

Plant-pollinator networks worldwide are composed by the same specific building blocks

Abstract

Ecological processes leave distinct structural imprints on indirect interactions in mutualistic networks. Detecting those relationships is not trivial since they go beyond pair-wise interactions, but may get blurred when considering full network descriptors. However, recent work has shown the network meso-scale can capture this important information. The meso-scale describes network subgraphs representing patterns of interactions between a small number of species (i.e. motifs) and those constitute the building blocks of the whole network. Here we have compiled 60 networks from 29 different studies and show that some motifs are consistently over-represented worldwide, suggesting that the building blocks of plant-pollinator networks are not random and are associated to ... Second, we show that the position of pollinator guilds and plant reproductive strategies is not random with respect to the positions occupied within each motif. ... Hence, we show that species ecology is shaping the building blocks that conform the web of life.

INTRODUCTION

The interaction between plants and pollinators can be studied at different scales, from species level interactions (micro-scale) to the full network structure (macro-scale). However, condensing to species level and the holistic view of the full network involves missing relevant information for the understanding of ecological processes. Intermediate levels to study plant-pollinator networks (meso-scale) have been little explored but can shed light of ecological processes not captured by the traditional approaches used in plant-pollinator network studies. [Explain what macro-descriptors have found i.e. Bascompte, and what pairwise interactions descriptors have accomplished i.e. trait-matching. End explaining Benno Simmons work]

Exploring the different and most frequent motifs could help our understanding of species interactions due to motifs consider direct and indirect interactions not accounted in single species metrics. [Explain this: <https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/1365-2435.13736> and say we don't know how motif distribution really looks like in the real world. Which is the expected pattern?]

However, motifs are abstract representations often decoupled from species ecology. [explain here the importance of relating reproductive strategies and pollinators behaviours to their positions within a motif. This is interesting because bridges pairwise interactions to the emergent network via motifs]

Here, we used ... blah... and find plant functional groups by using a comprehensive dataset with several reproductive traits and pollinator functional groups as the main differentiated guilds in life form and behaviour. Then, we explored the main motifs of these networks and we found [...]

METHODS

We have compiled 60 plant-pollinator networks from 29 different studies [...]

Plants were grouped into reproductive strategies [justify and explain the process]

Pollinators were grouped into guilds, because [justify and explain, it's important to convince the reader that Order level for pollinators and FG for plants makes sense].

We follow Simmons et al. to decompose networks on their constituent motifs. We compare observed motif frequencies to a null model, which constrains the number of plants, pollinators and the network connectance. For each network we compared the observed motif frequency with the expected frequencies and To summarize general patterns across networks we used a GLMM per motif with the percentile per network, ...

Next, we calculated which functional groups were over or under-represented in different motif positions by comparing with Again, we tested ...

[potential additions: Finally, for each motif, we recover the number of times a given combination of functional groups emerges ... Finally, we draw a network collapsing nodes per FG...]

RESULTS

0) Description of guilds and FG.

1) over and under represented motifs

```
library(lme4)

## Loading required package: Matrix
library(nlme)

##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##      lmList

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.3.3      v purrr   0.3.4
## v tibble  3.0.4      v dplyr   1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::collapse() masks nlme::collapse()
## x tidyr::expand()   masks Matrix::expand()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x tidyr::pack()     masks Matrix::pack()
## x tidyr::unpack()   masks Matrix::unpack()

source("../Scripts/Scripts_Alfonso/add_study_id.R")
d <- read_csv("../Data/Csv/Motifs_frequencies_and_null_models/Motifs_frequency_percentile.csv")

##
## -- Column specification -----
## cols(
##   Network_id = col_character(),
##   motif = col_double(),
##   nodes = col_double(),
##   observed_freq_sizeclass = col_double(),
##   expected_freq_sizeclass = col_double(),
##   percentil_sizeclass = col_double()
```

```
## )
```

```
d <- add_study_id(d)
str(d)
```

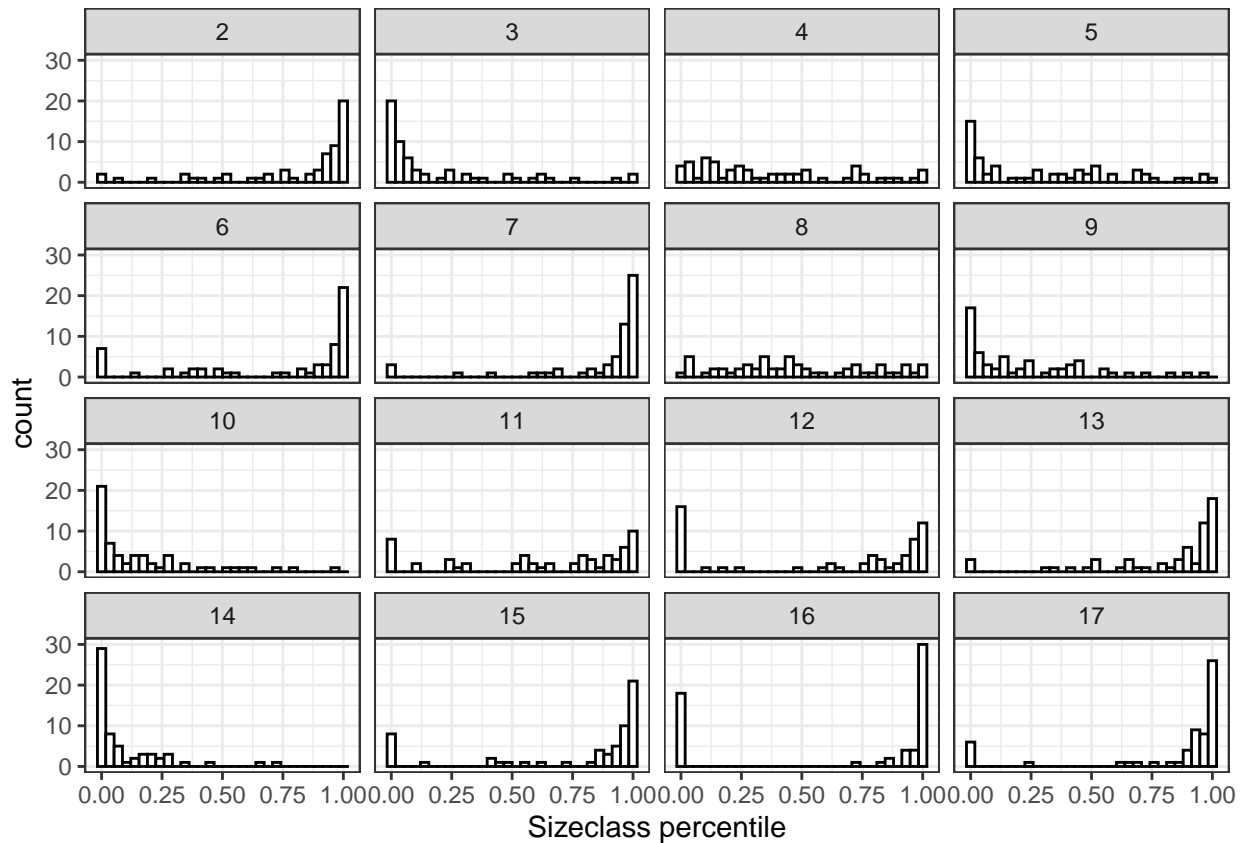
```
## tibble [1,020 x 7] (S3: tbl_df/tbl/data.frame)
##  $ study_id      : chr [1:1020] "bartomeus_2008" "bartomeus_2008" "bartomeus_2008" "bartomeus_2008" ...
##  $ Network_id    : chr [1:1020] "1_1_bartomeus_spain_2008_batca_1_2" "1_1_bartomeus_spain_2008_batca_1_2" ...
##  $ motif          : num [1:1020] 1 2 3 4 5 6 7 8 9 10 ...
##  $ nodes          : num [1:1020] 2 3 3 4 4 4 4 5 5 5 ...
##  $ observed_freq_sizeclass: num [1:1020] 1 0.142 0.858 0.58 0.365 ...
##  $ expected_freq_sizeclass: num [1:1020] 1 0.114 0.886 0.62 0.35 ...
##  $ percentil_sizeclass   : num [1:1020] 0 0.991 0.009 0.114 0.691 1 0.991 0.248 0.222 0.228 ...
```

```
head(d)
```

```
## # A tibble: 6 x 7
##   study_id Network_id motif nodes observed_freq_s~ expected_freq_s~
##   <chr>      <chr>      <dbl> <dbl>          <dbl>          <dbl>
## 1 bartome~ 1_1_barto~     1     2            1            1
## 2 bartome~ 1_1_barto~     2     3          0.142          0.114
## 3 bartome~ 1_1_barto~     3     3          0.858          0.886
## 4 bartome~ 1_1_barto~     4     4          0.580          0.620
## 5 bartome~ 1_1_barto~     5     4          0.365          0.350
## 6 bartome~ 1_1_barto~     6     4          0.0334         0.0204
## # ... with 1 more variable: percentil_sizeclass <dbl>
```

```
ggplot(d %>% filter(motif != 1), aes(percentil_sizeclass))+
  geom_histogram(color="black", fill="white")+
  facet_wrap(~motif)+
  theme_bw()+
  labs(x="Sizeclass percentile")
```

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



Motifs tend to be over- and under-represented in real networks

Mean percentile and SE taking into account the study system

```
motif_codes <- d %>% filter(motif != 1) %>% # We remove links (motif code: 1)
```

```
  select(motif) %>% unique() %>% pull()
```

```
motif_means <- tibble(motif = motif_codes)
```

```
motif_means$mean <- NA
```

```
motif_means$SE <- NA
```

```
for(i.motif in 1:length(motif_codes)){
```

```
  m <- lmer(percentil_sizeclass ~ 1+(1|study_id), #HERE WE NEED STUDY SYSTEM ONLY!!
    data = subset(d, motif == motif_codes[i.motif]))
```

```
  motif_means$mean[i.motif] <- fixed.effects(m) %>% unname()
```

```
  motif_means$SE[i.motif] <- sqrt(diag(vcov(m)))
```

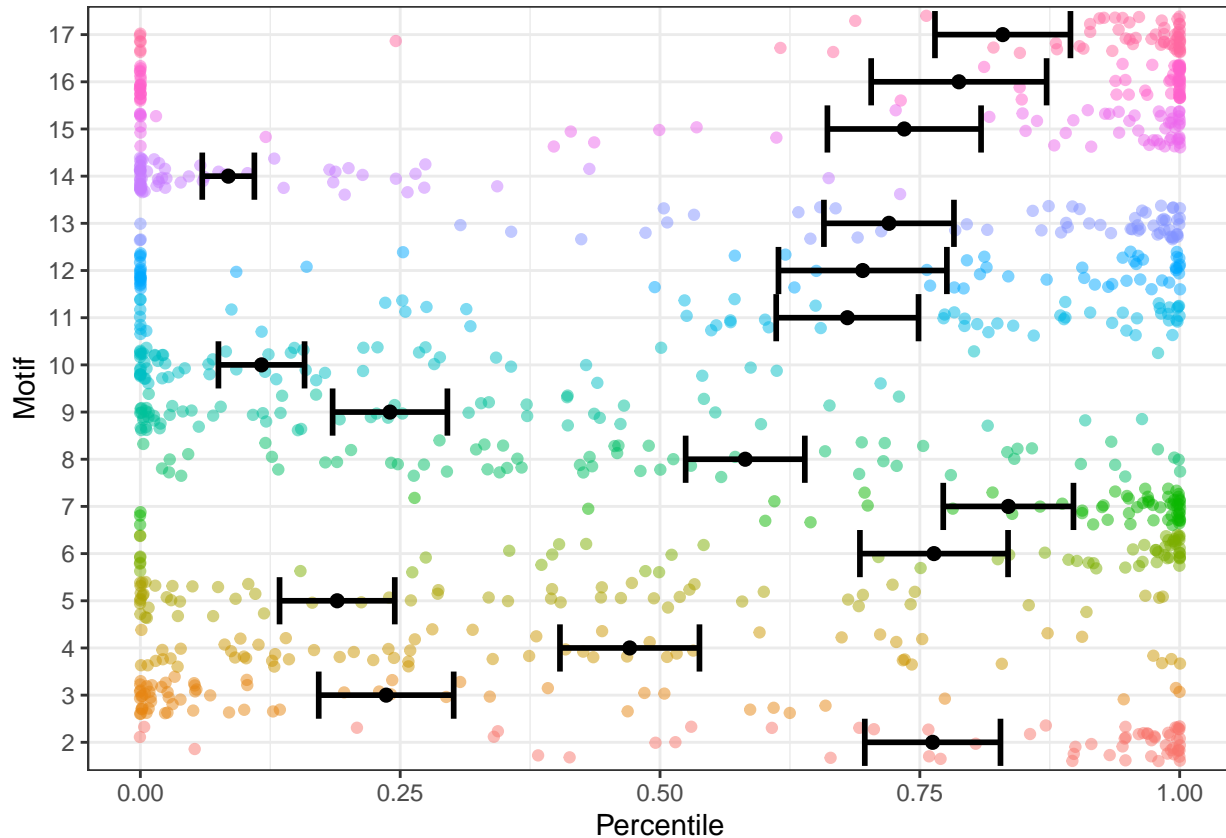
```
}
```

```
ggplot(NULL) +
```

```
  geom_point(data = d %>% filter(motif != 1),
    aes(y=as.factor(motif), x=percentil_sizeclass,
      color = as.factor(motif)),
    position = "jitter",alpha=0.5)+
```

```
  geom_errorbar(data = motif_means,aes(y = as.factor(motif), xmin=mean-SE, xmax=mean+SE),
    width=1.0,size=1)+
```

```
geom_point(data = motif_means, aes(y = as.factor(motif), x = mean),
           size = 2) +
labs(y = "Motif", x = "Percentile") +
theme_bw() +
theme(legend.position = "none")
```



*#Fig 1 Should be a dot plot (or forest plot) of the mean \pm SE of the 15 motifs.
#NICE! Can we put motifs in Y axes, and % in X?*

2) FG-position associations.

```
library(lme4)
library(nlme)
library(tidyverse)
library(stringi)
library(ggpubr)
```

```
## Registered S3 methods overwritten by 'car':
##   method                      from
##   influence.merMod             lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod      lme4
##   dfbetas.influence.merMod     lme4
```

```
source("../Scripts/Scripts_Alfonso/add_study_id.R")
```

```
all_position_percentiles <- read_csv("../Data/Csv/Motifs_positions_and_null_models/GF_positions_frequen
```

```
##
```

```

## -- Column specification -----
## cols(
##   Network_id = col_character(),
##   Node_FG = col_character(),
##   position = col_character(),
##   observed_freq = col_double(),
##   expected_freq_its_GF = col_double(),
##   expected_freq_all_GF = col_double(),
##   percentil_its_GF = col_double(),
##   percentil_all_GF = col_double()
## )

all_position_percentiles$position <- stri_extract_first_regex(all_position_percentiles$position,
  "[0-9]+") %>% as.numeric()

all_position_percentiles <- add_study_id(all_position_percentiles)
str(all_position_percentiles)

## tibble [26,082 x 9] (S3: tbl_df/tbl/data.frame)
## $ study_id      : chr [1:26082] "bartomeus_2008" "bartomeus_2008" "bartomeus_2008" "bartomeus_2008" ...
## $ Network_id    : chr [1:26082] "1_1_bartomeus_spain_2008_batca_1_2" "1_1_bartomeus_spain_2008_batca_1_2" ...
## $ Node_FG       : chr [1:26082] "1" "2" "3" "5" ...
## $ position      : num [1:26082] 1 1 1 1 1 1 1 1 1 2 ...
## $ observed_freq  : num [1:26082] 1 1 1 1 0 0 0 0 0 0 ...
## $ expected_freq_its_GF: num [1:26082] 1 1 1 1 NA NA NA NA NA NA ...
## $ expected_freq_all_GF: num [1:26082] 1 1 1 1 NA NA NA NA NA NA ...
## $ percentil_its_GF  : num [1:26082] 0 0 0 0 NA NA NA NA NA NA ...
## $ percentil_all_GF  : num [1:26082] 0 0 0 0 NA NA NA NA NA NA ...

head(all_position_percentiles)

## # A tibble: 6 x 9
##   study_id Network_id Node_FG position observed_freq expected_freq_i-
##   <chr>    <chr>    <chr>    <dbl>         <dbl>         <dbl>
## 1 bartome~ 1_1_barto~ 1          1             1             1
## 2 bartome~ 1_1_barto~ 2          1             1             1
## 3 bartome~ 1_1_barto~ 3          1             1             1
## 4 bartome~ 1_1_barto~ 5          1             1             1
## 5 bartome~ 1_1_barto~ Bee          1             0             NA
## 6 bartome~ 1_1_barto~ Coleop~    1             0             NA
## # ... with 3 more variables: expected_freq_all_GF <dbl>,
## #   percentil_its_GF <dbl>, percentil_all_GF <dbl>

list_plants_FG <- as.character(1:10)

plant_position_percentiles_filtered <- all_position_percentiles %>%
  filter(Node_FG %in% list_plants_FG, !position %in% c(1,2),
    !is.na(expected_freq_its_GF)) #ME DA ERROR
pollinator_position_percentiles_filtered <- all_position_percentiles %>%
  filter(!Node_FG %in% list_plants_FG, !position %in% c(1,2),
    !is.na(expected_freq_its_GF))

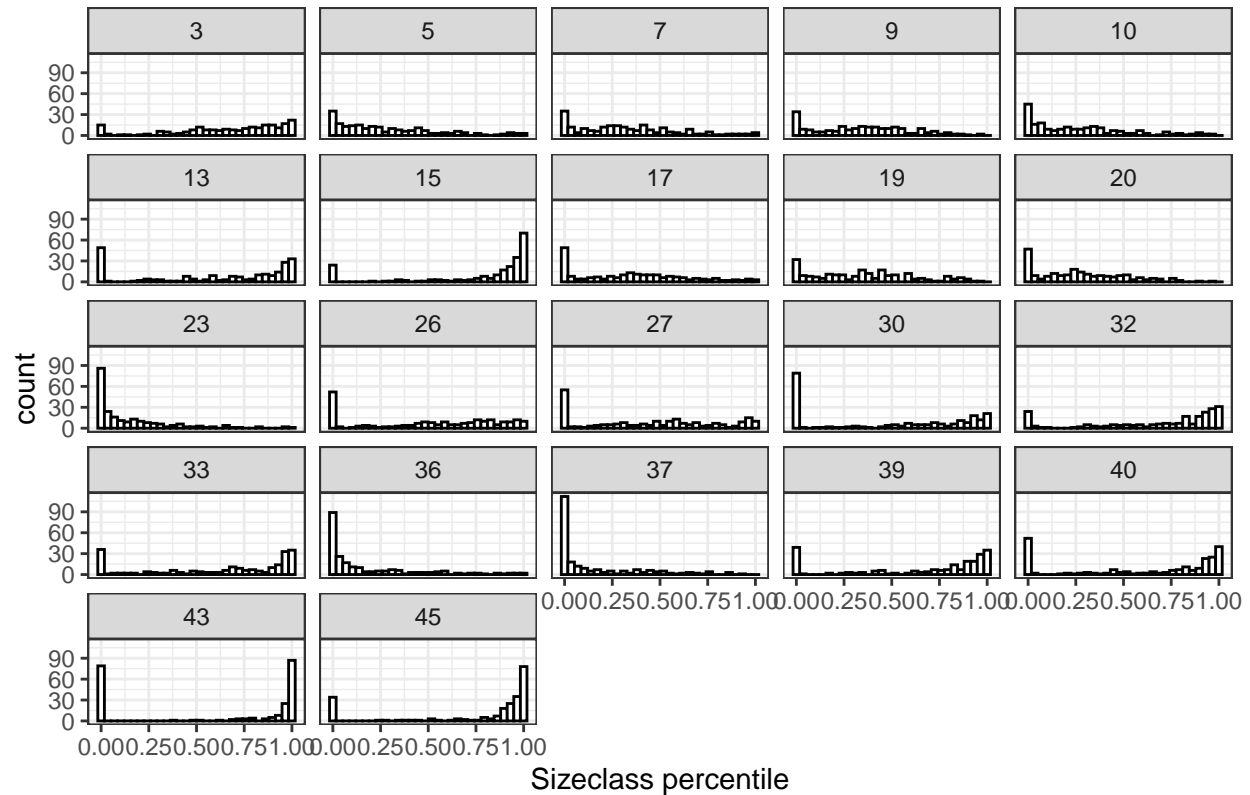
ggplot(plant_position_percentiles_filtered, aes(percentil_its_GF))+
  geom_histogram(color="black", fill="white")+
  facet_wrap(~position)+
  theme_bw()+

```

```
labs(x="Sizeclass percentile", title = "Histograms: Percentiles for plant positions")
```

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

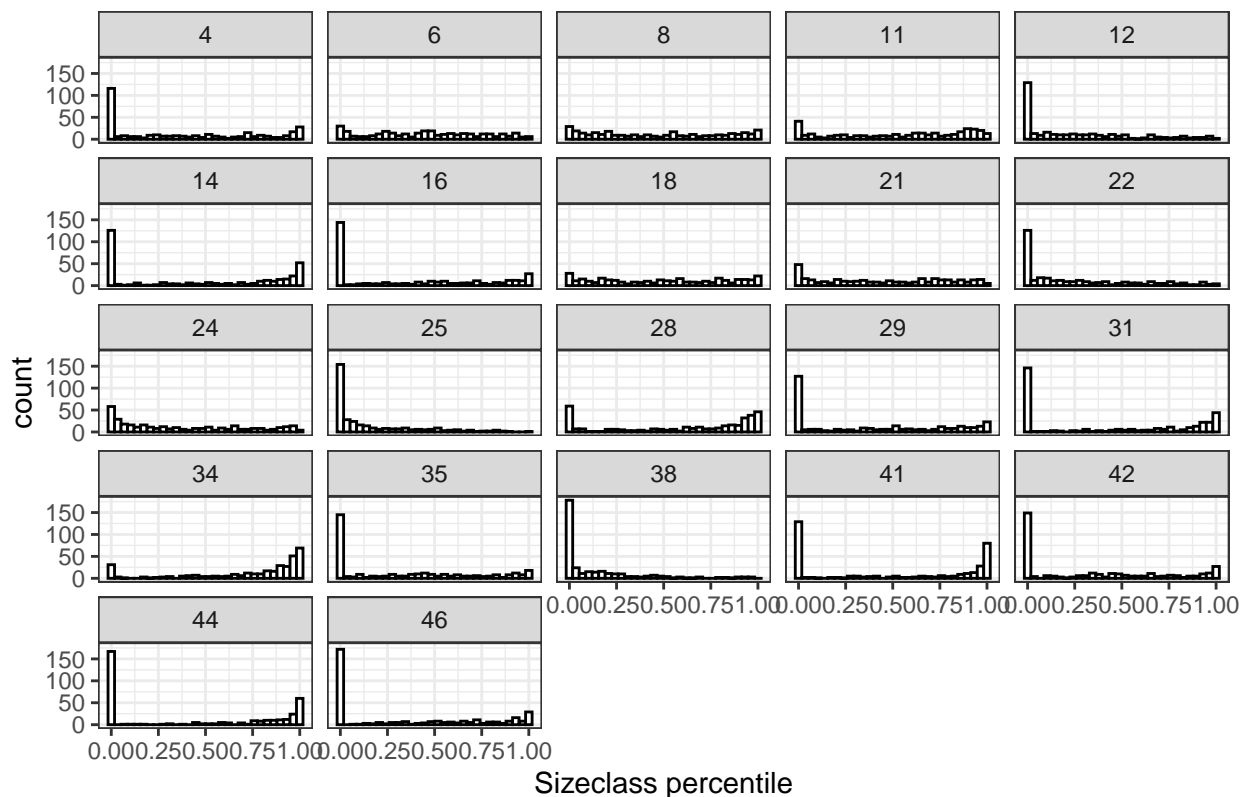
Histograms: Percentiles for plant positions



```
ggplot(pollinator_position_percentiles_filtered, aes(percentil_its_GF))+
  geom_histogram(color="black", fill="white")+
  facet_wrap(~position)+
  theme_bw()+
  labs(x="Sizeclass percentile",title = "Histograms: Percentiles for pollinator positions")
```

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

Histograms: Percentiles for pollinator positions



```
# estimated mean taking into account the random str is 0.XX +- 0.XX (look at the (Intercept) estimate)

# Plants-----

# Mean percentile and SE taking into account the study system
plant_codes <- plant_position_percentiles_filtered$position %>% unique()
plant_means <- plant_position_percentiles_filtered %>% select(position, Node_FG) %>% unique()
plant_means$mean <- NA
plant_means$SE <- NA

for(i.pos in 1:length(plant_codes)){

  m <- lmer(percentil_its_GF ~ Node_FG + (1|study_id),
            data = plant_position_percentiles_filtered %>%
              filter(position == plant_codes[i.pos]))

  temp <- fixed.effects(m) %>% unname()
  plant_means$mean[plant_means$position == plant_codes[i.pos]] <- c(temp[1],
                                                                    temp[1]+temp[2],
                                                                    temp[1]+temp[3],
                                                                    temp[1]+temp[4],
                                                                    temp[1]+temp[5]) ## to get real mean

  temp <- sqrt(diag(vcov(m)))
  plant_means$SE[plant_means$position == plant_codes[i.pos]] <- c(temp[1],
                                                                    temp[1]+temp[2],
```



```

temp[1]+temp[3],
temp[1]+temp[4],
temp[1]+temp[5])

}

GF_plant_pos <- ggplot(NULL) +
  geom_point(data = plant_position_percentiles_filtered,
    aes(x=as.factor(position), y=percentil_its_GF,
      color = as.factor(position)),
    position = "jitter",alpha=0.5)+
  geom_errorbar(data = plant_means,aes(x = as.factor(position), ymin=mean-SE, ymax=mean+SE),
    width=1.0,size=1)+
  facet_wrap(~Node_FG,ncol = 5)+
  geom_point(data = plant_means,aes(x = as.factor(position), y=mean),
    size=2)+
  labs(x="Position", y = "Percentile",title = "Plants")+
  theme_bw()+
  theme(legend.position = "none")+#,axis.text.x=element_text(angle=90,vjust=0.5, hjust=1))+
  coord_flip()

# Pollinators-----

# Mean percentile and SE taking into account the study system
pollinator_codes <- pollinator_position_percentiles_filtered$position %>% unique()
pollinator_means <- pollinator_position_percentiles_filtered %>% select(position,Node_FG) %>% unique()
pollinator_means$mean <- NA
pollinator_means$SE <- NA
pollinator_means_reordered <- NULL

for(i.pos in 1:length(pollinator_codes)){

  m <- lmer(percentil_its_GF ~ Node_FG + (1|study_id), #HERE WE NEED STUDY SYSTEM ONLY!!
    data = pollinator_position_percentiles_filtered %>%
      filter(position == pollinator_codes[i.pos]),
    control = lmerControl(optimizer = "Nelder-Mead")) # I changed the optimizer because motif 8

  temp <- fixed.effects(m) %>% unname()
  pollinator_means_aux <- pollinator_means[pollinator_means$position == pollinator_codes[i.pos],] %>% as.data.frame()

  pollinator_means_aux$mean <- c(temp[1],
    temp[1]+temp[2],
    temp[1]+temp[3],
    temp[1]+temp[4],
    temp[1]+temp[5],
    temp[1]+temp[6],
    temp[1]+temp[7],
    temp[1]+temp[8],
    temp[1]+temp[9]) ## to get real means per FG

  temp <- sqrt(diag(vcov(m)))

```

```

pollinator_means_aux$SE <- c(temp[1],
                             temp[1]+temp[2],
                             temp[1]+temp[3],
                             temp[1]+temp[4],
                             temp[1]+temp[5],
                             temp[1]+temp[6],
                             temp[1]+temp[7],
                             temp[1]+temp[8],
                             temp[1]+temp[9])

pollinator_means_reordered <- bind_rows(pollinator_means_reordered,pollinator_means_aux)
}

# There are lower limits (mean - SE) smaller than 0 (see Other insects)
# We set the lower limit of those error bars to zero

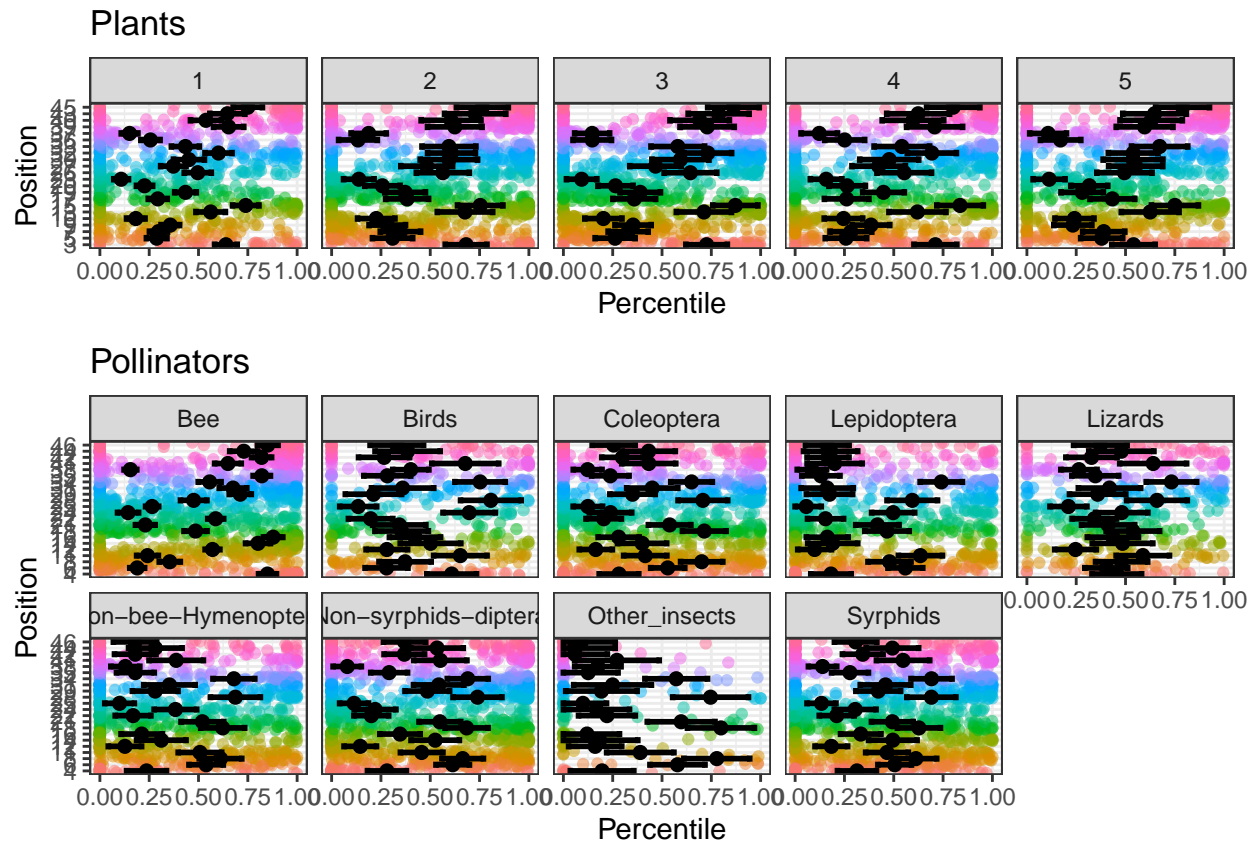
pollinator_means_reordered <- pollinator_means_reordered %>% mutate(lower = mean-SE)
pollinator_means_reordered$lower[pollinator_means_reordered$lower < 0] <- 0

GF_poll_pos <- ggplot(NULL) +
  geom_point(data = pollinator_position_percentiles_filtered,
            aes(x=as.factor(position), y=percentil_its_GF,
                color = as.factor(position)),
            position = "jitter",alpha=0.5)+
  geom_errorbar(data = pollinator_means_reordered, aes(x = as.factor(position), ymin=lower,
                                                       ymax=mean+SE),
               width=1.0,size=1)+
  facet_wrap(~Node_FG,ncol = 5)+
  geom_point(data = pollinator_means_reordered,aes(x = as.factor(position), y=mean),
            size=2)+
  labs(x="Position", y = "Percentile", title = "Pollinators")+
  theme_bw()+
  theme(legend.position = "none")+#,axis.text.x=element_text(angle=90,vjust=0.5, hjust=1))+
  coord_flip()

# Plant and Pollinators-----

ggarrange(GF_plant_pos,GF_poll_pos,
          ncol = 1, nrow = 2,heights = c(.90, 1.4))

```



#Fig 2 may be the same, but for the each positions-FG combination? Maybe heatmap (but we lose the SE's)