Figure 4 ReaChR experiments and analysis

Emily Jane Dennis

9/22/2017

Purpose:

To plot all single animal tracking data for the manuscript.

Background:

This is an R Markdown document. See here for more details on using R Markdown. I also found this site was useful for learning basic formatting with Markdown. I also enjoyed this site when learning how to use reshape and when learning how to tidy data in general.

The code required to get these graphs is in the gray boxes, along with information on how to run the statistical tests reported in the paper and any plots that were used to determine if these statistical tests were generally appropriate.

This was run in RStudio, using R version 3.3.1 (2016-10-31), Sincere Pumpkin Patch on a x86_64-apple-darwin13.4.0

```
library(plyr)
library(dplyr)
library(ggplot2)
library(reshape2)
library(MASS)
library(ggthemes)
library(car)
library(multcompView)
```

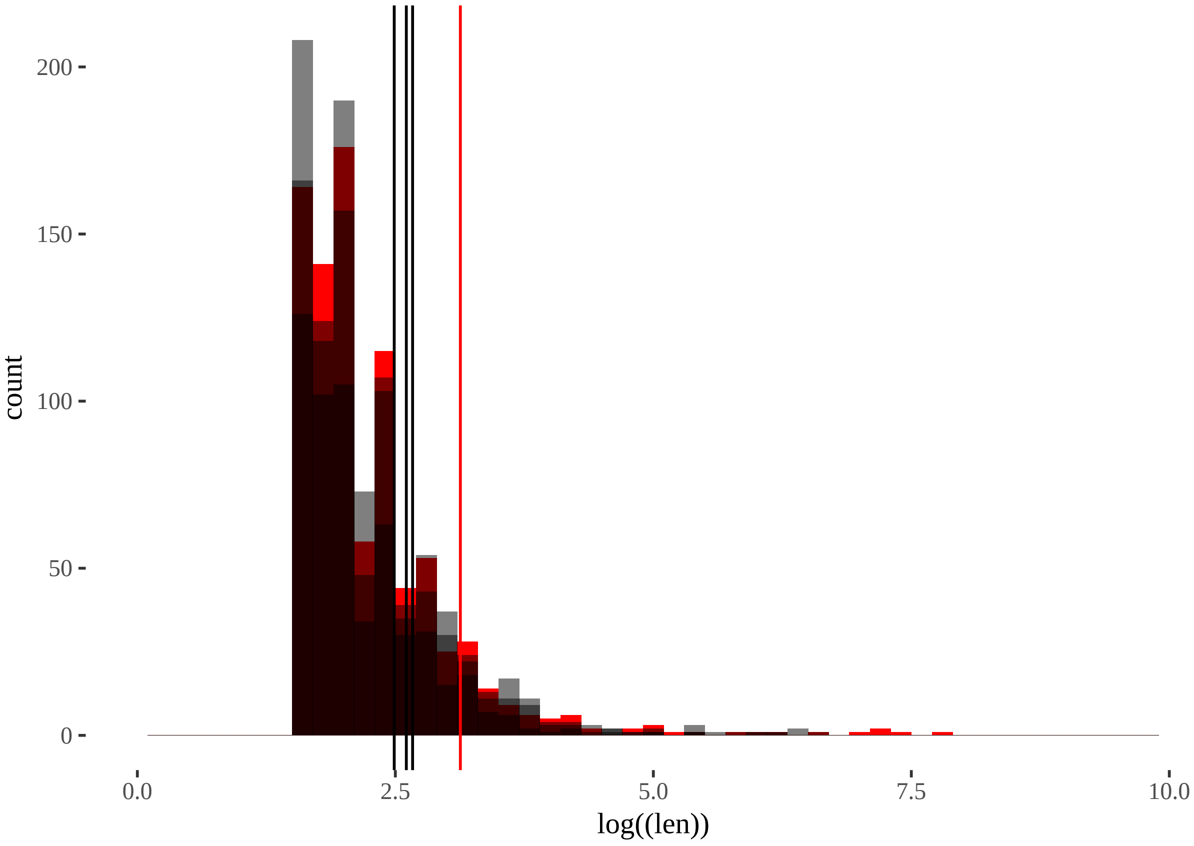
```
dfforplot <- read.csv('tracking_reachr.csv',header=TRUE)
```

```
fx <- subset(dfforplot,treatment=='Fx')
fr <- subset(dfforplot,treatment=='FR')
xr <- subset(dfforplot,treatment=='xR')
xx <- subset(dfforplot,treatment=='xx')

frend <- fr$fra+fr$len
fr <- cbind(fr,frend)
xxend <- xx$fra+xx$len
xx <- cbind(xx,xxend)
xrend <- xr$fra+xr$len
xr <- cbind(xr,xrend)
fxend <- fx$fra+fx$len
fx <- cbind(fx,fxend)

fron <- subset(fr,lightlist=='im')
froff <- subset(fr,lightlist=='FF')
xxon <- subset(xx,lightlist=='im')
xxoff <- subset(xx,lightlist=='FF')
xron <- subset(xr,lightlist=='im')
xroff <- subset(xr,lightlist=='FF')
```

```r
fxon <- subset(fx,lightlist=='im')
fxoff <- subset(fx,lightlist=='FF')

fxonsort <- fxon[ order(fxon$len), ]
fxonsort$y <- c(1:dim(fxonsort)[1])
fxonsort$yend <- c(1:dim(fxonsort)[1])
fxoffsort <- fxoff[ order(fxoff$len), ]
fxoffsort$y <- c(1:dim(fxoffsort)[1])
fxoffsort$yend <- c(1:dim(fxoffsort)[1])

fronsort <- fron[ order(fron$len), ]
fronsort$y <- c(1:dim(fronsort)[1])
fronsort$yend <- c(1:dim(fronsort)[1])
froffsort <- froff[ order(froff$len), ]
froffsort$y <- c(1:dim(froffsort)[1])
froffsort$yend <- c(1:dim(froffsort)[1])

xxonsort <- xxon[ order(xxon$len), ]
xxonsort$y <- c(1:dim(xxonsort)[1])
xxonsort$yend <- c(1:dim(xxonsort)[1])
xxoffsort <- xxoff[ order(xxoff$len), ]
xxoffsort$y <- c(1:dim(xxoffsort)[1])
xxoffsort$yend <- c(1:dim(xxoffsort)[1])


xronsort <- xron[ order(xron$len), ]
xronsort$y <- c(1:dim(xronsort)[1])
xronsort$yend <- c(1:dim(xronsort)[1])
xroffsort <- xroff[ order(xroff$len), ]
xroffsort$y <- c(1:dim(xroffsort)[1])
xroffsort$yend <- c(1:dim(xroffsort)[1])


m <- mean(dfforplot$len)
sdm <- sd(dfforplot$len)
minlength <- m+(3*sdm)

above3sdofmean <- subset(dfforplot,len>minlength)
xxabove <- subset(above3sdofmean,treatment=='xx')
xrabove <- subset(above3sdofmean,treatment=='xR')
fxabove <- subset(above3sdofmean,treatment=='Fx')
frabove <- subset(above3sdofmean,treatment=='FR')
```

plotting the histograms of all pauses

```r
ggplot(fronsort,aes(x=log((len)))) +
  geom_histogram(binwidth=.2,fill='red')+
  geom_histogram(data=fxonsort,binwidth=.2,fill='black',alpha=0.5)+
  geom_histogram(data=xxonsort,binwidth=.2,fill='black',alpha=0.5)+
  geom_histogram(data=xronsort,binwidth=.2,fill='black',alpha=0.5) +
  theme_tufte() +
  xlim(0,10) +
  geom_vline(xintercept=log(mean(fronsort$len)),color='red')+
  geom_vline(xintercept=log(mean(fxonsort$len)),color='black')+
  geom_vline(xintercept=log(mean(xxonsort$len)),color='black')+
  geom_vline(xintercept=log(mean(xronsort$len)),color='black')
```

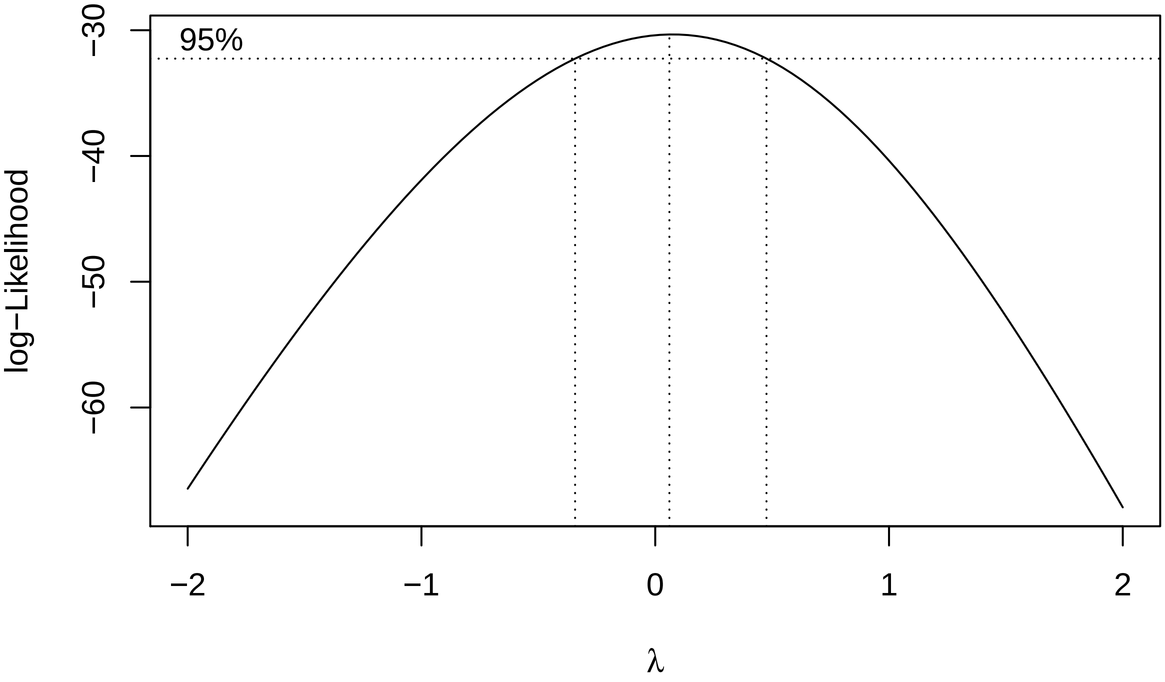


```
levels(dfforplot$lightlist) <- c('OFF','ON')

levels(dfforplot$treatment) <- c('ADL::ReaChR + retinal','ADL::ReaChR - retinal','wild type + retinal',

testcastnow <- dcast(dfforplot,treatment+datelist~lightlist,value.var='len',fun.aggregate=mean,na.rm=TRU

testcastnow$geno <- testcastnow$treatment
testcastnow$ret <- testcastnow$treatment
levels(testcastnow$geno) <- c('reachr','reachr','wildtype','wildtype')
levels(testcastnow$ret) <- c('retinal','noretinal','retinal','noretinal')



testmeltnow <- melt(testcastnow,id.vars=c('treatment','datelist'),variable.name='lightlist',value.name=

testcastnow$diff <- testcastnow$ON-testcastnow$OFF

summarycast <- ddply(testcastnow,.(treatment),summarize,mean=mean(diff),sd=sd(diff),N=length(diff),se=s

testcastnewst <- dcast(testmeltnow,treatment+datelist~lightlist,value.var='len',fun.aggregate=mean,na.r
testcastnewst$x = c(rep('OFF',length(testcastnewst$OFF)))
testcastnewst$xend = c(rep('ON',length(testcastnewst$ON)))
```

```
leveneTest(diff~geno*ret,data=testcastnow)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  3  4.2296 0.01809 *
##       20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
toadd <- 1-min(testcastnow$diff)
testcastnow$tnsl <- testcastnow$diff+toadd
bcr <- boxcox(tnsl~geno*ret,data=testcastnow)
```



```
#round up to nearest 0.5
lambdar <- 0.5
testcastnow$difflambdar <- testcastnow$tnsl^lambdar
leveneTest(difflambdar~geno*ret,data=testcastnow)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  1.8849 0.1647
##       20
```

```
#rounding to the tails of the 95% confidence interval when lambda is near 0 is

reachrtukey <- TukeyHSD(aov(difflambdar~geno*ret,data=testcastnow))
reachrtukey.levels <- reachrtukey$`geno:ret`[,4]
multcompLetters(reachrtukey.levels)['Letters']
```

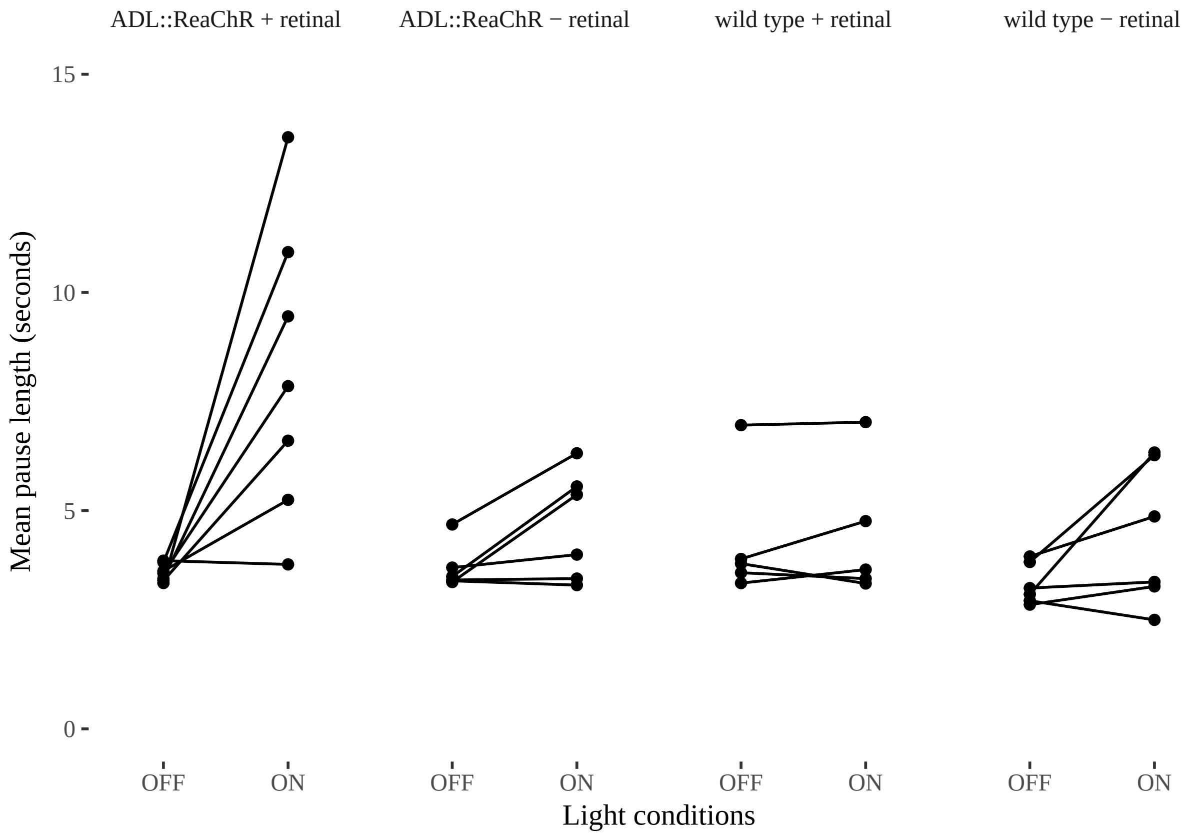
```
## $Letters
##   wildtype:retinal   reachr:noretinal wildtype:noretinal
##                "a"                "a"                "a"
##     reachr:retinal
##                 "b"
```

```
reachrstatstoplot <- ddply(testcastnow,.(geno,ret,treatment),summarize,median=median(difflambdar),N=leng
```

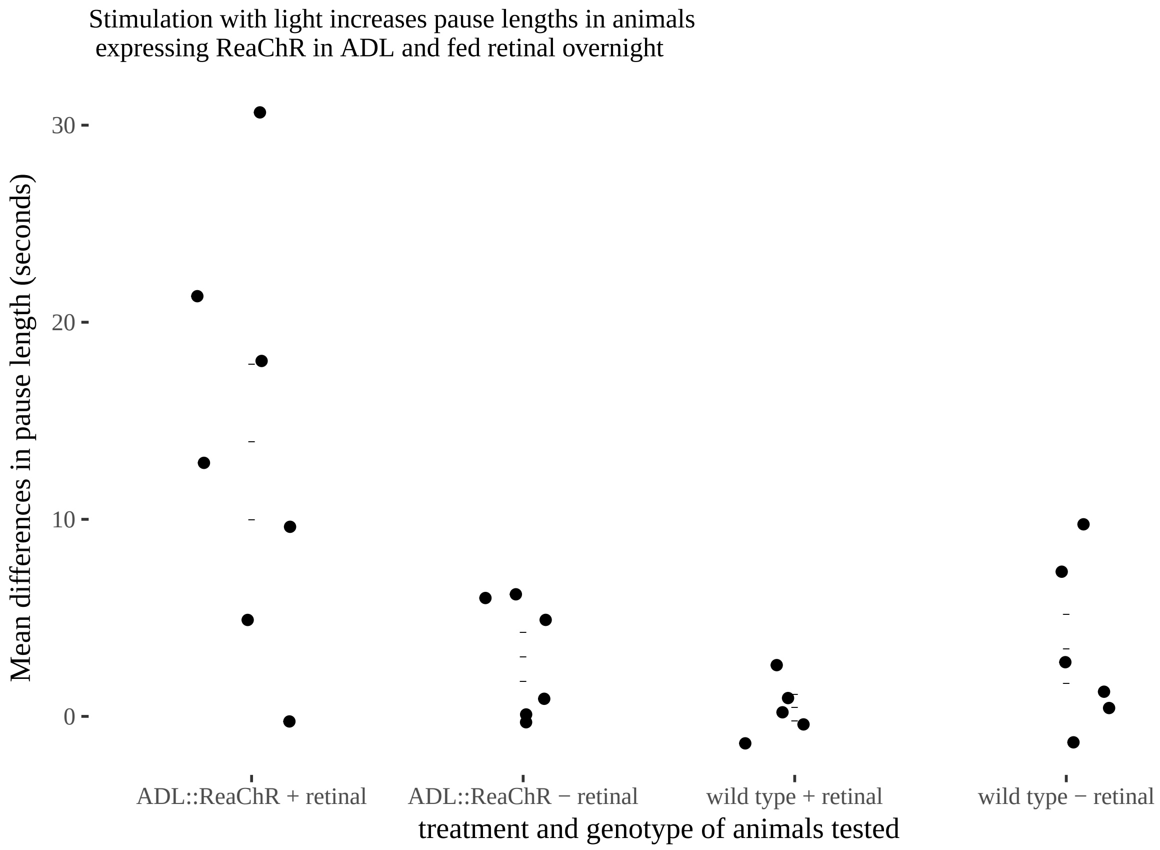
```
levels(dfforplot$lightlist) <- c('OFF','ON')

levels(dfforplot$treatment) <- c('ADL::ReaChR + retinal','ADL::ReaChR - retinal','wild type + retinal',

testcastnow <- dcast(dfforplot,treatment+datelist~lightlist,value.var='len',fun.aggregate=mean,na.rm=TRU

testmeltnow <- melt(testcastnow,id.vars=c('treatment','datelist'),variable.name='lightlist',value.name=

testcastnow$diff <- testcastnow$ON-testcastnow$OFF

summarycast <- ddply(testcastnow,.(treatment),summarize,mean=mean(diff),sd=sd(diff),N=length(diff),se=s

testcastnewst <- dcast(testmeltnow,treatment+datelist~lightlist,value.var='len',fun.aggregate=mean,na.r
testcastnewst$x = c(rep('OFF',length(testcastnewst$OFF)))
testcastnewst$xend = c(rep('ON',length(testcastnewst$ON)))
```

here I plotted the difference in mean length of pauses first, and computed summary statistics on those data, and then also showed the data 'blown apart' into light on and light off data

```
ggplot(testcastnewst,aes(x=x,y=(OFF/3))) +
  geom_segment(aes(x=x,xend=xend,y=(OFF/3),yend=(ON/3))) +
  geom_point(data=testmeltnow,aes(x=lightlist,y=(len/3))) +
  theme_tufte() +
  facet_grid(~treatment)+
  ylab("Mean pause length (seconds)")+
  xlab("Light conditions") +
  ylim(0,15)
```



```
ggplot(testcastnow,aes(x=treatment,y=diff)) +
  geom_jitter(width=0.2,height=0) +
  theme_tufte() +
  geom_point(data=summarycast,aes(x=treatment,y=mean),shape='-') +
  geom_point(data=summarycast,aes(x=treatment,y=mean+se),shape='-') +
  geom_point(data=summarycast,aes(x=treatment,y=mean-se),shape='-') +
  ylab("Mean differences in pause length (seconds)")+
  xlab("treatment and genotype of animals tested") +
  ggtitle("Stimulation with light increases pause lengths in animals \n expressing ReaChR in ADL and fe
  theme(plot.title=element_text(size=10))
```



```
multcompLetters(reachrtukey.levels)['Letters']
```

```
## $Letters
##   wildtype:retinal   reachr:noretinal wildtype:noretinal
##                "a"                "a"                "a"
##     reachr:retinal
##                "b"
```

Box GEP and Cox DR, An Analysis of Transformation.Journal of the Royal Statistical Society. Series B (Methodological), Vol. 26, No. 2. (1964), pp. 211-252.

http://links.jstor.org/sici?sici=0035-9246%281964%2926%3A2%3C211%3AAAOT%3E2.0.CO%3B2-6