Sandbox

Grace Saville

01/10/2021

Preamble

```
# Installing phylogenetics packages
# install.packages('ctv')
# library('ctv')
# install.views('Phylogenetics')
# update.views('Phylogenetics')
library(plyr)
library(ape)
library(phylotools)
library(stringr)
library(dplyr)
```

Looking at the data

```
data <- read.delim("Genotyping-007.010-01_copy2.txt")
dim(data) #478 rows, 333 columns
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 (function not working anymore??)
count(data[,206]) # how many of each base combination there are in a SNP column
```

Making a subset

```
sub <- data[1:10,17:30] # making subset to experiment with
str(sub)
# sub[sub=="?"] <- NA # not sure if I want to replace the ?'s with NA's
summary(sub)
str(sub)
```

Prep for making tree

```
x <- data.frame(a = c("asdfghjkl1", "asdf2ghjkl", "asdf3ghjkl", "asdfghjkl4"), b = c("CTAGTGACCCGTAG","
dat2phylip(x, outfile = "test.phy") # testing this function that saves phylip files
# after some experimenting, spaces in the base strings aren't necessary, but 10 character length names
```

IUPAC Ambiguity code:

Symbol	SNP bases
A	AA
T	TT
C	CC
G	GG
R (purine)	AG
Y (pyrimidine)	CT
W (weak)	AT
S (strong)	CG
M (amino)	AC
K (keto)	GT

```
y <- data # making a copy
# changing each SNP base combination to a single letter code
# y[y=="?"] <- "-" # "?" also accepted by splitstree
y[y=="A:A"] <- "A"
y[y=="T:T"] <- "T"
y[y=="C:C"] <- "C"
y[y=="G:G"] <- "G"
y[y=="A:G"] <- "R"
y[y=="G:A"] <- "R"
y[y=="C:T"] <- "Y"
y[y=="T:C"] <- "Y"
y[y=="A:T"] <- "W"
y[y=="T:A"] <- "W"
y[y=="C:G"] <- "S"
y[y=="G:C"] <- "S"
y[y=="A:C"] <- "M"
y[y=="C:A"] <- "M"
y[y=="G:T"] <- "K"
y[y=="T:G"] <- "K"

names(y)
y <- y[,-c(2:16)] # removing the rows with information other than the species key and the code above
y <- tidyr::unite(y, bases, -1, sep = "", remove = TRUE) # merging all the base columns into one column
y[,1] <- gsub("NBC.LAB.", "NBCLAB", y[,1]) # replacing part of the name of the rowname column so there

dat2phylip(y, outfile = "test2.phy")
# something wrong with the file when opened in splitstree, e.g. illegal characters or two of the same s
summary(duplicated(y[,1])) # no. 40 row name is duplicated
# duplicated(y[,1], fromLast = FALSE)
# duplicated(y[,1], fromLast = TRUE) # no. 39 is duplicated w no. 40
# summary(duplicated(y)) # checking row 40 doesn't have the same code string as well
# y[39:40,1] # "NBCLAB2005" "NBCLAB2005" need to replace one
# tail(y[,1], 20) # last specimen is labelled NBCLAB2436, will rename duplicate 2437
# y[40,1] <- "NBCLAB2437" # replacing duplicate w new name
# summary(duplicated(y[,1])) # double checking
# y[,1] # last 8 row names don't have 10 characters: "L0235" "L0234" "C0910" "GMI-4" "P0041" "R14018" "
# y[471,1] <- "L023500000" # replacing w new names
# y[472,1] <- "L023400000"
# y[473,1] <- "C091000000"
# y[474,1] <- "GMI0400000"
# y[475,1] <- "P004100000"
# y[476,1] <- "R140180000"
# y[477,1] <- "R675000000"
# y[478,1] <- "R712900000"
dat2phylip(y, outfile = "data.phy") # can be opened by splitstree when configured as proteins/amino aci
```

Try writing directly to a .nex file since splitstree only reads the .phy file as protein:

```
y2 <- as.matrix(y)
ape::write.nexus.data(y2, "nextestfile.nex", format = "dna") #doesn't format correctly, try again later
```

Mantel test prep

uses lat/long data, there are geospatial R packages that give exact distances on the globe between lat/long points which will be useful here

```
# install.packages("ade4") # I think this can be used for the mantel test
# install.packages("geosphere") # calculates distances between 2 points on a sphere
library(ade4)
library(geosphere)

# "need to generate two distance matrices: one containing spatial distances and one containing distance
# distHaversine() # assumes earth is a sphere
# distm() # makes distance matrix
# distGeo() # assumes earth is elliptical ish, can choose specific model

df <- data # make sure rows are samples and columns are variables
#longitude and latitude
names(df) # need "geo_lat" "geo_long"
df <- df[,c(1,11,12)]
names(df)
longlat <- as.matrix(df[,c(3,2)]) # distGeo function needs a matrix with 2 columns, col 1 longitude and
distGeo(longlat[c(1,12),]) # gives distance between the two rows

geo.dist <- distm(longlat, fun = distGeo)
dim(geo.dist) # 478 478


# genetic distances
getwd()
gen.dist <- read.delim("geneticdist.txt", sep = "\t", header = FALSE)
# ncol(gen.dist) #478
# gen.dist <- read.delim("geneticdist.txt", sep = "\t", col.names = paste0("V", seq_len(478)))
gen.dist <- as.matrix(gen.dist)
```

```
dim(gen.dist) # 114482    478 (needed to remove an extra text section from the matrix file in notepad)


#change matrix to distance matrix in R for mantel test
# col names and row names

# making a dummy matrix to test the function as.dist()
test <- matrix(data = c(0,3,6,9,4,
                        3,0,9,8,2,
                        6,9,0,4,2,
                        9,8,4,0,4,
                        4,2,2,4,0), nrow = 5, ncol = 5)
test <- as.dist(test,
        diag = TRUE, #include diagonal zeros
        upper = TRUE) #include upper triangle
test #works as expected, need to see why the function works strangely on my dataset

# subsetting one of my matrices
a <- gen.dist[1:5,1:5]
a
a2 <- as.dist(a, diag = TRUE, upper = TRUE)
a2


gen.dist <- as.dist(gen.dist, diag = TRUE, upper = TRUE)
gen.dist[1:5]
geo.dist <- as.dist(geo.dist, diag = TRUE, upper = TRUE)
geo.dist[20:30]

# Mantel test
mantel.rtest(gen.dist, geo.dist)
# Monte-Carlo test
# Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)

# Observation: 0.2807331 # genetic distance and geographic distance positively correlated

# Based on 99 replicates
# Simulated p-value: 0.01
# Alternative hypothesis: greater
#       Std.Obs   Expectation      Variance
# 26.5367570885 -0.0002961658  0.0001121521

# Plots
plot(geo.dist, gen.dist)

x <- data.frame(as.vector(gen.dist), as.vector(geo.dist))
colnames(x) <- c("Genetic_Distance", "Geographic_Distance")
library(ggplot2)
ggplot(data = x, aes(x = Geographic_Distance, y = Genetic_Distance)) + geom_point(shape = 1, colour = "
```

Heterozygousity experimenting

```
x <- c("A:A", "G:T", "C:C", "A:T", "T:A","T:T", "A:C", "C:A") # test vector
hetz <- c("A:G", "G:A", "C:T", "T:C", "A:T", "T:A", "C:G", "G:C", "A:C", "C:A", "G:T", "T:G") # vector
homz <- c("G:G", "A:A", "C:C", "T:T") # vector of homozygous combinations

# testing to see if I can identify what values are homozygous and heterozygous within an example vector
for (i in x){
  if (i %in% hetz){print("E")}
  else {print("O")}
}

# testing if I can replace the values in the above example vector
for (i in 1:length(x)){
  if (x[i] %in% hetz){x[i] <- "E"}
  else {x[i] <- "O"}
}

# trying another method for the subset dataframe:
for (i in 1:nrow(sub)){
  sub[i,][sub[i,] %in% hetz] <- "E"
  sub[i,][sub[i,] %in% homz] <- "O"
}

# applying above for loop to entire df
z <- data # making a copy of the data
for (i in 1:nrow(z)){
  z[i,][z[i,] %in% hetz] <- "E"
  z[i,][z[i,] %in% homz] <- "O"
}

count(z[,105]) #column count
sum(str_count(z[35,], "E")) # row count

# could also try 0 for homo and 1 for hetero, therefore the lower the totals the lower the heterozygous
z2 <- data # making a copy of the data
for (i in 1:nrow(z2)){
  z2[i,][z2[i,] %in% hetz] <- 1
  z2[i,][z2[i,] %in% homz] <- 0
}

z2[z2=="?"] <- NA # replacing ? with NA's
for (i in 16:332) {class(z2[,i]) <- "numeric"} # changed the class to numeric for each column so the to

z2 <- rowwise(z2) # making the df rowwise so i can make calculations by row easier
names(z2)
z2 <- mutate(z2, total = sum(c_across(X11_CHR1_101004452:X154_CHR9_95290356), na.rm = TRUE))
z2[,333]

# trying this method of adding row totals at the end to the E/O df
z <- rowwise(z)
summarise(z)
mutate(z, Etotal = sum(str_count(z, "E")))
```

```r
sum(str_count(z[35,], "?")) # row count
```