ANI Distace vs. Shared OGs

John Quensen

9/20/2021

Question: What is the relationship between ANI distances between genomes and the number of OGs (orthologous genes) that they share? Make a scatter plot for all pairwise comparisons in Jose Rodrigues' data of cattle and clinical *Campylobacter* genomes.

Load packages.

```
suppressPackageStartupMessages(library(vegan))
suppressPackageStartupMessages(library(RDPutils))
suppressPackageStartupMessages(library(QsNullModels))
suppressPackageStartupMessages(library(tidyverse))
```

Include a function for converting a distance matrix into table form.

```
dist_pairs <- function(d) {
  m <- as.matrix(d)
  xy <- t(combn(colnames(m), 2))
  d.pair <- data.frame(xy, dist=m[xy])
  return(d.pair)
}
```

And another function for converting the OGS table into binary form. I wrote this one before for making the OGs heatmap plots.

```
ogs2bin <- function(ogs) {
  ogs.bin <- ogs
  ogs.bin[ogs.bin!="-"] <- 1
  ogs.bin[ogs.bin=="-"] <- 0
  rownames(ogs.bin) <- RDPutils::make_otu_names(1:nrow(ogs.bin))
  rownames(ogs.bin) <- sub("OTU", "OG", rownames(ogs.bin))
  return(ogs.bin)
}
```

Read in the ANI distance matrix and the ortholog table from the MiGA project.

```
load("D:/data/github/MiGA-Plus/extra_data/jose-miga-project.dist.rdata")
ogs <- read.table("extra_data/jose-miga-project.ogs",
                  header = TRUE,
                  stringsAsFactors = FALSE,
                  sep = "\t")
```

Convert the distance matrix into tabular form.

```
d.pairs <- dist_pairs(ani.d) %>%
  as_tibble()
d.pairs
```

```
## # A tibble: 40,755 x 3
```

```
##    X1      X2        dist
##    <chr>   <chr>    <dbl>
##  1 TW16361 TW16362 0.0216
##  2 TW16361 TW16370 0.0210
##  3 TW16361 TW16373 0.0223
##  4 TW16361 TW16374 0.0186
##  5 TW16361 TW16397 0.0202
##  6 TW16361 TW16398 0.0188
##  7 TW16361 TW16399 0.0218
##  8 TW16361 TW16400 0.0218
##  9 TW16361 TW16401 0.0238
## 10 TW16361 TW16402 0.0189
## # ... with 40,745 more rows
```

This gives the proper number of rows; the combination of 286 genomes taken 2 at a time is 40,755.

Convert the OGs table into binary form.

```
ogs.bin <- ogs2bin(ogs) %>%
  as_tibble()
ogs.bin
```

```
## # A tibble: 10,594 x 286
##    TW16361 TW16636 TW16689 TW16693 TW16694 TW16728 TW16735 TW19114 TW19128
##    <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
##  1 1       1       1       1       1       1       1       1       1
##  2 1       1       1       1       1       1       1       1       1
##  3 1       1       1       1       1       1       1       1       1
##  4 1       1       1       1       1       1       1       1       1
##  5 1       1       1       1       1       1       1       1       1
##  6 1       1       1       1       1       1       1       1       1
##  7 1       1       1       1       1       1       0       1       1
##  8 1       1       1       1       1       1       0       1       1
##  9 1       1       1       1       1       1       1       1       1
## 10 1       1       1       1       1       1       1       1       1
## # ... with 10,584 more rows, and 277 more variables: TW19129 <chr>,
## #   TW19130 <chr>, TW19131 <chr>, TW19133 <chr>, TW19246 <chr>, TW19248 <chr>,
## #   TW19249 <chr>, TW19256 <chr>, TW19257 <chr>, TW19258 <chr>, TW19259 <chr>,
## #   TW19262 <chr>, TW19263 <chr>, TW19265 <chr>, TW19266 <chr>, TW19267 <chr>,
## #   TW19269 <chr>, TW19278 <chr>, TW19292 <chr>, TW19293 <chr>, TW19302 <chr>,
## #   TW19312 <chr>, TW19313 <chr>, TW19314 <chr>, TW19315 <chr>, TW19316 <chr>,
## #   TW19317 <chr>, TW19318 <chr>, TW19319 <chr>, TW19320 <chr>, TW19321 <chr>,
## #   TW19322 <chr>, TW19323 <chr>, US0782 <chr>, US0809 <chr>, US0857 <chr>,
## #   US0886 <chr>, US0893 <chr>, US0897 <chr>, US0900 <chr>, US0903 <chr>,
## #   US0913 <chr>, US0928 <chr>, US0940 <chr>, US0949 <chr>, US0957 <chr>,
## #   US0968 <chr>, US0970 <chr>, US0972 <chr>, US0975 <chr>, US0977 <chr>,
## #   US0980 <chr>, US0984 <chr>, US0987 <chr>, US0988 <chr>, US0991 <chr>,
## #   US0992 <chr>, US0994 <chr>, US0996 <chr>, US1003 <chr>, US1005 <chr>,
## #   US1006 <chr>, US1015 <chr>, US1016 <chr>, US1028 <chr>, US1034 <chr>,
## #   US1036 <chr>, US1039 <chr>, US1040 <chr>, US1044 <chr>, US1048 <chr>,
## #   US1050 <chr>, US1055 <chr>, US1058 <chr>, US1059 <chr>, US1060 <chr>,
## #   US1061 <chr>, US1065 <chr>, US1067 <chr>, US1069 <chr>, US1070 <chr>,
## #   US1071 <chr>, US1076 <chr>, US1078 <chr>, US1133 <chr>, US1134 <chr>,
## #   TW16362 <chr>, TW16400 <chr>, TW16401 <chr>, TW16444 <chr>, TW16445 <chr>,
## #   TW16446 <chr>, TW16451 <chr>, TW16452 <chr>, TW16453 <chr>, TW16455 <chr>,
## #   TW16463 <chr>, TW16464 <chr>, TW16467 <chr>, TW16491 <chr>, ...
```

I then used two functions I wrote previously from QsNullModels. The first calculates the number of shared species (OGS in this case) between all samples (genomes in this case). The result is a vector. I used the second function to put the vector output from the first into the same form as a distance matrix. I could then use the `dist_pairs()` function to make a tabular table of pairwise shared OGs.

```
t.ogs.bin <- t(ogs.bin)
shared.ogs.vector <- QsNullModels::shared_spp(t.ogs.bin)
shared.ogs.matrix <- QsNullModels::stat2dist(t.ogs.bin, shared.ogs.vector)
paired.shared.ogs <- dist_pairs(shared.ogs.matrix) %>%
rename(shared_ogs = dist) %>%
as_tibble()
paired.shared.ogs
```

```
## # A tibble: 40,755 x 3
##    X1      X2      shared_ogs
##    <chr>   <chr>        <dbl>
##  1 TW16361 TW16636       1473
##  2 TW16361 TW16689       1495
##  3 TW16361 TW16693       1504
##  4 TW16361 TW16694       1502
##  5 TW16361 TW16728       1529
##  6 TW16361 TW16735       1478
##  7 TW16361 TW19114       1509
##  8 TW16361 TW19128       1499
##  9 TW16361 TW19129       1508
## 10 TW16361 TW19130       1491
## # ... with 40,745 more rows
```

In order to merge `dist_pairs` and `paired.shared.ogs`, I created another variable `line_name` by pasting entries in columns X1 and X2 together.

```
paired.shared.ogs <- paired.shared.ogs %>%
  mutate(line_name = paste(X1, X2, sep = "_")) %>%
  dplyr::select(line_name, shared_ogs) %>%
  as_tibble()
paired.shared.ogs
```

```
## # A tibble: 40,755 x 2
##    line_name       shared_ogs
##    <chr>                <dbl>
##  1 TW16361_TW16636       1473
##  2 TW16361_TW16689       1495
##  3 TW16361_TW16693       1504
##  4 TW16361_TW16694       1502
##  5 TW16361_TW16728       1529
##  6 TW16361_TW16735       1478
##  7 TW16361_TW19114       1509
##  8 TW16361_TW19128       1499
##  9 TW16361_TW19129       1508
## 10 TW16361_TW19130       1491
## # ... with 40,745 more rows
```

```
paired.distances <- d.pairs %>%
  mutate(line_name = paste(X1, X2, sep = "_")) %>%
  dplyr::select(line_name, dist) %>%
                as_tibble()
```

```
paired.distances
```

```
## # A tibble: 40,755 x 2
##    line_name         dist
##    <chr>            <dbl>
##  1 TW16361_TW16362 0.0216
##  2 TW16361_TW16370 0.0210
##  3 TW16361_TW16373 0.0223
##  4 TW16361_TW16374 0.0186
##  5 TW16361_TW16397 0.0202
##  6 TW16361_TW16398 0.0188
##  7 TW16361_TW16399 0.0218
##  8 TW16361_TW16400 0.0218
##  9 TW16361_TW16401 0.0238
## 10 TW16361_TW16402 0.0189
## # ... with 40,745 more rows
```

But when I tried to join these two tibbles, I encountered a problem.

```
df2plt <- full_join(paired.distances, paired.shared.ogs)
```

```
## Joining, by = "line_name"
```

```
df2plt
```

```
## # A tibble: 59,229 x 3
##    line_name         dist shared_ogs
##    <chr>            <dbl>      <dbl>
##  1 TW16361_TW16362 0.0216       1417
##  2 TW16361_TW16370 0.0210       1481
##  3 TW16361_TW16373 0.0223       1508
##  4 TW16361_TW16374 0.0186       1529
##  5 TW16361_TW16397 0.0202       1500
##  6 TW16361_TW16398 0.0188       1505
##  7 TW16361_TW16399 0.0218       1498
##  8 TW16361_TW16400 0.0218       1501
##  9 TW16361_TW16401 0.0238       1483
## 10 TW16361_TW16402 0.0189       1511
## # ... with 59,219 more rows
```

There were too many rows. Why? Investigating, I determined that it was because some values of X1 and X2 were reversed between the two tibbles. (I still do not understand how this happens.)

```
a <- length(intersect(paired.distances$line_name, paired.shared.ogs$line_name))
b <- length(setdiff(paired.distances$line_name, paired.shared.ogs$line_name))
a
```

```
## [1] 22281
```

```
b
```

```
## [1] 18474
```

```
a+b
```

```
## [1] 40755
```

To rectify the problem, I set aside rows from `paired.distances` for which `line_names` did match those in `paired.shared.ogs`, reversed the order of X1 and X2 in those that did not, and then recombined the two to create a "patched" `paired.distances` which could be successfully joined with `paired.shared.ogs`.

```
a <- anti_join(paired.distances, paired.shared.ogs)
```

```
## Joining, by = "line_name"
```

```
b <- semi_join(paired.distances, paired.shared.ogs)
```

```
## Joining, by = "line_name"
```

```
a
```

```
## # A tibble: 18,474 x 2
##    line_name         dist
##    <chr>            <dbl>
##  1 TW16362_TW16636 0.0202
##  2 TW16362_TW16689 0.0243
##  3 TW16362_TW16693 0.0183
##  4 TW16362_TW16694 0.0165
##  5 TW16362_TW16728 0.0165
##  6 TW16362_TW16735 0.0382
##  7 TW16362_TW19114 0.0158
##  8 TW16362_TW19128 0.0150
##  9 TW16362_TW19129 0.0152
## 10 TW16362_TW19130 0.0162
## # ... with 18,464 more rows
```

```
b
```

```
## # A tibble: 22,281 x 2
##    line_name         dist
##    <chr>            <dbl>
##  1 TW16361_TW16362 0.0216
##  2 TW16361_TW16370 0.0210
##  3 TW16361_TW16373 0.0223
##  4 TW16361_TW16374 0.0186
##  5 TW16361_TW16397 0.0202
##  6 TW16361_TW16398 0.0188
##  7 TW16361_TW16399 0.0218
##  8 TW16361_TW16400 0.0218
##  9 TW16361_TW16401 0.0238
## 10 TW16361_TW16402 0.0189
## # ... with 22,271 more rows
```

```
c <- a %>%
  tidyr::separate(line_name, c("A", "B"), sep="_") %>%
  mutate(line_name = paste(B, A, sep = "_")) %>%
  select(line_name, dist) %>%
  rbind(b)
c
```

```
## # A tibble: 40,755 x 2
##    line_name         dist
##    <chr>            <dbl>
##  1 TW16636_TW16362 0.0202
##  2 TW16689_TW16362 0.0243
##  3 TW16693_TW16362 0.0183
##  4 TW16694_TW16362 0.0165
##  5 TW16728_TW16362 0.0165
```

```
##  6 TW16735_TW16362 0.0382
##  7 TW19114_TW16362 0.0158
##  8 TW19128_TW16362 0.0150
##  9 TW19129_TW16362 0.0152
## 10 TW19130_TW16362 0.0162
## # ... with 40,745 more rows
```

```
df2plt <- full_join(c, paired.shared.ogs)
```

```
## Joining, by = "line_name"
```

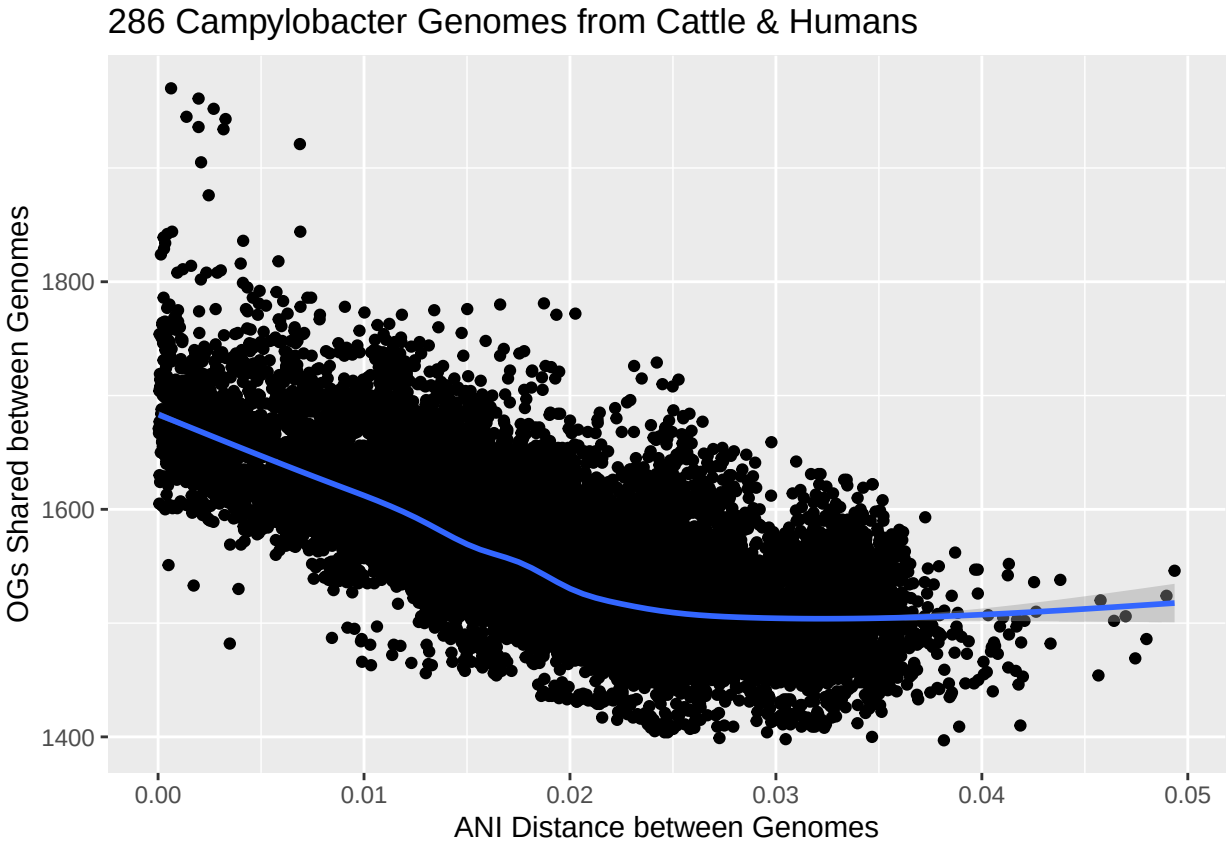
```
df2plt
```

```
## # A tibble: 40,755 x 3
##    line_name         dist shared_ogs
##    <chr>            <dbl>      <dbl>
##  1 TW16636_TW16362 0.0202       1520
##  2 TW16689_TW16362 0.0243       1482
##  3 TW16693_TW16362 0.0183       1480
##  4 TW16694_TW16362 0.0165       1456
##  5 TW16728_TW16362 0.0165       1472
##  6 TW16735_TW16362 0.0382       1459
##  7 TW19114_TW16362 0.0158       1461
##  8 TW19128_TW16362 0.0150       1527
##  9 TW19129_TW16362 0.0152       1533
## 10 TW19130_TW16362 0.0162       1464
## # ... with 40,745 more rows
```

I then made a scatter plot between `dist` and `shared_ogs`.

```
ggplot(data = df2plt, aes(x=dist, y = shared_ogs)) +
  geom_point() +
  geom_smooth() +
  xlab("ANI Distance between Genomes") +
  ylab("OGs Shared between Genomes") +
  ggtitle("286 Campylobacter Genomes from Cattle & Humans")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


286 Campylobacter Genomes from Cattle & Humans
OGs Shared between Genomes
1800
1600
1400
0.00
0.01
0.02
0.03
0.04
0.05
ANI Distance between Genomes