Data paper dataset processing

Museum of Southwestern Biology, University of New Mexico

This notebook documents simple manipulations of source files to prepare the master boood trait dataset for release and publication.

First, let's load tidyverse:

```r
library(tidyverse, quietly = TRUE)
```

Next, we'll load the three component .csv files: the single data file for Linck et al. *in prep*, and the two data files for Williamson et al. *in prep*.

```r
linck <- read_csv("~/Dropbox/andean_range_limits/data/blood_data.csv")
will_1 <- read_csv("~/Dropbox/andean_range_limits/data/hummingbird_spp_blood_raw.csv")
will_2 <- read_csv("~/Dropbox/andean_range_limits/data/patagona_blood_raw.csv")
```

Now, we'll manually select the columns from each of these, starting with the Linck et al. dataset.

```r
colnames_linck <- c("NK", "MSB_Cat_NUM", "Order", "Family", "Scientific name",
                    "Sex", "mass at death","Mass at capture",
                    "Mass for analyses","DAY","MONTH","YEAR",
                    "Elevation","LatDecDegS","LongDecDegW",
                    "Latitude  degrees S","Latitude minutes",
                    "Longitude degrees W",  "Longitude minutes",
                    "tHb", "tHbcorr", "Uno-corr-factor",
                    "Uno-corr-fact2","Hct-column-after", "Hct-top",
                    "Hct-bottom","Hct-middle-calculated","Hct-ratio",
                    "2ndHct-column","2ndHct-top",
                    "2ndHct-bottom","2ndHct-middle-calculated",
                    "2ndHct-ratio","wtAVG-Hct",
                    "HctBestEstimate","Blood_Notes")
linck <- linck[, names(linck) %in% colnames_linck]
```

I've included more columns than we'll use for the final dataset because we want to make that the "best" estimates for mass, hemoglobin and hematocrit haven't been left blank unnecessarily. First, let's tweak raw hemoglobin values so we can use them if we need to:

```r
linck$`tHb` <- as.numeric(linck$`tHb`)
linck$`tHbcorr` <- as.numeric(linck$`tHbcorr`)
unopette_correction_factor <- 1
linck$`tHbcorr` <- (linck$`tHbcorr`-unopette_correction_factor)
```

Now, we'll apply the `coalesce()` function to fill in the blanks. This function selects a value for its given column (e.g., `hb_final`) from the columns provided to its arguments, *in that order*. For example: the new column `mass_final` draws on data in `Mass for analyses` first, but if left blank, will next look to `Mass at capture` and then lastly `mass at death`.

```r
linck$`mass at death` <- as.numeric(linck$`mass at death`)
linck$`2ndHct-ratio` <- as.numeric(linck$`2ndHct-ratio`)
linck <- linck %>% mutate(mass_final =
                            coalesce(`Mass for analyses`, `Mass at capture`,
                                     `mass at death`))
linck <- linck %>% mutate(hct_final =
                            coalesce(`HctBestEstimate`,`wtAVG-Hct`,
                                     `2ndHct-ratio`))
linck <- linck %>% mutate(hb_final =
                            coalesce(tHb, tHbcorr))
```

Next, let's convert latitude and longitude from minutes to decimal degrees, and use `coalesce()` again to make sure we don't leave anything blank:

```
linck <- linck %>% mutate(lat_dec_deg = -((linck$`Latitude  degrees S`) +
                                            (linck$`Latitude minutes`/60)))
linck <- linck %>% mutate(long_dec_deg = -((linck$`Longitude degrees W`) +
                                             (linck$`Longitude minutes`/60)))
linck <- linck %>% mutate(lat_fin = coalesce(lat_dec_deg, `LatDecDegS`))
linck <- linck %>% mutate(long_fin = coalesce(long_dec_deg, `LongDecDegW`))
```

Let's drop columns we no longer need, add a dummy column for RBC (missing in my dataset), and standardize column names:

```
colnames_drop <- c("mass at death","Mass at capture","Mass for analyses",
                   "LatDecDegS","LongDecDegW","Latitude  degrees S","Latitude minutes",
                   "Longitude degrees W",   "Longitude minutes","tHb", "tHbcorr",
                   "Uno-corr-factor", "Uno-corr-fact2",
                   "Hct-column-after","Hct-top", "Hct-bottom",
                   "Hct-middle-calculated","Hct-ratio","2ndHct-column",
                   "2ndHct-top","2ndHct-bottom",
                   "2ndHct-middle-calculated","2ndHct-ratio","wtAVG-Hct",
                   "HctBestEstimate", "lat_dec_deg",
                   "long_dec_deg")
linck <- linck[, !names(linck) %in% colnames_drop]
linck$rbc <- NA
colnames(linck) <- c("nk","msb_cat_no","sex","order","family","species","day",
                     "month","year","elevation",
                     "notes","mass","hct","hb","lat","long", "rbc")
```

A character-to-numeric column check is always good:

```
linck$hb <- as.numeric(linck$hb)
```

Lastly, we'll scan the "Notes" field and exclude any observation with common phrases that indicate problematic data. We'll also attempt to distinguish between issues that affect Hb and Hct, but if specific blood traits are not mentioned in the comment, we'll get rid of the entire record:

```
patterns <- c("unusable","discrepancy","died","poor.*", "broken", "problem.*",
              "clot.*", "death", "failed", "bubble",
              "bad", "stressed", "switched", "not so good", "inferred", "did not",
              "didn't", "no dilution", "leaked",
              "unspun", "wouldn't", "unspun", "insufficient", "no blood")
linck_temp <- filter(linck, grepl(paste(patterns, collapse="|"), notes, ignore.case=TRUE))
linck_hb_issues <- filter(linck_temp, grepl(paste("hb", collapse="|"), notes, ignore.case=TRUE))
linck_hb_issue_index <- linck_hb_issues$nk
linck[linck$nk %in% linck_hb_issue_index,]$hb <- NA
linck_hct_issues <- filter(linck_temp, grepl(paste("hct", collapse="|"), notes, ignore.case=TRUE))
linck_hct_issue_index <- linck_hct_issues$nk
```

```r
linck[linck$nk %in% linck_hct_issue_index,]$hct <- NA
linck_issues_remainder_1 <- setdiff(linck_temp$nk, linck_hb_issue_index)
linck_issues_remainder_2 <- setdiff(linck_temp$nk, linck_hct_issue_index)
linck_issues <- coalesce(linck_issues_remainder_1, linck_issues_remainder_1)
linck_final <- linck[!linck$nk %in% linck_issues,]
```

How many records did this drop?

```r
nrow(linck) - nrow(linck_final)
```

```
## [1] 349
```

How many records and species remain?

```r
nrow(linck_final) # number of records
```

```
## [1] 5578
```

```r
linck_final$species %>% unique() %>% length() # number of species
```

```
## [1] 634
```

Next, let's process the other two hummingbird-specific datasets the same way. First, we subset columns for the full hummingbird data…

```r
colnames_will_1 <- c("nk", "msb_cat_no", "order", "family", "species",
                     "sex", "mass_at_death",
                    "mass_at_capture","mass_for_analyses","day","month",
                    "year","elev","lat_dec_deg_S",
                    "lon_dec_deg_W","lat_degrees_S","lat_mins","lon_degrees_W",
                    "lon_mins","hb", "hb_corr",
                    "uno_corr_factor", "uno_corr_factor2",
                    "Hct-column-after","hct1_column",
                    "hct1_top","hct1_bottom","0_hct1_middle_calculated",
                    "0_hct1_ratio","hct2_column",
                    "hct2_top","hct2_bottom","0_hct2_middle_calculated",
                    "0_hct2_ratio","0_wtAVG_hct",
                    "hct_best","blood_notes","RBCx10^6mm^3","RBC2",
                    "RBC_best_estimate")
will_1 <- will_1[, names(will_1) %in% colnames_will_1]
```

…then we correct raw Hb values, and consolidate columns:

```r
will_1$hb <- (will_1$hb-unopette_correction_factor)
will_1$mass_at_death <- as.numeric(will_1$mass_at_death)
will_1$`0_wtAVG_hct` <- as.numeric(will_1$`0_wtAVG_hct`)
will_1$`0_hct2_ratio` <- as.numeric(will_1$`0_hct2_ratio`)
will_1 <- will_1 %>% mutate(mass_final = coalesce(mass_for_analyses,
                                                  mass_at_capture, mass_at_death))
will_1 <- will_1 %>% mutate(hct = coalesce(hct_best, `0_wtAVG_hct`, `0_hct2_ratio`))
will_1 <- will_1 %>% mutate(hb = coalesce(hb_corr, hb))
will_1 <- will_1 %>% mutate(rbc = coalesce(`RBC_best_estimate`,`RBCx10^6mm^3`,`RBC2`))
will_1 <- will_1 %>% mutate(lat_dec_deg = -((will_1$lat_degrees_S) +
                                              (will_1$lat_mins/60)))
will_1 <- will_1 %>% mutate(long_dec_deg = -((will_1$lon_degrees_W) +
                                               (will_1$lon_mins/60)))
will_1 <- will_1 %>% mutate(lat = coalesce(lat_dec_deg, `lat_dec_deg_S`))
will_1 <- will_1 %>% mutate(long = coalesce(long_dec_deg, `lon_dec_deg_W`))
```

…trim down to just the data we need, and reorder columns to match the `linck` dataset…

```
colnames_drop <- c("mass_at_death","mass_at_capture","mass_for_analyses", "lat_dec_deg_S",
                   "lon_dec_deg_W","lat_degrees_S","lat_mins","lon_degrees_W",
                   "lon_mins","hb_corr","uno_corr_factor",
                   "uno_corr_factor2","hct1_column","hct1_top","hct1_bottom",
                   "0_hct1_middle_calculated","0_hct1_ratio",
                   "hct2_column","hct2_top","hct2_bottom","0_hct2_middle_calculated",
                   "0_hct2_ratio","0_wtAVG_hct",
                   "hct_best","lat_dec_deg","long_dec_deg","RBCx10^6mm^3",
                   "RBC2","RBC_best_estimate")
will_1 <- will_1[, !names(will_1) %in% colnames_drop]
will_1 <- will_1[, c(1, 2, 6, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14, 11, 16, 17,15)]
colnames(will_1) <- c("nk","msb_cat_no","sex","order","family","species",
                      "day","month","year","elevation",
                     "notes","mass","hct","hb","lat","long","rbc")
```

…and drop problematic ata by scanning the notes field:

```
patterns <- c("unusable","discrepancy","died","poor.*", "broken", "problem.*",
              "clot.*", "death", "failed", "bubble",
              "bad", "stressed", "switched", "not so good", "inferred", "did not",
              "didn't", "no dilution", "leaked",
              "unspun", "wouldn't", "unspun", "insufficient", "no blood")
will_1_temp <- filter(will_1, grepl(paste(patterns, collapse="|"), notes, ignore.case=TRUE))
will_1_hb_issues <- filter(will_1_temp, grepl(paste("hb", collapse="|"), notes, ignore.case=TRUE))
will_1_hb_issue_index <- will_1_hb_issues$nk
will_1[will_1$nk %in% will_1_hb_issue_index,]$hb <- NA
will_1_hct_issues <- filter(will_1_temp, grepl(paste("hct", collapse="|"), notes, ignore.case=TRUE))
will_1_hct_issue_index <- will_1_hct_issues$nk
will_1[will_1$nk %in% will_1_hct_issue_index,]$hct <- NA
will_1_issues_remainder_1 <- setdiff(will_1_temp$nk, will_1_hb_issue_index)
will_1_issues_remainder_2 <- setdiff(will_1_temp$nk, will_1_hct_issue_index)
will_1_issues <- coalesce(will_1_issues_remainder_1, will_1_issues_remainder_1)
will_1_final <- will_1[!will_1$nk %in% will_1_issues,]
```

How many records lost?

```
nrow(will_1) - nrow(will_1_final)
```

```
## [1] 102
```

How many records and species remain?

```
nrow(will_1) # number of records
```

```
## [1] 1201
```

```
will_1$species %>% unique() %>% length() # number of species
```

```
## [1] 77
```

Now, we subset Jessie's super special *Patagona* data to columns that match the other datasets, and add dummy variables for the info we're missing:

```
colnames_will_2 <- c("museum_cat_num","nk","species","elev","month","year","lat",
                     "lon","sex", "mass","hb", "hct","TRBC")
will_2 <- will_2[, names(will_2) %in% colnames_will_2]
will_2$order <- "Apodiformes"
```

```
will_2$family <- "Trochilidae"
will_2$day <- NA
will_2$notes <- NA
will_2$species <- gsub("_", " ", will_2$species)
will_2 <- will_2[, c(2, 1, 9, 14, 15, 3, 16, 5, 6, 4, 17, 10, 12, 11, 7, 8, 13)]
colnames(will_2) <- c("nk","msb_cat_no","sex","order","family","species",
                      "day","month","year","elevation",
                     "notes","mass","hct","hb","lat","long","rbc")
will_2_final <- will_2
```

Let's remove the prefix "MSB:Birds" so the catalog numbers match our other data:

```
will_2_final$msb_cat_no <-
  sapply(strsplit(will_2_final$msb_cat_no, split=':'), "[", 3) %>%
  as.numeric()
```

We next merge all three datasets, removing duplicate rows:

```
# bind together
df1 <- rbind.data.frame(linck_final, will_1_final, will_2_final)

# function to squish rows together
coalesce_all_columns <- function(df) {
  return(coalesce(!!! as.list(df)))
}

# do the squishing
df2 <- df1 %>%
  group_by(nk) %>%
  summarise_all(coalesce_all_columns)

# make sure hb is numeric
df2$hb <- as.numeric(df2$hb)
```

Finally, we calculate secondary blood indices, add an Arctos URL, reorder in a pleasing way, and export as a `.csv`:

```
df2 <- df2 %>% mutate(hct_percent = hct*100)
df2 <- df2 %>% mutate(mchc = (hb/hct_percent)*100)
df2 <- df2 %>% mutate(mcv = (hct_percent/rbc)*10)
df2 <- df2 %>% mutate(mch = (hb/rbc)*10)
df2$arctos_url <- NA
df2$arctos_url <- paste0("https://arctos.database.museum/guid/","MSB:Bird:",df2$msb_cat_no)
df2$arctos_url[df2$arctos_url=="https://arctos.database.museum/guid/MSB:Bird:NA"] <- NA
blood_data_final <- df2[, c(1,2,22,4,5,6,7,8,9,15,16,10,3,12,14,13,18,17,19,20,21,11)]
blood_data_final <- blood_data_final[order(blood_data_final$species),]
write.csv(blood_data_final, "~/Dropbox/andean_range_limits/data/blood_data_final.csv")
```

What's it look like?

```
head(blood_data_final)
```

```
## # A tibble: 6 x 22
##       nk msb_cat_no arctos_url order family species   day month  year    lat
##    <dbl>      <dbl> <chr>      <chr> <chr>  <chr>   <dbl> <chr> <dbl>  <dbl>
## 1 175040      36209 https://a~ Acci~ Accip~ Accipi~    17 June   2011  -7.42
## 2 176716      41684 https://a~ Acci~ Accip~ Accipi~     9 June   2012  -5.21
## 3 175425      36494 https://a~ Acci~ Accip~ Accipi~    29 June   2011  -7.41
## 4 161047      27280 https://a~ Apod~ Troch~ Adelom~     2 April  2007 -13.1
## 5 161071      27303 https://a~ Apod~ Troch~ Adelom~     3 April  2007 -13.1
## 6 161072      27304 https://a~ Apod~ Troch~ Adelom~     3 April  2007 -13.1
## # ... with 12 more variables: long <dbl>, elevation <dbl>, sex <chr>,
## #   mass <dbl>, hb <dbl>, hct <dbl>, hct_percent <dbl>, rbc <dbl>, mchc <dbl>,
## #   mcv <dbl>, mch <dbl>, notes <chr>
```

How many records and species?

```r
nrow(blood_data_final) # number of records
```

```
## [1] 5769
```

```r
blood_data_final$species %>% unique() %>% length() # number of species
```

```
## [1] 635
```