UHA_Pairwise_Dist_Matrix

Mikaely Evans

2024-01-10

Loading in necessary libraries and data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(geosphere)

wd <- getwd()

UHA_Tissue_Database <- read.csv(paste0(wd,"/../../Data_Files/Clean_Data_Files/Clean_csv/ARCHIVED_USBG_Hy
    filter(!is.na(Longitude) | !is.na(Latitude)) #filter any individuals that we don't have Lat/Long fo

parentage_results <- read.csv(paste0(wd,"/../../Data_Files/Clean_Data_Files/Clean_csv/UHA_full_parentag
  rename(Mother_ID = Mother.ID,
         Father_ID = Candidate.father.ID) %>%
  filter(!is.na(Father_ID)) #something weird happened in the cervus bit that led to this file having a
```

Get all possible combinations of trees in the Tissue Database and save it in a df called “potential_combo_dedup”

```
all_potential_combo <- crossing(UHA_Tissue_Database$Tissue.ID, UHA_Tissue_Database$Tissue.ID) #create e
colnames(all_potential_combo) <- c("Parent_1", "Parent_2") #assign names to columns of all the possible

# No longer deduplicating because it makes downstream analyses more cimplicated and only takes a little
#combo_dedup <- all_combo[!duplicated(apply(all_combo, 1, sort), MARGIN = 2),] #remove any duplicates o
```

```r
potential_combo_dedup <- filter(all_potential_combo, Parent_1 != Parent_2)  #remove rows where the pare

#make columns to store important data (distances and parent species)
potential_combo_dedup$dist <- NA
potential_combo_dedup$Parent_1_species <- NA
potential_combo_dedup$Parent_2_species <- NA

#loop through every pair of individuals in the deduplicated combinations data set (combo_dedup) and get
for(d in 1:nrow(potential_combo_dedup)){
  Parent_1 <- potential_combo_dedup$Parent_1[d]
  Parent_2 <- potential_combo_dedup$Parent_2[d]

  #access the original tissue database via the parents
  Parent_1_Database_row <- filter(UHA_Tissue_Database, Tissue.ID == Parent_1)
  Parent_2_Database_row <- filter(UHA_Tissue_Database, Tissue.ID == Parent_2)

  #use distGeo to get distance in m between the lat long points of the 2 parents
  potential_combo_dedup$dist[d] <- distGeo(c(Parent_1_Database_row$Longitude, Parent_1_Database_row$Lati
                                c(Parent_2_Database_row$Longitude, Parent_2_Database_row$Latitude))

  #record the species of both parents
  potential_combo_dedup$Parent_1_species[d] <- Parent_1_Database_row$Species
  potential_combo_dedup$Parent_2_species[d] <- Parent_2_Database_row$Species

}
```

Edit the potential_combo_dedup df to ensure consistent naming and create a column that indicates if the 2 parents are conspecifics or heterospecifics

```r
potential_combo_info <- potential_combo_dedup %>%
  mutate(Parent_1_species = str_replace_all(Parent_1_species, "Quercus", "Q."),
         Parent_2_species = str_replace_all(Parent_2_species, "Quercus", "Q.")) %>% #replace all instan
  mutate(Parental_species_match = case_when(Parent_1_species == Parent_2_species ~ "Conspecific",
                                            Parent_1_species != Parent_2_species ~ "Heterospecific")) #
```

Make some changes to the potential_combo_info and parentage_results dfs to make them more relevant to analyses

```r
mom_IDs <- unique(parentage_results$Mother_ID) #get all unique mother ID's from the df that contains th

relevant_potential_combos <- potential_combo_info %>%
  filter(Parent_1 %in% mom_IDs) #filter from the total set of combinations for all possible trees down


relevant_parentage_results <- left_join(parentage_results, select(relevant_potential_combos, c(Parent_1
```

Make temporary dfs that contain the mean distance of the 5 shortest distances from the mothers to the fatners for both the real (relevant_parentage_results) and potential (relevant_potential_combos) datasets

```
#create df that has the mean distance of the 5 closest potential conspecific and heterospecific fathers
potential_fathers_mean_small_summary <- relevant_potential_combos%>%
  group_by(Parent_1, Parental_species_match) %>%
  top_n(-5, dist) %>% #keeps only the smallest 5 dist values in each of the unique groups
  summarise(Mean_smallest_potential_dists = mean(dist, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'Parent_1'. You can override using the
## `.groups` argument.
```

```
#make a dataset that contains only the mothers from which there are less than 5 unique dads so I can ma
#top_n isn't actually going to work since it allows ties and doesn't count them toward the 5 so we need
inds_w_few_fathers <- relevant_parentage_results %>%
  group_by(Mother_ID, Father_ID) %>%
  summarize(num_off = n()) %>%
  group_by(Mother_ID) %>%
  summarize(num_dads = n()) %>%
  filter(num_dads < 5)
```

```
## `summarise()` has grouped output by 'Mother_ID'. You can override using the
## `.groups` argument.
```

```
#SHOULD I BE TAKING THE MEAN OF THE SUCCESSFUL DADS OR THE MEAN OF ALL OF THE OFFSPRING (bc repeats wil

#summarize the mean distance of all the distances for the mothers with less than 5 unique dads bc these
inds_w_few_fathers_summary <- relevant_parentage_results %>%
  filter(Mother_ID %in% inds_w_few_fathers$Mother_ID) %>%
  group_by(Mother_ID) %>%
  summarise(Mean_smallest_real_dists = mean(dist, na.rm=TRUE))


#create df that has the mean distance of the 5 closest real fathers to each maternal tree (including th
real_fathers_mean_small_summary <- relevant_parentage_results%>%
  filter(!Mother_ID %in% inds_w_few_fathers$Mother_ID) %>%
  group_by(Mother_ID) %>%
  slice_min(dist, n =5, with_ties = F) %>%
  summarise(Mean_smallest_real_dists = mean(dist, na.rm=TRUE)) %>%
  rbind(., inds_w_few_fathers_summary) #rbind the dataframe that has the mean distances of the individu

#create df that has the mean distance of the 5 closest possible fathers to each maternal tree (includin
potential_fathers_mean_small_summary <- relevant_potential_combos%>%
  group_by(Parent_1, Parental_species_match) %>%
  slice_min(dist, n =5, with_ties = F) %>%
  summarise(Mean_smallest_potential_dists = mean(dist, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'Parent_1'. You can override using the
## `.groups` argument.
```

Make dataset ("real_fathers_summary") containing all of the desired summarized categories from the real parentage data (mean distance of real fathers, minimum distance of real fathers, mean distance of closest 5 real fathers). NOTE: this is based off of the offspring data meaning that the 5 closest fathers might be the same 5 fathers and the mean distance is weighted by the number of offspring fathered by each dad.

```
#summarize data by mean and min distance to real fathers and by proportion of offspring that are hybrid
real_fathers_summary <- relevant_parentage_results%>%
  group_by(Mother_ID) %>%
  summarise(Mean_real_dist = mean(dist, na.rm=TRUE),
            Min_real_dist = min(dist, na.rm=TRUE),
            Prop_hybrids = mean(Hybrid, na.rm = TRUE)) %>%
  left_join(., real_fathers_mean_small_summary, join_by(Mother_ID == Mother_ID)) #add the Mean_smallest
```

Make dataset ("real_fathers_summary") containing all of the desired summarized categories from the possible combinations of mothers and fathers data (mean distance of potential fathers, minimum distance of potential fathers, mean distance of closest 5 potential fathers). NOTE: this is NOT based off of the offspring data meaning that the 5 closest fathers will be unique and the mean distance to fathers will be unweighted

```
#summarize data from data with all combos of possible conspecific and heterospefic individuals with eac
potential_fathers_summary <- relevant_potential_combos %>%
  group_by(Parent_1, Parental_species_match) %>%
  summarise(Mean_potential_dist = mean(dist, na.rm=TRUE),
            Min_potential_dist = min(dist, na.rm=TRUE))  %>%
  left_join(., potential_fathers_mean_small_summary, join_by(Parent_1 == Parent_1, Parental_species_mat
  left_join(., select(real_fathers_summary, c(Mother_ID, Prop_hybrids)), join_by(Parent_1 == Mother_ID)
```

```
## `summarise()` has grouped output by 'Parent_1'. You can override using the
## `.groups` argument.
```

The linear model for the mean distance of real fathers to a given mother

```
summary(lm(formula = Prop_hybrids~Mean_real_dist, data=real_fathers_summary))
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_real_dist, data = real_fathers_summary)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09325 -0.03719  0.02394  0.04263  0.04865
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0628119  0.0603046  -1.042   0.3377
## Mean_real_dist  0.0006941  0.0002812   2.469   0.0485 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05832 on 6 degrees of freedom
## Multiple R-squared:  0.5039, Adjusted R-squared:  0.4212
## F-statistic: 6.094 on 1 and 6 DF,  p-value: 0.04855
```

The linear model for the smallest distance of a real father to a given mother

```
summary(lm(formula = Prop_hybrids~Min_real_dist, data=real_fathers_summary))
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Min_real_dist, data = real_fathers_summary)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.080729 -0.066683 -0.003176  0.036629  0.121525
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.180e-02  3.663e-02   2.233    0.067 .
## Min_real_dist -8.814e-05  4.146e-04  -0.213    0.839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08249 on 6 degrees of freedom
## Multiple R-squared:  0.007476,   Adjusted R-squared:  -0.1579
## F-statistic: 0.04519 on 1 and 6 DF,  p-value: 0.8387
```

The linear model for the mean distance of the shortest 5 distances of real fathers to a given mother

```
summary(lm(formula = Prop_hybrids~Mean_smallest_real_dists, data=real_fathers_summary))
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_smallest_real_dists, data = real_fathers_summary)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.082601 -0.074547  0.003231  0.041411  0.117966
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.151e-02  4.882e-02   1.465    0.193
## Mean_smallest_real_dists 4.932e-05  3.462e-04   0.142    0.891
##
## Residual standard error: 0.08266 on 6 degrees of freedom
## Multiple R-squared:  0.00337,    Adjusted R-squared:  -0.1627
## F-statistic: 0.02029 on 1 and 6 DF,  p-value: 0.8914
```

The linear model for the mean distance of a potential fathers to a given mother split by conspecific and heterospecific fathers

```
summary(lm(formula = Prop_hybrids~Mean_potential_dist, data=filter(potential_fathers_summary, Parental_s
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_potential_dist, data = filter(potential_fathers_summary,
##     Parental_species_match == "Conspecific"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10133 -0.04937  0.01949  0.05852  0.06126
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.0871141  0.1280344  -0.680    0.522
## Mean_potential_dist  0.0005308  0.0004054   1.309    0.238
##
## Residual standard error: 0.07303 on 6 degrees of freedom
## Multiple R-squared:  0.2222, Adjusted R-squared:  0.0926
## F-statistic: 1.714 on 1 and 6 DF,  p-value: 0.2383
```

```
summary(lm(formula = Prop_hybrids~Mean_potential_dist, data=filter(potential_fathers_summary, Parental_s
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_potential_dist, data = filter(potential_fathers_summary,
##     Parental_species_match == "Heterospecific"))
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08460 -0.07141 -0.00043  0.04235  0.12104
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.252e+00  2.126e+01  -0.247    0.813
## Mean_potential_dist  9.279e-05  3.703e-04   0.251    0.810
##
## Residual standard error: 0.08237 on 6 degrees of freedom
## Multiple R-squared:  0.01036,    Adjusted R-squared:  -0.1546
## F-statistic: 0.0628 on 1 and 6 DF,  p-value: 0.8105
```

The linear model for the smallest distance of a potential father to a given mother split by conspecific and heterospecific fathers

```
summary(lm(formula = Prop_hybrids~Min_potential_dist, data=filter(potential_fathers_summary, Parental_sp
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Min_potential_dist, data = filter(potential_fathers_summary,
##     Parental_species_match == "Conspecific"))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.095327 -0.050501  0.001991  0.048984  0.094070
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.0448270  0.0309327   1.449    0.197
## Min_potential_dist 0.0005516  0.0003299   1.672    0.146
##
## Residual standard error: 0.06839 on 6 degrees of freedom
## Multiple R-squared:  0.3178, Adjusted R-squared:  0.2041
## F-statistic: 2.796 on 1 and 6 DF,  p-value: 0.1456
```

```
summary(lm(formula = Prop_hybrids~Min_potential_dist, data=filter(potential_fathers_summary, Parental_sp
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Min_potential_dist, data = filter(potential_fathers_summary,
##     Parental_species_match == "Heterospecific"))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.099658 -0.033385 -0.002741  0.045200  0.097936
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         0.1159695  0.0379327   3.057   0.0223 *
## Min_potential_dist -0.0005529  0.0003994  -1.384   0.2156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07209 on 6 degrees of freedom
## Multiple R-squared:  0.242,  Adjusted R-squared:  0.1157
## F-statistic: 1.916 on 1 and 6 DF,  p-value: 0.2156
```

The linear model for the mean distance of the shortest 5 distances of potential fathers to a given mother split by conspecific and heterospecific fathers

```
summary(lm(formula = Prop_hybrids~Mean_smallest_potential_dists, data=filter(potential_fathers_summary,
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_smallest_potential_dists, data = filter(potential_fathers_summary,
##     Parental_species_match == "Conspecific"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10592 -0.04235  0.01020  0.04775  0.07853
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.0326971  0.0338246   0.967    0.371
## Mean_smallest_potential_dists 0.0004786  0.0002625   1.823    0.118
##
## Residual standard error: 0.06642 on 6 degrees of freedom
## Multiple R-squared:  0.3565, Adjusted R-squared:  0.2493
## F-statistic: 3.324 on 1 and 6 DF,  p-value: 0.1181
```

```
summary(lm(formula = Prop_hybrids~Mean_smallest_potential_dists, data=filter(potential_fathers_summary,
```

```
##
## Call:
## lm(formula = Prop_hybrids ~ Mean_smallest_potential_dists, data = filter(potential_fathers_summary,
##     Parental_species_match == "Heterospecific"))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.097062 -0.036501  0.000295  0.040084  0.108617
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.1294906  0.0474181   2.731   0.0341 *
## Mean_smallest_potential_dists -0.0005579  0.0004237  -1.317   0.2359
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07293 on 6 degrees of freedom
## Multiple R-squared:  0.2242, Adjusted R-squared:  0.09495
## F-statistic: 1.734 on 1 and 6 DF,  p-value: 0.2359
```

A for loop that, across each mom, randomly samples the pool of the potential dads 1000 times and records the distance between that dad and the given mom and stores the results in "parents_dists_table_full"

```
set.seed(2024) #since we are using the sample function, set the seed for repeatability

#create the table that will hold the results of the for loop with 3 columns: Mom, Dad, and dist
parents_dists_table_full <- tibble(Mom = character(), Dad = character(), dist = character())


for(i in 1:length(unique(relevant_potential_combos$Parent_1))){
  parents_dists_table_small <- tibble(Mom = character(), Dad = character(), dist = character()) #make a
  mom <- unique(relevant_potential_combos$Parent_1)[i]  #get the ID of the given mom
  mom_specific_combos <- relevant_potential_combos %>%
    filter(Parent_1 == mom) %>%  #filter all possible combos to only those with the given mom as Parent_
    filter(Parental_species_match == "Conspecific")  #filter all possible combos with the given mom to

  possible_dads <- mom_specific_combos$Parent_2 #make a vector with a list of all of the possible dads

  dads <- sample(possible_dads, size=1000, replace=T)  #randomly draw 1000 samples from all possible da

  #a for loop that will loop through each of the 1000 sampled dads will make and add a row for the pare
  for(x in 1:length(dads)){
    dad <- dads[x] #get the ID of the given dad
    parent_combo_row_for_table <- tibble(Mom = character(), Dad = character(), dist = character()) #cre
    parent_combo_row_for_table$Mom <- mom #add ID of the given mom to table
    parent_combo_row_for_table$Dad <- dad #add ID of the given dad to table
    parent_combo_row_for_table$dist <- filter(mom_specific_combos, Parent_1 == mom & Parent_2 == dad)$di

    parents_dists_table_small <- rbind(parents_dists_table_small, parent_combo_row_for_table) #bind the
  }
  parents_dists_table_full <- rbind(parents_dists_table_full, parents_dists_table_small)  #bind the tab
}
```

Make a plot that compares the real distances between parents (from relevant_parentage_results) to the potential distance between parents (from parents_dists_table_full)

```r
#edit relevant_parentage_results to have the same formatting as parents_dists_table_full for easier his
parentage_results_for_hists <- relevant_parentage_results %>%
  rename(Mom = Mother_ID,
         Dad = Father_ID) %>%
  select(Mom, Dad, dist, Hybrid) %>%
  filter(Hybrid == F) %>%  #we are only looking at distance between conspecific pairs in this analysis
  filter(!is.na(dist))  #remove entries where dist is NA because some trees that we have tissue from do

#Create the a histogram that shows the difference in the real and theoretical distances between parents
ggplot() +
  geom_histogram(data = parents_dists_table_full, aes(x = dist, y = ..density.., fill = "theoretical"),
  geom_histogram(data = parentage_results_for_hists, aes(x = dist, y = ..density.., fill = "real"), alp
  labs(fill = "data set") +
  scale_fill_manual(values = c("theoretical" = "red", "real" = "blue")) +
  #facet_wrap(~Mom) +  # facet_wrap by Mom to see how each individual Maternal tree did compared to boo
  theme_classic()
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```