# Filtering and summarizing pollen data for analysis

Kara Leimberger

## Step 1: Import pollen data

In this dataset, pollen samples were collected from captured hummingbirds and mounted on microscope slides ("sample" = "slide"). Since pollen grains on a hummingbird suggest that they visited the plant associated with those pollen grains ("pollen morphotype"), this is another way to quantify plant-hummingbird interactions - but without directly observing the interaction.

Pollen slides were then reviewed and any pollen morphotypes were identified; this is the resulting dataset. At a minimum, there is one row per microscope slide. If >1 pollen types were observed on a slide, then there are additional rows per slide.

```
# 1. Rename 'site' in anticipation of matching column name in camera data
# (patch = site) 2. Make a column for the year + patch combination (year_patch
# = replicate)
pollen <- read.csv("../data/import/Pollen_data_2012-2018_20220714.csv") %>%
    rename(patch = site) %>%
    mutate(year_patch = paste(year, patch, sep = "_"))
```

## Step 2: Process data

### Initial filtering

```
# Imported data is All pollen data, from all years of the CR hummingbird
# project. Here, I am only interested in the years associated with my project.
# Also need to: 1. Remove pollen slides from sites outside of experiment (i.e.,
# captures in P33/garden) 2. Remove pollen slides not associated with a
# particular hummingbird species (need species to create networks!) 3. Remove
# unmarked birds to prevent unintentional resampling of birds recaptured within
# same capture session
pollen02 <- pollen %>%
    filter(year >= 2016) %>%
    filter(!is.na(control_treatment)) %>%
    filter(!is.na(bird_species)) %>%
    filter(!is.na(band_number) | (is.na(band_number) & !is.na(colors)))
```

### Add up pollen counts from same morphotypes within a slide ID

```
# Identify slides with multiple rows for the same pollen morphotype. This
# doesn't happen frequently, but rows need to be combined when it does.
check01 <- pollen02 %>%
    group_by(year, bird_id, day_bird_id, patch, control_treatment, exp_phase, date,
        slide_number, slide_id, band_number, colors, pollen_yes_or_no, bird_species,
        bird_sex, pollen_morphotype) %>%
    summarise(n = n()) %>%
    filter(n > 1)

# Summarize pollen data so that there are not duplicate pollen morphotypes on a
# single slide (pollen sample)
pollen03 <- pollen02 %>%
    select(year, patch, year_patch, bird_id, day_bird_id, control_treatment, exp_phase,
        date, slide_number, slide_id, band_number, colors, pollen_yes_or_no, bird_species,
        bird_sex, contains("name"), pollen_morphotype, record_in_capture_data, collection_time,
        pollen_count) %>%
    group_by(across(!pollen_count)) %>%
    summarise(pollen_count = sum(as.numeric(pollen_count), na.rm = TRUE)) %>%
    ungroup()
```

### Remove pollen samples that do not correspond to any captures in capture dataset

This is an additional quality check that was created during an earlier processing stop. Here, I filter out any pollen samples that couldn't be matched to a corresponding bird in the capture dataset. Records were matched based on year, site, bird identifier (either band number or color marks), date. The resulting column, 'record_in_capture_data', is intended to be a flag for data quality and three options:

- Yes = everything looks good!
- No = mismatch between pollen and capture data. Mismatches suggest that there was an error in either the pollen slide label (and subsequently in the pollen dataest) or a typo in the capture dataset. Any mismatches from 2016-2018 have been investigated and resolved when possible; the same attention has not been dedicated to the earlier years.
- NA = not enough identifying information (i.e., band number or color marks) to make the link between the two datasets.

```
# Remove samples where there is a clear issue. Could also remove the NA
# situations, but this risks removing important data. Given the overall low
# number of captures (samples) for this analysis, I've decided to err on the
# side of only removing known problems.
pollen04 <- pollen03 %>%
    filter(record_in_capture_data != "no")
```

### Deal with 'day recaptures', i.e., multiple captures of same bird (and therefore multiple pollen samples) from same capture session

I want to only sample each bird once per capture session. Birds with >1 capture per capture session (and ultimately >1 pollen sample) = "day recaps"

```
# How many day recaps are there in the pollen dataset? Need to identify these
# birds. Steps: 1. Just look at slides that be traced back to particular bird
# based on band number/color mark 2. Calculate number of slides per individual
# bird 3. Filter to birds with >1 pollen sample (slide) per capture session.
# These are the day recaptures.
day_recaps <- pollen04 %>%
    filter(!is.na(band_number) | (is.na(band_number) & !is.na(colors))) %>%
    distinct(year, patch, control_treatment, day_bird_id, bird_id, slide_id) %>%
    group_by(year, patch, control_treatment, day_bird_id, bird_id) %>%
    summarise(num_slides = n()) %>%
    ungroup() %>%
    filter(num_slides > 1)
```

There are 14 birds with >1 sample per capture session. How to resolve?

- In 2018, slides have associated timestamp. **DECISION:** use sample from first capture of the day
- In 2016-2017, slides do not have any timestamps. **DECISION:** randomly choose sample

```
# Get capture times from capture data and order by time captured. For initial
# captures, order = 1
capture_times_2018 <- pollen04 %>%
    filter(year == "2018" & !is.na(collection_time)) %>%
    mutate(collection_time = lubridate::hm(collection_time)) %>%
    distinct(year, patch, day_bird_id, bird_id, collection_time, slide_number) %>%
    group_by(year, patch, day_bird_id) %>%
    arrange(collection_time) %>%
    mutate(order = 1:length(day_bird_id)) %>%
    ungroup() %>%
    arrange(year, patch, day_bird_id, order)

# Identify slides that will be removed
slides_to_remove_2018 <- capture_times_2018 %>%
    filter(order > 1) %>%
    mutate(slide_id = paste(year, slide_number, sep = "-"))

# For remaining slides, randomly pick. dplyr::sample_n randomly samples a
# dataframe - will keep 1 slide per bird Need to start with pollen data to get
# list of slides associated with day recaps
set.seed(1)
day_recaps_slides_2016_2017 <- pollen04 %>%
    filter(day_bird_id %in% day_recaps$day_bird_id) %>%
    filter(year == "2016" | year == "2017") %>%
    distinct(year, patch, control_treatment, date, band_number, day_bird_id, bird_id,
        slide_id)

slides_to_keep_2016_2017 <- day_recaps_slides_2016_2017 %>%
    group_by(year, patch, control_treatment, date, band_number, day_bird_id, bird_id) %>%
    sample_n(1)

# Need this step because one bird was captured three times in same day (can't
# just keep one each -- need to frame in terms of slides to remove)
slides_to_remove_2016_2017 <- day_recaps_slides_2016_2017 %>%
    filter(!(slide_id %in% slides_to_keep_2016_2017$slide_id))

# Remove duplicate slides associated with day recaptures
pollen05 <- pollen04 %>%
    filter(!(slide_id %in% slides_to_remove_2018$slide_id)) %>%
    filter(!(slide_id %in% slides_to_remove_2016_2017$slide_id))

# Check for day recaptures again
check02 <- pollen05 %>%
    distinct(year, patch, control_treatment, date, day_bird_id, bird_id, slide_id) %>%
    group_by(year, patch, control_treatment, date, day_bird_id, bird_id) %>%
    summarise(num_slides = n()) %>%
    ungroup() %>%
    filter(num_slides > 1)
```

### Create dataset for experiment

Remove samples with no pollen

```
# Filter to data from Heliconia removal experiment (there was a pollen slide
# from the establishment period, which needed to be removed)
data_experiment <- pollen05 %>%
    filter(exp_phase == "capture_1" | exp_phase == "capture_2")

# How many total slides?
(total_slides <- unique(data_experiment$slide_id) %>%
    length())
```

```
## [1] 318
```

```
# How many unique hummingbirds?
(total_hbird_indls <- unique(data_experiment$bird_id) %>%
    length())
```

```
## [1] 291
```

```
# How many slides with no pollen?
slides_no_pollen <- data_experiment %>%
    filter(pollen_yes_or_no == "N")

(total_slides_no_pollen <- unique(slides_no_pollen$slide_id) %>%
    length())
```

```
## [1] 12
```

```
# What percentage of slides have pollen?
1 - (total_slides_no_pollen/total_slides)
```

```
## [1] 0.9622642
```

```
# Remove slides with no pollen, because these slides do not provide any
# information about what flowers hummingbirds are visiting
data_experiment <- data_experiment %>%
    filter(!(slide_id %in% slides_no_pollen$slide_id))
```

Indicate which birds are pre-post recaps, i.e., caught in both capture sessions (and have pollen data)

```
# Individual birds with pollen data pre and post
pre_post_recaps <- data_experiment %>%
    distinct(year, patch, control_treatment, exp_phase, band_number, bird_species,
        bird_id, slide_id) %>%
    group_by(year, patch, control_treatment, exp_phase, band_number, bird_species,
        bird_id) %>%
    summarise(num_slides = n()) %>%
    ungroup() %>%
    pivot_wider(names_from = exp_phase, values_from = num_slides) %>%
    filter(capture_1 == 1 & capture_2 == 1)

# Add recap status and bird group to data. Bird group is useful because I'll
# want to analyze Heliconia specialists (green hermits and violet sabrewings)
# separately from rest of hummingbird species
data_experiment <- data_experiment %>%
    mutate(recap_y_n = ifelse(bird_id %in% pre_post_recaps$bird_id, "yes", "no")) %>%
    mutate(bird_group = ifelse(bird_species == "GREH" | bird_species == "VISA", "greh_visa",
        "other"))

# How many recaps were there? Recaptures of all species
recap_summary <- data_experiment %>%
    distinct(slide_id, bird_species, bird_id, exp_phase, recap_y_n) %>%
    group_by(recap_y_n, exp_phase) %>%
    summarise(num_slides = n())

recap_summary
```

```
## # A tibble: 4 x 3
## # Groups:   recap_y_n [2]
##   recap_y_n exp_phase num_slides
##   <chr>     <chr>          <int>
## 1 no        capture_1        140
## 2 no        capture_2        112
## 3 yes       capture_1         27
## 4 yes       capture_2         27
```

```
# Recaptures of GREH/VISA only
recap_summary_greh_visa <- data_experiment %>%
    filter(bird_group == "greh_visa") %>%
    distinct(recap_y_n, bird_group, exp_phase, slide_id) %>%
    group_by(recap_y_n, bird_group, exp_phase) %>%
    summarise(num_slides = n())

recap_summary_greh_visa
```

```
## # A tibble: 4 x 4
## # Groups:   recap_y_n, bird_group [2]
##   recap_y_n bird_group exp_phase num_slides
##   <chr>     <chr>      <chr>          <int>
## 1 no        greh_visa  capture_1         63
## 2 no        greh_visa  capture_2         34
## 3 yes       greh_visa  capture_1         16
## 4 yes       greh_visa  capture_2         16
```

## Step 3: Summarize data for analysis

These data will be be used for multiple analyses that largely mirror those of the camera data (but see #3, below).

To build interaction networks, I first need an interaction rate between each hummingbird species and plant species. Here, the interaction rate will be the umber of interactions between each hummingbird species and pollen morphotype ( = plant type/species), following other studies of hummingbird pollen networks (e.g., Ramirez-Burbano et al. 2017, Maglianesi et al. 2015, Morrison & Mendenhall 2020).

To calculate this rate, I will calculate the number of times that a pollen morphotype appears on each hummingbird species, regardless of the number of pollen grains on the bird.

At this point it's also good to think ahead about the different datasets I'll need:

1. To understand how hummingbird visitation changes as result of our experimental manipulation (Heliconia removal), I'll need to summarize data at the level of EXPERIMENTAL PERIOD (pre vs. post) and REPLICATE (i.e., site + year combination). To explore how sampling method (camera observations vs. pollen samples) influences network metrics from individual networks, I can use this same dataset, but filtered to the 'pre' period only.

2. To visualize 'normal' interactions within the study system, I'll need to just look at unmanipulated data from the 'pre' period. Here, I am interested in the interactions across sites and years. This is the "meta-network" (network of networks) approach.

3. Individual specialization = number of pollen morphotypes per individual hummingbird, i.e., pollen richness. This is not a network measure.

### Summary for network analysis

Number of times a hummingbird carried the pollen of each plant species

```r
# Interaction frequency = number of times a hummingbird carried the pollen of
# each plant species, i.e., number of slides containing a given pollen
# morphotype
network_data_experiment <- data_experiment %>%
    group_by(year, patch, year_patch, control_treatment, exp_phase, bird_species,
        pollen_morphotype) %>%
    summarise(num_slides = n()) %>%
    ungroup()

# This is similar to summary for experiment...EXCEPT that it is not summarized
# to level of pre/post. This is the 'meta-network', i.e., summary network of
# sub-sampled sites/networks
metanetwork_data <- data_experiment %>%
    filter(exp_phase == "capture_1") %>%
    group_by(bird_species, pollen_morphotype) %>%
    summarise(num_slides = n()) %>%
    ungroup()
```

### Summary for individual specialization

Number of pollen morphotypes per individual bird (richness of flower types per sample)

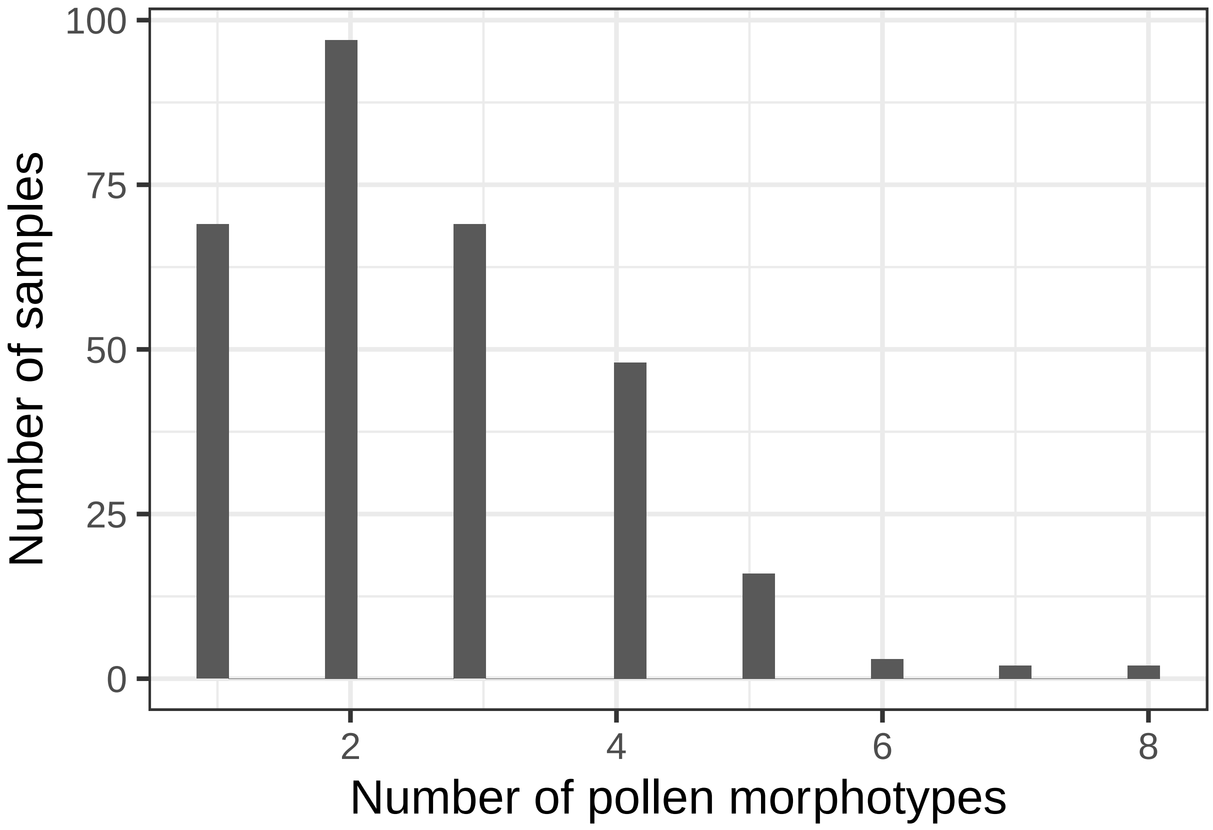
```
# Remember, slide = microscope slide = pollen sample
morphotypes_per_slide_experiment <- data_experiment %>%
    select(year, patch, control_treatment, exp_phase, date, slide_number, slide_id,
        bird_id, day_bird_id, band_number, bird_species, bird_sex, recap_y_n, pollen_yes_or_no,
        pollen_morphotype) %>%
    group_by(year, patch, control_treatment, exp_phase, date, slide_number, slide_id,
        bird_id, day_bird_id, band_number, bird_species, bird_sex, recap_y_n, pollen_yes_or_no) %>%
    summarise(num_morphotypes = n()) %>%
    ungroup() %>%
    arrange(num_morphotypes)

max(morphotypes_per_slide_experiment$num_morphotypes)
```

```
## [1] 8
```

```
# Max of 8 morphotypes per slide, including unknown morphotypes

# Plot morphotypes per bird - including birds with no pollen (EXPERIMENT)
ggplot(aes(x = num_morphotypes), data = morphotypes_per_slide_experiment) + geom_histogram() +
    theme_bw(base_size = 18) + labs(x = "Number of pollen morphotypes", y = "Number of samples")
```



### Step 4: Summarize data for general results about hummingbird natural history

Decided to just look at 'normal' data, but results are basically the same if decide to include all slides (i.e., treatment post)

```
# Note: have already removed birds with no pollen, so number of morphotypes
# will always be >0
morphotypes_per_slide_normal <- morphotypes_per_slide_experiment %>%
    filter(exp_phase == "capture_1")

data_normal_visitation <- data_experiment %>%
    filter(exp_phase == "capture_1")
```

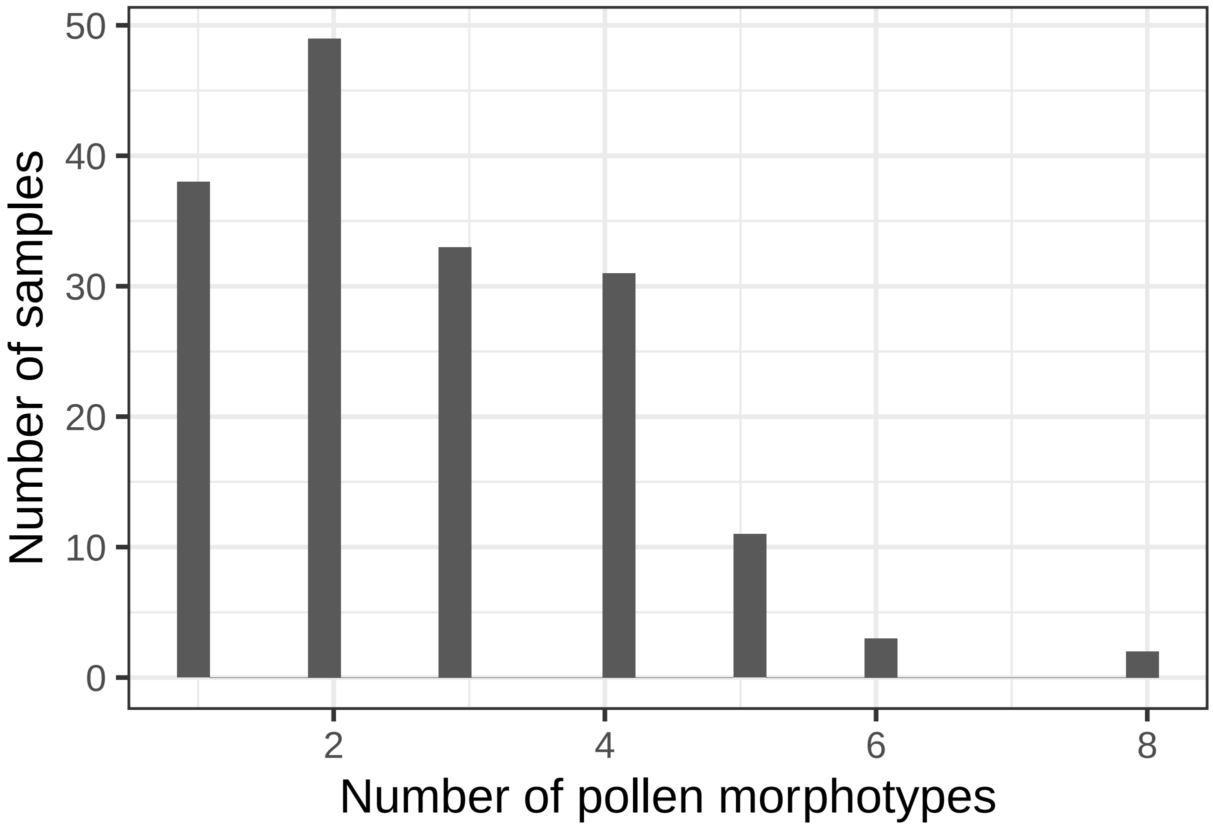
**How many pollen morphotypes per bird?**

```
# Summary of number of morphotypes per bird
num_morphotypes_per_bird_sum <- morphotypes_per_slide_normal %>%
    summarise(mean = mean(num_morphotypes), sd = sd(num_morphotypes), median = median(num_morphotypes),
        min = min(num_morphotypes), max = max(num_morphotypes))

num_morphotypes_per_bird_sum
```

```
## # A tibble: 1 x 5
##    mean    sd median   min   max
##   <dbl> <dbl>  <int> <int> <int>
## 1  2.68  1.42      2     1     8
```

```
# Plot morphotypes per bird
ggplot(aes(x = num_morphotypes), data = morphotypes_per_slide_normal) + geom_histogram() +
    theme_bw(base_size = 18) + labs(x = "Number of pollen morphotypes", y = "Number of samples")
```



### Of the pollen samples (slides) with Heliconia pollen, how many are from GREH/VISA?

```
# One row per slide with HETO
slides_with_heto_pollen <- data_normal_visitation %>%
    filter(pollen_morphotype == "HELICONIA01")

(num_slides_with_heto_pollen = unique(slides_with_heto_pollen$slide_id) %>%
    length())
```

```
## [1] 129
```

```
sum_slides_with_heto_pollen <- slides_with_heto_pollen %>%
    distinct(slide_id, bird_group) %>%
    group_by(bird_group) %>%
    summarise(num_slides = n()) %>%
    ungroup() %>%
    mutate(prop_of_total = num_slides/num_slides_with_heto_pollen)

sum_slides_with_heto_pollen
```

```
## # A tibble: 2 x 3
##   bird_group num_slides prop_of_total
##   <chr>           <int>         <dbl>
## 1 greh_visa          78         0.605
## 2 other              51         0.395
```

### Of the pollen samples (slides) from GREH/VISA, how many contain Heliconia pollen?

```
slides_greh_visa <- data_normal_visitation %>%
    filter(bird_group == "greh_visa") %>%
    distinct(year, patch, control_treatment, exp_phase, bird_group, bird_species,
        slide_id, pollen_morphotype)

(num_slides_greh_visa = unique(slides_greh_visa$slide_id) %>%
    length())
```

```
## [1] 79
```

```
slides_greh_visa_with_heto_pollen <- slides_greh_visa %>%
    filter(pollen_morphotype == "HELICONIA01")

(num_slides_greh_visa_with_heto_pollen = unique(slides_greh_visa_with_heto_pollen$slide_id) %>%
    length())
```

```
## [1] 78
```

```
# Proportion of total
num_slides_greh_visa_with_heto_pollen/num_slides_greh_visa
```

```
## [1] 0.9873418
```

```
# There is only one bird that does not have Heliconia pollen? Yes.
lone_bird <- slides_greh_visa %>%
    filter(!(slide_id %in% slides_greh_visa_with_heto_pollen$slide_id))
```

**Of the pollen samples (slides) from GREH/VISA, how many contain ONLY Heliconia pollen?**

```
# Slides where Heliconia is only morphotype present
slides_greh_visa_with_heto_pollen_only <- morphotypes_per_slide_normal %>%
    filter(slide_id %in% slides_greh_visa_with_heto_pollen$slide_id) %>%
    filter(num_morphotypes == 1)

(num_slides_greh_visa_with_heto_pollen_only = unique(slides_greh_visa_with_heto_pollen_only$slide_id) %
    length())
```

```
## [1] 15
```

```
# Proportion of total that had Heliconia as only pollen type
num_slides_greh_visa_with_heto_pollen_only/num_slides_greh_visa
```

```
## [1] 0.1898734
```

```
# Proportion of total that did *not* have Heliconia as only pollen type
1 - num_slides_greh_visa_with_heto_pollen_only/num_slides_greh_visa
```

```
## [1] 0.8101266
```

**Of the pollen samples (slides) from GREH/VISA, how many contain Heliconia pollen + 1 other type?**

```
# Slides that have Heliconia + one other morphotype
num_slides_greh_visa_with_heto_pollen_plus_1_other <- morphotypes_per_slide_normal %>%
    filter(slide_id %in% slides_greh_visa_with_heto_pollen$slide_id) %>%
    filter(num_morphotypes == 2)

(num_slides_greh_visa_with_heto_pollen_plus_1_other = unique(num_slides_greh_visa_with_heto_pollen_plus_
    length())
```

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

```
# Proportion of total that had Heliconia as only pollen type
num_slides_greh_visa_with_heto_pollen_plus_1_other/num_slides_greh_visa
```

```
## [1] 0.3291139
```

```
# Proportion of total that did *not* have Heliconia as only pollen type
1 - num_slides_greh_visa_with_heto_pollen_plus_1_other/num_slides_greh_visa
```

```
## [1] 0.6708861
```

**Of the pollen samples (slides) from GREH/VISA, how many contain ONLY Heliconia pollen -OR- ONLY Heliconia pollen + 1 other type?**

```
(num_slides_greh_visa_with_heto_pollen_only + num_slides_greh_visa_with_heto_pollen_plus_1_other)
```

```
## [1] 41
```

```
# Proportion of total
(num_slides_greh_visa_with_heto_pollen_only + num_slides_greh_visa_with_heto_pollen_plus_1_other)/num_sl
```

```
## [1] 0.5189873
```

## Step 5: Summarize data for methods

### Summary of the overall number of morphotypes detected - across all birds - and the taxonomic resolution at which they were identified

Across ALL slides collected for experiment

```
(num_morphotypes_total <- unique(data_experiment$pollen_morphotype) %>%
    length())
```

```
## [1] 45
```

```
# All morphotypes found.
morphotypes_list <- data_experiment %>%
    distinct(plant_name_family, plant_name_genus, plant_name_species, pollen_morphotype)

# How many are completely unknown? I.e. don't even know the family?
morphotypes_unknown <- morphotypes_list %>%
    filter(is.na(plant_name_family))

# Summary of lowest taxonomic resolution. For how many morphotypes do we know
# family/genus/species?
morphotypes_known_species <- morphotypes_list %>%
    filter(!is.na(plant_name_species))

morphotypes_known_genus <- morphotypes_list %>%
    filter(!is.na(plant_name_genus)) %>%
    filter(!(pollen_morphotype %in% morphotypes_known_species$pollen_morphotype))

morphotypes_known_family <- morphotypes_list %>%
    filter(!is.na(plant_name_family)) %>%
    filter(!(pollen_morphotype %in% morphotypes_known_species$pollen_morphotype)) %>%
    filter(!(pollen_morphotype %in% morphotypes_known_genus$pollen_morphotype))

# Numbers of morphotype per category

# Completely unknown. Family not even ID'ed
(num_morphotypes_unknown <- unique(morphotypes_unknown$pollen_morphotype) %>%
    length())
```

```
## [1] 15
```

```
# ID'ed to species
(num_morphotypes_known_species <- unique(morphotypes_known_species$pollen_morphotype) %>%
    length())
```

```
## [1] 17
```

```
# ID'ed to genus, but not species
(num_morphotypes_known_genus <- unique(morphotypes_known_genus$pollen_morphotype) %>%
    length())
```

```
## [1] 7
```

```
# ID'ed to famly, but not genus or species
(num_morphotypes_known_family <- unique(morphotypes_known_family$pollen_morphotype) %>%
    length())
```

```
## [1] 6
```

```
# Calculate stats for manuscript...
(num_morphotypes_known = num_morphotypes_known_species + num_morphotypes_known_genus +
    num_morphotypes_known_family)
```

```
## [1] 30
```

```
num_morphotypes_unknown/num_morphotypes_total
```

```
## [1] 0.3333333
```

```
num_morphotypes_known_species/num_morphotypes_total
```

```
## [1] 0.3777778
```

```
num_morphotypes_known_genus/num_morphotypes_total
```

```
## [1] 0.1555556
```

```
num_morphotypes_known_family/num_morphotypes_total
```

```
## [1] 0.1333333
```

## Step 6: Export data

```
# Unsummarized data (intermediate step, not for analysis)
write.csv(data_experiment, "../data/export/intermediate/Pollen_data_filtered_for_analysis.csv")

# Summarized data for analysis

# Summarized morphotype richness
write.csv(morphotypes_per_slide_experiment, "../data/export/for_analysis/Pollen_data_summarized_for_pp_

# Interaction frequencies between plants + hummingbirds (per replicate)
write.csv(network_data_experiment, "../data/export/for_analysis/Pollen_data_summarized_for_pp_networks.

# Interaction frequencies between plants + hummingbirds (meta-network)
write.csv(metanetwork_data, "../data/export/for_analysis/Pollen_data_summarized_for_metanetwork.csv")
```