

Figure 3c

```
library(readxl)
library(lubridate)

##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
library(mgcv)

## Loading required package: nlme
## This is mgcv 1.8-34. For overview type 'help("mgcv-package")'.
library(ggplot2)
library(ggsignif)
library(readr)
library(emmeans)
library(tidyr)
library(plyr)
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:plyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
## The following object is masked from 'package:nlme':
##
##   collapse
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
#load in raw data
dat <- data.frame(read_csv("3c_data.csv"))

##
## -- Column specification -----
## cols(
##   `Repeat(Date)` = col_date(format = ""),
##   Algae = col_character(),
```

```
## Treatment = col_character(),
## Well_ID = col_character(),
## Larvae = col_character(),
## Time_to_expulsion = col_double()
## )
```

```
## Warning: 14 parsing failures.
```

```
## row      col      expected      actual      file
## 43 Repeat(Date) valid date 2020/11/31 '3c_data.csv'
## 44 Repeat(Date) valid date 2020/11/31 '3c_data.csv'
## 45 Repeat(Date) valid date 2020/11/31 '3c_data.csv'
## 46 Repeat(Date) valid date 2020/11/31 '3c_data.csv'
## 47 Repeat(Date) valid date 2020/11/31 '3c_data.csv'
## ... .....
## See problems(...) for more details.
```

```
head(dat)
```

```
##   Repeat.Date. Algae Treatment      Well_ID Larvae Time_to_expulsion
## 1  2020-07-16 SSB01      DMSO S_D_2020-07-16      1           720
## 2  2020-07-16 SSB01      DMSO S_D_2020-07-16      2           720
## 3  2020-07-16 SSB01      DMSO S_D_2020-07-16      3           720
## 4  2020-07-16 SSB01      DMSO S_D_2020-07-16    4-1           720
## 5  2020-07-16 SSB01      DMSO S_D_2020-07-16    4-2           720
## 6  2020-07-16 SSB01      DMSO S_D_2020-07-16      5           720
```

```
#clean up raw data
```

```
dat$Algae <- factor(dat$Algae, levels= c("SSB01", "M. gaditana", "N. oculata", "C. velia"))
dat$Treatment <- factor(dat$Treatment)
dat$Repeat.Date. <- factor(as.character(dat$Repeat.Date.))
dat$Well_ID <- factor(dat$Well_ID)
dat$Time_to_expulsion <- ifelse(dat$Time_to_expulsion<720,1,0)
head(dat)
```

```
##   Repeat.Date. Algae Treatment      Well_ID Larvae Time_to_expulsion
## 1  2020-07-16 SSB01      DMSO S_D_2020-07-16      1           0
## 2  2020-07-16 SSB01      DMSO S_D_2020-07-16      2           0
## 3  2020-07-16 SSB01      DMSO S_D_2020-07-16      3           0
## 4  2020-07-16 SSB01      DMSO S_D_2020-07-16    4-1           0
## 5  2020-07-16 SSB01      DMSO S_D_2020-07-16    4-2           0
## 6  2020-07-16 SSB01      DMSO S_D_2020-07-16      5           0
```

```
#model data with generalised linear mixed modes
```

```
summary <- data.frame()
```

```
for(i in unique(dat$Algae))
```

```
{
```

```
#i <- unique(dat$Algae)[1]
```

```
Dat <- droplevels(dat[dat$Algae==i,])
```

```
##Regression model:
```

```
mod <- gam(Time_to_expulsion ~ Treatment + s(Well_ID,bs='re'),data=Dat,family=binomial())
```

```
S <- summary(mod)$p.table
```

```
if(S[2,4]>0.1)
```

```
{
```

```
  p <- ""
```

```
}
```

```
if(S[2,4]<0.1)
```

```

{
  p <- ""
}
if(S[2,4]<0.05)
{
  p <- "*"
}
if(S[2,4]<0.01)
{
  p <- "**"
}
if(S[2,4]<0.001)
{
  p <- "***"
}
newline <- data.frame(Algae=i,outcome=S[1,1],SE=S[1,2],type="DMSO",p=NA,pvalue=NA)
summary <- rbind(summary,newline)
newline <- data.frame(Algae=i,outcome=S[1,1] + S[2,1],SE=S[2,2],type="XMD17-109",p=p, pvalue=S[2,4])
summary <- rbind(summary,newline)
}

summary$up <- plogis(summary$outcome + 1.96*summary$SE)
summary$down <- plogis(summary$outcome - 1.96*summary$SE)
summary$outcome <- plogis(summary$outcome)

summary

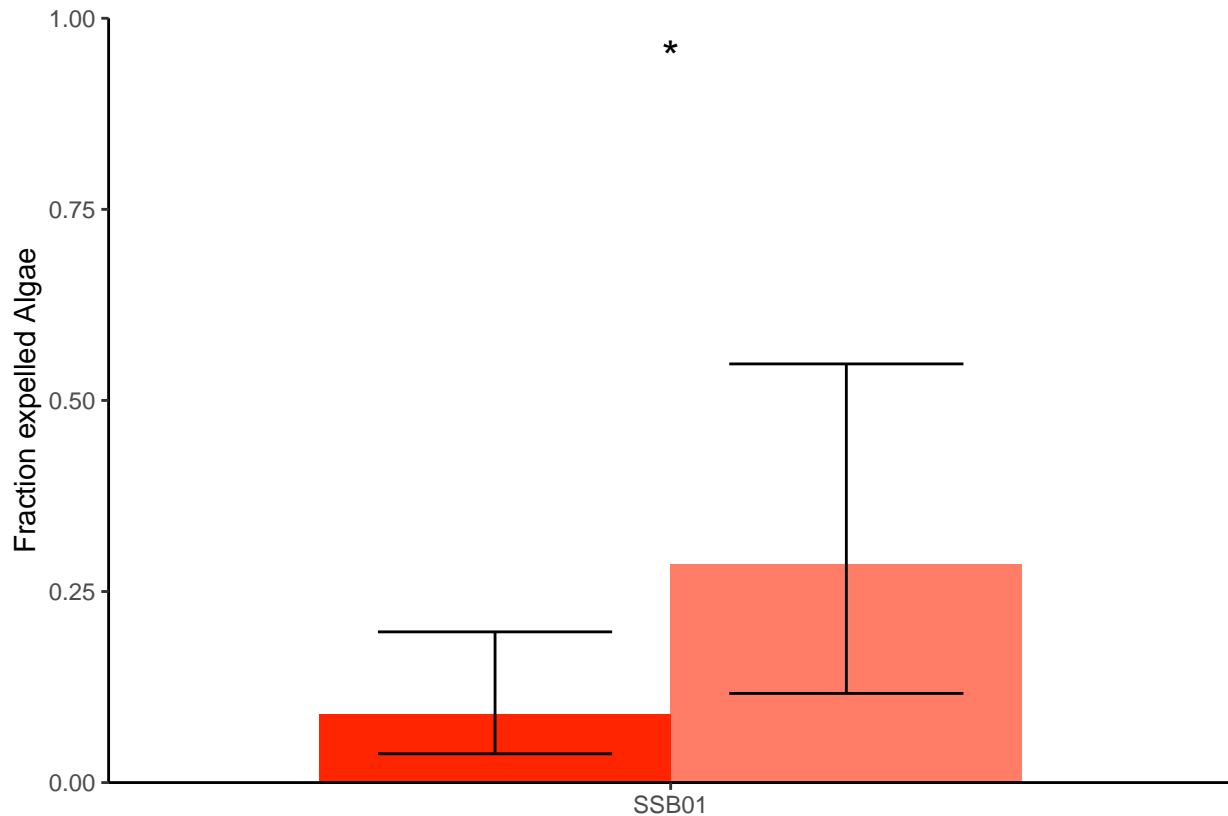
##   Algae   outcome      SE      type    p      pvalue      up      down
## 1 SSB01 0.08928574 0.4686244    DMSO <NA>      NA 0.1972006 0.03765558
## 2 SSB01 0.28571430 0.5653405 XMD17-109    * 0.01287645 0.5477978 0.11666912

write_csv(summary, "3c_summary.csv")

#Plot graph
P1<-ggplot(data = summary,aes(x = Algae, y = outcome, group = type)) +
  geom_bar(stat = "identity", width = 0.75, size=1.5, position = "dodge", fill = c("#FF2500",alpha("#FF2500",0.5))) +
  geom_errorbar(aes(ymin=down, ymax=up), width=.5, position =position_dodge(.75)) +
  geom_text(data = summary, aes(x=Algae,y=0.95),label=summary$p,size=6) +
  theme_classic() +
  labs(x = "", y = "Fraction expelled Algae") +
  scale_y_continuous(expand = c(0, 0), limits = c(0,1), labels = scales::number_format(accuracy = 0.01))
plot(P1)

## Warning: Removed 1 rows containing missing values (geom_text).

```



```
ggsave("Figure_3c.pdf",width=5.75,height=11,units="cm")
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
# get n count per treatment
```

```
ndat <- ddpoly(dat, c("Algae", "Treatment", "Repeat.Date."), summarise,  
              N     = length(Larvae))
```

```
n_Figure <- ddpoly(ndat, c("Algae", "Treatment"), summarise,  
                  N     = length(Repeat.Date.))
```

```
n_Figure
```

```
##   Algae Treatment N
```

```
## 1 SSB01      DMSO 5
```

```
## 2 SSB01 XMD17-109 5
```

```
write_csv(n_Figure, "3c_n.csv")
```