Killer whale Clock

Caila Kucheravy

2024-10-28

Run the killer whale skin clock developed by Parsons et al. (2023).

Prep:

```
setwd("~/Documents/Master's/Analysis/Epigenetic Aging/Killer Whales")

library(tidyverse)
#library(glmnet)
```

Load updated sample sheet:

```
sample_sheet <- readRDS('output/updated_sample_sheet_combined_KillerWhale_array.rds')
```

Load the killer whale skin clock, select the correct columns and filter out NAs to have only CpGs used in the clock:

```
clock <- read_csv('input/Table.WhaleS3.SkinClockCoef.csv')

kw_clock <- clock %>%
  select(var, Coef.Killerwhale.Skin.Sqrt) %>%      # Select the correct column for the KW clock
  filter(!is.na(Coef.Killerwhale.Skin.Sqrt))       # Filter out CpGs not used for this clock (50 CpGs)
```

Load normalized beta values:

```
kw_betas <- readRDS('output/tbetas_corrected_combined_KillerWhale_array_August2024_redo3.rds')

# Filter for CpGs used in clock - 50 for killer whales
kw_betas_filtered <- kw_betas %>%
  select(any_of(kw_clock$var))
```

The age transformation used in the paper is: sqrt(Age+1)=Age. Need to back-transform for final age.

Form a weighted linear combination of the CpGs for killer whales:

```
# Pivot clock wider to match beta table
kw_clock_wide <- kw_clock %>%
  pivot_wider(names_from = var, values_from = Coef.Killerwhale.Skin.Sqrt)

# Multiply beta values by the clock weights:
est_ages_kw <- data.frame(mapply('*', kw_betas_filtered, kw_clock_wide[,2:51]))

# Sum values, add intercept
est_ages_kw <- est_ages_kw %>%
  mutate(Sum = rowSums(est_ages_kw)) %>%
  mutate(Intercept = kw_clock_wide$`(Intercept)`)

# Sum intercept and weighted beta values
```

```
est_ages_kw <- est_ages_kw %>%
  mutate(Ages = rowSums(est_ages_kw[,c("Sum", "Intercept")]))

# Age transformation: DNAmAge = F^(-1) (x*beta)
est_ages_kw <- est_ages_kw %>%
  mutate(Age_Transformed = (est_ages_kw$Ages^2) - 1)

est_ages_kw$Age_Transformed
```

```
##  [1] 23.615180  8.218794  9.065264  4.183960 11.200262  3.177588  4.612530
##  [8] 10.369020 12.701103  3.385554 20.186232  6.627101 14.988458 14.256375
## [15]  3.913672  7.734235  7.347518  6.786813  7.470849  9.314062 15.930368
## [22] 12.712197 15.634721  2.887993  9.315785 12.841012  7.294429 11.350434
## [29]  9.176903 25.289485  5.437012  7.140396  7.932776  3.434888 11.790230
## [36] 12.000812  9.948638  5.922878  4.298312  6.956046 23.257862  8.779063
## [43]  5.988418  5.765844  5.030084 22.468361 10.667820 17.570467  8.731627
## [50]  3.650937  6.414204  7.789091 18.925497 12.108077 29.251329  3.102473
## [57]  5.459565  4.480056  4.984759  8.101788  9.068439  6.308725 10.764665
## [64] 12.387284  6.667874  2.960607 14.696543  3.009963  3.344104 12.910515
## [71] 12.964907  6.070538  9.738125  8.691357 11.665770 11.033129 14.122635
## [78]  8.734187 10.284856  9.164254 16.533763 15.735843  8.255537  9.297317
## [85] 12.497050 11.071359 10.038349  7.628992  9.070211  5.612963  7.631208
## [92] 13.087478 14.853402  9.355239 13.765679 14.110114  2.756090 15.160602
## [99] 12.051822
```

Add ages to sample sheet:

```
# Add column with basename back to dataframe with ages
est_ages_kw$chip.ID.loc <- sample_sheet$chip.ID.loc

# Select chip ID and ages to join with sample sheet
DNAm_ages_kw <- est_ages_kw %>%
  select(chip.ID.loc, Age_Transformed)

# Join ages with sample sheet
kw_ages <- sample_sheet %>%
  left_join(DNAm_ages_kw, by = "chip.ID.loc")

# Save csv file of all ages
# write.csv(kw_ages, "kw_ages_August2024")
```

Remove the "bad" samples from the epigenetic array:

```
bad <- c("KW-2021-PG-03",
         "KW-2021-PG-04",
         "KW-2021-PG-05",
         "KW-2021-PG-06",
         "KW-2021-PG-08",
         "KW-2021-PG-09",
         "KW-2021-PG-11",
         "KW-2021-PG-XX",
         "KW-2019-01",
         "KW-2019-02",
         "GRNL-KW-2021-01",
         "GRNL-KW-2021-03",
```

```
          "OR21-1",
          "OR21-2",
          "KW-Nfld-22-25",
          "KW-2022-PI-01"
          )

kw_ages <- kw_ages  %>%
  filter(!sampleId %in% bad)
```

Take a look at the duplicates:

```
# Note that KW-2019-06 was a technical replicate (from the same DNA sample), while the others were from
duplicates <- kw_ages %>%
  select(block, sampleId, Year, Location, Sex, Age_Transformed) %>%
  arrange(block) %>%
  group_by(block) %>%
  filter(n() > 1)
duplicates
```

```
## # A tibble: 41 x 6
## # Groups:   block [19]
##    block        sampleId       Year Location         Sex   Age_Transformed
##    <chr>        <chr>         <int> <chr>            <chr>           <dbl>
##  1 ARPI-2013-01 ARPI-2013-01   2013 Eclipse Sound    F                7.73
##  2 ARPI-2013-01 KW-2020-PG-21  2020 Cumberland Sound F                9.30
##  3 ARPI-2013-03 ARPI-2013-04   2013 Eclipse Sound    F                6.79
##  4 ARPI-2013-03 ARPI-2013-03   2013 Eclipse Sound    F                7.47
##  5 ARPI-2013-06 ARPI-2013-06   2013 Eclipse Sound    F               11.4
##  6 ARPI-2013-06 KW-2020-PG-06  2020 Cumberland Sound F               12.4
##  7 ARPI-2013-06 KW-2020-PG-18  2020 Cumberland Sound F               16.5
##  8 ARPI-2018-10 ARPI-2018-10   2018 Eclipse Sound    F                7.14
##  9 ARPI-2018-10 ARPI-2018-14   2018 Eclipse Sound    F                7.93
## 10 ARPI-2018-15 ARPI-2018-16   2018 Eclipse Sound    M               15.9
## # i 31 more rows
```

```
#write.csv(duplicates, "duplicate_epigenetic_ages_August2024.csv")
```

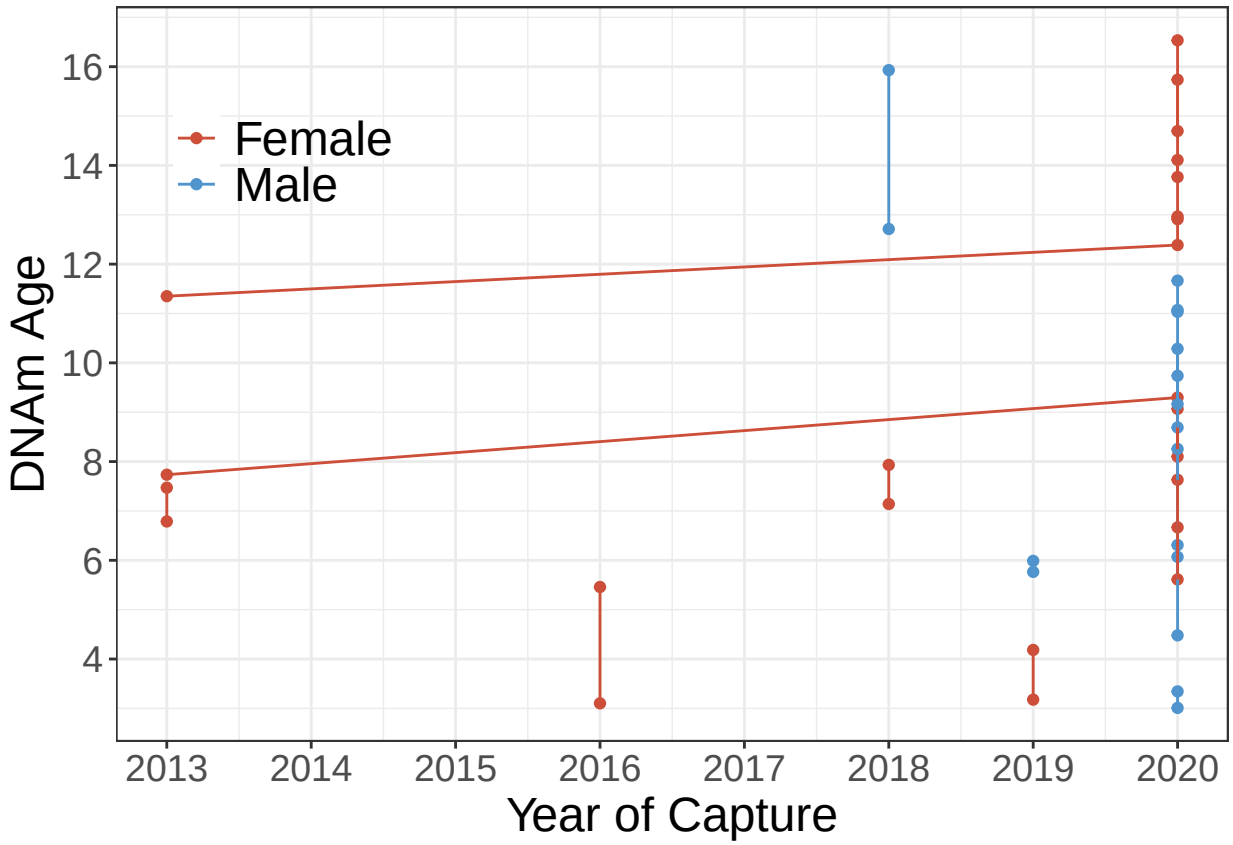
Most of the ages estimate for duplicate samples are quite close, but some of the recaptures don't reflect the difference in age between captures.

Try plotting the differences in estimated ages across time.

```
cols <- c("tomato3", "steelblue3")

# Plot difference in estimated ages across time
ggplot(duplicates, aes(x = Year,
                       y = Age_Transformed,
                       group = block,
                       #label = sampleId,
                       color = Sex)) +
  geom_point() +
  geom_line() +
  #geom_text(hjust = -0.1, vjust = -0.1, cex = 2) +
  scale_x_continuous(limits = c(2013, 2020), breaks = seq(2013, 2020, by = 1)) +
  scale_y_continuous(breaks = seq(0, 30, by = 2)) +
  scale_color_manual(values = cols,
```

```
                     labels = c("Female", "Male")) +
  labs(x = "Year of Capture", y = "DNAm Age") +
  theme_bw() +
  theme(axis.text = element_text(size=14),
        axis.title = element_text(size=18),
        legend.title = element_blank(),
        legend.text = element_text(size = 18),
        legend.position = c(0.15,0.79),
        legend.background = element_blank())
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
#ggsave("Plots/age_difference_by_year.png", width = 8, height = 6, dpi = 300)
```

We also have a few known-age individuals. Unfortunately, two of the known-age individuals (ARSQ-xx-1379, 34yo and KW-CH-2011, 35 yo) had to be removed in the quality control stage, and the 2022 sample was mixed in the great sample disaster.

We can compare the DNAm estimated age to the known age (determined from GLGs):

```
known_ages <- data.frame(
  sampleId = c("ARRB-xx-1291", "ARSQ-xx-1397"),
  Known_Age = c(28, 6))
```

```
est_ages <- kw_ages %>%
  select(sampleId, Age_Transformed) %>%
  filter(sampleId %in% c("ARRB-xx-1291", "ARSQ-xx-1397"))

compare_known_ages <- known_ages %>%
  left_join(est_ages, by = "sampleId") %>%
  mutate(Difference = Age_Transformed - Known_Age)
compare_known_ages
```

```
##       sampleId Known_Age Age_Transformed Difference
## 1 ARRB-xx-1291        28       29.251329  1.2513292
## 2 ARSQ-xx-1397         6        3.102473 -2.8975267
## 3 ARSQ-xx-1397         6        5.459565 -0.5404349
```

Take the first of duplicate samples, then remove duplicates and individuals not in the "High Arctic" population.

```
# Remove duplicates - take the first sample
kw_ages_dupsRemoved <- kw_ages %>%
  arrange(block) %>%
  mutate(duplicate = duplicated(block)) %>%
  filter(!duplicate == "TRUE")

# Keep only Northern Baffin Island locations since this is where the mark recapture is run
kw_ages_HA <- kw_ages_dupsRemoved  %>%
  filter(Location %in% c("Cumberland Sound", "Eclipse Sound")) %>%
  # And the one 2013 CS sample that grouped with the Greenland samples
  filter(!sampleId == "ARPG-2013-01")
```

Plot the data (unadjusted - age at year of capture):

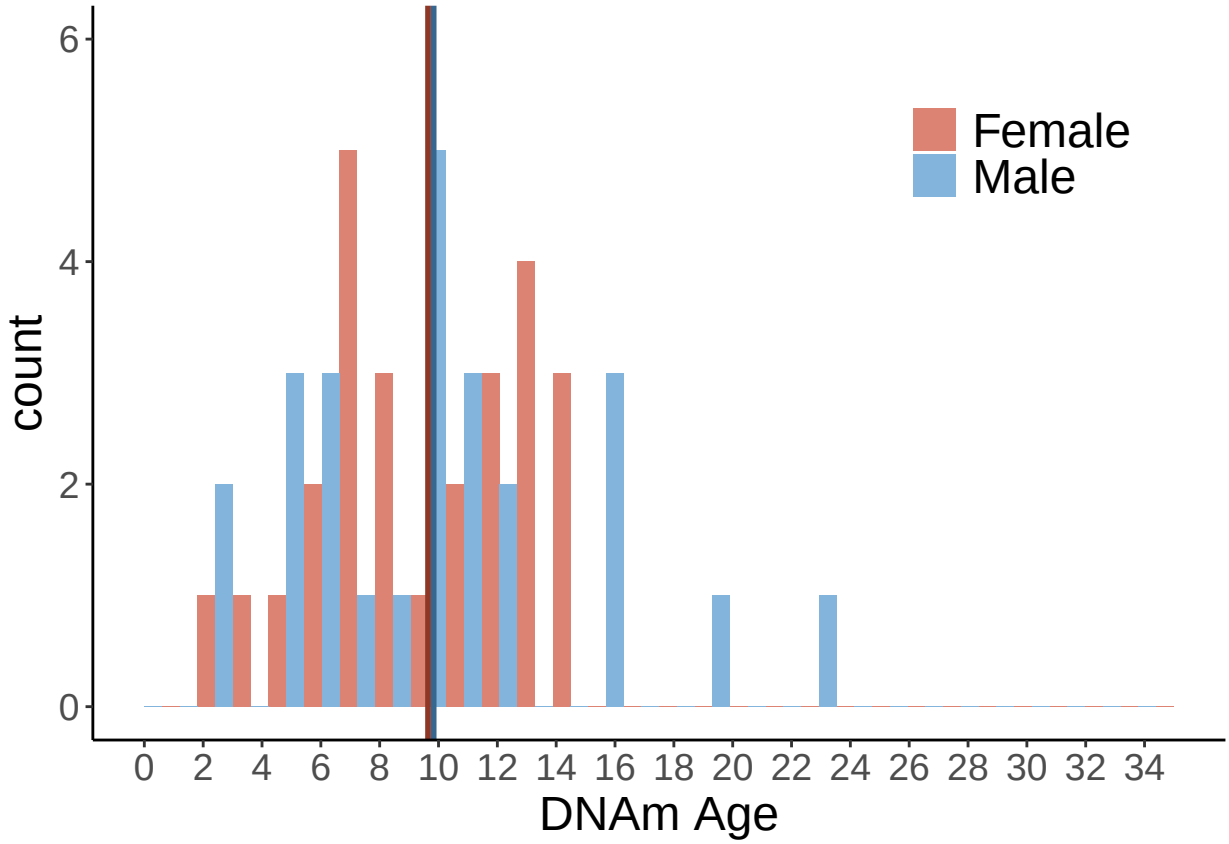
```
# Set plot colors
#cols <- c("tomato3", "steelblue3")
cols <- c("#dc8374", "#83b4dc")

# Calculate means for males & females
males <- kw_ages_HA %>%
  filter(Sex == "M")
mean_males <- mean(males$Age_Transformed)
median_males <- median(males$Age_Transformed)

females <- kw_ages_HA %>%
  filter(Sex == "F")
mean_females <- mean(females$Age_Transformed)
median_females <- median(females$Age_Transformed)

#tiff("Plots/Age_Distribution_August2024_2.tiff", units="in", width=8, height=5, res=400)

# Plot
ggplot(kw_ages_HA, aes(x = Age_Transformed, fill = Sex)) +
  geom_histogram(position = "dodge") +
  scale_x_continuous("DNAm Age", limits = c(0,35), breaks = seq(0, 35, by = 2)) +
  scale_y_continuous(limits = c(0,6), breaks = seq(0, 6, by = 2)) +
  scale_fill_manual(values = cols,
                    labels = c("Female", "Male")) +
  geom_vline(xintercept = mean_males, col = "#37678f", lty = 1, size = 1) +
```

```
  geom_vline(xintercept = mean_females, col = "#8f3727", lty = 1, size = 1) +
  theme_classic() +
  theme(axis.text = element_text(size=14),
        axis.title = element_text(size=18),
        legend.title = element_blank(),
        legend.text = element_text(size = 18),
        legend.position = c(0.82,0.8),
        legend.background = element_blank())
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` 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`.
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```




```
#dev.off()
```

To get what the age structure would be in a given year, adjust ages to year 2020:

```
kw_ages_HA <- kw_ages_HA %>%
  mutate(diff_Year = (2022 - Year)) %>%
  mutate(adj_age = (Age_Transformed + diff_Year))

#write.csv(kw_ages_HA, "kw_ages_HA_August2024.csv")
```

Plot the adjusted data (age corrected to 2020):

```
# Calculate means for males & females
mean_adj_males <- mean(males$adj_age)
```

```
## Warning in mean.default(males$adj_age): argument is not numeric or logical:
## returning NA
```

```
median_adj_males <- median(males$adj_age)

mean_adj_females <- mean(females$adj_age)
```

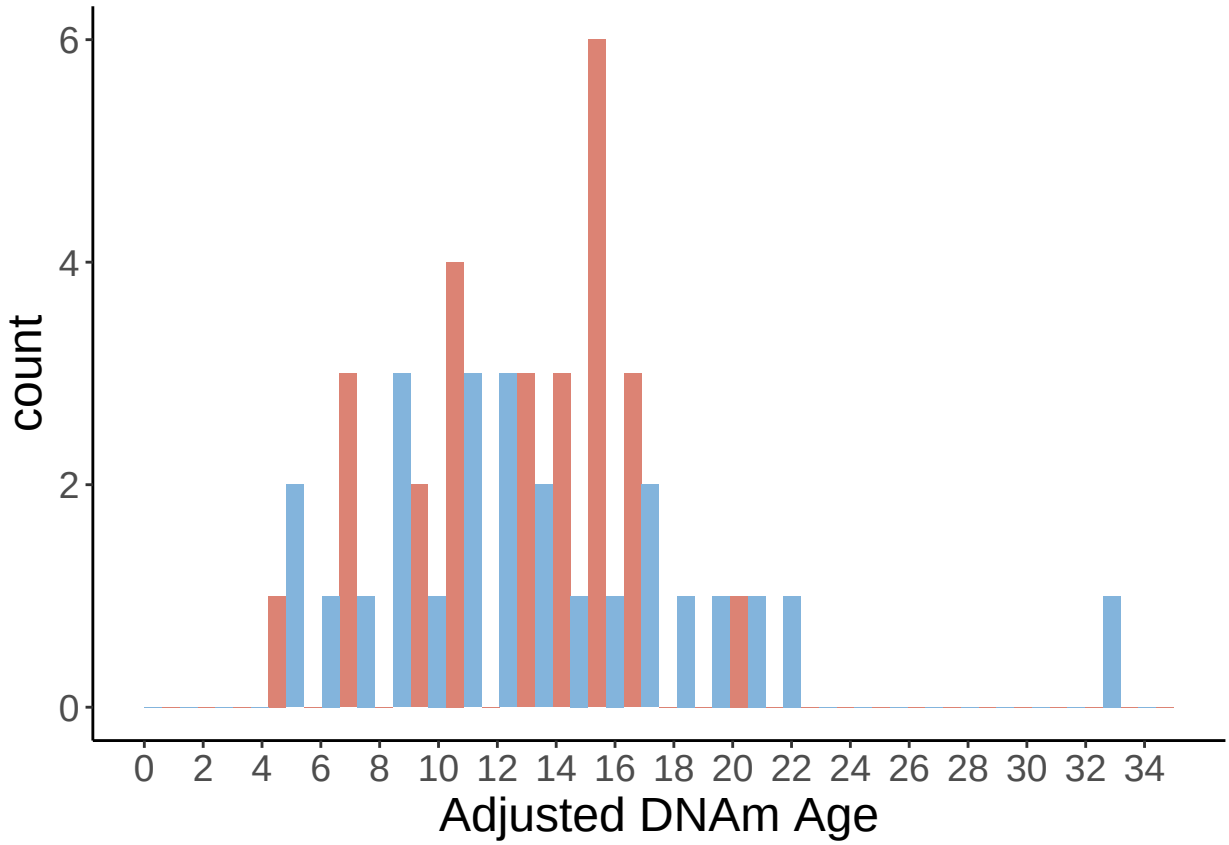
```
## Warning in mean.default(females$adj_age): argument is not numeric or logical:
## returning NA
```

```
median_adj_females <- median(females$adj_age)

#tiff("Plots/Adj_Age_Distribution_August2024_2.tiff", units="in", width=8, height=5, res=400)

# Plot
ggplot(kw_ages_HA, aes(x = adj_age, fill = Sex)) +
  geom_histogram(position = "dodge") +
  xlab("DNAm Age") +
  scale_x_continuous("Adjusted DNAm Age", limits = c(0,35), breaks = seq(0, 35, by = 2)) +
  scale_y_continuous(limits = c(0,6), breaks = seq(0, 6, by = 2)) +
  scale_fill_manual(values = cols,
                    labels = c("Female", "Male")) +
  geom_vline(xintercept = mean_adj_males, col = "#37678f", lty = 1, size = 1) +
  #geom_vline(xintercept = median_males, col = "steelblue3", lty = 2) +
  geom_vline(xintercept = mean_adj_females, col = "#8f3727", lty = 1, size = 1) +
  #geom_vline(xintercept = median_females, col = "tomato3", lty = 2) +
  theme_classic() +
  theme(axis.text = element_text(size=14),
        axis.title = element_text(size=18),
        legend.title = element_blank(),
        legend.position = "none")
```

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_vline()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_vline()`).
```




```
#dev.off()
```

Summarize data:

```
# Number of samples
length(kw_ages_HA$sampleId)
```

```
## [1] 51
```

```
# Summary of unadjusted age
summary(kw_ages_HA$Age_Transformed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.756   6.707   9.355   9.738  12.303  23.258
```

```
# Summary of adjusted age
summary(kw_ages_HA$adj_age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.756   9.865  12.765  13.110  15.921  32.258
```

```
# Summary for males:
males <- kw_ages_HA %>%
  filter(Sex == "M")

length(males$sampleId)
```

```
## [1] 25
```

```
summary(males$Age_Transformed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.756   5.988   9.738   9.835  11.666  23.258
```

```
summary(males$adj_age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.756   9.030  12.038  13.235  17.161  32.258
```

```
# Proportion of males
length(males$adj_age)/length(kw_ages_HA$adj_age)
```

```
## [1] 0.4901961
```

```
# Summary for females
females <- kw_ages_HA %>%
  filter(Sex == "F")

length(females$sampleId)
```

```
## [1] 26
```

```
summary(females$Age_Transformed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.961   7.192   9.025   9.644  12.755  14.853
```

```
summary(females$adj_age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.961  10.417  13.860  12.990  15.827  20.350
```

```
# Proportion of females
length(females$adj_age)/length(kw_ages_HA$adj_age)
```

```
## [1] 0.5098039
```

```
# Juveniles
kw_ages_HA %>%
  dplyr::count(adj_age < 13)
```

```
##   adj_age < 13  n
## 1        FALSE 25
## 2         TRUE 26
```

```
26/51
```

```
## [1] 0.5098039
```

```
# Reproductive adults
kw_ages_HA %>%
  dplyr::count(adj_age < 35)
```

```
##   adj_age < 35  n
## 1         TRUE 51
```

```
25/51
```

```
## [1] 0.4901961
```

```
# Post-reproductive adults
kw_ages_HA %>%
  dplyr::count(adj_age > 35)
```

```
##   adj_age > 35  n
## 1        FALSE 51
```

```
g <- ggplot() +
  geom_histogram( data = females,
    aes(x = Age_Transformed, y = after_stat(count)),
    fill="#dc8374") +
  geom_histogram( data = males,
    aes(x = Age_Transformed, y = -after_stat(count)),
    fill= "#83b4dc") +
  theme_bw()

#tiff("Age_Distribution_2.tiff", units="in", width=8, height=5, res=400)

age_plot <- g +
  coord_flip() +
  scale_x_continuous("DNAm Age", limits = c(0,35), breaks = seq(0, 35, by = 5)) +
  scale_y_continuous("Count", limits = c(-5,5), breaks = seq(-5,5, by = 1)) +
  geom_segment(aes(x = mean_males, xend = mean_males, y = 0, yend = -5), col = "#37678f", lty = 1, size
  geom_segment(aes(x = mean_females, xend = mean_females, y = 0, yend = 5), col = "#8f3727", lty = 1, si
  #annotate(geom = "text", label = "Male", x = 0.5, y = -5, color = "#37678f", cex = 7, hjust = 0) +
  theme(axis.text = element_text(size=14),
        axis.title = element_text(size=16))

#dev.off()
```

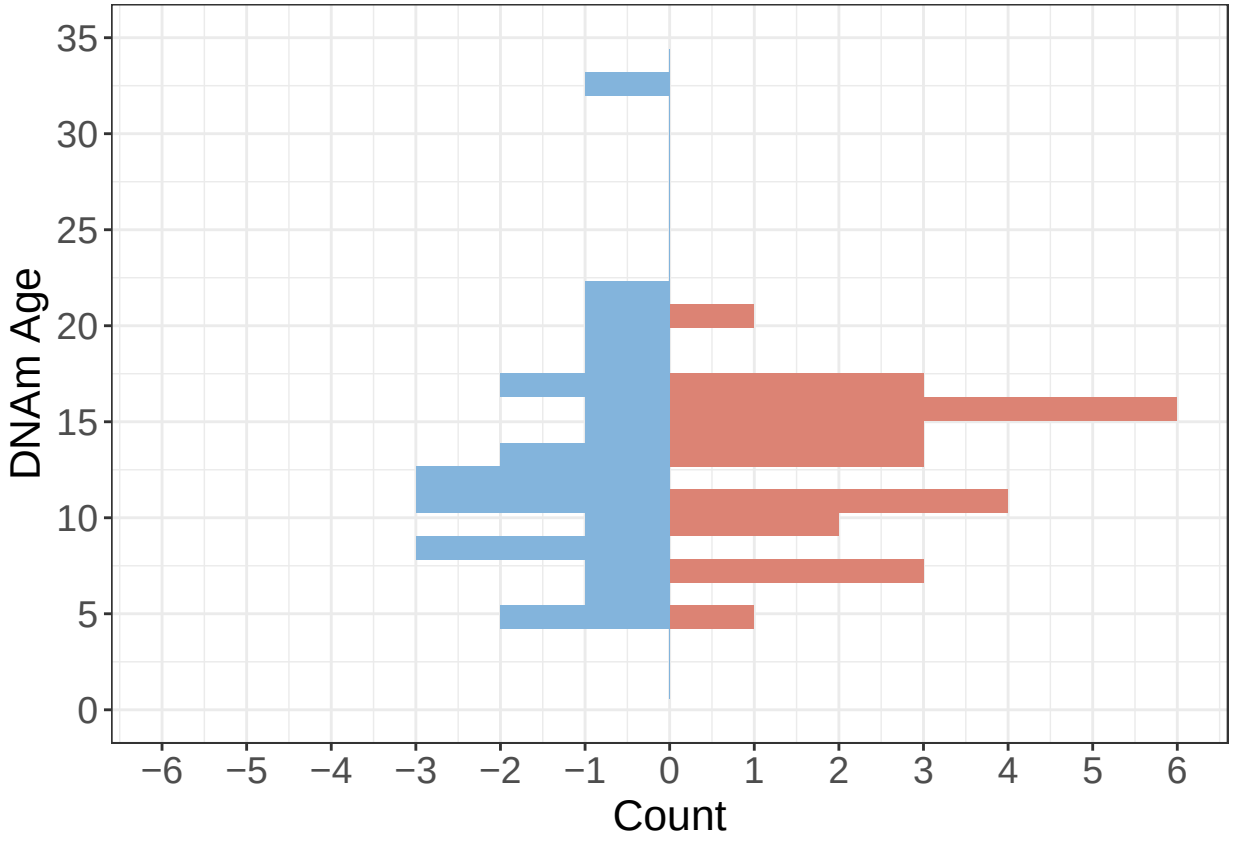
```
g2 <- ggplot() +
  geom_histogram( data = females,
    aes(x = adj_age, y = after_stat(count)),
    fill="#dc8374") +
  geom_histogram( data = males,
    aes(x = adj_age, y = -after_stat(count)),
    fill= "#83b4dc") +
  theme_bw()

#tiff("Age_Distribution_2.tiff", units="in", width=8, height=5, res=400)

g2 +
  coord_flip() +
  scale_x_continuous("DNAm Age", limits = c(0,35), breaks = seq(0, 35, by = 5)) +
  scale_y_continuous("Count", limits = c(-6,6), breaks = seq(-6,6, by = 1)) +
  geom_segment(aes(x = mean_adj_males, xend = mean_adj_males, y = 0, yend = -6), col = "#37678f", lty =
  geom_segment(aes(x = mean_adj_females, xend = mean_adj_females, y = 0, yend = 6), col = "#8f3727", lty
  #annotate(geom = "text", label = "Male", x = 0.5, y = -5, color = "#37678f", cex = 7, hjust = 0) +
  theme(axis.text = element_text(size=14),
        axis.title = element_text(size=16))
```

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

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
```

```
## (`geom_segment()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_segment()`).
```




```
#dev.off()
```