Predicting Plant-Pollinator Interactions in Montane Meadows Using a Multinomial Model

Ivan Pyzow
University of Chicago

Kaitlin Horan
Middlebury College

Abstract

We seek to understand the relationship between flower frequency and observed plant-pollinator interactions in the Western Cascade Range. We create our own probabilistic model to represent the process of flower selection by pollinators using a multinomial distribution. Using Pearson's chi-squared test for goodness-of-fit, we demonstrate that the pollinators in the study do not exhibit flower selection strategies that are based solely on relative flower abundance (i.e. strictly frequency-dependent flower selection strategies). We generate our own optimized flower preference list for each pollinator, and show that though they are still unreliable in predicting pollinator interactions in most cases, they have more predictive strength than uniform flower preference list that represents a strictly frequency-dependent flower selection strategy.

Definitions

plot: a 3m by 3m section of a meadow marked for data collection. Each meadow is represented by ten plots.
meadow: a particular type of ecosystem whose vegetation consists mostly of grass and wildflowers. Different meadows from the data set vary in size and specific sorts of terrain (soil moisture, soil type, grassy or rocky etc.).
(meadow) complex: a set of several meadows grouped together by location. Five different complexes appear in our data.
watch: a subset of the data, in which each meadow in each complex is surveyed. Between five and seven watches are performed incrementally throughout a growing season for each year.
meadow-watch-year: a collection of all the data from one meadow, on one watch, in one year
watch-year: a collection of all the data from one watch in one year
flower selection strategy: the exhibited behavior of a pollinator with regard to flower interactions
strictly frequency-dependent: based only on the relative abundances of flowers, used to describe flower selection strategies
pollinator's preference (for a flower): a value from 0 to 1 that quantifies the level of interactions with a flower, independent of the flower's relative abundance. In other words, we use preference to refer to all factors that influence flower selection besides frequency-dependence, which might be the nectar resource of the flower or an innate bias by the pollinator.
pollinator's preference list: a list of a pollinator's preferences for different flowers, scaled so that the preference list sums to 1. The list is comprised of all potential flowers for a single pollinator.
flower's desirability (to a pollinator): refers to the same value as a pollinator's preference, with the semantic difference being that pollinators have preferences, while flowers have desirability.

Introduction

The study of pollination networks has become increasingly important in response to growing evidence of pollinator declines. Understanding these networks is critical to fostering management of pollinators and the ecosystem services they provide to agriculture. Some evidence indicates that pollinators have distinct preferences for particular plant species; for example, native bees in Yolo

County, California, prefer native plants to exotic plants (Morandin & Kremen 2013). It has also been suggested that pollinator preferences influence the evolution of floral traits (Gong & Huang 2011). Furthermore, understanding pollinator networks is important in predicting systems' responses to climate change. It is suggested that such mutualistic systems might be especially damaged by "phenological desynchronization" (Benadi, Hovestadt, Poethke & Blüthgen 2014). In this case, understanding pollinator preferences could once again be important in attempting to manage such environmental transformations.

Floral rewards (pollen and/or nectar) are the driving force behind pollinator visitation of flowers. It seems logical that pollinators might select flowers with the greatest rewards, or those with floral traits indicative of greater rewards (Sutherland & Vickery 1993). Pollinator preferences have also been identified previously with respect to radish populations. Radishes with white flowers were shown to be less desirable by several major pollinator taxa (Lee & Snow 1998). This, as well as first-hand experiences observing meadow interactions, contribute to the hypothesis that pollinators in montane meadows also express preferences for specific flowers.

Our study addresses the following two questions:
1) To what extent do the pollinators exhibit flower selection strategies that are strictly frequency-dependent?
2) For each pollinator, how accurately does our optimized preference list predict the observed interactions?

We anticipate that pollinators have flower selection strategies that are not strictly frequency-dependent and predict that our general preference list for each pollinator will accurately predict observed interactions.

To answer Question 1), we apply Pearson's chi-squared test for goodness-of-fit between the observed interactions and the interactions predicted by a strictly frequency-dependent flower selection strategy. To answer Question 2), we first generate a flower preference list for each pollinator using our multinomial likelihood model that that simulates pollinator interactions in the meadows. We then use this preference list to predict interactions, and then compare them to observed interactions by similarly applying Pearson's chi-squared test for goodness-of-fit.

Study Site

The data used in this paper were recorded in montane meadows in the western Cascades Range in Oregon. Data were collected between the years 2011 and 2013, during which time five meadow complexes were surveyed. Though we use data from all five meadow complexes, not every complex was surveyed each year. In 2011 and 2012, five complexes were surveyed: Carpenter, Frissell, Cone, Bunchgrass and Lookout. In 2013, complexes surveyed included Carpenter, Frissell and Lookout.

Ten 3m by 3m plots were constructed for each meadow and systematically placed to best represent the entire meadow. Data were collected in points referred to as watches, which occurred incrementally during the growing season. There were seven watches recorded in 2011, five in 2012, and seven in 2013.

The data contain 63,668 interactions between 96 plant species and 436 pollinator species. Flower abundance data includes flower counts of 130 different species of plants.

Omitted data include entries that failed to identify either a pollinator or plant species. There were also several concerns with data that appeared to have column entries swapped; these were also excluded from our analysis.

Pollinators documented include bumblebees, honeybees, solitary bees, butterflies, moths, hummingbirds, bee flies, beetles, wasps, syrphids, hornets, etc. The ten most commonly recorded insects throughout all three years of data are presented in the table below. The insect with an asterisk are those considered in this analysis.

Figure 1: Number of sightings (i.e. number of times an insect is recorded in the data set, not total number of interactions. This avoids the problem of weighting numbers of social bees, or insects which often visit multiple flowers in a vicinity at the same time, over those that do not) of the ten most frequently observed pollinator species from 2011 to 2011 in montane meadows in the western Cascades. Asterisks indicate insects which are closely considered in the following analysis.

Pollinator	Description	Number of Sightings (Note this is different from the total number of interactions as the same pollinator might have had multiple interactions)
Apis mellifera*	Social bee	4101
Bombus mixtus*	Social bee	2835
Epicauta puncticollis	Beetle	1534
Bombus bifarius	Social bee	945
Muscoid genus 3	Fly	731
Bombylius major*	Beefly	677
Eristalis hirtus*	Syrphid	572
Coccinella septempunctata	Ladybird	459
Bombus vosnesenskii	Social bee	370
Chrysotoxum fasciatum	Syrphid	358

Methods

Field Methods
As noted above, meadows were visited between five and seven times for each year of data collection, between the months of May and August. One plot watch was performed for each plot in each meadow per visit.

A plot watch begins with a flower survey, during which an observer identified all flowers in a given plot and estimated their abundance (unknown flowers were sampled and brought back for identification). The observer also recorded information about the weather, temperature and time of day. This was followed by a sampling period of 15 minutes. During these 15 minutes, the observer recorded each pollinator-plant interaction he or she observed. An interaction was considered any contact between the reproductive parts of the flower and a given pollinator. Relatively common species of insects such as *Apis mellifera*, *Bombus mixtus*, *Bombus bifarius* and *Bombus vosnesenskii* were identified by the observer in the field. Other pollinators were captured with a net and identified at a later date by Andy Moldenke (Faculty Botany and Plant Pathology, Oregon State University).

While recording flower visitation is a valuable tool in attempting to identify relationships between plants and pollinators, we note that this method might not be entirely representative of the true process of pollination. It is possible that some "pollinators" are merely visiting flowers, but not pollinating them (King, C , Ballantyne, GA & Willmer, PG 2013). The limitations of the observer also prevent any data collected from being a totally accurate depiction of a plot. Thus, though we work with an imperfect data set, we hope to glean some insight into the true dynamic network of plant-pollinator interactions.

Statistical Methods

In order to model the behavior of a pollinator in the meadow, we created a multinomial function, L, that gives the likelihood of a particular combination of flower visitations by a given pollinator.

Equation 1: Likelihood function.

$$L(d_1, d_2, ..., d_n) = \prod_{j=1}^{m} c_j \left(\prod_{i=1}^{n} \lambda_{ij}^{b_{ij}} \right)$$

where
m = number of meadow-watch-years
n = number of flower species
a_{ij} = abundance of flower i on meadow-watch-year j
b_{ij} = number of interactions between pollinator p and flower i on meadow-watch-year j
d_i = desirability of flower i for pollinator p

$$\lambda_{ij} = \frac{d_i a_{ij}}{\sum_{k=1}^{n} d_k a_{kj}}$$

$$c_j = \frac{\left(\sum_{h=1}^{n} b_{hj} \right)!}{\prod_{h=1}^{n} (b_{hj}!)}$$

This function *L* takes three parameters: 1) the pollinator's preference list for flower species 2) the abundance of the flower species, and 3) the number of interactions by the pollinator on each flower species. When we maximize L, however, we will fix the second two parameters according to our observed data for interactions and flower abundances. This allows us to solve for the preference list (the first parameter).

The model is based on the assumption that each observed interaction between a pollinator and a flower is an independent event with a probability that is directly proportional to 1) the relative abundance of the flower and 2) the pollinator's preference for the flower (i.e., the flower's desirability value). This probability is expressed by the expression λ_{ij}. For example, in a meadow with equal abundances of *Gilia capitata* and *Eriophylum lanatum*, if we observed twice as many interactions on *Gilia* as on *Eriophylum* for the pollinator *Apis mellifera*, we would assign *Gilia* a desirability value of .67 and *Eriophylum* a desirability value of .33 (for the pollinator *Apis mellifera*). (The problem becomes more complex below, when we consider 138 flowers instead of two.)

In reality, it is possible that each interaction is not an independent event. For instance, we know that some species of pollinators tend to visit only one species in a day according to their selected "search image", which would imply that each interaction is not independent. However, in a large sample of insects of the same species, we could imagine that that aggregation of all the interactions dictated by individual search images would produce data that resembles data generated by interactions with independent probabilities.

The coefficient *Cj* in *L* is the multinomial coefficient that takes into consideration the fact that we do not consider the order of interactions for a given meadow-watch-year.

When generating the likelihood for a given pollinator's preference list using our function *L*, we consider every meadow-watch-year from the three years of data for which the pollinator had more than five sightings, and weight each remaining meadow-watch-year equally.

For our analysis, we selected four of the most common pollinators (belonging to different genera): *Apis mellifera*, *Bombus mixtus*, *Bombylius major*, and *Eristalis hirtus*. Different genera were selected because of the different social habits attributed to the separate genera. *Apis mellifera* is a social bee that lives in a colony. It has been asserted that *Apis mellifera* learns to associate flower traits with a nectar reward in order to increase foraging productivity (Honeybee Genome Sequencing Consortium 2006). The genus *Bombus*, similarly, is distinguished for the social behavior of its species (Dukas & Real 1991). Although *Bombylius* (bee flies) are not social, studies have suggested that bee flies exhibit "short-term learning paths"; once they have visited a certain flower, they are likely to visit another flower of the same species (Boesi, Polidori & Andrietti 2008). Finally, the syrphid *Eristalis hirtus*, like *Bombylius major*, is a solitary insect.

Predicting Interactions

We predict interactions for each watch-year by maximizing *L* after inputting the given preferences list and the flower abundances.
Next, we generate our own preference lists for each pollinator based on the data. In order to calculate the most likely set of preferences for a given pollinator, we fix the flower abundance data

and interaction data, and maximize the likelihood function L for the preference list. We achieved this by minimizing the negative log likelihood function (Figure 20) using the R function "optim."

The Five Tests

We ran the following five tests to determine the predictive strength of the two preference lists in consideration: the uniform preference list and the optimized preference list. For the first four tests, we used Pearson's chi-squared test for goodness-of-fit to compare the expected (predicted) interactions to the observed interactions. For the fifth test, we used a likelihood ratio test to compare the respective likelihoods of the two preference lists directly.

Test 1: Pearson's chi-squared test for goodness-of-fit for *each* watch-year for each insect, comparing interactions predicted by a uniform preference list (null model) to observed interactions.

Test 2: Pearson's chi-squared test for goodness-of-fit for *all* watch-years for each insect, comparing interactions predicted by a uniform preference list (null model) to observed interactions.

Test 3: Pearson's chi-squared test for goodness-of-fit for *each* watch-year for each insect, comparing interactions predicted by our optimized preference list (null model) to observed interactions.

Test 4: Pearson's chi-squared test for goodness-of-fit for *all* watch-years for each insect, comparing interactions predicted by our optimized preference list (null model) to observed interactions.

Test 5: The likelihood ratio test to determine, for each pollinator, whether a strictly frequency-dependent flower selection strategy (from Question 1) is more accurate than a preference-dependent strategy (from Question 2) in describing the interaction behavior exhibited by the pollinators in this study. Our null hypothesis is that of Question 1), and the maximum likelihood for our alternative hypothesis is the result of the likelihood function using the preferences generated in Question 2).

Results

Regarding Question 1), we observe that under the null hypothesis that each pollinator exhibits a strictly frequency-dependent flower selection strategy, Pearson's chi-squared test for goodness-of-fit returns extremely low p-values for each watch-year.

Regarding Question 2), under the null hypothesis that each pollinator exhibits a flower selection strategy that is predicted by our generalized preference list for that pollinator, Pearson's chi-squared test for goodness-of-fit returns extremely low p-values for most watch-years for *Apis mellifera*, *Bombylius major*, and *Bombus mixtus*. For *Eristalis hirtus*, the test returned p-values above 0.05 for 6 of the 10 watch-years that we examined. We ran the test for the goodness-of-fit for all ten watch-years together due to the promising nature of the p-values of the tests on individual watch-years for *Eristalis hirtus*. However, this aggregate test for *Eristalis hirtus* returned an extremely low p-value.

Comparing both hypotheses using the likelihood ratio test for each pollinator, we see that we obtain low p-values.

Presented below in figure 2 we visualize the flower preference list generated by our model for *Apis mellifera*. The model suggests that *Agoseris aurantiaca* is much more desirable than any other flower.
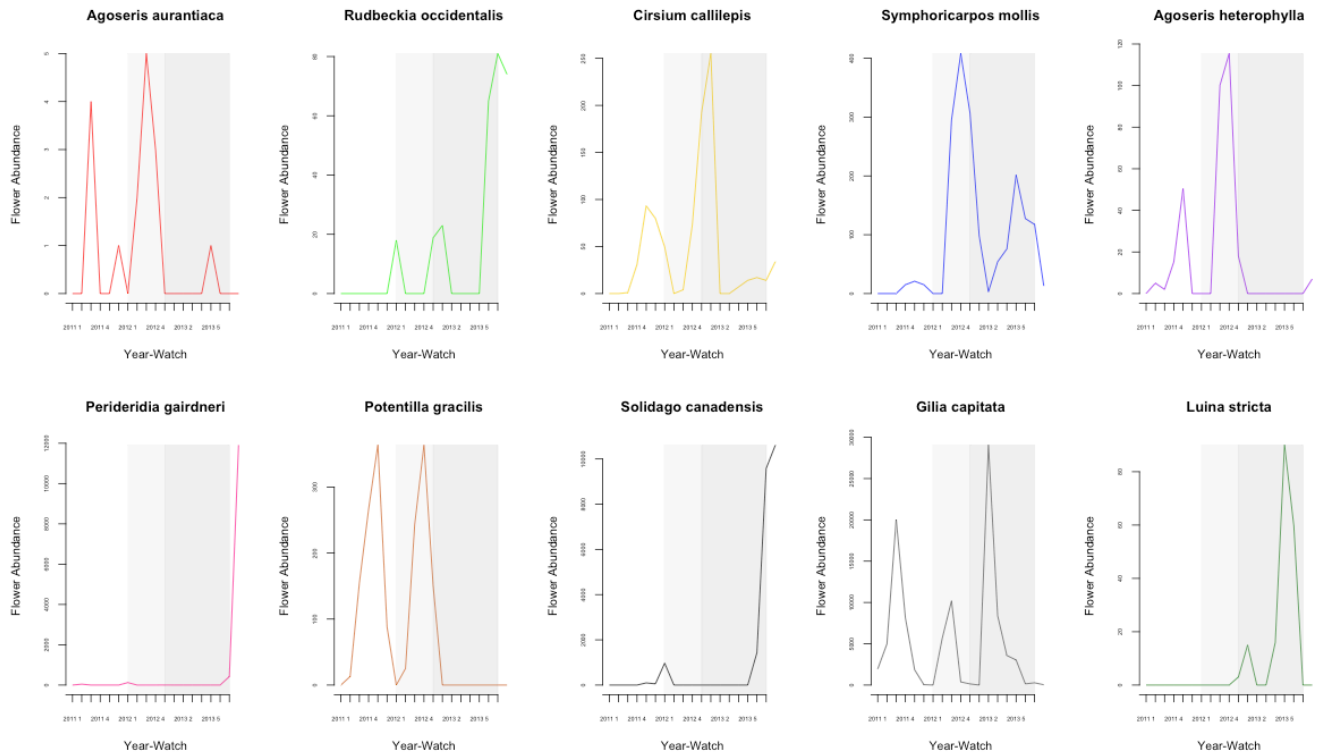
Figure 2: Flower preference list graph for Apis mellifera. Flowers presented are the top ten most preferable, as outputted from our maximization of the likelihood function.
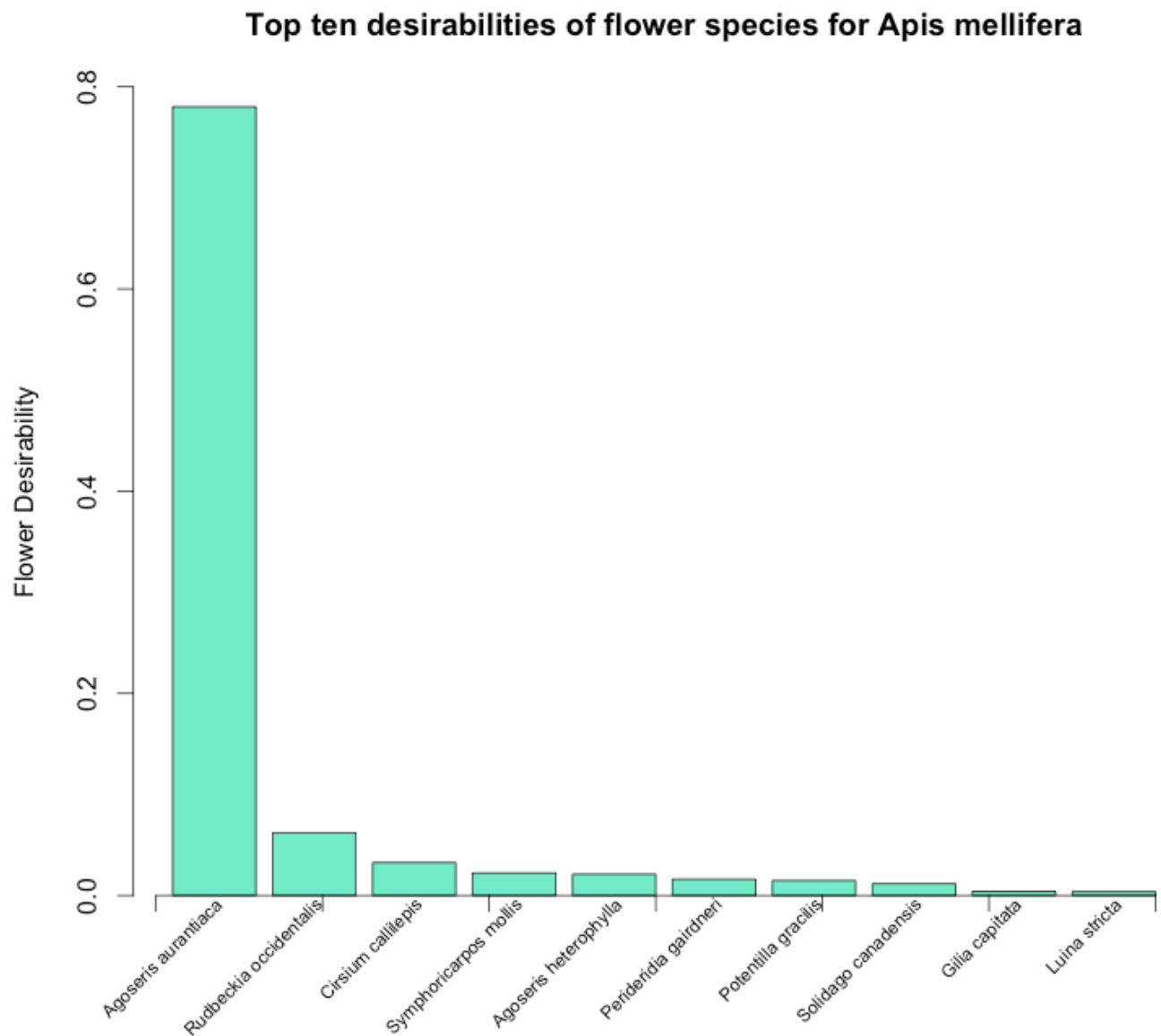
The flower abundance graph, figure 3, for *Apis mellifera* shows huge variability in flower populations from year to year. *Perideridia gairdneri* and *Luina stricta* were never observed in significant numbers until 2013. *Rudbeckia occidentalis* and *Solidago candensis* demonstrate a similar, though less extreme, trend.

Figure 3: We present the abundances of the top ten most preferable flowers for Apis mellifera. Shading denotes years 2011, 2012 and 2013 respectively (as shading goes from lightest to darkest). Flowers are listed left to right, and top to bottom from most desirable to less desirable. Here we would expect to see three peaks in the graph, representing the height of the season in 2011, 2012 and 2013. Graphs which do not depict this trend suggest that the flower was not in bloom some years, or that its abundance was significantly lower for some reason.

Flower Abundances over Time for Most Preferred Flowers of *Apis Mellifera*

In figures 4 and 5, we show the the flower preference list generated by our model for *Eristalis hirtus*. The desirabilites here are more evenly distributed than the desirabilites seen for *Apis mellifera*. Furthermore, the abundances presented show more consistent patterns from year to year.

Figure 4: Flower preference list graph for Eristalis hirtus. Flowers presented are the top ten most preferable, as outputted from our maximization of the likelihood function.

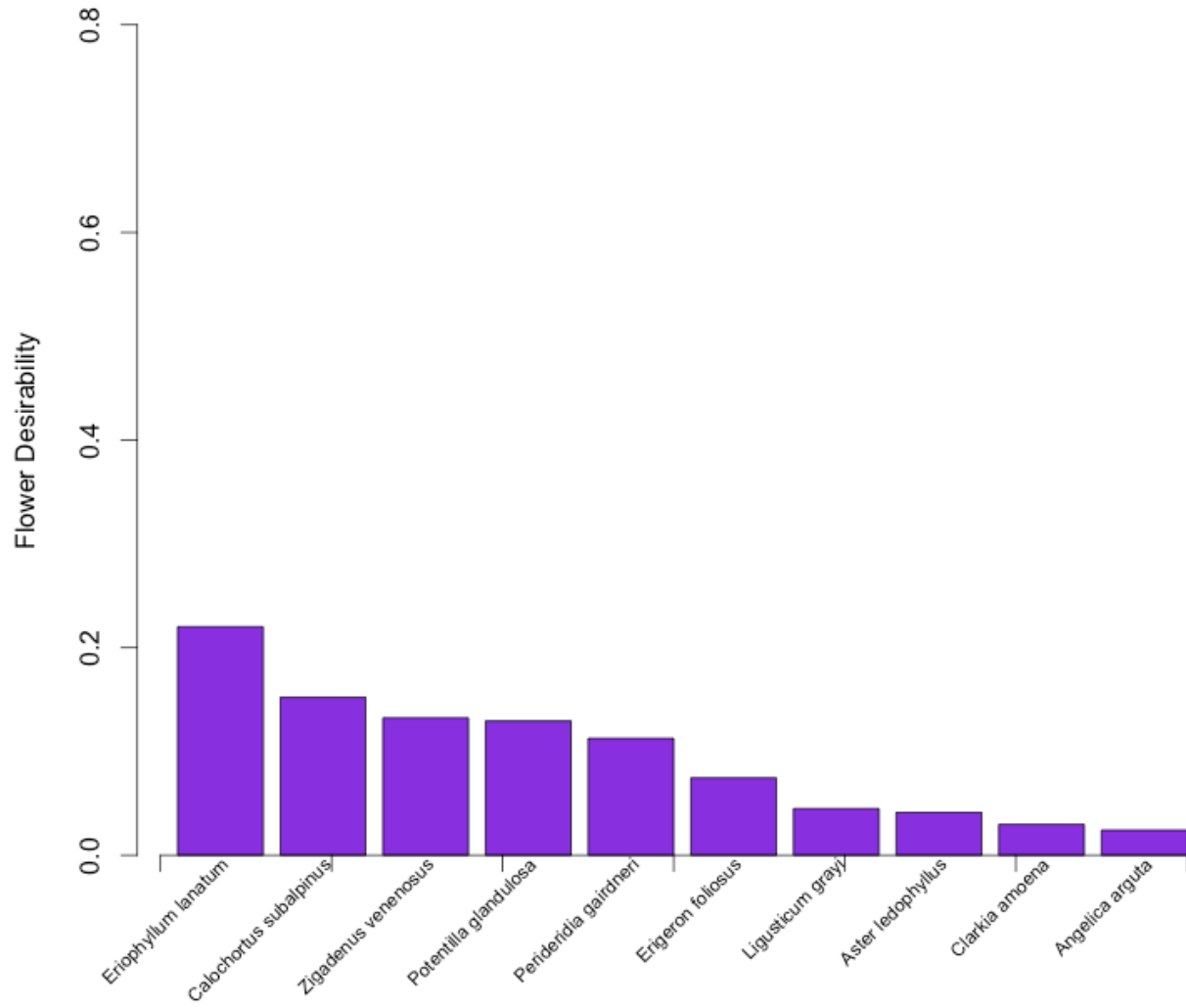**Top ten desirabilities of flower species for Eristalis hirtus**

Figure 5: Below are the abundances of the top ten most preferable flowers for Eristalis hirtus. Shading denotes years 2011, 2012 and 2013 respectively (as shading goes from lightest to darkest). Flowers are listed left to right, and top to bottom from most desirable to less desirable.

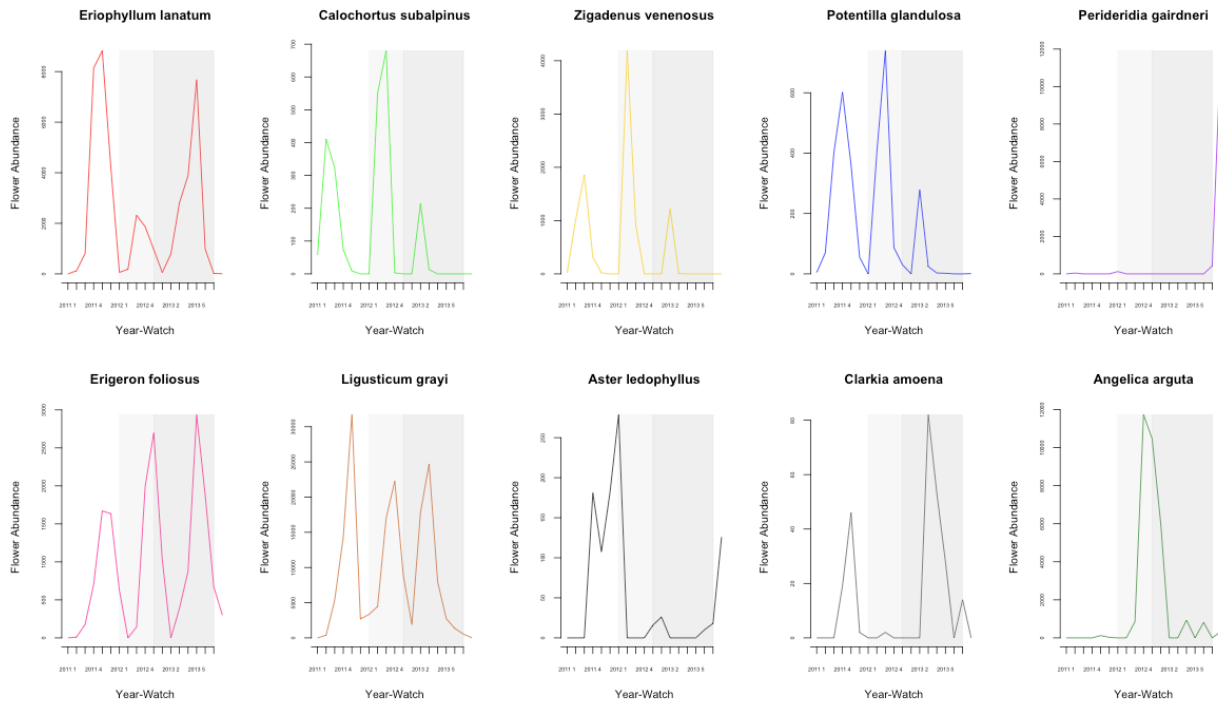**Flower Abundances over Time for Most Preferred Flowers of *Eristalis hirtus***

Figure 5.5: Below are the observed interactions (orange) and the interactions predicted by the optimized preferences list (blue) for Eristalis hirtus in Watch 8. While the chi-square test for goodness-of-fit rejects this optimized preferences list as a good predictor of interactions, we can see that the predictions are relatively close to the observed values.
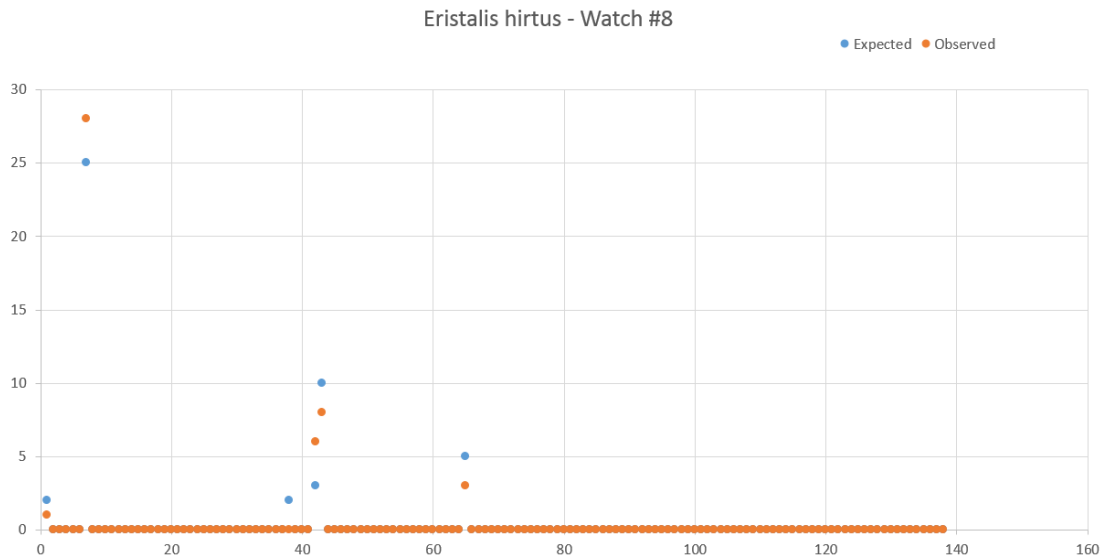
Discussion

Interpretation of Results

The low p-values for our tests on all four pollinators for Question 1) suggest that we reject the null hypothesis that pollinators in the study exhibit strictly frequency-dependent flower selection strategies. In other words, flower species frequency is not the only factor that determines a pollinator's selection strategy, though it could be one factor among many. It could be the case that pollinators are influenced by an innate, fixed preference list that is independent of external factors, or instead a fluid preference list that is determined by unknown factors.

The low p-values for our tests for *Apis mellifera*, *Bombylius major*, and *Bombus mixtus* for Question 2) (Figure 5.6 in the Appendix) suggest that we reject the null hypothesis that the interactions for these pollinators can be predicted accurately from their generalized preference list generated by our model.

The higher p-values for *Eristalis hirtus* (Figure 5.6 in the Appendix) suggest that the preference list for this pollinator generated by our model has predictive strength. It is interesting to note that the watch-years with good fits occur primarily in 2012 and 2013. It could be that the data in 2012 and 2013 was more consistent. Or, it could be the case that our preference list is a good predictor of a typical year for *Eristalis hirtus*, and 2011 represented an unusual year in the flower selection strategies of this species, either due to climate considerations or due to an unknown phenomenon in the local population of *Eristalis hirtus*. Furthermore, we note that almost all of the top ten most

preferred flowers for *Eristalis hirtus* are bowl-shaped, perhaps implying a general preference of this type of syrphid.

Areas for Improvement

This study could be improved in several ways. Though data used span three years, having a larger data set would make results of the analysis more accurate. It would hopefully then be possible to analyze preferences for pollinators whose current number of recorded interactions are relatively low.

Consistency for flower counting techniques between years is also an important element of the study to preserve. There are some discrepancies in methods of counting both flowers and number of interactions between years. Resolving these would result in more accurate data and a superior analysis.

Conclusion

We conclude from the data that pollinators do not express flower preferences which are based solely on flower frequency, i.e. that this data refutes the theory of strictly frequency-dependent foraging.

This result is especially interesting because of its implications for further research. If certain flowers are more desirable than others, what causes the desirability of a flower to be greater? Do certain flower traits influence desirability more than others? Furthermore, do different guilds of pollinators express similar preferences for flowers? Our analysis includes information about a honeybee, bumblebee, beefly and syrphid. If we were able to obtain preference lists for *Bombus mixtus*, *Bombylius major*, and *Apis mellifera* that had predictive strength, would we see a contrast in the preferred flowers?

We also conclude that our preference list for *Eristalis hirtus* generated by our model is a potential predictive tool for its flower selection strategy, though we cannot show rigorously that the list is a good fit.

References

Benadi, Gita, Hovestadt, Thomas, Poethke, Hans-Joachim & Blüthgen, Nico (2014) Specialization and phenological synchrony of plant-pollinator interactions along an altitudinal gradient. *Journal of Animal Ecology*, vol. 83, pp. 639-650.

Boesi, Roberto, Polidori, Carlo & Andrietti, Francesco (2008) Searching for the Right Target: Oviposition and Feeding Behavior in *Bombylius* Bee Flies (Diptera: Bombyliidae). *Zoological Studies*, vol. 48, part 2, pp. 141-150.

Dukas, Reuven & Real, Leslie A. (1991) Learning foraging tasks by bees: a comparison between social and solitary species. *Animal Behaviour*, vol. 42, issue 2, pp. 269-276.

Gong, Yan-Bing & Huang, Shuang-Quan (2011) Temporal stability of pollinator preference in an alpine plant comunity and its implications for the evolution of floral traits. *Oecologia*, DOI: 10.1007/s00442-011-1910-7.

Honeybee Genome Sequencing Consortium (2006) Insights into social insects from the genome of the honeybee *Apis mellifera*. *Nature*, DOI: 10.1038/nature05260.

Lee, Ted N. & Snow, Allison A. (1998) Pollinator Preferences and the Persistence of Crop Genes in Wild Radish Populations (*Raphanus raphanistrum*, Brassicaceae). *American Journal of Botany*, vol. 85, pp. 333-339.

Morandin, Lora A. & Kremen, Claire (2013) Bee Preference for Native versus Exotic Plants in Restored Agricultural Hedgerows. *Restoration Ecology*, DOI: 10.1111/j.1526-100X.2012.00876.x.

King, C , Ballantyne, GA & Willmer, PG (2013) Why flower visitation is a poor proxy for pollination: measuring single-visit pollen deposition, with implications for pollination networks and conservation. *Methods in Ecology and Evolution* , vol. 4, no. 9, pp. 811-818.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Sutherland, Steven D. & Vickery, Robert K. Jr. (1993). On the Relative Importance of Floral Color, Shape, and Nectar Rewards in Attracting Pollinators to *Mimulus*. *Great Basin Naturalist*, vol. 53, part 2. pp. 107-117.

Figures and Tables

Figure 5.6: Results of chi-squared tests under the null hypothesis that our generalized preference list for each pollinator correctly predicts observed interactions

APIS MELLIFERA

Watch-Year	p-value
2011 1	0.108
2011 2	0.527
2011 3	0
2011 4	0
2011 5	0
2011 6	0
2012 1	0.043
2012 2	0
2012 3	0
2012 4	0
2012 5	0
2013 1	0.947
2013 2	0
2013 3	0
2013 4	0
2013 5	0
2013 6	0
2013 7	0.044

BOMBUS MIXTUS

Watch-Year	p-value
2011 1	0
2011 2	0
2011 3	0
2011 4	0
2011 5	0
2011 6	0.007
2012 1	0
2012 2	0
2012 3	0
2012 4	0
2012 5	0.001
2013 1	0
2013 2	0
2013 3	0
2013 4	0
2013 5	0
2013 6	0
2013 7	0.07

BOMBYLIUS MAJOR

Watch-Year	p-value
2011 1	0
2011 2	0
2011 3	0
2012 1	0
2012 2	0
2012 5	1
2013 1	0
2013 2	1
2013 3	0.994

ERISTALIS HIRTUS

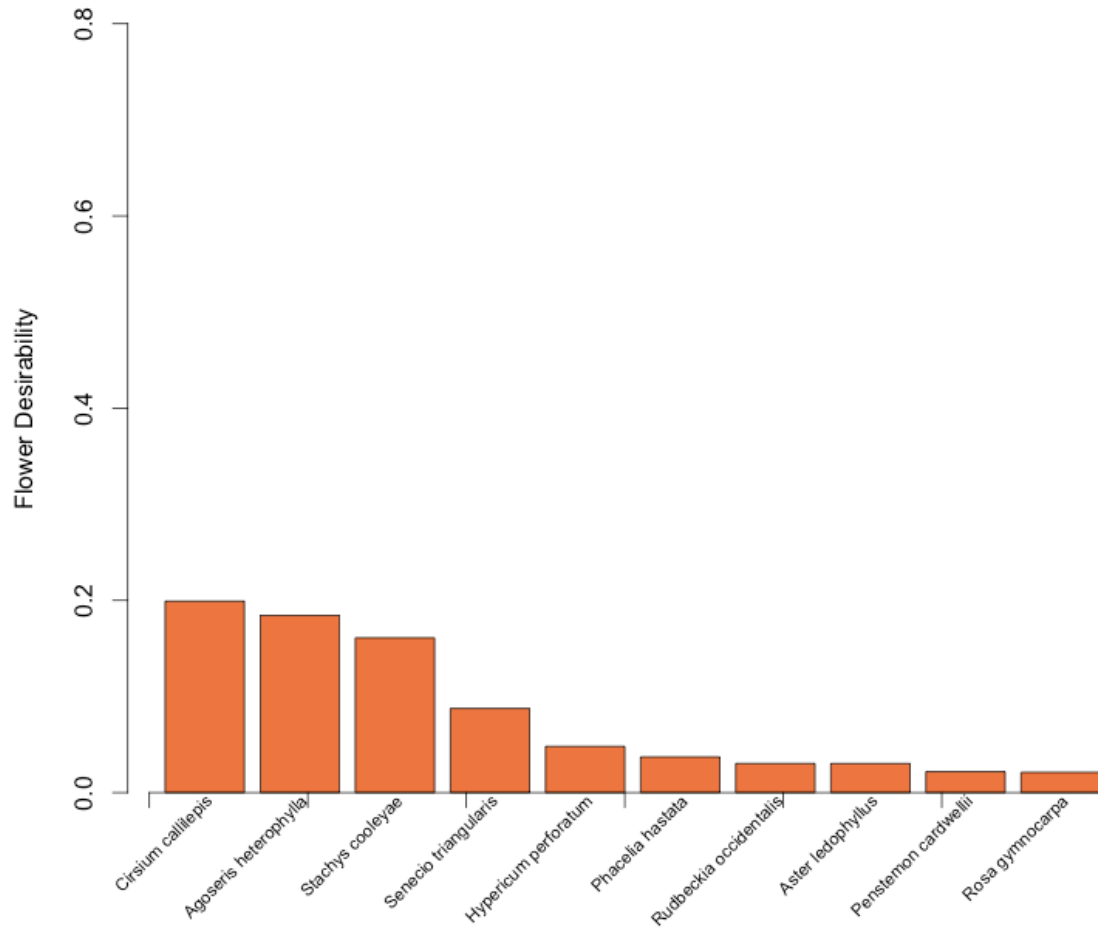| Watch-Year	p-value
2011 2	0
2011 3	0
2011 4	0
2011 5	0.762
2011 6	0.131
2012 1	0.974
2012 2	0
2012 3	1
2012 4	0.639
2013 7	0.986

Figure 6: Flower preference list graph for Bombus mixtus. Flowers presented are the top ten most preferable, as outputted from our maximization of the likelihood function.
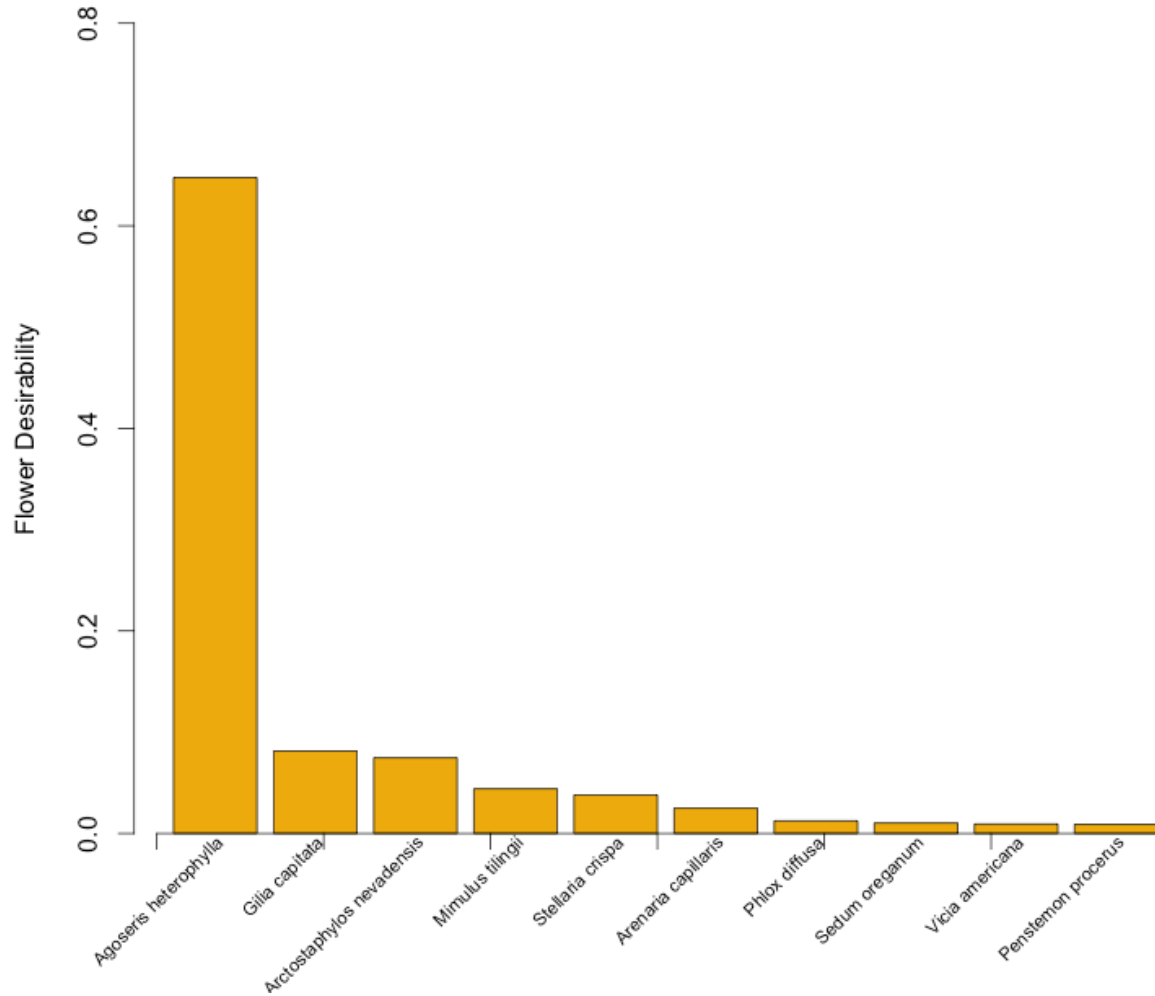
Figure 7: Flower preference list graph for Bombylius major. Flowers presented are the top ten most preferable, as outputted from our maximization of the likelihood function.

Appendix

Here is the R Code used for this study.

####################

pollinator_preferences_R_code.txt

by Kaitlin Horan and Ivan Pyzow

8/18/14

####################

INITIALIZATION

####################

setwd("/Users/Boss/Desktop/EISI_TESTING")

intData <- read.csv("interactions_v4.csv", stringsAsFactors = FALSE)

flowersData <- read.csv("flowers_v6.csv", stringsAsFactors = FALSE)

####################

DATABASE STUFF

####################

intData2 <- intData[order(intData$year, intData$DATE, intData$COMPLEX, intData$MEADOW),]

#gets rid of flagged rows

intData3 <- intData2[intData2$COMPLEX != "" & is.na(intData2$FLAGGED),]

flowersData <- flowersData[is.na(flowersData$FLAGGED),]

```
#adds a meadow-watch-year column

intData3$meadowWatchYear <- paste(intData3$WATCH, intData3$MEADOW, intData3$year)

flowersData$meadowWatchYear <- paste(flowersData$WATCH, flowersData$MEADOW,
flowersData$YEAR)


#adds a watch-year column

intData3$watchYear <- paste(intData3$year,intData3$WATCH)

flowersData$watchYear <- paste(flowersData$YEAR, flowersData$WATCH)


dwInt <- c()

z <- 1

for(i in 1:nrow(intData3)){

  if(!(intData3[i, 'meadowWatchYear'] %in% dwInt)){

    dwInt[z] <- intData3[i, 'meadowWatchYear']

    z <- z + 1

    }

}


dwPlant <- c()

q <- 1

for(k in 1:nrow(flowersData)){

  if(!(flowersData[k, 'meadowWatchYear'] %in% dwPlant)){

    dwPlant[q] <- flowersData[k, 'meadowWatchYear']

    q <- q + 1

  }

}


#Final list of meadow-watch-years that are present in both flower survey and interactions file
```

```r
dwFINAL <- intersect(dwPlant, dwInt)


#generate list of flowers using flowersData
flowerList <- c()
y <- 1
for(p in 1:nrow(flowersData)){
  if(!(flowersData[p, 'SPP_NAME'] %in% flowerList)){
    flowerList[y] <- flowersData[p, 'SPP_NAME']
    y <- y + 1
  }
}


newFlowers <- sort(flowerList)
flowerList2 <- newFlowers[-c(1)]
#flowerList2 <- flowerList1[-c(1)]
#for me, I don't need to remove two from the beginning, so I changed it.



##################################################
## MAKING INTERACTION AND ABUNDANCE MATRICES ##
##################################################


selectMeadowWatchYearList <- function(pollID, INTDATA, FLOWDATA){
  intDataSelected <- INTDATA[INTDATA$VISSP_CODE == pollID, ]
  mdw <<- intersect(unique(intDataSelected$meadowWatchYear),
unique(FLOWDATA$meadowWatchYear))
  return(mdw)
}
```

```r
selectWatchYearList <- function(pollID, INTDATA, FLOWDATA){
  intDataSelected <- INTDATA[INTDATA$VISSP_CODE == pollID, ]
  wy <<- intersect(unique(intDataSelected$watchYear), unique(FLOWDATA$watchYear))
  return(sort(wy))
}




makeAdjustedIntMatrix <- function(pollID, INTDATA, FLOWDATA, cutoff=0, cutoff2=5){
  intDataSelected <- INTDATA[INTDATA$VISSP_CODE == pollID, ]
  mdw <- selectMeadowWatchYearList(pollID, INTDATA, FLOWDATA)

  #this part eliminates the meadowWatchYears for which their are not enough rows of interactions
  for (m in mdw) {
    rowCount <- nrow(intDataSelected[intDataSelected$meadowWatchYear==m,]);
    if (rowCount < cutoff2) {mdw <- mdw[which(mdw!=m)]}
  }

  intMatrix <- matrix(data = 0, nrow = length(mdw), ncol = length(flowerList2))
  for(i in 1:nrow(intDataSelected)){
    intIndex <- match(intDataSelected[i, 'meadowWatchYear'], mdw)
    flowIndex <- match(intDataSelected[i, 'PLTSP_NAME'], flowerList2)
    intMatrix[intIndex, flowIndex] <- intMatrix[intIndex, flowIndex] + intDataSelected[i, 'NO_INT']
  }
  highRows <- rowSums(intMatrix) >= cutoff
  finalIntMatrix <- intMatrix[highRows, ]


  flowMatrix <- matrix(data = 0, nrow = length(mdw), ncol = length(flowerList2))
```

```r
  for(j in 1:nrow(FLOWDATA)){


    flowIndex <- match(FLOWDATA[j, 'SPP_NAME'], flowerList2)

    meadowWatchIndex <- match(FLOWDATA[j, 'meadowWatchYear'], mdw)

    flowMatrix[meadowWatchIndex, flowIndex] <- flowMatrix[meadowWatchIndex, flowIndex] +
as.numeric(FLOWDATA[j, 'TOT_FLW'])

    }
  finalFlowMatrix <- flowMatrix[highRows, ]


  final <- list()
  final[[1]] <- finalIntMatrix
  final[[2]] <- finalFlowMatrix
  final[[3]] <- mdw


  return(final)

}


makeWatchYearMatrix <- function(pollID, INTDATA, FLOWDATA, cutoff2=5){
  intDataSelected <- INTDATA[INTDATA$VISSP_CODE == pollID, ]
  mdw <- selectMeadowWatchYearList(pollID, INTDATA, FLOWDATA)
  wy  <- selectWatchYearList(pollID, INTDATA, FLOWDATA)


  #this part eliminates the meadowWatchYears for which their are not enough rows of interactions
  for (m in mdw) {
    rowCount <- nrow(intDataSelected[intDataSelected$meadowWatchYear==m,]);
    if (rowCount < cutoff2) {mdw <- mdw[which(mdw!=m)]}
  }


  intDataSelected2 <- intDataSelected[which (intDataSelected$meadowWatchYear %in% mdw), ]
```

```r
intMatrix <- matrix(data = 0, nrow = length(wy), ncol = length(flowerList2))
for(i in 1:nrow(intDataSelected2)){
    intIndex <- match(intDataSelected2[i, 'watchYear'], wy)
    flowIndex <- match(intDataSelected2[i, 'PLTSP_NAME'], flowerList2)
    intMatrix[intIndex, flowIndex] <- intMatrix[intIndex, flowIndex] + intDataSelected2[i,
'NO_INT']
  }


  finalIntMatrix <- intMatrix[which(rowSums(intMatrix)!=0),]


  flowDataSelected <- FLOWDATA[which (FLOWDATA$meadowWatchYear %in% mdw), ]


  flowMatrix <- matrix(data = 0, nrow = length(wy), ncol = length(flowerList2))
  for(j in 1:nrow(flowDataSelected)){


    flowIndex <- match(flowDataSelected[j, 'SPP_NAME'], flowerList2)
    watchyearIndex <- match(flowDataSelected[j, 'watchYear'], wy)
    flowMatrix[watchyearIndex, flowIndex] <- flowMatrix[watchyearIndex, flowIndex] +
as.numeric(flowDataSelected[j, 'TOT_FLW'])
  }


  finalFlowMatrix <- flowMatrix[which(rowSums(intMatrix)!=0),]


  finalwy <- wy[which(rowSums(intMatrix)!=0)]


  final <- list()
  final[[1]] <- finalIntMatrix
  final[[2]] <- finalFlowMatrix
```

```r
  final[[3]] <- finalwy


  return(final)

}



##############
## MODELING ##
##############


#modified logarithm function to prevent NaN
newlog <- function (x) ifelse(x==0,0,log(x))


#our negative log likelihood function
lik7 <- function (desList, xInt, xAbu)
{
  result <- 0;
  for (j in (1:dim(xInt)[1]) )
  {      result = result + sum (xInt[j,] * ( desList + newlog(xAbu[j,]) - newlog ( sum(exp(desList) *
xAbu[j,]) ) ) )
  }
  return(-result)
}



createPredictedInts2 <- function (desList, xInt, xAbu) {
        step1 <- t(t(xAbu)*desList)
        step2 <- step1/rowSums(step1)
        step3 <- rowSums(xInt)*step2
```

```
    return(step3)

    }



################################
## THE CHI-SQUARED FUNCTIONS ##
################################


chibaby <- function (optimized, matrix, i) {
  desAndAbu <- (exp(optimized$par)) * matrix[[2]][i,];
  observed <- matrix[[1]][i,][which(matrix[[2]][i,] != 0)];
  probs <- normalize(desAndAbu[which(matrix[[2]][i,] != 0)]);
  return(chisq.test(observed, p = probs))
}


chiBig <- function (optimized, matrix) {
  numWY <- length(matrix[[3]])
  result <- matrix(,numWY,2)
  result[,1] <- matrix[[3]]
  for (i in 1:numWY) {
    run <- chibaby(optimized, matrix, i)
    result[i,2] <- run$p.value
  }
  result[,2] <- round(as.numeric(result[,2]), digits =3)
  return(result)
}



chiAdult2 <- function (optimized, matrix) {
```

```r
  numWY <- length(matrix[[3]]);
  allexpected <- c()
  allobserved <- c()
  pred <- createPredictedInts2(normalize(exp(optimized$par)), matrix[[1]], matrix[[2]])
  pred2 <- pred
  for (i in 1:numWY) {
  observed <- matrix[[1]][i,][which(pred2[i,] != 0)];
  expected <- (pred[i,])[which(pred2[i,] != 0)];
  allexpected <-  c(allexpected,expected)
  allobserved <-  c(allobserved, observed)

  }
  #diff <- abs(allexpected-allobserved)
  #print(max(diff,10))
  #allexpected2 <- allexpected[which(diff != max(diff))]
  #allobserved2 <- allobserved[which(diff != max(diff))]
  #print(allobserved2)
  if (length(allexpected) == length(allobserved)) {
  return(chiOwn(allobserved, allexpected,length(allobserved)-1))
  } else return("no entiendo")
}

chiOwn <- function(obs0,exp0,df) {
  exp <- exp0[which(exp0 != 0)]
  obs <- obs0[which(exp0 != 0)]
  x2stat <- sum(((obs - exp)^2)/exp)
  result <- pchisq(x2stat,df,lower.tail=FALSE)
  print(round(exp,digits=2))
  print(obs)
  print (sum(round(exp,digits=2)))
```

```r
  print (sum(obs))

  print (round(exp,digits=2)-obs)

  print (df)

  return(result)

  }



##########################
## LOOKING AT SPECIFICS ##
##########################


#generate data for most frequent pollinators
listPoll <- sort(table(intData3$VISSP_NAME), decreasing = TRUE)
# get top ten most frequent pollinators
listPoll[1:20]


ints_mell2 <- makeAdjustedIntMatrix("APISMELL", intData3, flowersData)

ints_mixt2 <- makeAdjustedIntMatrix("BOMBMIXT", intData3, flowersData)

ints_hirt2 <- makeAdjustedIntMatrix("ERISHIRT", intData3, flowersData)

ints_majo2 <- makeAdjustedIntMatrix("BOMBMAJO", intData3, flowersData)


ints_mell3 <- makeWatchYearMatrix("APISMELL", intData3, flowersData)

ints_mixt3 <- makeWatchYearMatrix("BOMBMIXT", intData3, flowersData)

ints_hirt3 <- makeWatchYearMatrix("ERISHIRT", intData3, flowersData)

ints_majo3 <- makeWatchYearMatrix("BOMBMAJO", intData3, flowersData)


apismell2 <- optim(c(rep(0, ncol(ints_mell2[[1]]))), lik7, method="BFGS", xInt = ints_mell2[[1]],
xAbu = ints_mell2[[2]], control=list(trace=10))

bombmixt2 <- optim(c(rep(0, ncol(ints_mixt2[[1]]))), lik7, method="BFGS", xInt = ints_mixt2[[1]],
xAbu = ints_mixt2[[2]], control=list(trace=10))
```

```r
erishirt2 <- optim(c(rep(0, ncol(ints_hirt2[[1]]))), lik7, method="BFGS", xInt = ints_hirt2[[1]], xAbu
= ints_hirt2[[2]], control=list(trace=10))

bombmajo2 <- optim(c(rep(0, ncol(ints_majo2[[1]]))), lik7, method="BFGS", xInt =
ints_majo2[[1]], xAbu = ints_majo2[[2]], control=list(trace=10))



dlikrattest <- 2 * ( lik7(uniform$par, ints_hirt2[[1]], ints_hirt2[[2]]) - lik7(erishirt2$par,
ints_hirt2[[1]], ints_hirt2[[2]]) )


pchisq (dlikrattest, df=136, lower.tail=FALSE)


plot(ints_hirt3[[1]])


##############
## NEW SHIT ##
##############

makeColNames <- function (vec) {
  result <- c()
  for (n in 1:length(vec)) {
    result[3*n-2] <- vec[n]
    result[3*n-1] <- vec[n]
  }
}


makeCSV <- function (matrix1, matrix2, filename) {
  matrix1 <- (round(matrix1,digits=0))
  result <- matrix(,3*nrow(matrix1),ncol(matrix1) )
  for (i in 1:nrow(matrix1)) {
    result[3*i-2,] <- 1:ncol(matrix1)
```

```r
    result[3*i-1,] <- (matrix1[i,])
    result[3*i,] <- (matrix2[i,])
  }
  result <- round(result, digits=0)
  write.csv(result, filename)
}


pred <- createPredictedInts2(normalize(exp(erishirt2$par)),ints_hirt3[[1]], ints_hirt3[[2]] )
makeCSV( pred,ints_hirt3[[1]], "erisHirtwatches.csv")




##############################
## SAMPLE VALUES AND TESTS ##
##############################


aaa <- createPredictedInts2(normalize(exp(erishirt2$par)), ints_hirt3[[1]], ints_hirt3[[2]] )
chiOwn(ints_hirt3[[1]][1,],aaa,33)


uniform <- NULL
uniform$par = c(rep(0,138))



######################
## EXTRA FUNCTIONS ##
######################


normalize <- function (x) {x*(1/sum(x))}


extractMax <- c()
```

```r
q <- normalize(exp(test$par))
for(i in 1:10){
  extractMax[i] <- which.max(q)
  q[which.max(q)] <- 0
}


extractMax2 <- function (optimized) {
  extract <- c();
  q <- normalize(exp(optimized$par));
    for(i in 1:10){
    extract[i] <- which.max(q);
    q[which.max(q)] <- 0
  }
  return(flowerList2[extract])
}


removeZeroes <- function(pref) {
        pref <- pref[which(pref != 0)]
        return(pref)
        }


barplot(normalize(exp(test$par))[extractMax], names.arg = flowNames, xlab = "Flower Species",
ylab = "Flower Desirability", col= "blue", log="y", cex.axis = .01)


#check number of NAs is actually 0
sum(is.na(FLOWERS))



##########################
```

VALIDATION FUNCTIONS

###########################

```
checkCountInt <- c(rep(0, length(dwInt)))
for(i in 1:nrow(intData3)){
  if(intData3[i, 'meadowWatchYear'] %in% dwInt){
    q <- match(intData3[i, 'meadowWatchYear'], dwInt)

    checkCountInt[q] <- checkCountInt[q] + 1
  }
}


checkCountPlant <- c(rep(0, length(dwPlant)))
cate <- c()
for(i in 1:nrow(flowersData)){
  if(flowersData[i, 'meadowWatchYear'] %in% dwPlant & !flowersData[i, 'meadowWatchYear'] %in% cate){
    q <- match(flowersData[i, 'meadowWatchYear'], dwPlant)

    checkCountPlant[q] <- checkCountPlant[q] + 1
  }
}


#check to make sure flowers appearing in interaction file are in the flower survey
flowFromInt <- c()
y <- 1
for(p in 1:nrow(intData2)){
  if(!(intData2[p, 'PLTSP_NAME'] %in% flowFromInt)){
    flowFromInt[y] <- intData2[p, 'PLTSP_NAME']
```

```
    y <- y + 1
  }
}
flowFromInt1 <- sort(flowFromInt)
flowFromInt2 <- flowFromInt1[-c(1)]



length(intersect(flowFromInt2, flowerList2))

#find out what flowers are in interactions file that aren't in flower survey
p <- c()
s <- 1
for(i in 1:length(flowFromInt2)){
  if(!(flowFromInt2[i] %in% flowerList2)){
    p[s] <- flowFromInt2[i]
    s <- s + 1
  }
}
```