

Mantel Test Script

Grace Saville

01/02/2022

Preamble

```
library(ade4)
library(geosphere)
library(ggplot2)
getwd()
setwd("C:/Users/airhe/OneDrive/Documents/Masters/Project 3/kiore-project")
```

Loading the data

```
data <- read.csv("../data/RStudio/ratsSNPs_clean.csv")
gen.matrix <- as.matrix(read.delim("../data/SplitsTree/geneticdist_SplitsTree_output.txt", sep = "\t", header = TRUE))
# Make sure the text file is the distances only, remove any extras e.g. column vector at bottom of file
```

Creating geographical distance matrix

Using the mapview funtion in the last code chunk of this script I noticed that Tahanea is plotted in Australia, and found out the longitude number is missing a negative sign, and that Reiono and Honuea have the same coordinates even though they're different islands. Fixing that here:

```
x <- grep("Tahanea", data$island.1, value = FALSE) # finding rows of Tahanea
names(data) # finding column number for "geo_long"
data[x, 12]
data[x, 12] <- -144.97 # replacing the number

# the coordinates given for Reiono and Honuea point to an island in French Polynesia called Moorea-Maia
# Reino should be approx. -17.046, -149.546
# Honuea should be approx -17.009, -149.585
# Checking other islands close by:
# Rimatu'u is -17, -149.57 (over the sea) but should be approx -17.03, -149.558

x <- grep("Reiono", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.046 # replcing geo_lat
data[x, 12] <- -149.546 # replacing geo_long
```

```

x <- grep("Honuea", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.009 # replcing geo_lat
data[x, 12] <- -149.585 # replacing geo_long

x <- grep("Rimatuu", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.03 # replcing geo_lat
data[x, 12] <- -149.558 # replacing geo_long

rm(x)

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

```

(only need to do the above code once since it saves the edited df to file)

Different distance functions:

- distHaversine() assumes earth is a sphere
- distm() makes distance matrix
- distGeo() assumes earth is elliptical (ish), can choose specific model

```

names(data) # need "geo_lat" "geo_long"
longlat <- data[,c(1,11,12)]
head(longlat) # checking correct columns are used

longlat <- as.matrix(longlat[,c(3,2)]) # distGeo function needs a matrix with 2 columns, col 1 longitud

geo.matrix <- distm(longlat, fun = distGeo) # converting to pairwise distance matrix
dim(geo.matrix) # 370 370

geo.dist <- as.dist(geo.matrix, diag = TRUE, upper = TRUE) # converting to dist object
# diag = TRUE #includes diagonal zeros
# upper = TRUE #includes upper triangle

```

Creating genetic distance matrix

```

dim(gen.matrix) # 370 370
gen.dist <- as.dist(gen.matrix, diag = TRUE, upper = TRUE) # converting to dist object

```

Mantel test

```

mantel.rtest(gen.dist, geo.dist, nrepet = 999)

```

Results:

Monte-Carlo test

Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)

Observation: 0.4987612

Based on 999 replicates

Simulated p-value: 0.001

Alternative hypothesis: greater

Std.Obs: 23.7703702908

Expectation: -0.0004626768

Variance: 0.0004410815

- -1 suggests strong negative correlation, e.g. closer islands mean further genetically or further islands means closer genetically
- 0 suggests no correlation, e.g. genetic difference is not correlated to island distance
- 1 suggests strong positive correlation e.g. closer islands mean closer genetically

Therefore the observed correlation of 0.4987612 suggests that there is a positive correlation between genetic distance and geographic distance (and the null hypothesis of no correlation is rejected).

Results before I fixed the missing negative sign in Tahanea and other location issues:

Monte-Carlo test

Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)

Observation: 0.4345602

Based on 999 replicates

Simulated p-value: 0.001

Alternative hypothesis: greater

Std.Obs: 2.329675e+01

Expectation: 7.916798e-04

Variance: 3.466772e-04

Results from uncleaned dataset:

Monte-Carlo test

Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)

Observation: 0.2807331

Based on 99 replicates

Simulated p-value: 0.01

Alternative hypothesis: greater

Std.Obs: 26.5367570885

Expectation: -0.0002961658

Variance: 0.0001121521

Plots

```

specimens <- read.delim("../data/SplitsTree/geneticdist_SplitsTree_output_taxa_only.txt", header = FALSE)
specimens <- as.vector(specimens[,2])

colnames(gen.matrix) <- specimens
rownames(gen.matrix) <- specimens # naming the rows and columns by the order given in the SplitsTree ou

colnames(geo.matrix) <- data[,1]
rownames(geo.matrix) <- data[,1] # naming the rows and columns here by the order of lat/long, which cam

gen.matrix[lower.tri(gen.matrix)] <- NA # keeping only the upper triangle of each matrix
geo.matrix[lower.tri(geo.matrix)] <- NA

dist.summary <- data.frame(
  col = colnames(gen.matrix)[col(gen.matrix)],
  row = rownames(gen.matrix)[row(gen.matrix)],
  gen.dist = c(gen.matrix)
) # converting the genetic matrix into a df with columns describing which combos result in the distance

x <- data.frame(
  col = colnames(geo.matrix)[col(geo.matrix)],
  row = rownames(geo.matrix)[row(geo.matrix)],
  geo.dist = c(geo.matrix)
) # doing the same with the geographic matrix

dist.summary <- merge(dist.summary, x, by = 1:2, all = TRUE) # merging the 2 dfs
rm(x)

dist.summary <- tidyr::unite(dist.summary, islands.combo, 1:2, sep = ":", remove = TRUE) # combining th

dist.summary$geo.dist <- dist.summary$geo.dist/1000 # going from metres to km

dist.summary <- na.omit(dist.summary) # removing NA's that were in the bottom triangle

head(dist.summary, 30)

getwd()
write.csv(dist.summary, "../data/RStudio/gen_geo_distance_matrices_df.csv", row.names = FALSE)

```

The above df “dist.summary” is not ideal since it’s not space efficient (~12MB), however it provides an easy way to link the row and column specimens that generated the distances, therefore making points on the graph label-able.

```

z <- dist.summary
z[z == 0] <- NA
z <- na.omit(z)
# rows which have either or both gen and geo dist at zero will sit on the axis and potentially pull on

LM <- lm(gen.dist ~ geo.dist, data = z) # model
summary(LM) # model results
par(mfrow = c(2, 2)) # changing the number of plots visible at once
plot(LM) # diagnostic plots
par(mfrow = c(1, 1))
plot(log10(gen.dist + 1) ~ geo.dist, data = z)

```

```

abline(coef = coef(LM), col = 4, lwd = 2)
MASS::boxcox(LM)

LM <- lm(log10(gen.dist + 1) ~ geo.dist, data = z) # model
summary(LM) # model results
qqnorm(LM$residuals)
qqline(LM$residuals)
hist(LM$residuals)
ks.test(LM$residuals, 'pnorm')
par(mfrow = c(2, 2)) # changing the number of plots visible at once
plot(LM) # more diagnostic plots
par(mfrow = c(1, 1))
plot(log10(gen.dist + 1) ~ geo.dist, data = z)
abline(coef = coef(LM), col = 4, lwd = 2)
MASS::boxcox(LM)

GLM <- glm(gen.dist ~ geo.dist, data = z, family = Gamma(link = "inverse"))
summary(GLM) # model results

```

```

summary(dist.summary$gen.dist)

LM <- lm(gen.dist ~ geo.dist, data = dist.summary) # model
summary(LM) # model results
par(mfrow = c(2, 2)) # changing the number of plots visible at once
plot(LM) # diagnostic plots, not normal, distribution may be skewed towards the left side ("Right skew")

# Using these tests and plots to test normality in the variables and the model residuals:
qqnorm(LM$residuals)
qqline(LM$residuals)
hist(LM$residuals)
ks.test(LM$residuals, 'pnorm') # testing for normality, p-value < 0.05 t/f rejects H0 that the residuals

par(mfrow = c(1, 1))
MASS::boxcox(LM) # closest to 0, therefore could try log(x) transformation (alternatively exp(x))
plot(gen.dist ~ geo.dist, data = dist.summary)
abline(coef = coef(LM), col = 4, lwd = 2)

```

```

LM <- lm(log10(gen.dist + 1) ~ geo.dist, data = dist.summary) # model (with 1 added everywhere to avoid
summary(LM) # model results
par(mfrow = c(2, 2)) # changing the number of plots visible at once
plot(LM) # diagnostic plots, looks promising
par(mfrow = c(1, 1))
plot(log10(gen.dist + 1) ~ log10(geo.dist + 1), data = dist.summary)
abline(coef = coef(LM), col = 4, lwd = 2)

```

Results for log transformed model:

Residuals:

Min	1Q	Median	3Q	Max
-0.81095	-0.21637	-0.00449	0.17447	0.87210

Coefficients:

	Estimate	Std. Error	t value	p-value
(Intercept)	-1.559641	0.006220	-250.7	<2e-16 ***
log10(geo.dist)	0.215728	0.001731	124.6	<2e-16 ***

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

Residual standard error: 0.2644 on 65254 degrees of freedom

Multiple R-squared: 0.1922

Adjusted R-squared: 0.1922

F-statistic: 1.553e+04 on 1 and 65254 DF, p-value: < 2.2e-16

- Higher (or further from 0) t-value is generally more significant, related to p-value.
- Multiple R-squared indicates ~20% of the variance is explained by the model
- Significant p-value(s), can reject H0 where x and y are not correlated.
- F-test indicates if the variance that is explained by the model is significant, however it is sensitive to non-normality

Notes on diagnostic plots for log transformed model:

- Residuals vs. Fitted plot shows a horizontal almost flat t/f indicates the relationship is linear.
- Normal QQ plot shows residuals generally in line, indicating normality. A few points out of line at the lower end.
- Scale-Location plot is used to indicate constant residual variance with a line that does not trend up or down overall. Constant residual variance appears present.
- Residuals vs Leverage (Cook's distance) plot assesses outliers. None of the points are over 1 (Cook's distance line which would make them statistical outliers) nor are any over 0.5 (which would make them questionable).

```
ggplot(data = dist.summary, aes(x = geo.dist, y = gen.dist)) +
  geom_point(shape = 1, colour = "grey") +
  geom_smooth(method = "lm") +
  geom_quantile(quantiles = c(0.05, 0.95)) +
  ggtitle("Correlation between Geographic and Genetic Distance between Islands") +
  xlab("Geographic Distance between 2 islands (km)") + ylab("Genetic Distance between 2 islands") +
  theme_light()
# linear smooth regression line and 5/95 quantile lines added

# log transformed:
ggplot(data = dist.summary, aes(x = log10(geo.dist), y = log10(gen.dist))) +
  geom_point(shape = 1, colour = "grey") +
  geom_smooth(method = "lm") +
  ggtitle("Geographic versus Genetic Distance between Islands") +
  xlab("log10 Geographic Distance between 2 islands (km)") + ylab("log10 Genetic Distance between 2 islands") +
  theme_light()
```

Experimenting with plotting the coordinates on a map

```
names(data)
longlat <- data[,c(1,8,11,12)]
longlat <- longlat[!duplicated(longlat$island.1),] # keeping only 1 coordinate for each island
longlat <- longlat[,-1]
longlat

library(sf)
x <- st_as_sf(longlat, coords = c("geo_long", "geo_lat"), crs = 4326) # 4326 is WGS84 Coordinate Refer

library(mapview)
mapview(
  x, grid = FALSE, map.types = "Esri.WorldGrayCanvas",
  col.regions = c(
    "red",
    "grey40",
    "indianred1",
    "blue",
    "grey91",
    "darkorchid1",
    "moccasin",
    "darkorange1",
    "navy",
    "goldenrod1",
    "deeppink",
    "yellow",
    "slategray3",
    "yellowgreen",
    "chocolate4",
    "palegreen",
    "coral4",
    "green",
    "darkmagenta",
    "seagreen3",
    "lightsalmon",
    "darkgreen",
    "pink",
    "cyan",
    "plum",
    "peru"
  )
)
```

For the code above I would need to find an alternative to mapview because there appears to be a bug where the legend colours don't match up to the map point colours