Analysis of Human Origins (plus Pacific) data

Alejandro Ochoa and John D. Storey

2019-05-03

Introduction

In this R markdown file we demonstrate the basic analysis of Human Origins dataset presented in our paper "New kinship and FST estimates reveal higher levels of differentiation in the global human population" by Ochoa and Storey. The analysis presented here relies exclusively on the data files provided in this repository (https://github.com/StoreyLab/human_differentiation_analysis), which are the public subset of Human Origins and Pacific datasets.

R package dependencies

To run this code, you will need the following packages:

```r
library(BEDMatrix)    # to load genotype matrices
library(popkin)       # to estimate "population kinship" matrices
library(genio)        # to parse FAM file
library(readr)        # to read additional tables
library(tibble)       # to create tables
library(dplyr)        # for sanity checks
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(RColorBrewer) # for colors
```

Estimate the kinship matrix

Load the genotype matrix (smartly, actually doesn't load into memory):

```r
X <- BEDMatrix('data/human_origins_and_pacific_public', simple_names = TRUE)
```

```
## Extracting number of samples and rownames from human_origins_and_pacific_public.fam...

## Extracting number of variants and colnames from human_origins_and_pacific_public.bim...
```

This function from `genio` loads individual annotations (in plink FAM format), including subpopulations.

```r
fam <- read_fam('data/human_origins_and_pacific_public')
```

```
## Reading: data/human_origins_and_pacific_public.fam
```

Now estimate the kinship matrix. We rely on the subpopulation labels to estimate the minimum kinship value to be treated as unrelated (zero). This step takes about 7 minutes.

```
kinship <- popkin(X, subpops = fam$fam)
```

Sub-subpopulation annotations

Before plotting, we need to reorder individuals so things look reasonable. To achieve this and other tasks, we load an additional table with subpopulation annotations.

```
info <- read_tsv(
    'data/human_origins_and_pacific_public_subpops.txt',
    col_types = 'ccddc'
)
# inspect
info
```

```
## # A tibble: 248 x 5
##    subsubpop     subpop      x     y country
##    <chr>         <chr>   <dbl> <dbl> <chr>
##  1 Ju_hoan_South SAfrica  20.7 -21.2 Botswana
##  2 Ju_hoan_North SAfrica  21.5 -18.9 Namibia
##  3 Taa_West      SAfrica  20.3 -23.6 Botswana
##  4 Taa_East      SAfrica  22.8 -24.2 Botswana
##  5 Taa_North     SAfrica  22.4 -23   Botswana
##  6 Naro          SAfrica  21.6 -22   Botswana
##  7 Gui           SAfrica  23.3 -21.5 Botswana
##  8 Hoan          SAfrica  23.4 -24   Botswana
##  9 Xuun          SAfrica  19.7 -18.7 Namibia
## 10 Gana          SAfrica  23.4 -21.7 Botswana
## # ... with 238 more rows
```

This table has one row per sub-subpopulation, for each of the 248 sub-subpopulations in the full dataset (including singleton and other sub-subpopulations removed in our publication). The table contains geographical coordinates (`x` and `y`) and country annotations, which we won't use here. These sub-subpopulations are grouped into 11 subpopulations (second column) that represent continental-level regions. We will use the **`subsubpop`** and **`subpop`** labels, and their order in this table, to sort individuals and create plots. The order and subpopulation assignments were defined manually, informed by geography and refined iteratively using the kinship plots to keep the most similar sub-subpopulations together.

Reorder individuals using sub-subpopulation annotations

Let's reorder individuals. First we ensure that every sub-subpopulation in the genotype table is present in the annotations table:

```
stopifnot(
    fam$fam %in% info$subsubpop
)
```

Now we reorder individuals so their sub-subpopulations appear in the desired order

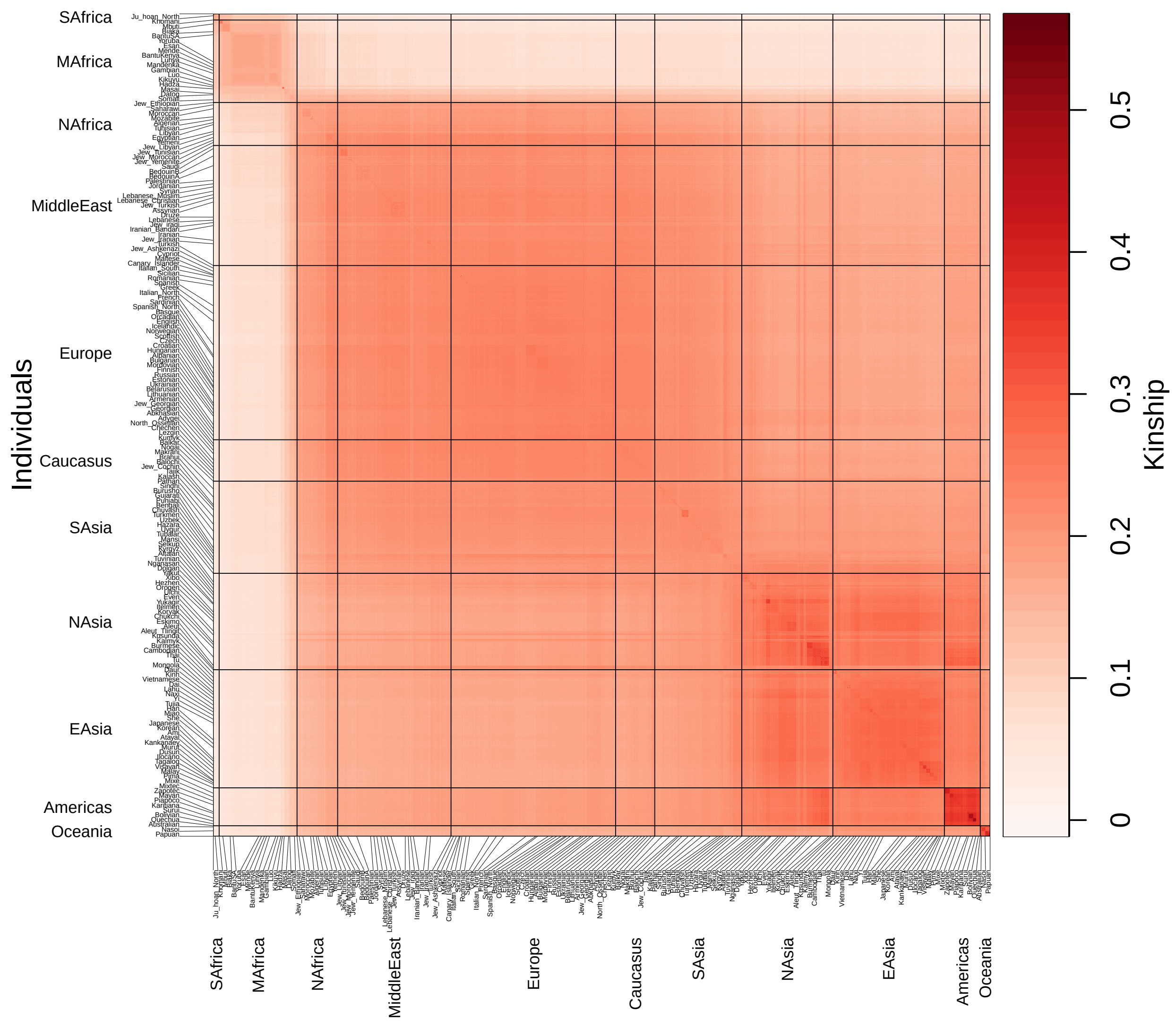
```
# the rank ties every individual in the same sub-subpopulation
fam_rank <- match(fam$fam, info$subsubpop)
# this order breaks ties by order of appearance
```

```r
indexes <- order(fam_rank)
# before reordering, let's map subpopulations onto the FAM table
fam$subpop <- info$subpop[fam_rank]
# apply order to kinship matrix (both dimensions)
kinship <- kinship[indexes, indexes]
# and to individual annotations (rows only)
fam <- fam[indexes, ]
```

Kinship matrix plot

We are ready to visualize the estimated kinship matrix:

```r
# line position for outer labels
line <- 3
# inner margins
par(mar = c(0, 0, 0, 0) + 0.2)
# outer margins
par(oma = c(line + 2.2, line + 3.2, 0, 3))

# main plot with all labeling bells and whistles
plot_popkin(
    # plot inbreeding along diagonal
    kinship = inbr_diag( kinship ),
    # leave a lot of space for labels
    ylab_line = line + 2,
    # two levels of labels
    labs = cbind(fam$fam, fam$subpop),
    labs_cex = c(0.25, 0.6),
    labs_las = 2,
    labs_line = c(1, line),
    labs_lwd = c(0.1, 0.5),
    labs_sep = c(FALSE, TRUE),
    labs_even = c(TRUE, FALSE)
)
```



Compared to the full dataset, this version limited to the public data has very few samples from `SAfrica` and `Oceania`.

Weights to balance subpopulations and sub-subpopulations

This function calculates weights for individuals such that:

- every subpopulation has equal weight, and
- every sub-subpopulation has equal weight within its subpopulation.

```r
# wrap code around function
get_weights_human_origins <- function(fam, info) {
    # this processing will be wrong if info contains more subpops than are present in fam,
    # so let's make them agree internally
    info <- info[ info$subsubpop %in% fam$fam, ]

    # get counts for sub-subpops and subpops
    # number of individuals in each sub-subpopulation
    subsubpop_to_counts <- table( fam$fam )
    # number of sub-subpopulations in each subpopulation
    subpop_to_counts <- table( info$subpop )
    # number of unique subpopulations
    K <- length( subpop_to_counts )

    # construct weights!
    weights <- 1 / (K * subsubpop_to_counts[fam$fam] * subpop_to_counts[fam$subpop] )
}
# apply function
weights <- get_weights_human_origins(fam, info)

# sanity checks

# do the weights sum to one?  Yes!
sum( weights )
```

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

```
# next steps are best performed on a tibble
weights_tibble <- tibble(
    weight = as.numeric( weights ),
    subpop = fam$subpop,
    subsubpop = fam$fam
)

# inspect
weights_tibble
```

```
## # A tibble: 2,124 x 3
##     weight subpop  subsubpop
##      <dbl> <chr>   <chr>
##  1 0.00909 SAfrica Ju_hoan_North
##  2 0.00909 SAfrica Ju_hoan_North
##  3 0.00909 SAfrica Ju_hoan_North
##  4 0.00909 SAfrica Ju_hoan_North
##  5 0.00909 SAfrica Ju_hoan_North
##  6 0.00413 SAfrica Khomani
##  7 0.00413 SAfrica Khomani
##  8 0.00413 SAfrica Khomani
##  9 0.00413 SAfrica Khomani
## 10 0.00413 SAfrica Khomani
## # ... with 2,114 more rows
```

```
# are subpopulations weighed equally?  Also yes!
weights_tibble %>%
    group_by( subpop ) %>%
    summarize(sum = sum(weight))
```

```
## # A tibble: 11 x 2
##    subpop        sum
##    <chr>       <dbl>
##  1 Americas   0.0909
##  2 Caucasus   0.0909
##  3 EAsia      0.0909
##  4 Europe     0.0909
##  5 MAfrica    0.0909
##  6 MiddleEast 0.0909
##  7 NAfrica    0.0909
##  8 NAsia      0.0909
##  9 Oceania    0.0909
## 10 SAfrica    0.0909
## 11 SAsia      0.0909
```

```
# lastly, the sub-subpopulations are weighed equally within their subpopulation.
# We test that with 'Americas' only, for brevity:
weights_tibble %>%
    filter( subpop == 'Americas' ) %>%
    group_by( subsubpop ) %>%
    summarize(sum = sum(weight))
```

```
## # A tibble: 10 x 2
##    subsubpop     sum
##    <chr>       <dbl>
##  1 Bolivian  0.00909
##  2 Karitiana 0.00909
##  3 Mayan     0.00909
##  4 Mixe      0.00909
##  5 Mixtec    0.00909
##  6 Piapoco   0.00909
##  7 Pima      0.00909
##  8 Quechua   0.00909
##  9 Surui     0.00909
## 10 Zapotec   0.00909
```

FST estimate

This is the FST estimate we obtain, which is made more robust by using the weights we previously calculated to balance subpopulations and sub-subpopulations:

```
fst_estimate <- fst( kinship, weights )

# inspect
fst_estimate
```

```
## [1] 0.2622629
```

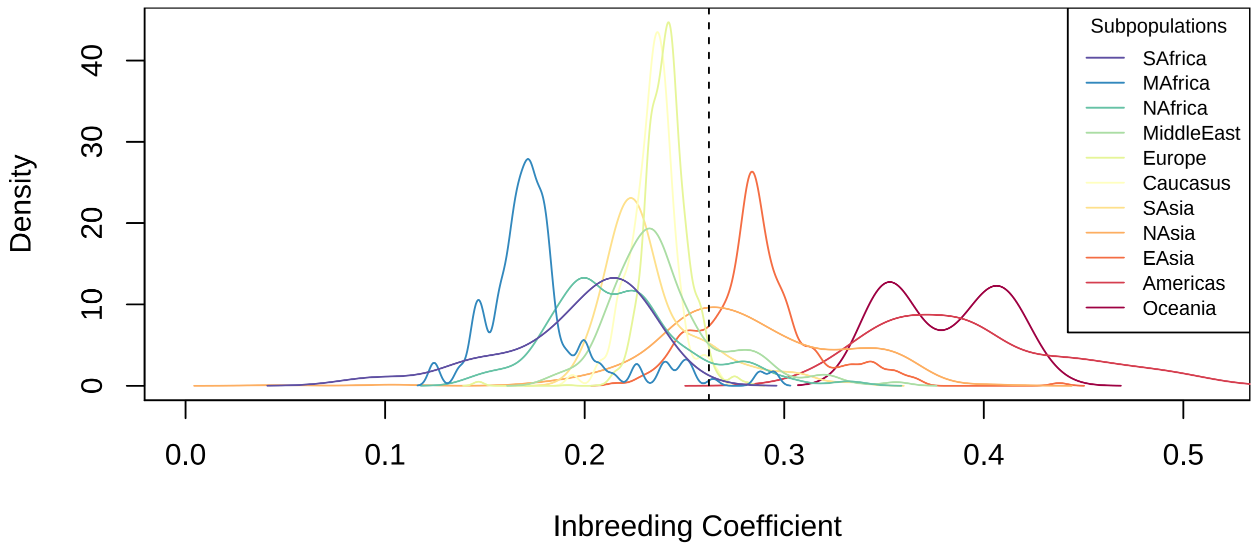
Nice! The value we estimated in the publication, using the full dataset, was 0.260.

Density of inbreeding coefficients

This code plots density curves of inbreeding coefficients for each subpopulation in a different color:

```
subpops <- unique( info$subpop )
# color palette for subpopulations
colors_subpops <- rev(brewer.pal(length(subpops), 'Spectral'))

# extract inbreeding vector
inbrs <- inbr( kinship )
```

```r
# initialize y max, will grow below
y_max <- 0
# collect densities of interest.
# Complicated because samples are weighed to balance sub-subpopulations
# within each subpopulation
subpop_to_density <- list()
for (subpop in subpops) {
    # filter of data
    indexes <- fam$subpop == subpop
    # get subset weights too
    weights_subpop <- weights[indexes]
    # renormalize when filters are done!
    weights_subpop <- weights_subpop / sum(weights_subpop)
    # estimate density using these weights
    density_subpop <- density( inbrs[indexes], weights = weights_subpop )
    # add to list of densities
    subpop_to_density[[subpop]] <- density_subpop
    # grow y max as needed
    y_max <- max(y_max, density_subpop$y)
}

# shrink default margins
par( mar = c(4, 4, 0, 0) )

# initialize plotting region
plot(
    NA,
    xlim = range(0, inbrs),
    ylim = c(0, y_max),
    xlab = 'Inbreeding Coefficient',
    ylab = 'Density'
)

# add each subpopulation density
for (subpop in rev(subpops)) {
    # get color for this subpopulation
    color_subpop <- colors_subpops[match(subpop, subpops)]
    # plot density
    lines( subpop_to_density[[subpop]], col = color_subpop )
}

# add new Fst estimate (dashed line)
abline(v = fst_estimate, lty = 2)

# legend
legend(
    'topright',
    subpops,
    title = 'Subpopulations',
    lty = 1,
    col = colors_subpops,
    cex = 0.7
)
```


Subpopulations
SAfrica
MAfrica
NAfrica
MiddleEast
Europe
Caucasus
SAsia
NAsia
EAsia
Americas
Oceania
Density
Inbreeding Coefficient
0
10
20
30
40
0.0
0.1
0.2
0.3
0.4
0.5