imaging_analysis

Emily Jane Dennis

5/26/2017

updated 9/22/2017

Purpose:

To plot all ADL imaging data for manuscript.

Background:

This is an R Markdown document. http://rmarkdown.rstudio.com.

The scripts 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 qqPlots 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

First, load libraries:

```r
library(ggthemes)
library(ggplot2)
library(multcompView)
library(reshape2)
library(plyr)
library(dplyr)
```

Then, load data and add a column called 'frames' from 1:1500 (at 10fps, 150s experiment is 1500 frames)

```r
frames <- c(1:1500)

adlc9f <- read.csv('smoothed_dfoverfmax/adlgcamp_c9_smooth5F.csv',header=FALSE)
adlc9f <- cbind(frames, adlc9f)

adlpt0f <- read.csv('smoothed_dfoverfmax/adlgcamp_solvent_smooth5F.csv',header=FALSE)
adl0f <- cbind(frames, adlpt0f)

adlpt15f <- read.csv('smoothed_dfoverfmax/adlgcamp_pt15DEET_smooth5F.csv',header=FALSE)
adl15f <- cbind(frames, adlpt15f)

mutc9f <- read.csv('smoothed_dfoverfmax/mutantgcamp_c9_smooth5F.csv',header=FALSE)
mutc9f <- cbind(frames, mutc9f)

mutpt0f <- read.csv('smoothed_dfoverfmax/mutantgcamp_solvent_smooth5F.csv',header=FALSE)
mut0f <- cbind(frames, mutpt0f)

mutpt15f <- read.csv('smoothed_dfoverfmax/mutantgcamp_pt15DEET_smooth5F.csv',header=FALSE)
mut15f <- cbind(frames, mutpt15f)
```

```
resc9f <- read.csv('smoothed_dfoverfmax/rescuegcamp_c9_smooth5F.csv',header=FALSE)
resc9f <- cbind(frames, resc9f)

respt0f <- read.csv('smoothed_dfoverfmax/rescuegcamp_solvent_smooth5F.csv',header=FALSE)
res0f <- cbind(frames, respt0f)

respt15f <- read.csv('smoothed_dfoverfmax/rescuegcamp_pt15DEET_smooth5F.csv',header=FALSE)
res15f <- cbind(frames, respt15f)
```

To format each dataframe for analysis I completed the following steps:

1. 'Melted' the data frame: I in the .csv file, each column represents a single animal. Melting these data takes each column and assigns it a variable name (V1, V2…) and then moves each variable into the 'variable' column and each value into the 'value' column. This allows for easier handling later.
2. Subset to keep only frames 300-500 (30-50s, or the first pulse of stimulus)
3. Extract the mean value for each animal during this 30-50s period
4. Add these data to a new data frame and label it with the genotype and treatment.

I repeated the above formatting 1-4 for all 0.15% DEET data:

```
meltadl15f <- melt(adl15f,id.vars = c('frames'))
first <- subset(meltadl15f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('adl',length(x))
z= rep(15,length(x))
dfadl15 <- data.frame(x,y,z)

meltmut15f <- melt(mut15f,id.vars = c('frames'))
first <- subset(meltmut15f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)[1:17]
y= rep('mut',length(x))
z= rep(15,length(x))
dfmut15 <- data.frame(x,y,z)


meltrescue15f <- melt(res15f,id.vars = c('frames'))
first <- subset(meltrescue15f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('rescue',length(x))
z= rep(15,length(x))
dfrescue15 <- data.frame(x,y,z)
```

I repeated the above formatting 1-4 for all 100nM C9 data:

```
meltadlc9f <- melt(adlc9f,id.vars = c('frames'))
first <- subset(meltadlc9f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
```

```
x = (firstmean$mean)
y= rep('adl',length(x))
z= rep(100,length(x))
dfadlc9 <- data.frame(x,y,z)

meltmutc9f <- melt(mutc9f,id.vars = c('frames'))
first <- subset(meltmutc9f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('mut',length(x))
z= rep(100,length(x))
dfmutc9 <- data.frame(x,y,z)


meltrescuec9f <- melt(resc9f,id.vars = c('frames'))
first <- subset(meltrescuec9f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('rescue',length(x))
z= rep(100,length(x))
dfrescuec9 <- data.frame(x,y,z)
```

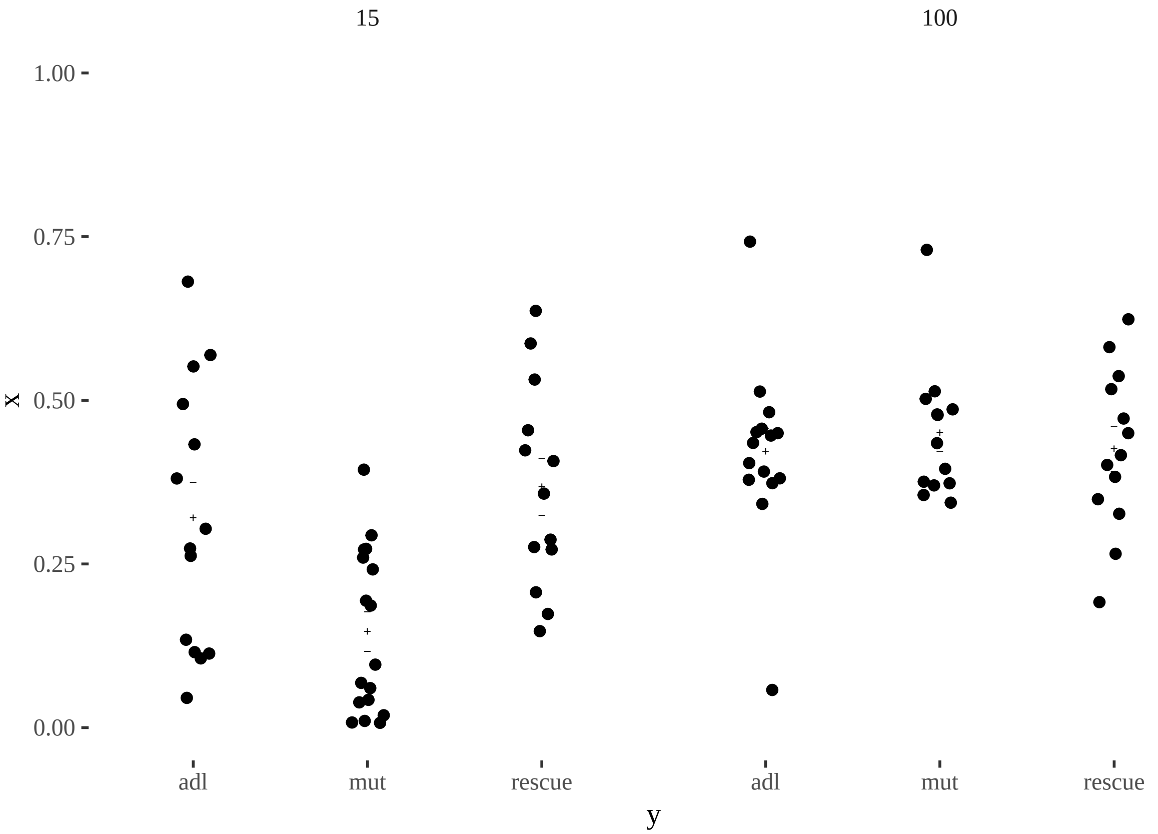
I repeated the above formatting 1-4 for all solvent data: This was a separate control experiment not reported in the paper which looks at ADL responses (or lack thereof) to solvent (ethanol) and/or switching between stimuli.

```
meltadl0 <- melt(adl0f,id.vars = c('frames'))
first <- subset(meltadl0,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('adl',length(x))
z= rep(0,length(x))
dfadl0 <- data.frame(x,y,z)

meltmut0 <- melt(mut0f,id.vars = c('frames'))
first <- subset(meltmut0,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('mut',length(x))
z= rep(0,length(x))
dfmut0 <- data.frame(x,y,z)


meltrescue0f <- melt(res0f,id.vars = c('frames'))
first <- subset(meltrescue0f,frames>300)
first <- subset(first,frames<500)
firstmean <- ddply(first,c('variable'),summarise,mean=mean(value,na.rm=TRUE))
x = (firstmean$mean)
y= rep('rescue',length(x))
z= rep(0,length(x))
```

```
dfrescue0 <- data.frame(x,y,z)
```

I then made a new data frame with all of these summary data together, and also calculated the mean and standard error for plotting.

```
dftoplot <- rbind(dfadl15,dfmut15,dfrescue15,dfadlc9,dfmutc9,dfrescuec9)

#calculate means, standard deviation, and standard error
meanstoplot <- ddply(dftoplot,c('y','z'),summarise,means=mean(x,na.rm = TRUE),N=length(x),sd=sd(x,na.rm=


#plot
ggplot(dftoplot,aes(x=y,y=x)) +
  geom_jitter(height=0,width=0.1) +
  geom_point(data=meanstoplot,aes(x=y,y=means),shape='+') +
  geom_point(data=meanstoplot,aes(x=y,y=means+se),shape='-') +
  geom_point(data=meanstoplot,aes(x=y,y=means-se),shape='-') +
  theme_tufte() +
  ylim(0,1) +
  facet_grid(~z)
```

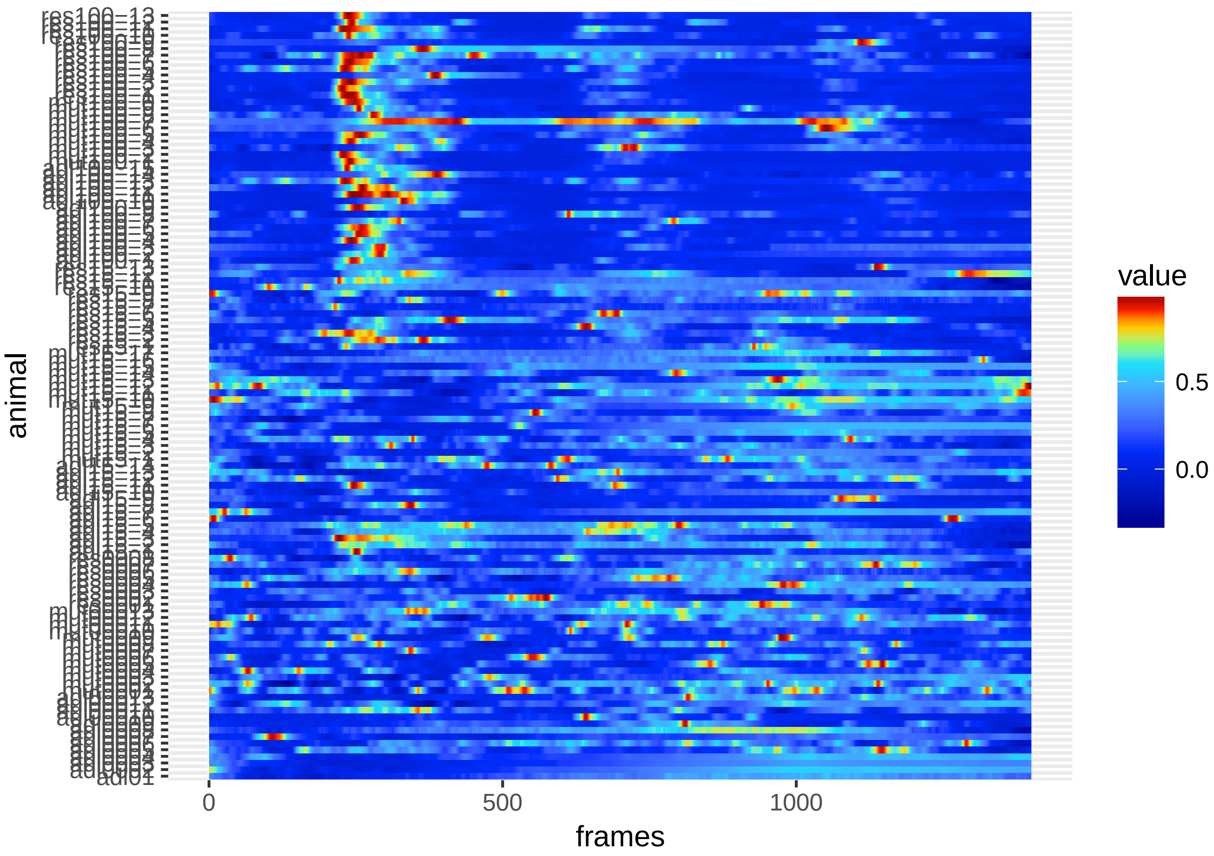


```
#statistics: first need to convert z/y to factors
dftoplot$z <- as.factor(dftoplot$z)
dftoplot$y <- as.factor(dftoplot$y)
imagingtukey <- TukeyHSD(aov(x~y*z,dftoplot))
imagingtukey.levels <- imagingtukey$`y:z`[,4]
multcompLetters(imagingtukey.levels)['Letters']
```

```
## $Letters
##     mut:15  rescue:15    adl:100    mut:100 rescue:100     adl:15
##        "a"        "b"        "b"        "b"        "b"        "b"
```

```
adl15mf <- t(adl15f)
res15mf <- t(res15f)
mut15mf <- t(mut15f)
adl0mf <- t(adl0f)
res0mf <- t(res0f)
mut0mf <- t(mut0f)
adl100mf <- t(adlc9f)
res100mf <- t(resc9f)
mut100mf <- t(mutc9f)


row.names(adl0mf) <- c('adlframes','adl01','adl0002','adl0003','adl0004','adl0005','adl0006','adl0007',
row.names(res0mf) <- c('frames','res0001','res0002','res0003','res0004','res0005','res0006','res0007','
row.names(mut0mf) <- c('mutframes','mut0001','mut0002','mut0003','mut0004','mut0005','mut0006','mut0007

row.names(adl15mf) <- c('adlframes','adl15-1','adl15-2','adl15-3','adl15-4','adl15-5','adl15-6','adl15-
row.names(res15mf) <- c('frames','res15-1','res15-2','res15-3','res15-4','res15-5','res15-6','res15-7',
row.names(mut15mf) <- c('mutframes','mut15-1','mut15-2','mut15-3','mut15-4','mut15-5','mut15-6','mut15-

row.names(adl100mf) <- c('adlframes','adl100-1','adl100-2','adl100-3','adl100-4','adl100-5','adl100-6',
row.names(res100mf) <- c('frames','res100-1','res100-2','res100-3','res100-4','res100-5','res100-6','re
row.names(mut100mf) <- c('mutframes','mut100-1','mut100-2','mut100-3','mut100-4','mut100-5','mut100-6',



#sort by greatest increase in first 10 seconds of pulse vs previous 10 seconds (plotted values)

sortvectorforadlf <- rowMeans(adl15mf[,201:300]) - rowMeans(adl15mf[,301:400])
adl15mf2 <- adl15mf[order(sortvectorforadlf),]

sortvectorformutf <- rowMeans(mut15mf[,201:300]) - rowMeans(mut15mf[,301:400])
mut15mf2 <- mut15mf[order(sortvectorformutf),]

sortvectorforresf <- rowMeans(res15mf[,201:300]) - rowMeans(res15mf[,301:400])
res15mf2 <- res15mf[order(sortvectorforresf),]

allf <- rbind(adl0mf,mut0mf,res0mf,adl15mf,mut15mf,res15mf,adl100mf,mut100mf,res100mf)
subfullf<- allf[,101:1500]

allf <- t(subfullf)
allf2 <- melt(allf)
names(allf2) <- c('frames','animal','value')

matlabrainbow <- c('#00008f','#002fff','#00dfff','#7fff7f','#ffcf00','#ff1f00','#7f0000')
allf3 <- subset(allf2,animal!='mutframes')
allf3 <- subset(allf3,animal!='frames')
allf3 <- subset(allf3,animal!='adlframes')
allf3 <- subset(allf3,animal!='mut15-18')
```

```r
ggplot(allf3, aes(frames, animal)) + geom_tile(aes(fill = value)) + scale_fill_gradientn(values=c(0, .7
```


res100-13
adl01
animal
frames
value