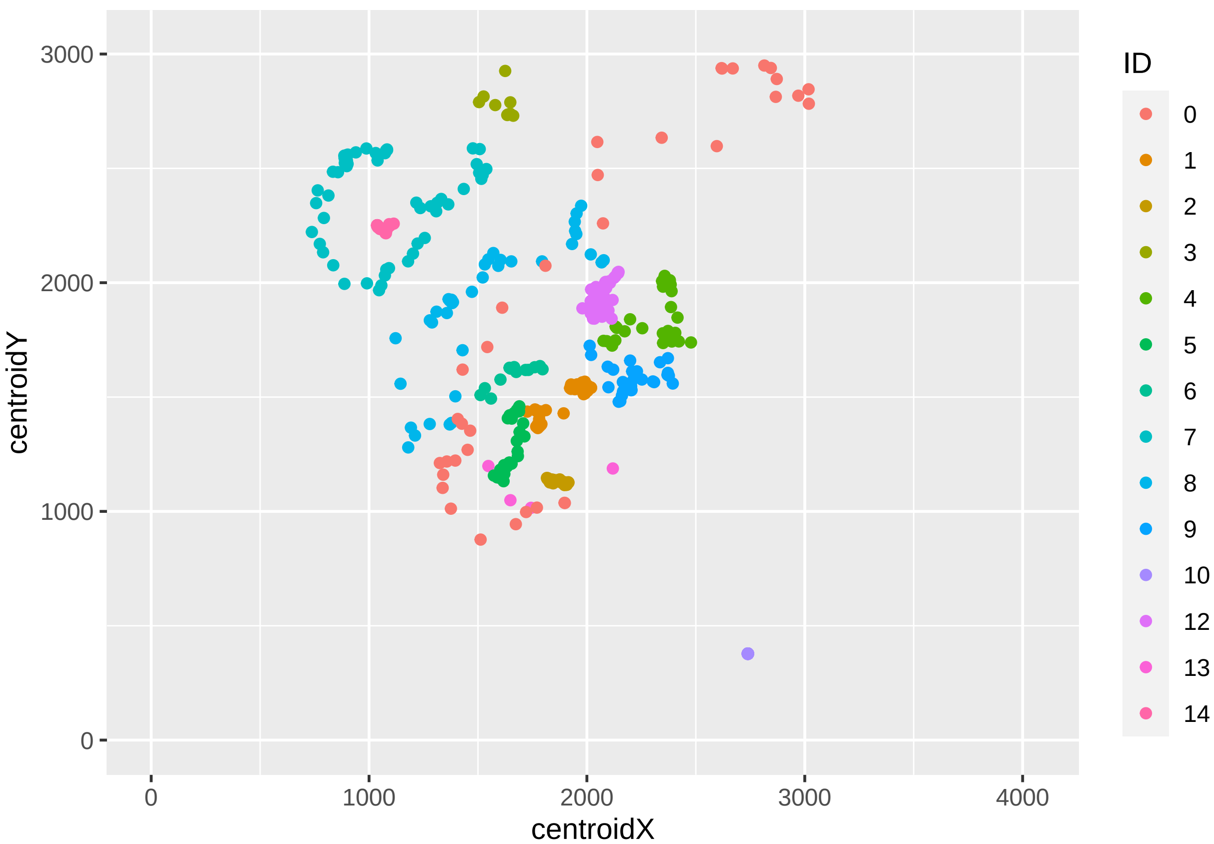
BumbleboxExampleAnalysis

2023-11-07

Load and plot data tracking data from single video

```
#Get list of unique trials (using recursive search in case there are subfolders)
trials <- list.files(pdir, pattern = '*averages.csv', full.names = TRUE, recursive = TRUE)

#Select a random trial - trial 10 in this list
i <- 10

#Load background photo
background_im <- readPNG(str_replace(trials[i], '_averages.csv', '.png'))

#Clip image excess
background_im <- background_im[1:3040,]

#Optional: plot image (this can be slow)
#image(background_im, col = gray.colors(33))


#load and plot tag tracking data
tracking_data <- read.csv(str_replace(trials[i], '_averages.csv', '_updated.csv'))
tracking_data$ID <- as.factor(tracking_data$ID)
ggplot(tracking_data, aes(x = centroidX, y = centroidY, colour = ID)) + geom_point()+xlim(c(0,xl)) + yli
```




```r
#average data
summary_data <- read.csv(trials[i])
summary_data
```

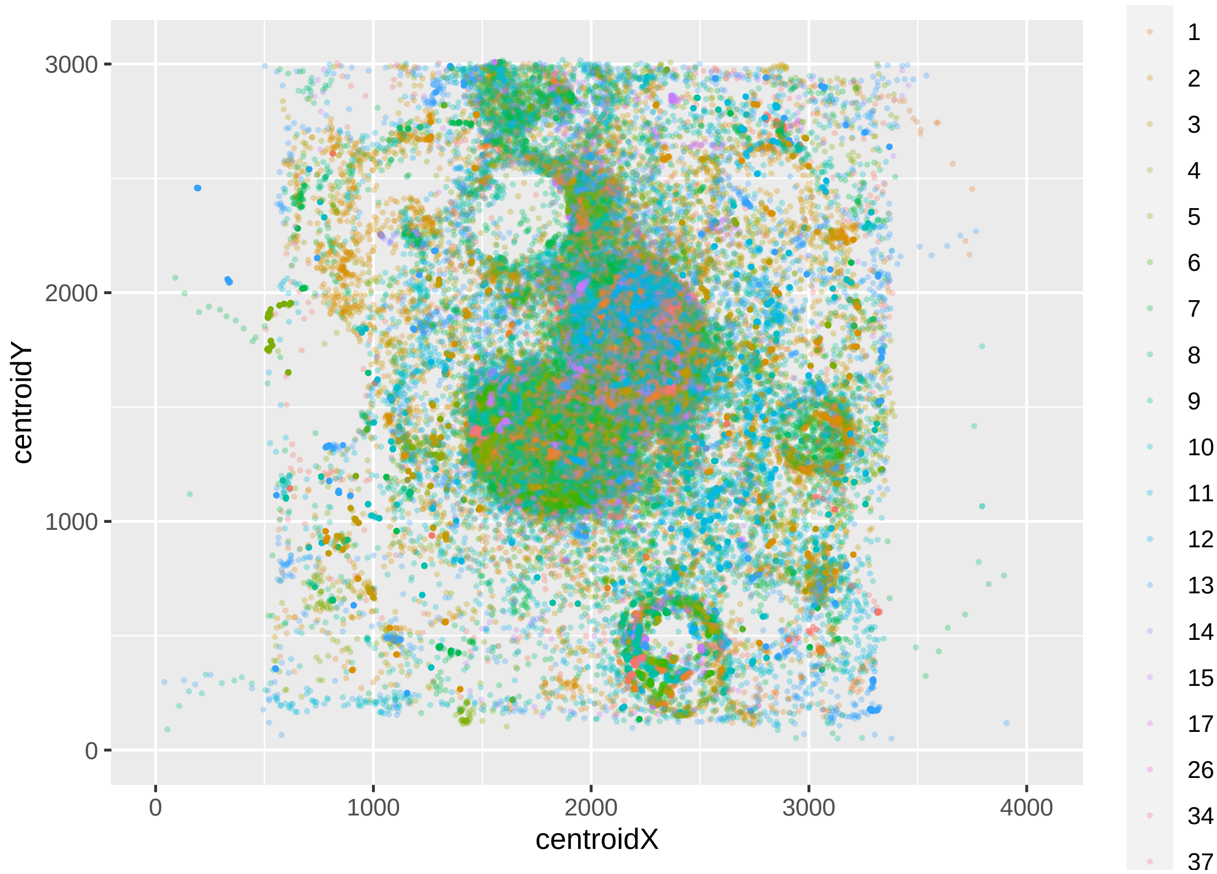
```
##                             filename ID average.distance.from.center
## 1  bumblebox-01_2023-10-28_00-55-03  0                          783
## 2  bumblebox-01_2023-10-28_00-55-03  1                          194
## 3  bumblebox-01_2023-10-28_00-55-03  2                          492
## 4  bumblebox-01_2023-10-28_00-55-03  3                         1060
## 5  bumblebox-01_2023-10-28_00-55-03  4                          497
## 6  bumblebox-01_2023-10-28_00-55-03  5                          433
## 7  bumblebox-01_2023-10-28_00-55-03  6                          177
## 8  bumblebox-01_2023-10-28_00-55-03  7                         1086
## 9  bumblebox-01_2023-10-28_00-55-03  8                          571
## 10 bumblebox-01_2023-10-28_00-55-03  9                          334
## 11 bumblebox-01_2023-10-28_00-55-03 10                         1522
## 12 bumblebox-01_2023-10-28_00-55-03 12                          413
## 13 bumblebox-01_2023-10-28_00-55-03 13                          537
## 14 bumblebox-01_2023-10-28_00-55-03 14                          973
##    average.speed frames.tracked.in.video
## 1            130                      35
## 2             20                      48
## 3              6                      45
## 4             52                       9
## 5             49                      28
## 6             20                      48
```

```
## 7            43                       13
## 8            47                       52
## 9            75                       40
## 10           63                       28
## 11            1                       52
## 12           32                       52
## 13          140                        4
## 14            9                       23
```

Explore data across all trials

First loop and compile across individual tracking files to compile data

```
#Make sure you don't have a 'comb_data' object in memory
rm('tracking_data')

for(file in trials){

  data <- read.csv(str_replace(file, '_averages.csv', '_updated.csv'))
  ## Compile average data
  #Check to see if output data frame exists
  if(!exists('tracking_data')){
    tracking_data <- data #If not, create it from the first loop
  } else {
    tracking_data <- rbind(tracking_data, data) #Otherwise append new data to the end
  }

}

tracking_data$ID <- as.factor(tracking_data$ID)
ggplot(tracking_data, aes(x = centroidX, y = centroidY, colour = ID)) + geom_point(alpha = 0.3, cex = 0
```



Compile data on within-trial average data

```
#Make sure you don't have a 'comb_data' object in memory
rm('comb_data')
```

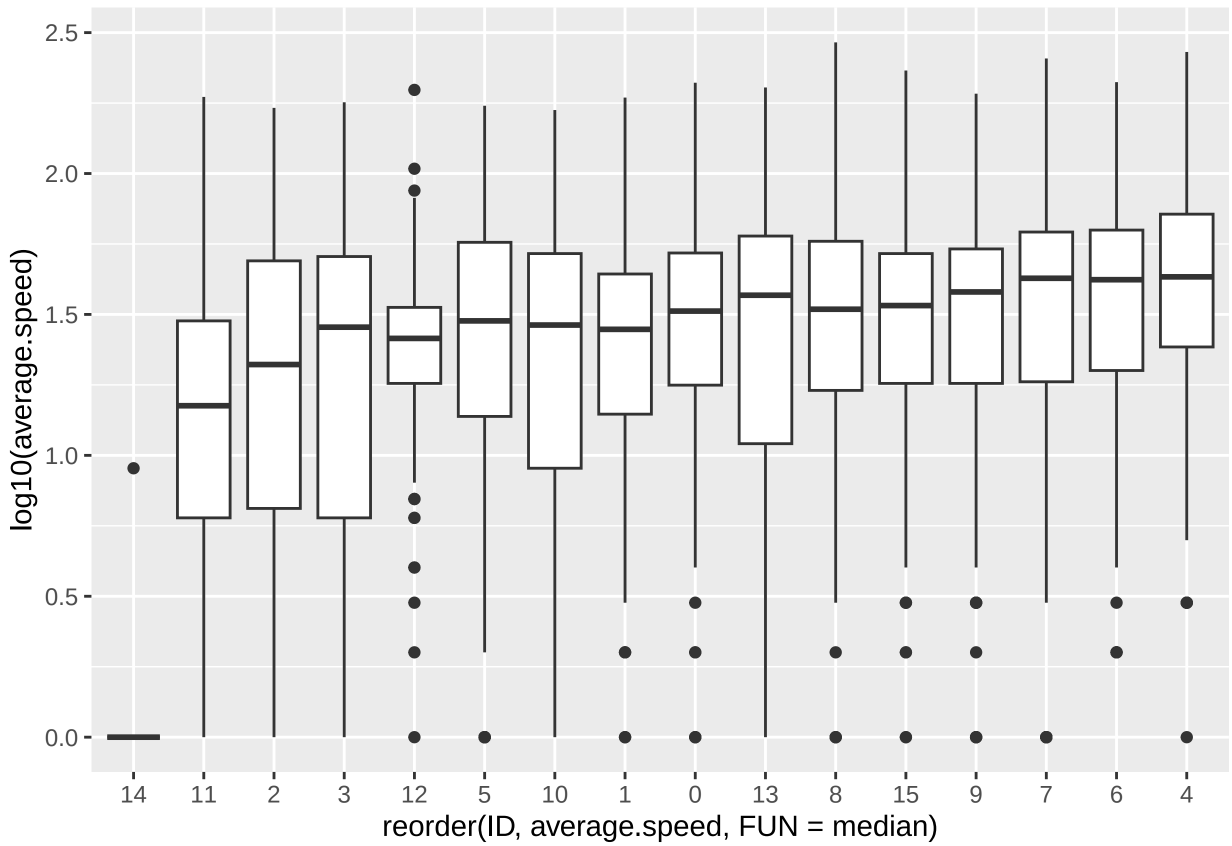
```
## Warning in rm("comb_data"): object 'comb_data' not found
```

```
for(file in trials){

  data <- read.csv(file)
  ## Compile average data
  #Check to see if output data frame exists
  if(!exists('comb_data')){
    comb_data <- data #If not, create it from the first loop
  } else {
    comb_data <- rbind(comb_data, data) #Otherwise append new data to the end
  }

}

comb_data$ID <- as.factor(comb_data$ID)
```

Plot variation in speed and distance from center across individuals

```
#Clean out missing data
comb_data <- comb_data[complete.cases(comb_data),]

#Boxplot of speed vs. individual
ggplot(comb_data, aes(x = reorder(ID, average.speed, FUN = median), y = log10(average.speed)))+geom_box
```

```
## Warning: Removed 103 rows containing non-finite values (`stat_boxplot()`).
```



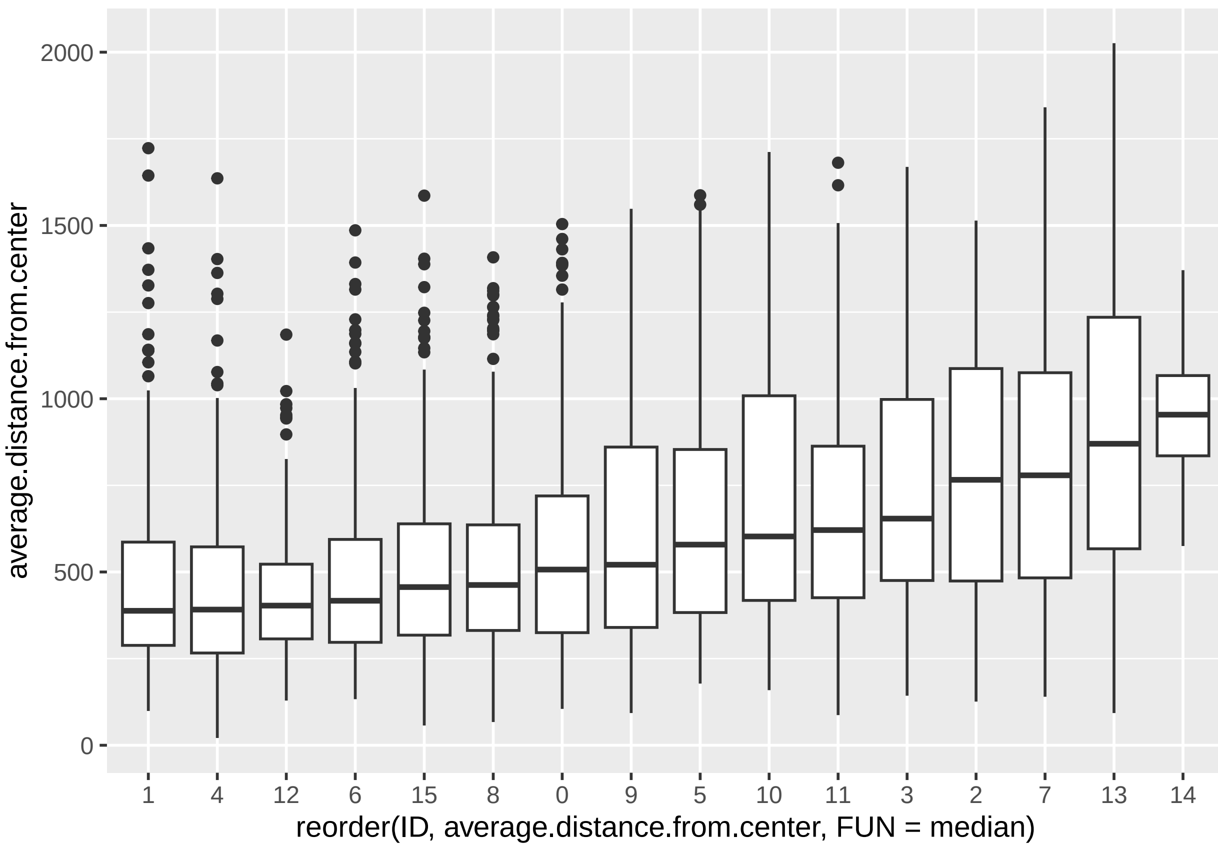

```
#Check for evidence of individual variation
model <- lmer(log10(1+average.speed)~ID + (1|filename), data = comb_data)
model.1 <- lmer(log10(1+average.speed)~1 + (1|filename), data = comb_data)
anova(model, model.1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: comb_data
## Models:
## model.1: log10(1 + average.speed) ~ 1 + (1 | filename)
## model: log10(1 + average.speed) ~ ID + (1 | filename)
##         npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## model.1    3 4660.9 4678.9 -2327.5   4654.9
```

```
## model     18 4199.4 4307.3 -2081.7   4163.4 491.56 15  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

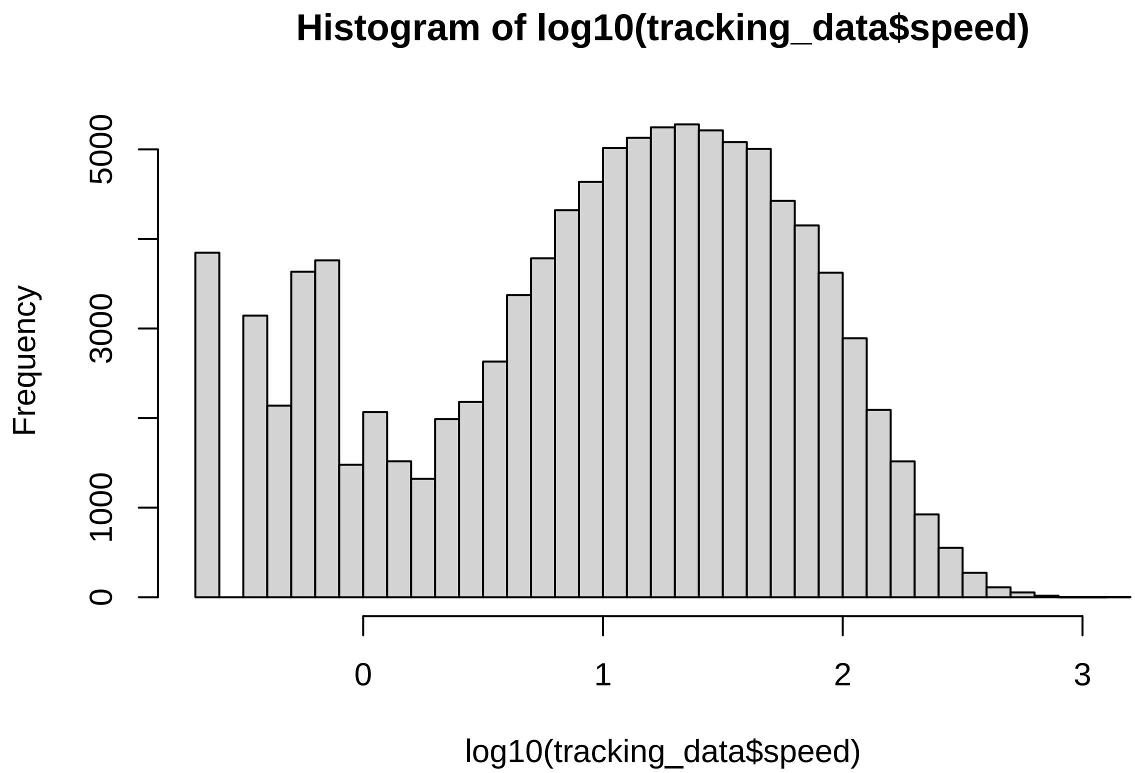
```
#Show distance from center by individual
ggplot(comb_data, aes(x = reorder(ID, average.distance.from.center, FUN = median), y = average.distance
```




```
#Get individual averages
individual_averages <- aggregate(cbind(average.distance.from.center, average.speed, frames.tracked.in.vi
colnames(individual_averages) <- c('ID', 'dist', 'speed', 'frames_tracked')

# Remove outlier with zero velocity
individual_averages <- subset(individual_averages, speed > 1)
```

Determine activity (moving vs. not moving) by thresholding speed

```
#Plot histogram of framewise speeds to determine threshold for 'real' movement
hist(log10(tracking_data$speed), breaks = 30)
```

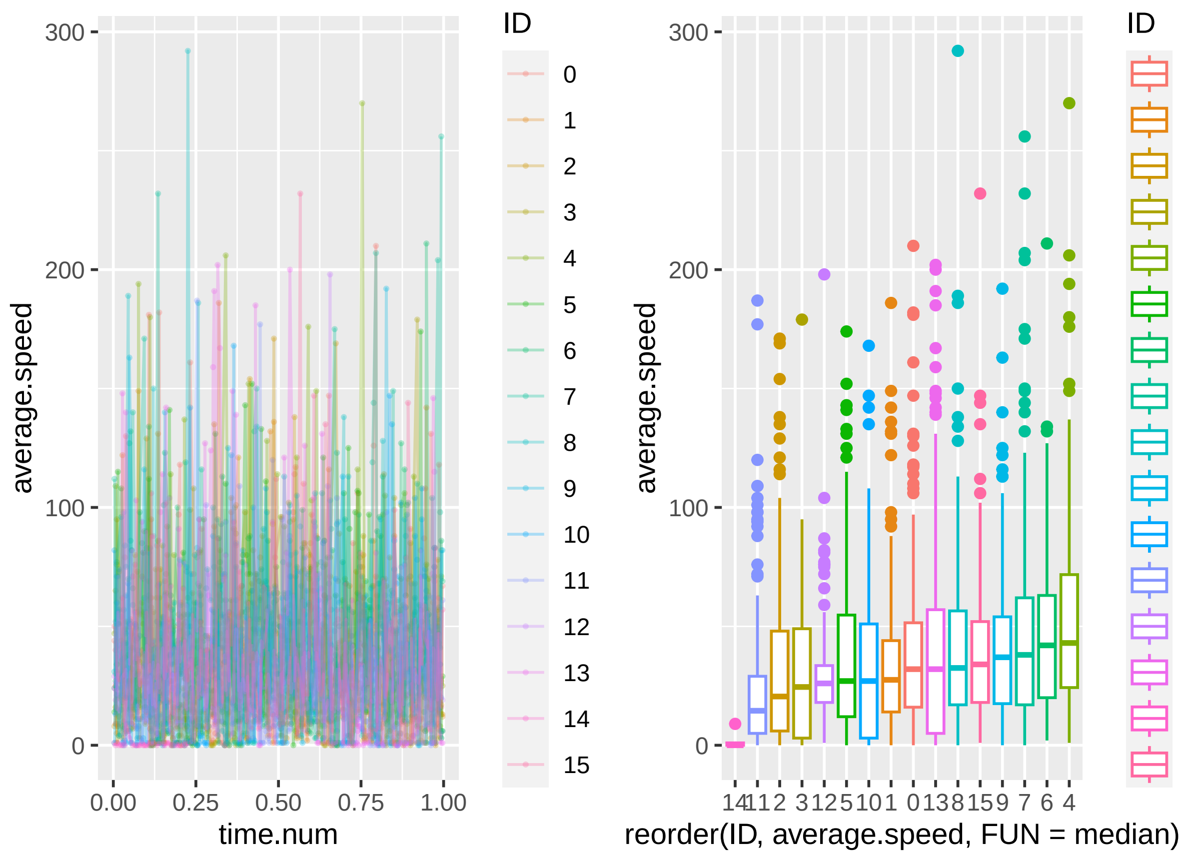



```
#Looks like ~3.1 pixels/frame (log10(3.1)~= 0.5)
speed_thresh <- 3.1
tracking_data$moving[tracking_data$speed > speed_thresh] <- 1
tracking_data$moving[tracking_data$speed < speed_thresh] <- 0
```

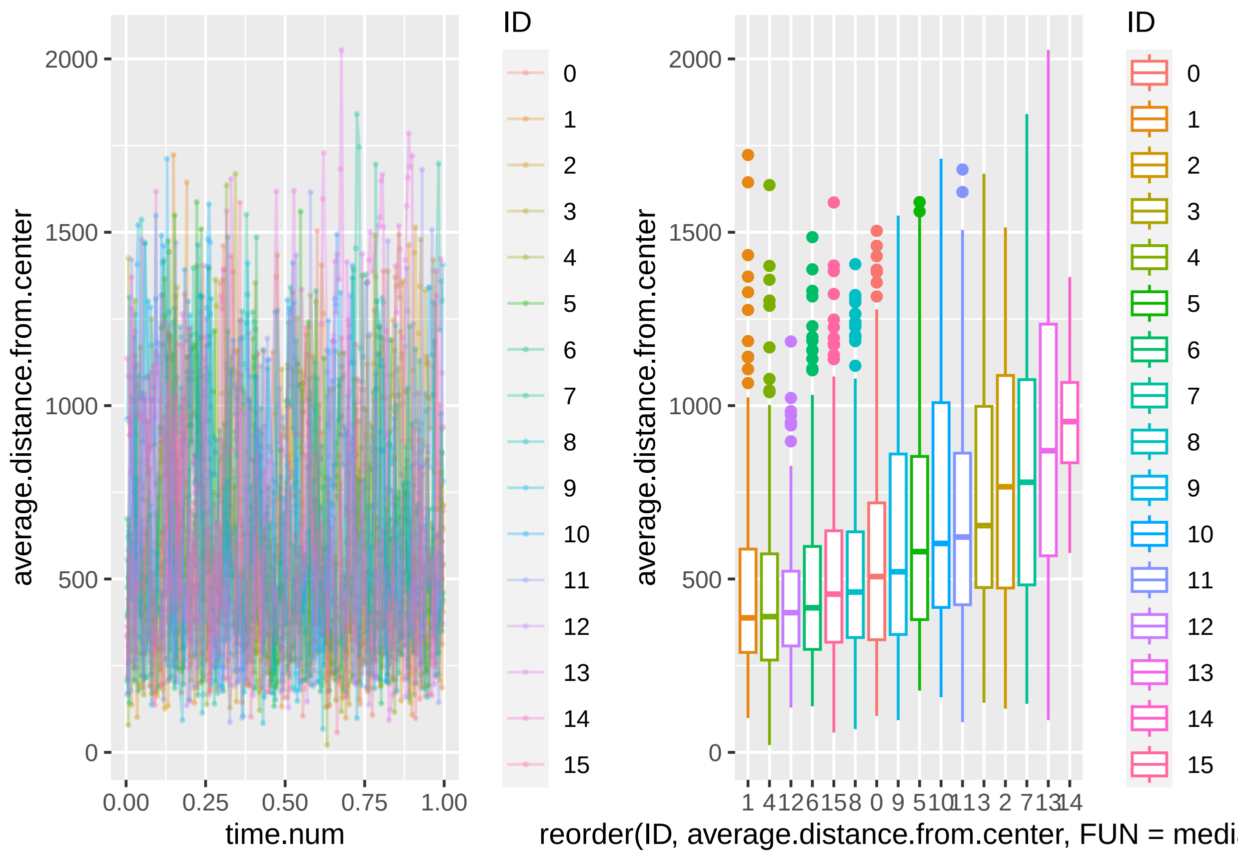
Visualize changes in individual behavior over time

```
#Visualize individual speed/distance over time

#Set up timestamps
#Reference start time
start.date <- parse_date_time('2023-10-28 00:00:00', "%Y-%m-%d %H:%M:%S")

#Create timestamp variable
comb_data$datetime <- parse_date_time(substr(comb_data$filename, 14, 32), "%Y-%m-%d_%H-%M-%S")

#Convert into 'days since the start time'
comb_data$time.num <- as.numeric(difftime(comb_data$datetime, start.date, units = 'days'))

#Reorder
comb_data <- comb_data[order(comb_data$ID, comb_data$time.num),]

#Plots over time and individual variation for speed
```

```
p1 <- ggplot(comb_data, aes(x = time.num, y = average.speed, colour = ID)) + geom_point(alpha = 0.3, ce
p2 <- ggplot(comb_data, aes(x = reorder(ID, average.speed, FUN = median), y = average.speed, colour = I
grid.arrange(p1, p2, ncol=2)
```




```
#Plots over time and individual variation for distance from center
p1 <- ggplot(comb_data, aes(x = time.num, y = average.distance.from.center, colour = ID)) + geom_point(
p2 <- ggplot(comb_data, aes(x = reorder(ID, average.distance.from.center, FUN = median), y = average.di
grid.arrange(p1, p2, ncol=2)
```

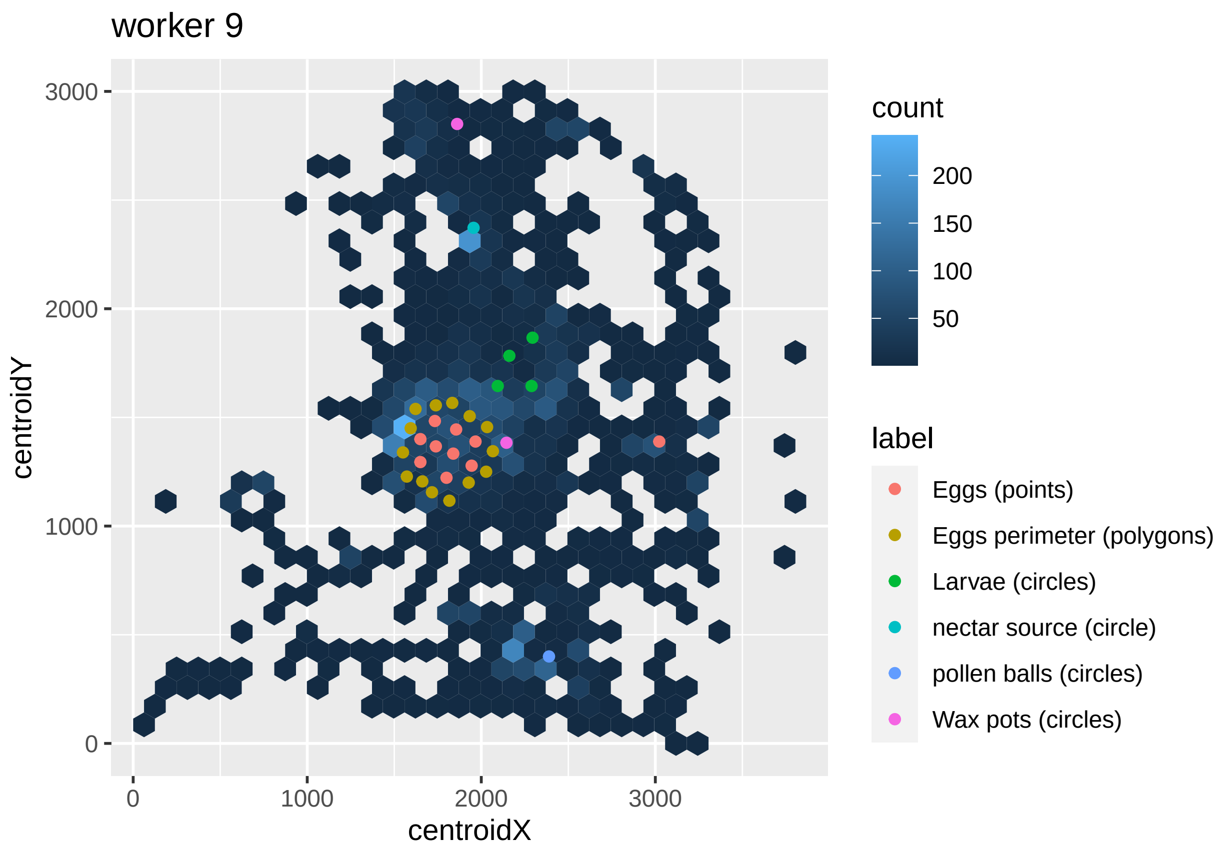



```
#Get average time spent moving for each bee
aggregate(moving~ID, data = tracking_data, FUN = mean)
```

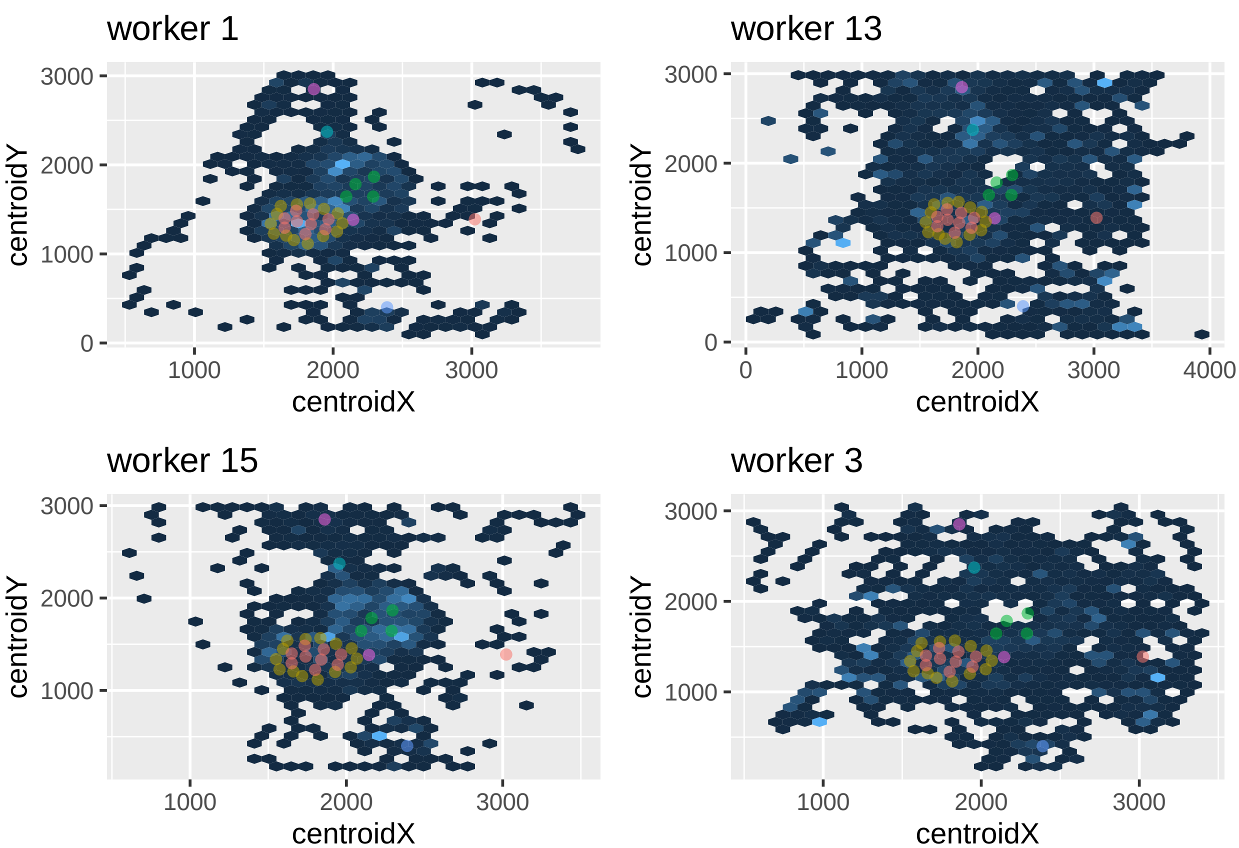
```
##    ID      moving
## 1   0 0.820146223
## 2   1 0.841329966
## 3   2 0.648259994
## 4   3 0.561511140
## 5   4 0.914539920
## 6   5 0.727098956
## 7   6 0.889504950
## 8   7 0.764985260
## 9   8 0.824413818
## 10  9 0.765464770
## 11 10 0.591617345
## 12 11 0.579634465
## 13 12 0.892954723
## 14 13 0.608617095
## 15 14 0.005407654
## 16 15 0.876669635
```

Individual spatial variation

```r
#Plot for single representative individual, with brood locations overlaid
ggplot(subset(tracking_data, ID == 9), aes(x = centroidX, y = centroidY)) + geom_hex()+ggtitle('worker 9
```




```r
#Plot spatial distribution for select individuals (representing centrality range)
p1 <- ggplot(subset(tracking_data, ID == 1), aes(x = centroidX, y = centroidY)) + geom_hex(show.legend =
p2 <- ggplot(subset(tracking_data, ID == 13), aes(x = centroidX, y = centroidY)) + geom_hex(show.legend
p3 <- ggplot(subset(tracking_data, ID == 15), aes(x = centroidX, y = centroidY)) + geom_hex(show.legend
p4 <- ggplot(subset(tracking_data, ID == 3), aes(x = centroidX, y = centroidY)) + geom_hex(show.legend =

grid.arrange(p1,p2,p3,p4, ncol = 2)
```



Contact-based social networks

```
#Load contact data by looping over source files

#Create empty output data structure
out_data <- as.data.frame(matrix(ncol = 17, nrow = 0))
colnames(out_data) <- as.character(c('ID', as.character(seq(0,15))))

#loop over trials
for(file in trials){

  #Get filename specifically for 'contacts.csv' data
  data <- read.csv(str_replace(file, '_averages.csv', '_contacts.csv'))

  #Get rid of 'X' in filenames
  names(data) <- gsub("X", "", names(data))

  #Remove columns that aren't in output data
  data <- data[,colnames(data) %in% colnames(out_data)]

  #Create and populate dummy intermediate data structure to align columns with output
  empty_data <- as.data.frame(matrix(ncol = 17, nrow = length(data[,1])))
  colnames(empty_data) <- colnames(out_data)
  empty_data[,match(colnames(data), colnames(out_data))] <- data
```

```
  #Aggregate data
  out_data <- rbind(out_data, empty_data)

}

#Now, get average interaction rates across all pairwise individuals
interactions <- aggregate(out_data[,2:17], by = list(out_data$ID), FUN = mean, na.rm = TRUE)

#Remove data from errant tags and restructure into adjacency matrix
interactions <- interactions[1:16,2:17]
interactions <- as.matrix(interactions)
interactions[is.na(interactions)] <- 0
interactions[interactions == 1] <- 0

#Make graph
net <- graph.adjacency(interactions, mode = 'undirected', weighted=TRUE, diag=FALSE)
plot.igraph(net,layout=layout.fruchterman.reingold, edge.width=E(net)$weight*50, vertex.size = 20)
```