Muzz\_Thesis 2016-17

M. Abdur-Razak

May 26, 2017

# Chapter 2 - Impact of Biocontrol

Purple loosestrife (Lythrum *salicaria*) is an invasive species, introduced in North America from Europe in the 1800's. Since that time, it has rapidly propagated and evolved across its introduced range. In the 1990's, a biocontrol beetle was released to (effectively) maintain the fitness of North American Lythrum populations. The novel stress of the biocontrol beetle is of interest to evolutionary ecologists as Lythrum has undergone local adaptation and rapid evolution sans its main antagonist for ~200 years.

The following report contains the initial findings of biocontrol impact on a Lythrum population at the Queen's University Biological Station (QUBS). The conducted study, in Summer 2016, experimentally removed the biocontrol beetle from half of the marked individuals (n = 500) using the insecticide deltamethrin. The insecticide was applied in a checkerboard design across 22 transects. From June to August, individuals were visited every 4 days and measurements were taken on growth and phenological traits. Fitness was measured in seeds per plant - a function of seeds per fruit and fruit per plant measurements.

### load packages and data

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.2.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.5

library(zoo)

## Warning: package 'zoo' was built under R version 3.2.5

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(lubridate)

## Warning: package 'lubridate' was built under R version 3.2.5

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

ldata<-read.csv("oe3c\_dat.csv", header=T)  
names(ldata)

## [1] "indiv" "date" "trt" "fruit"   
## [5] "flower" "bud" "adult" "egg"   
## [9] "node" "total" "dmg\_leave" "avg\_dmg"   
## [13] "dmg\_index" "ap\_dmg" "initial\_beetle" "sec\_branch"   
## [17] "sec\_node" "sec\_branch\_rep" "sec\_ap1" "shoot\_height"   
## [21] "leaf\_length" "stem\_width" "cumulative"

**indiv** - the unique identifier for an indiviual. For example, C001 and C002.  
**date** - the visit date *number*. For example, June 10th is given a value of 1, July 4th is given 7, and August 17th is given 16.  
**trt** - the treatment assigned to an indiviual, either C (**unsprayed**) or P (**sprayed**). For example, C065 and P065.  
**bud** - the height (cm) of the bud.  
**flower** - the height (cm) of the flower. Used in determining time to flowering.  
**fruit** - the height (cm) of the fruit. Used in determining flower duration.  
**adult** - the number of adults found on an individual at that visit.  
**egg** - the number of eggs found on an individual at that visit.  
**node** - Lythrum produces two opposite leaves at 'nodes' in intervals across each stem. Here, a node represents a couplet of leaves, used in estimating plant growth and in calculating the damage index.  
**total** - total leaves on a plant. Estimated through node count.  
**dmg\_leave** - A count of individual leaves that had evidence of characteristic biocontrol "shot hole" damage.  
**avg\_dmg** - An estimate of average percent removal across all damaged leaves. Key fractions of percent removal, for example 1%, 5%, 10%, 25%, 50%, 75%, and 90% were used as a guide.  
**dmg\_index** - A standardized measurement of damage, from a scale of 0 to 1, that all plants could be placed on. **Index = (Damaged leaves / Total leaves) \* (Average Damage / 100)**  
**ap\_dmg** - if a plant's apical mersitem was damaged (characteristic to the beetle's foraging behaviour), ap\_dmg was given a value of 1. Else, it was kept as zero. Ap\_dmg was always reserved for the primary stem.  
**initial\_beetle** - once a plant had branched due to apical mersitem damage on the primary, to keep record of the apical meristem damage across the plant and to track the change in new primary stems, initial\_beetle (meaning the initial primary stem had its apical meristem damaged by beetles) was given a value of 1. The shift from ap\_dmg = 1 & initial\_beetle = 0 to ap\_dmg = 0 & initial\_beetle = 1 indicated a change in primary stem.  
**sec\_branch** - the number of secondary branches.  
**sec\_node** - similar to node count, but on the secondary branches **sec\_branch\_rep** - the number of secondary branches that have reached a reproductive stage **sec\_ap1** - the number of damaged meristems (value of 1) on secondary branches (i.e. non-primary stems) **shoot\_height** - the vegetative height (cm) of the primary stem (irrespective of ap\_dmg) from the soil surface to the beginning of the reproductive developement.  
**leaf\_length** - the length (cm) of the longest leaf of the plant, usually taken in the mid section of the shoot\_height.  
**stem\_width** - the width (mm) of the primary stem, taken at the thickest section along shoot\_height.  
**cumulative** - the cumulative reproductive height (cm) of the secondary branches.

## Add Julian date to dataset based on visit day number

# Make a new dataframe to hold true dates of the year, rather than only visit date number  
Visitdata<-ldata  
Visitdata$date<-as.factor(Visitdata$date)  
  
# Visit dates from 1 to 16 will be bound to actual dates in visitdate  
date<-c(1:16)  
visitdate<-c("2016/06/10","2016/06/14","2016/06/18","2016/06/22","2016/06/26","2016/06/30","2016/07/04","2016/07/12","2016/07/16","2016/07/20","2016/07/24", "2016/08/01", "2016/08/05", "2016/08/09", "2016/08/13", "2016/08/17")  
  
# intermediate dataframe Vdat used to add actual dates into Visitdata, which will be used from hence  
Vdat<-as.data.frame(cbind(date,visitdate))  
Visitdata<- left\_join(Visitdata, Vdat)

## Joining, by = "date"

Visitdata$yday <- yday(Visitdata$visitdate)

# Chapter 2.1 [Codename: Lucy Loosestrife] - How much damage does the biocontrol do?

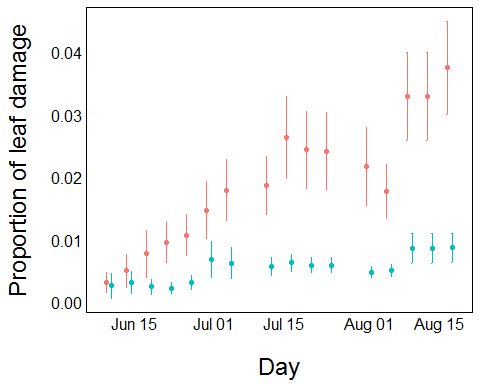
### Calculating leaf damage using dmg\_index

As explained above, we can use the data collected from the summer to calculate overall leaf damage across each entire plant using the equation: Index = (Damaged leaves / Total leaves) \* (Average Damage / 100). The way it is coded below is by calculating the standard error (SE) using the standard deviation (SD) and the total number of plants (N) per treatment. We can then plot the SE to show the error bars (see leaf damage plot).

# Variable to hold the actual Julian dates  
ldata$visitdate<-NA  
for(i in 1:nrow(Vdat)){  
 ldata$visitdate[ldata$date==Vdat$date[i]]<-paste0(Vdat$visitdate[i])  
}  
ldata$visitdate<-as.factor(ldata$visitdate)  
# Calculate SE for dmg  
dmg\_sd<-aggregate(dmg\_index ~ trt + visitdate,data=ldata,FUN=sd,na.action=na.omit)  
names(dmg\_sd)[3]<-"dmg\_sd"  
dmg\_N<-aggregate(dmg\_index ~ trt + visitdate,data=ldata,FUN=length,na.action=na.omit)  
names(dmg\_N)[3]<-"dmg\_N"  
  
dmgsum<-aggregate(dmg\_index ~ trt + visitdate,data=ldata,FUN=mean,na.action=na.omit)  
dmgsum<-merge(dmgsum,dmg\_sd,by=c("trt","visitdate"))  
dmgsum<-merge(dmgsum,dmg\_N,by=c("trt","visitdate"))  
dmgsum$dmg\_se<-dmgsum$dmg\_sd/sqrt(dmgsum$dmg\_N - 1)

### Plotting leaf damage

dmgsum$visitdate<-as.Date(dmgsum$visitdate)  
pd <- position\_dodge(width = 2)  
  
ggplot(aes(x=visitdate,y=dmg\_index,colour=trt),data=dmgsum)+  
 geom\_errorbar(aes(ymin=dmg\_index-1.96\*dmg\_se, ymax=dmg\_index+1.96\*dmg\_se),position=pd, width=.1)+geom\_point(position=pd) +  
 theme\_simple() +  
 xlab("\nDay") +  
 ylab("Proportion of leaf damage\n")



### Significance test

mod1 <- lm(dmg\_index ~ trt\*visitdate, data=dmgsum)  
anova(mod1)

## Analysis of Variance Table  
##   
## Response: dmg\_index  
## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 1 0.00150519 0.00150519 164.37 3.081e-13 \*\*\*  
## visitdate 1 0.00097074 0.00097074 106.00 5.037e-11 \*\*\*  
## trt:visitdate 1 0.00045403 0.00045403 49.58 1.168e-07 \*\*\*  
## Residuals 28 0.00025641 0.00000916   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

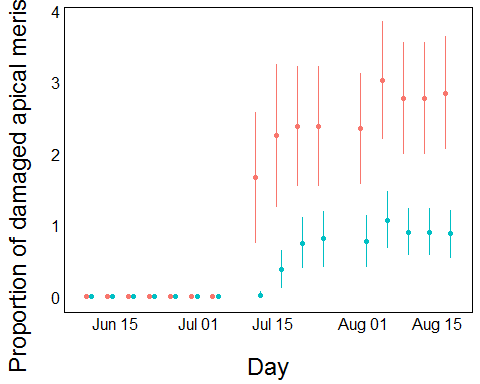
### Calculating apical mersitem damage

Another important way of expressing overall damage is to look at how many apical meristems had a damage of one. Here, our code looks at the spread of apical meristem damage (or accumulated apical meristem damage) across the secondary branches (i.e when sec\_ap1 >0). We can use the same method of calculating a plotting the standard error of apical meristem damage.

# Undamaged secondary meristems  
ldata$sec\_ap\_un<-ldata$sec\_branch-ldata$sec\_ap1  
# Rename damaged apical meristems to punctured apical meristem (pct)  
ldata$sec\_ap\_pct<-ldata$sec\_ap1  
  
  
# Bind the average number of secondary branches with average damaged secondary meristems for each treatment per date   
apsum<-aggregate(cbind(sec\_branch,sec\_ap\_pct) ~ trt + visitdate,data=ldata,FUN=mean,na.action=na.omit)  
# Calculate SE for secondary branches  
sec\_branch\_sd<-aggregate(sec\_branch ~ trt + visitdate,data=ldata,FUN=sd,na.action=na.omit)  
names(sec\_branch\_sd)[3]<-"sec\_branch\_sd"  
sec\_branch\_N<-aggregate(sec\_branch ~ trt + visitdate,data=ldata,FUN=length,na.action=na.omit)  
names(sec\_branch\_N)[3]<-"sec\_branch\_N"  
# Calculate SE for proportion damaged secondary branches  
sec\_ap\_pct\_sd<-aggregate(sec\_ap\_pct ~ trt + visitdate,data=ldata,FUN=sd,na.action=na.omit)  
names(sec\_ap\_pct\_sd)[3]<-"sec\_ap\_pct\_sd"  
sec\_ap\_pct\_N<-aggregate(sec\_ap\_pct ~ trt + visitdate,data=ldata,FUN=length,na.action=na.omit)  
names(sec\_ap\_pct\_N)[3]<-"sec\_ap\_pct\_N"  
  
# Merge everything  
apsum<-merge(apsum,sec\_branch\_sd,by=c("trt","visitdate"))  
apsum<-merge(apsum,sec\_branch\_N,by=c("trt","visitdate"))  
apsum$sec\_branch\_se<-apsum$sec\_branch\_sd/sqrt(apsum$sec\_branch\_N - 1)  
  
apsum<-merge(apsum,sec\_ap\_pct\_sd,by=c("trt","visitdate"))  
apsum<-merge(apsum,sec\_ap\_pct\_N,by=c("trt","visitdate"))  
apsum$sec\_ap\_pct\_se<-apsum$sec\_ap\_pct\_sd/sqrt(apsum$sec\_ap\_pct\_N - 1)

### Plotting secondary apical meristem damage

# Change date from factor to date. This is for plotting purposes  
apsum$visitdate<-as.Date(apsum$visitdate)  
# Add dodge to the points  
pd <- position\_dodge(width = 2)  
  
ggplot(aes(x=visitdate,y=sec\_ap\_pct,colour=trt),data=apsum)+  
 geom\_errorbar(aes(ymin=sec\_ap\_pct-1.96\*sec\_ap\_pct\_se, ymax=sec\_ap\_pct+1.96\*sec\_ap\_pct\_se),position=pd, width=.1)+geom\_point(position=pd) +  
 theme\_simple()+  
 xlab("\nDay") +  
 ylab("Proportion of damaged apical meristems\n")



### Sigificance test

mod1 <- lm(sec\_ap\_pct ~ trt \* visitdate, data=apsum)  
anova(mod1)

## Analysis of Variance Table  
##   
## Response: sec\_ap\_pct  
## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 1 7.9538 7.9538 58.91 2.320e-08 \*\*\*  
## visitdate 1 19.8026 19.8026 146.67 1.199e-12 \*\*\*  
## trt:visitdate 1 5.1090 5.1090 37.84 1.219e-06 \*\*\*  
## Residuals 28 3.7804 0.1350   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

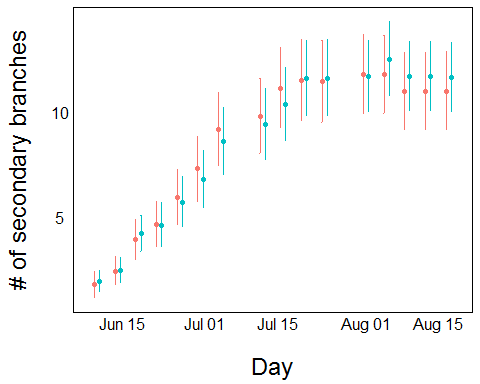
# Chapter 2.2 [Codename: Mykenzie the Summer Frenzy] - What was the effect of the beetle on growth and phenology

### The Predictive Model

Because there was a significant difference in damage for our treatments, we can proceed with a predictive model. This model postulates that more damaged Lythrum will experience **more** branching, **delayed** flowering time, and **lower** fitness (seeds per plant).

### Plotting secondary branching over the season

ggplot(aes(x=visitdate,y=sec\_branch,colour=trt),data=apsum)+  
 geom\_errorbar(aes(ymin=sec\_branch-1.96\*sec\_branch\_se, ymax=sec\_branch+1.96\*sec\_branch\_se),position=pd, width=.1)+geom\_point(position=pd) +  
 theme\_simple() +  
 xlab("\nDay") +  
 ylab("# of secondary branches\n")



### Significance test

mod1 <- lm(sec\_branch ~ trt\*visitdate, data=apsum)  
anova(mod1)

## Analysis of Variance Table  
##   
## Response: sec\_branch  
## Df Sum Sq Mean Sq F value Pr(>F)   
## trt 1 0.03 0.03 0.0128 0.9107   
## visitdate 1 332.24 332.24 144.8960 1.385e-12 \*\*\*  
## trt:visitdate 1 0.44 0.44 0.1939 0.6631   
## Residuals 28 64.20 2.29   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We can see here that there was no significant difference between branching. **NB: This is just for flowering plants. You need to check to see if there is a significant difference in unflowering plants, or just total plants together. Getting a proportion of plants that reached flowering could also be included as a graph, or perhaps as a line in your manuscript**

### Calculating date of first flower

Part of understanding how the beetle impacts plant fitness is understanding how it delays the time to flowering. In order to calculate the time to first flowering, we can create a for loop to go through each individual and scan for the first value where flower > 0 and it's corresponding date. We can also calculate flowering duration by scanning each indivual looking for when fruit > 0 and flower & bud == 0.

## calculate time to flowering, flowering date (relative to start), flowering duration   
fdata<-as.data.frame(unique(Visitdata$indiv))  
names(fdata)<-"indiv"  
fdata$ftime<-NA ## time to flowering  
fdata$fday<-NA ## day of flowering  
fdata$fdur<-NA ## duration of flowering  
  
for (ind in unique(Visitdata$indiv)) {  
 # subset all rows for this individual  
 individualdata<-Visitdata[Visitdata$indiv==ind,]  
 # subset out rows with no measurements  
 individualdata<-individualdata[!is.na(individualdata$node),]  
 # make indicator for days with a flower measurement  
 individualdata$indicatorfstart<-ifelse(individualdata$flower>0, TRUE, FALSE)  
 # find the first time that flowers is positive   
 floweringday<-min(which(individualdata$indicatorfstart == TRUE))  
 # ftime is time to flowering, the day of first flower growth minus the start day of measurement  
 fdata[fdata$indiv==ind,]$ftime <- individualdata$yday[floweringday]- individualdata$yday[1]  
 # fday is the day that flower growth first appears  
 fday<- individualdata$yday[floweringday]  
 fdata[fdata$indiv==ind,]$fday<- fday  
 # subset data for only the visits with flowering, fruiting, or buds  
 individualdata<-individualdata[individualdata$fruit>0 | individualdata$flower >0 | individualdata$bud>0,]  
 # new indicator where bud and flowers both equal zero, there will only be fruit measurements  
 individualdata$indicatorfend<-ifelse(individualdata$bud==0 & individualdata$flower==0, TRUE, FALSE)  
 # find the first day with only fruit measurements, no flowers or buds, taken as maximum from last visit  
 durend<- max(which(individualdata$indicatorfend == FALSE))  
 # duration of flowering is the last day of flowering minus the first day of flower growth (as calculated above)  
 fdata[fdata$indiv==ind,]$fdur<-individualdata$yday[durend] - fday  
}

## Warning in min(which(individualdata$indicatorfstart == TRUE)): no non-  
## missing arguments to min; returning Inf

## Warning in max(which(individualdata$indicatorfend == FALSE)): no non-  
## missing arguments to max; returning -Inf

## Warning in min(which(individualdata$indicatorfstart == TRUE)): no non-  
## missing arguments to min; returning Inf

## Warning in max(which(individualdata$indicatorfend == FALSE)): no non-  
## missing arguments to max; returning -Inf

## Warning in min(which(individualdata$indicatorfstart == TRUE)): no non-  
## missing arguments to min; returning Inf

## Warning in max(which(individualdata$indicatorfend == FALSE)): no non-  
## missing arguments to max; returning -Inf

## Warning in min(which(individualdata$indicatorfstart == TRUE)): no non-  
## missing arguments to min; returning Inf

## Warning in max(which(individualdata$indicatorfend == FALSE)): no non-  
## missing arguments to max; returning -Inf

# Organize the data by treatment (for easy graphing)  
fdata<-na.omit(fdata)  
fdata$trt<-gsub("[0-9]","",fdata$indiv)  
test<-fdata[order(fdata$trt, fdata$ftime),]  
test$cumulative<-c(1:109, 1:129)  
test<-test[order(test$trt, decreasing=TRUE),]  
  
# Remove point -- CHECK DATASET FOR PROBLEM  
test2 <- test %>%   
 filter(fday <230)

## Flowering time Plot

ggplot(test2, aes(x=fday, y=cumulative, color=trt)) +  
 geom\_smooth() +  
 theme\_simple() +  
 xlab("\nDay") +  
 ylab("# of flowering plants\n")

## `geom\_smooth()` using method = 'loess'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 206

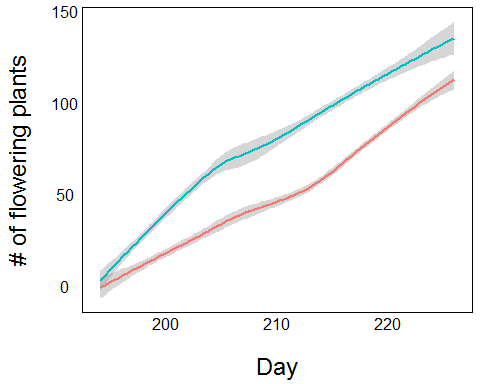
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 8

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 2.8031e-016

## Warning in predLoess(object$y, object$x, newx = if  
## (is.null(newdata)) object$x else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used  
## at 206

## Warning in predLoess(object$y, object$x, newx = if  
## (is.null(newdata)) object$x else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
## 8

## Warning in predLoess(object$y, object$x, newx = if  
## (is.null(newdata)) object$x else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal  
## condition number 2.8031e-016



## Significance test

We can use a linear model to see if there is a significant difference between treatment. Cumulative (seperate from initial 'cumulative' defined at the beginning) here is the number of cumulative flowering plants, seperated per treatment, per day.

mod1 <- lm(fday ~ cumulative \* trt, data = test2)  
anova(mod1)

## Analysis of Variance Table  
##   
## Response: fday  
## Df Sum Sq Mean Sq F value Pr(>F)   
## cumulative 1 15307.3 15307.3 2869.703 < 2.2e-16 \*\*\*  
## trt 1 2957.5 2957.5 554.456 < 2.2e-16 \*\*\*  
## cumulative:trt 1 142.4 142.4 26.687 5.135e-07 \*\*\*  
## Residuals 233 1242.8 5.3   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

As we see here, there is a significant difference in that damaged plants were delayed in their flowering.

# Chapter 2.3 [Codename: Sierra salicaria] - How does the beetle affect fitness, and does it interfere with pollination?

seed\_dat <- read.csv("seeds\_per\_plant.csv", header=T)  
names(seed\_dat)

## [1] "trt" "plantid" "site"   
## [4] "prim\_nfrts" "sec\_nfrts" "plantid.1"   
## [7] "prim\_avgseed" "sec\_avgseed" "prim\_seeds\_per\_plant"  
## [10] "sec\_seeds\_per\_plant" "dmg\_index"

Seeds per plant is used here as the