(grep("This locus is monomorphic", df[, 1],
        value = FALSE, fixed = TRUE))  # making list of rows that only say these 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.Het", "P.value", "s.d",
        "Steps.done"))  # splitting column into multiple (with a extra column because of the space at t
    df <- df[, -1]  # removing extra row
    for (ii in 1:ncol(df)) {
        df[, ii] <- as.numeric(df[, ii])
    }  # converting to numeric rather than character/factor
    assign(paste(filenames[i]), df)  # renaming object
    # write.table(df, paste('df', filenames[i], sep = '_'),
    # row.names = FALSE, sep = '\t') # save to file
}
```

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

Checking the HWE P-values

```
objectnames <- as.vector(ls())  # should be islands only, otherwise remove extras from vector
objectnames
# If necessary: objectnames <- objectnames[-c(3, 27, 28)] #
# removing non-island objects

# 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
```

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

```
nrow(hwe.all)  # 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, "./data/HWEanalysis_allresults.csv", row.names = FALSE)
```

Examining significant hwe p-values

```
hwe.all <- read.csv("./data/HWEanalysis_allresults.csv")
hwe.signif <- hwe.all[which(hwe.all$p.value.adjusted <= 0.05),
    ]
hwe.signif[, c(1, 8, 9)]
plyr::count(hwe.signif$Locus)  # 2 at locus 41 (Kaikura and Reiono), rest are singles
plyr::count(hwe.signif$islandpop)  # concerning that 17 of 23 are Kayangel
```

Adjusted p-values per Locus

Locus	islandpop	p.value.adjusted
127	honuea	0.02613
182	honuea	0.00000
41	kaikura_island	0.02613
16	kayangel	0.02613
37	kayangel	0.02613
50	kayangel	0.02613
54	kayangel	0.02613
61	kayangel	0.02613
62	kayangel	0.00000
67	kayangel	0.00000
84	kayangel	0.00000
93	kayangel	0.02613
122	kayangel	0.00000
170	kayangel	0.02613
171	kayangel	0.02613
177	kayangel	0.00000
211	kayangel	0.02613
252	kayangel	0.02613
266	kayangel	0.00000
278	kayangel	0.00000
107	rakiura	0.02613
128	rakiura	0.02613
41	reiono	0.00000

Number of islands with a significant adjusted p-value at a particular locus

loci column	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
211	1
252	1
266	1
278	1

Number of loci per island population with significant adjusted p-values

island population	no. of signif. loci
honuea	2
kaikura_island	1
kayangel	17
rakiura	2
reiono	1

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

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

# need to remove Kamaka_008, and Rimatuu_19 and Rimatuu_20
# due to position in Structure.  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_008", "Rimatuu_19", "Rimatuu_20", "Kayangel11",
    "Kayangel13", "Kayangel15", "Kayangel17", "Kayangel19", "Kayangel21")  # names of specimens to remo
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
clean <- halfclean[-c(x), ]  # removing rows described above

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

rm(x, remove)
```

Double checking for monomorphic columns (SNP loci)

```
ncol(clean)  #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  # 29  30  43  52  54  76  84  86  89 105 113 115 123 127 136 149 154 155 156 170 179 182 183 
for (i in monocols) {
    print(count(clean[, i]))
}
# none with only 1 unique SNP in each column. It's possible
# since the SNP loci were selected for their differences,
# but double check this
```

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

rm(i, monocols)
```

Specimens per Island after data clean-up

```
plyr::count(clean$island.1)
```

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	19	2
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.