

Associative learning in Nematostella vectensis

Analysis R code for the DeepLabCut tracking data

Gaëlle Botton-Amiot

2023-01-19

For each video, the tracking output data from DLC consists in a set of coordinates (x,y) and a confidence score ("likelihood", continuous from 0 to 1) for every timepoint (i.e., for every video frame), for every individual. All videos from an experiment were processed using a custom-made function for batch-analysis.

Initialization

```
# set the working directory

WorkingDirectory<- "____"
setwd(WorkingDirectory)

# Metadata 1 - set video files parameters -----

# set the number of videos to analyze and the date of the experiment
nbVids<-8
paths<-0 # Leave 0
date<- '220427'

# set the names of individuals - keep consistent with the output DLC file
Anims<-c('individual1', 'individual2', 'individual3', 'individual4', 'individual5', 'individual6', 'individual7', 'individual8', 'individual9', 'individual10')

# set the condition and group corresponding to each video file
AnimsList<-replicate(nbVids, list(Anims))
names<-0
conditions<-c('pretest', 'pretest', 'pretest', 'pretest',
               'unpaired', 'paired', 'unpaired', 'paired')
group<-c('up1', 'p1', 'up2', 'p2')
groups<-rep(group, 2)
```

```
# set the path to access each video file - they were all determined to finish
by the same suffixe, except the video number at the beginning

for (i in 1:nbVids) {
  paths[i]<- paste(date, "_csv files/",i,"_",date, "x6croppedDLC_dlcnetms5_3
pts_for220427Apr28shuffle1_45000_el.csv", sep="")
  names[i]<-paste("Vid", i, sep="")
}

#in case of non-continuous video names use the following commands:
# idVids<-c("1", "2", "3", "5", "6", "7", "8")
# for (i in 1:nbVids) {
#   paths[i]<-paste(date, "videoFolder/", idVids[i],"_",date, "suffixe_DL
C_output_File.csv.csv", sep="")
#   names[i]<-paste("Vid", idVids[i], sep="")
# }

# generate the first part of the metadata file with the video files parameter
s
metadata1<-cbind(paths, names, conditions, groups, AnimsList)
metadata1<- as.data.frame(metadata1)

# If necessary, it's possible to modify the individuals composition for a vid
eo (ex. in case of a bad tracking, if one animal needs to be excluded from th
e analysis)
# For example, we can exclude individual 7 in the video 10 with the command b
elow:
#metadata1$AnimsList[[10]]<-c('individual1', 'individual2', 'individual3','in
dividual4', 'individual5', 'individual6', 'individual8', 'individual9', 'individ
ual10')

# Metadata 2 - set testing parameters -----
# initial frame (0 in most cases)
MinF=0
MinFlist<-replicate(nbVids, MinF)
#duration of the video in seconds
MaxF<-240
MaxFlist<-replicate(nbVids, MaxF)
MaxPlot.y<-250
#window to determine the initial body length (BL) in seconds
InitWindow<-10
InitWindowlist<-replicate(nbVids, InitWindow)
# specify the seconds intervals of the CS application during testing
CSaplBegin=120
CSaplEnd=180
CSaplBlist<-replicate(nbVids, CSaplBegin)
CSaplElist<-replicate(nbVids, CSaplEnd)
```

```
# specify the seconds intervals around CS application for the initial and final CS application BL values
rangeDifCS=1
rangeDifCSlist<-replicate(nbVids, rangeDifCS)
# specify the window before and after CS application for reference values Pre /Post CS application
rangeDifPP=30
rangeDifPPlist<-replicate(nbVids, rangeDifPP)
# specify the CUTOFF in BL percentage reduction for retraction
cutoffBL=-10
cutoffBLlist<-replicate(nbVids, cutoffBL)
# specify the CUTOFF in BL percentage reduction before CS - animals already retracting before CS --> to discard!
cutoffBLpre=-10
cutoffBLprelist<-replicate(nbVids, cutoffBLpre)
# likelihood cutoff in DLC tracking data
likelihood=0.9

metadata2<-data.frame('Begin video'=MinFlist, 'Duration video'=MaxFlist, "Initial BL window (sec)"=InitWindowlist, "Occurrence CS (sec)"= CSap1Blist, "End CS (sec)"= CSap1Elist, "Interval around CSap1 (sec)"=rangeDifCSlist, "Pre/post CS window"= rangeDifPPlist, "CutOff BL retraction"=cutoffBLlist, "CutOff BL retraction preCS"=cutoffBLprelist)

# Merge and save metadata files together ----
metadata<-cbind(metadata1, metadata2)

metadata3<-metadata
metadata3$AnimsList<-as.character(metadata$AnimsList)

library(dplyr)

metadata3 %>%
  rowwise() %>%
  mutate_if(is.list, ~paste(unlist(.), collapse = '|')) %>%
  write.csv(paste(getwd(), '/', date, '_Categorical', cutoffBL, 'percent_1min CS_preCS10perc_metadata.csv', sep=""), row.names = TRUE)
```

Function to calculate body length variation for each animal in each video

The function performed the following steps and computed the following parameters:

- filter out missing or low-confidence results for each point (likelihood<0.9).
- calculate time variables (seconds during the recording and in the accelerated video) depending on the recording frame-rate and playback speed of the video.
- bin the position of each point per second: calculate the mean position for the 5 frames tracked per second (recording at 5fps).
- plot the coordinates of all animals, to allow for identity checking of the animals in the corresponding video if necessary.
- calculate the absolute body length (BL_{abs}) in pixels of each animal at any time point: sum of the Euclidian distances between the mouth and the midpoint and between the midpoint and the foot.
- calculate the body length in variation of % of initial body length ($\%BL_{variation}$) for each animal. Since the length of anemones can considerably vary among individuals and at different time points during the day for a same individual, we used this value to normalize the observed changes during this 4-minute recording to each animal's baseline. The initial body length (BL_{init}) was defined as the mean body length in the first 10 seconds of the recording and $\%BL_{variation}$ was calculated as follow:

$$\%BL_{variation} = \frac{BL_{abs} - BL_{init}}{BL_{init}} \times 100$$

- plot the $\%BL_{variation}$ of each animal for each video in order to check for any aberrant values.
- calculate the following values:
 - BL_{initCS} : $\%BL_{variation}$ value at the beginning of CS application.
 - Max_{preCS} : maximum value of $\%BL_{variation}$ in a 30 seconds window before CS application.
 - Δ_{preCS} : a pre-CS application maximum variation:

$$\Delta_{preCS} = BL_{initCS} - Max_{preCS}$$

If $\Delta_{preCS} \leq -10\%$ the animal was discarded and not used for subsequent analysis.

- Δ_{CS} : a maximum retraction value during the course of the CS application. The minimum value for $\%BL_{variation}$ during the 1 minute of CS application (Min_{CS}) was determined and Δ_{CS} was calculated as follows:

$$\Delta_{CS} = Min_{CS} - \%BL_{initCS}$$

A cutoff of -10% was chosen and if $\Delta_{CS} \leq -10\%$ the animal was attributed the factor "retract". Otherwise, the factor "immobile" was given.

```
MeanBLdifVid_Categorical<- function(DLC_data){  
  
  # Prepare the raw data -----  
  
  # remove the "scorer" row  
  DLC_data<-DLC_data[-c(1),]  
  # re-organize the rows  
  DLC_data<- DLC_data[c(3,1,2,4:nrow(DLC_data)), ]  
  DLC_data <- as.data.frame(t(DLC_data))  
  
  names(DLC_data)      # Look at only the names of the variables  
  colnames(DLC_data, do.NULL = TRUE)  
  colnames(DLC_data) <- DLC_data[1,]  
  DLC_data<-DLC_data[2:nrow(DLC_data),]  
  
  str(DLC_data)        # check the structure  
  
  #reshape in long format  
  
  library(tidyr)  
  DLC_data <- gather(DLC_data,  
                     key = "frame",  
                     value = "value.px",  
                     4:ncol(DLC_data))  
  
  str(DLC_data)  
  DLC_data<- spread(DLC_data, key="coords", value="value.px")  
  str(DLC_data)  
  
  # reorganize the columns  
  DLC_data<-DLC_data[c(3,1,2,5, 6, 4)]  
  str(DLC_data)  
  
  #convert the values into numbers and factors  
  DLC_data$x<- as.numeric(DLC_data$x)  
  DLC_data$y<- as.numeric(DLC_data$y)  
  DLC_data$likelihood<- as.numeric(DLC_data$likelihood)  
  DLC_data$frame<- as.integer(DLC_data$frame)  
  DLC_data$individuals<- as.factor(DLC_data$individuals)  
  DLC_data$bodyparts<- as.factor(DLC_data$bodyparts)  
  
  # organize per frame number  
  library(dplyr)  
  DLC_data<- DLC_data %>% arrange(DLC_data$frame)
```

```
#Exclude individuals and/or frames given the metadata parameters

DLC_data<-DLC_data %>%
  dplyr::filter(individuals %in% Anims, frame>=MinF & frame<=MaxF*5)
str(DLC_data)

nf_DLC_data=nlevels(as.factor(DLC_data$frame))
print(nf_DLC_data)

#
# Bring the arena in a square with a 0,0 origin
minX<-min(DLC_data$x, na.rm=TRUE)
minY<-min(DLC_data$y, na.rm=TRUE)

DLC_data$x<-(DLC_data$x)-minX
DLC_data$y<-(DLC_data$y)-minY

dimension<- c(max(DLC_data$x, na.rm=TRUE), max(DLC_data$y, na.rm=TRUE))

#
# selection of the rows only with a sufficient likelihood (determined in me
tadata)

library(dplyr)
highlik<-DLC_data %>%
  dplyr::filter(likelihood >= likelihood)

drops<-"likelihood"
highlik <- highlik[, ! names(highlik) %in% drops, drop = F]

#Time in secondes and minutes
# --> acquisition at 5fps
sec<-(highlik$frame/5)
secVid<-as.numeric(sec/6)

minutes<-(highlik$frame/(5*60))
minVid<- (minutes/6)

sec.int<-as.integer(sec)
sec.int<-as.numeric(sec.int)

highlik<- cbind(highlik, sec, sec.int, minutes, secVid, minVid)
```

```
#remove duplicated columns (keep the last one)
highlik <- highlik[, !duplicated(colnames(highlik), fromLast = TRUE)]

#nb of rows highlik
nf_highlik=nrow(highlik)

#DATAFRAME WITH MEAN POSITION PER SECOND - bin the position of each point
per second

meanSecX<-aggregate(highlik$x, by=list(Seconds=highlik$sec.int, highlik$ind
ividuals, highlik$bodyparts), FUN=mean)
meanSecY<-aggregate(highlik$y, by=list(Seconds=highlik$sec.int, highlik$ind
ividuals, highlik$bodyparts), FUN=mean)

meanMin<-as.numeric(meanSecX$Seconds)/60
meanSecvid<-as.numeric(meanSecX$Seconds)/6

meanSec<- cbind(meanSecX, meanSecY$x, meanMin, meanSecvid)

rm('meanMin')
rm('meanSecvid')

colnames(meanSec)<-c("Seconds", "individuals", "bodyparts", "x", "y", "minu
tes", "secVid")
colnames(meanSec)

max_meanSec=nlevels(as.factor(meanSec$Seconds))

#PLOT COORDINATES -----

#plot coordinates of each individuals and save the file in the working dire
ctory

library(ggplot2)
Subt<-paste(name, condition, date, sep="-")

nameplot1<-paste(date, 'coords','_', name, '_', condition, '.png', sep="")
png(file=nameplot1, width=300, height=250, unit='mm', res=100)
cords<-ggplot(highlik) + geom_point(data=highlik, aes(x = x, y = y, color=i
ndividuals), size = 1)+
  theme(legend.position = "right")+
  labs(y="Y coordinates (A.U.)", x="X coordinates (A.U.)", color="Individua
ls")+
  labs(subtitle = paste(Subt), caption = Sys.time())
print (cords)
dev.off()
```

```
# BODYLENGTH CALCULATION -----

#create dataframe with only rows for each second and animal and remove duplicated rows
DF_bodylength<-meanSec[, -c(3,4,5)]

DF_bodylength<- DF_bodylength %>%
  arrange(individuals, Seconds)
DF_bodylength<- DF_bodylength %>%
  distinct()

bodyparts<-c('head', 'foot', 'mid')

head<-subset(meanSec, bodyparts=="head")
head<- head %>%
  arrange(individuals, Seconds)

foot<-subset(meanSec, bodyparts=="foot")
foot<- foot %>%
  arrange(individuals, Seconds)

mid<-subset(meanSec, bodyparts=="mid")
mid<- mid %>%
  arrange(individuals, Seconds)

lenCord<-nrow(DF_bodylength)

DF_bodylength <- DF_bodylength %>% left_join(head, by=c("Seconds","individuals", 'minutes', 'secVid'))
DF_bodylength <- DF_bodylength %>% left_join(foot, by=c("Seconds","individuals", 'minutes', 'secVid'))
DF_bodylength <- DF_bodylength %>% left_join(mid, by=c("Seconds","individuals", 'minutes', 'secVid'))

colnames(DF_bodylength)<-c('Seconds', 'individuals', 'minutes', 'secVid', 'a', 'head.x', 'head.y', 'b', 'foot.x', 'foot.y', 'c', 'mid.x', 'mid.y')

DF_bodylength<-DF_bodylength[, -c(5, 8, 11, 14, 17)]

# body segments length calculation

headmid<-rep(1, lenCord)
midfoot<-rep(1, lenCord)

sumBL<-rep(1, lenCord)

DF_bodylength<-cbind(DF_bodylength, headmid, midfoot, sumBL)
```



```
for (i in 1: lenCord ) {
  DF_bodylength$headmid[i]=sqrt((DF_bodylength$head.x[i]-DF_bodylength$mid.
x[i])^2+(DF_bodylength$head.y[i]-DF_bodylength$mid.y[i])^2)
  DF_bodylength$midfoot[i]=sqrt((DF_bodylength$mid.x[i]-DF_bodylength$foot.
x[i])^2+(DF_bodylength$mid.y[i]-DF_bodylength$foot.y[i])^2)
  DF_bodylength$sumBL[i]=sum(DF_bodylength$headmid[i],DF_bodylength$midfoot
[i])
}

# BL variation in percentage of initial BL

nAnim=nlevels(as.factor(Anims))
print(nAnim)

initBL<-0
indivIndx<-0
sumBL_percent<-0

for (i in 1:nAnim) {
  anim<-subset(DF_bodylength, individuals==Anims[i])
  initBL[i]<- mean(anim$sumBL[0:InitWindow],na.rm=T)
}

for (i in 1: lenCord ) {
  indivIndx<-match(DF_bodylength$individuals[i], Anims)
  sumBL_percent[i]<-((DF_bodylength$sumBL[i]-initBL[indivIndx])/initBL[indi
vIndx]*100)
}

DF_bodylength<-cbind(DF_bodylength, sumBL_percent)
```

```
# Plot bodylength -----

MaxPlotPercent.y<- max(DF_bodylength$sumBL_percent)
MinPlotPercent.y<- min(DF_bodylength$sumBL_percent)

# plot Bodylength over time (seconds) of all animals
library(ggplot2)

nameplot2<-paste(date, 'BL','_', name, '_', condition, '.png', sep="")
png(file=nameplot2, width=300, height=225, unit='mm', res=100)
BLplot<-ggplot(DF_bodylength, aes(x=Seconds, y=sumBL, color=individuals)) +
  geom_line(size=1, alpha=0.9, linetype=1) +
  geom_vline(xintercept=(CSapl[1]), color='deeppink', linetype='longdash')+
  geom_vline(xintercept=(CSapl[2]), color='deeppink', linetype='longdash')+
  annotate("rect", xmin = (CSapl[1]-rangeDifPP) , xmax = (CSapl[1]), ymin =
0, ymax = MaxPlot.y, alpha = .2)+
  annotate("rect", xmin = (CSapl[2]) , xmax = (CSapl[2]+rangeDifPP), ymin =
0, ymax = MaxPlot.y, alpha = .2)+
  annotate("rect", xmin = (CSapl[1]-rangeDifCS) , xmax = (CSapl[1]+rangeDif
CS), ymin = 0, ymax = MaxPlot.y, alpha = .2, fill='deeppink')+
  annotate("rect", xmin = (CSapl[2]-rangeDifCS) , xmax = (CSapl[2]+rangeDif
CS), ymin = 0, ymax = MaxPlot.y, alpha = .2, fill='deeppink')+
  ggtitle("Bodylength over time") +
  xlab("Seconds") +
  ylab("Bodylength (A.U.)")+
  xlim(0,MaxF)+
  ylim(0,MaxPlot.y)+
  theme_minimal()+
  labs(subtitle = paste(Subt), caption = Sys.time())
print(BLplot)
dev.off()
```

```
# plot Bodylength percentage variation over time (seconds) of individual animals in separated panels
```

```
nameplot3<-paste(date, 'BLperc', '_', name, '_', condition, '.png', sep='')  
png(file=nameplot3, width=400, height=175, unit='mm', res=100)
```

```
BLplotIndiv<-ggplot(DF_bodylength, aes(x=Seconds, y=sumBL_percent, color=individuals)) +  
  geom_line(size=1, alpha=0.9, linetype=1) +  
  facet_wrap(facets = vars(individuals), ncol = 5)+  
  geom_vline(xintercept=(CSapl[1]), color='deeppink', linetype='longdash')+  
  geom_vline(xintercept=(CSapl[2]), color='deeppink', linetype='longdash')+  
  annotate("rect", xmin = (CSapl[1]-rangeDifPP), xmax = (CSapl[1]), ymin =  
MinPlotPercent.y, ymax = MaxPlotPercent.y, alpha = .2)+  
  annotate("rect", xmin = (CSapl[2]), xmax = (CSapl[2]+rangeDifPP), ymin =  
MinPlotPercent.y, ymax =MaxPlotPercent.y, alpha = .2)+  
  annotate("rect", xmin = (CSapl[1]-rangeDifCS), xmax = (CSapl[1]+rangeDif  
CS), ymin = MinPlotPercent.y, ymax = MaxPlotPercent.y, alpha = .2, fill='deep  
pink')+  
  annotate("rect", xmin = (CSapl[2]-rangeDifCS), xmax = (CSapl[2]+rangeDif  
CS), ymin = MinPlotPercent.y, ymax = MaxPlotPercent.y, alpha = .2, fill='deep  
pink')+  
  ggtitle("Initial Bodylength percentage variation over time") +  
  xlab("Seconds") +  
  ylab("Bodylength percentage")+  
  xlim(0,MaxF)+  
  ylim(MinPlotPercent.y, MaxPlotPercent.y)+  
  theme_bw()+  
  theme(legend.position="none")+  
  labs(subtitle = paste(Subt), caption = Sys.time())  
print(BLplotIndiv)  
dev.off()
```

```
# BODYLENGTH VARIATION DURING CS APPLICATION-----
```

```
# Initialization
```

```
MeanCSinit<-0  
MaxPre<-0  
MinCS<-0  
DeltaCS<-0  
DeltaPreCS<-0  
mvt<-0  
Cname<-c(rep(name,length(Anims)))  
Cname<-data.frame(Cname, stringsAsFactors = TRUE)  
Ccondition<-c(rep(condition,length(Anims)))  
Ccondition<-data.frame(Ccondition, stringsAsFactors = TRUE)  
Cgroup<-c(rep(group,length(Anims)))  
Cgroup<-data.frame(Cgroup, stringsAsFactors = TRUE)
```

```
# Calculation of DeltapreCS and DeltaCS

for (i in 1:nAnim) {
  anim<-subset(DF_bodylength, individuals==Anims[i])
  MeanCSinit<-mean(anim$sumBL_percent[(CSapl[1]-rangeDifCS):(CSapl[1]+range
DifCS)])
  MaxPre<-max(anim$sumBL_percent[(CSapl[1]-rangeDifPP):(CSapl[1]]), na.rm=T
)
  MinCS<-min(anim$sumBL_percent[CSapl[1]:CSapl[2]], na.rm=T)
  DeltaCS[i]<-MinCS-MeanCSinit
  DeltaPreCS[i]<-MeanCSinit-MaxPre

  if (DeltaPreCS[i]<= cutoffBLpre |is.na(DeltaPreCS[i])|is.na(DeltaCS[i])){
    mvt[i]<-NA
  }
  else if ( DeltaCS[i]<= cutoffBL){
    mvt[i]<-"retract"
  }
  else if ( DeltaCS[i]>= -cutoffBL){
    mvt[i]<-"extend"
  }
  else {
    mvt[i]<-"immobile"
  }
}

DeltaCS<-unlist(DeltaCS)
DeltaPreCS<-unlist(DeltaPreCS)
mvt<-unlist(mvt)
BLdif<-cbind(Anims, DeltaCS, DeltaPreCS, mvt, Cname, Ccondition, Cgroup)
BLdif<-as.data.frame(BLdif)

colnames(BLdif)<-c("Individuals", "DeltaCS", "PreCSretraction", "MvtCategor
y", "Video", 'Condition', 'Group')
print(BLdif)

#Calculate the mean bodylength difference of the 10 animals and the total
number of animals retracting
DeltaCSMean<-mean(as.numeric(BLdif$DeltaCS), na.rm=TRUE)
DeltaPreCSMean<-mean(as.numeric(BLdif$PreCSretraction), na.rm=TRUE)
CountRetract<-length(which(BLdif$MvtCategory == "retract"))
```

```
# Print results for each video and add a summary row at the end of the calculations table for each video

BLdif<-rbind(BLdif, c('Mean',DeltaCSMean, DeltaPreCSMean, CountRetract))
print(BLdif)

print("The mean bodylength difference of the 10 animals between the beginning and the end of the video is:")
print(DeltaCSMean)
print("The number of animals retracting is:")
print(CountRetract)

# Print plots
print(cords)
print(BLplot)
print(BLplotIndiv)

BLdif<- as.list.data.frame(BLdif)
return(BLdif)

}
```

It's now time to use the function to perform the batch analysis of all the videos. It will automatically generate plots and save them in the working directory.

```
OutputData<- list()
DLC_data<- 0
Data<-0

for (i in 5:nbVids) {
  DLC_data<- read.csv2 (as.character(metadata$path[[i]]), sep=',', header=FALSE)
  name<-metadata$names[[i]]
  condition<-metadata$conditions[[i]]
  group<-metadata$groups[[i]]
  Anims<-metadata$AnimsList[[i]]
  MinF<-metadata$Begin.video[[i]]
  MaxF<-metadata$Duration.video[[i]]
  InitWindow<-metadata$Initial.BL.window..sec. [[i]]
  CSap1<-c(metadata$Occurence.CS..sec. [[i]], metadata$End.CS..sec. [[i]])
  rangeDifCS<-metadata$Interval.around.CSap1..sec. [[i]]
  rangeDifPP<-metadata$Pre.post.CS.window [[i]]
  rangeDif<-metadata$RangeBLdif [[i]]
  cutoffBL<-metadata$CutOff.BL.retraction [[i]]
  cutoffBLpre<-metadata$CutOff.BL.retraction.preCS [[i]]
  Data<-MeanBLdifVid_Categorical(DLC_data)
  OutputData[[i]] <- Data
}
```

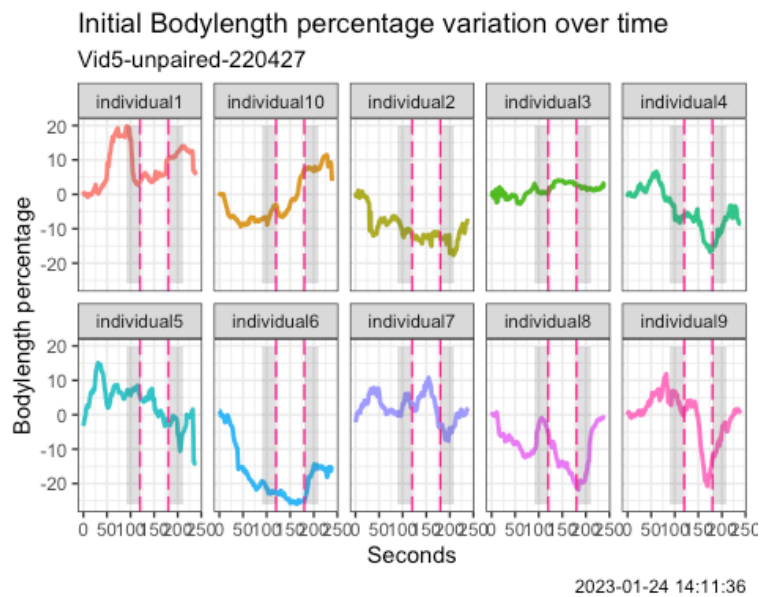
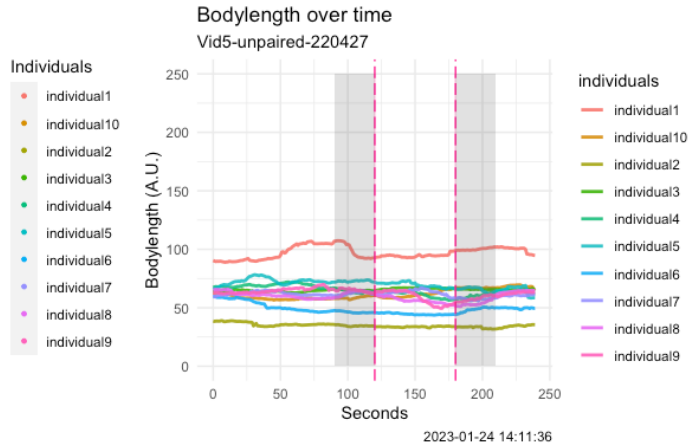
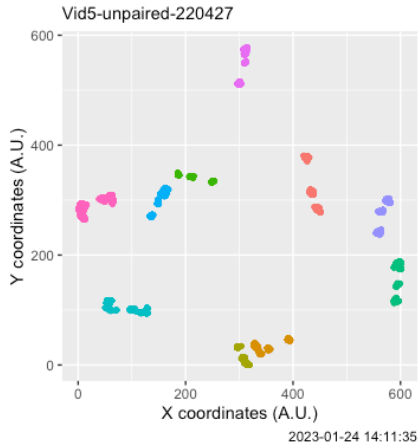
Output from the function:

First, a table with the results for the video analyzed: the values Δ_{preCS} and Δ_{CS} , the MvtCategory attributed, namely “retract” if $\Delta_{CS} < -10\%$ & $\Delta_{preCS} \geq -10\%$, “immobile” if $\Delta_{CS} \geq -10\%$ and NA if $\Delta_{preCS} < -10\%$.

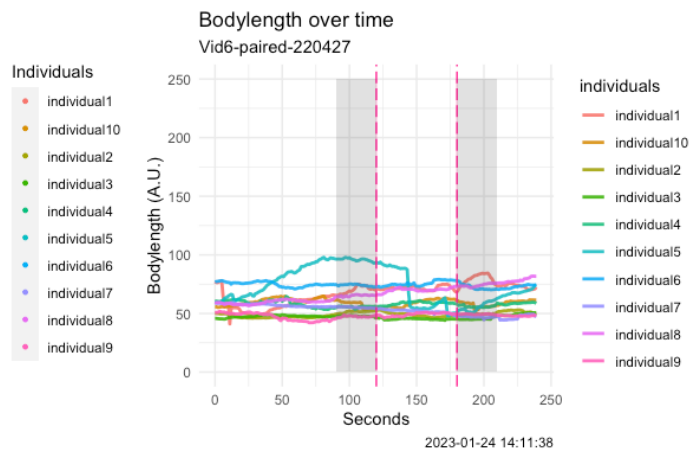
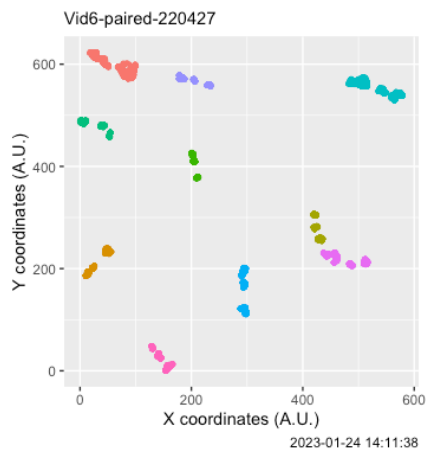
During the function execution, 3 plots per video are also generated (see below):

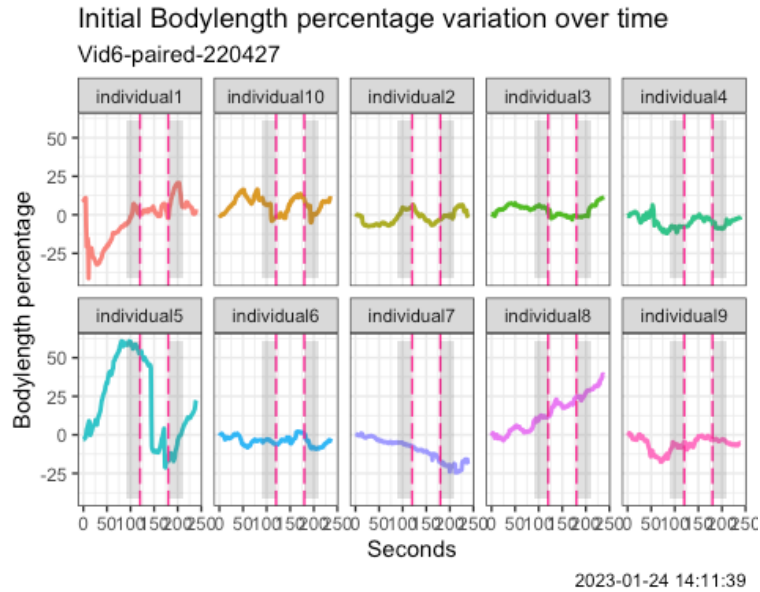
- a plot of the all the coordinates of all animals,
- a plot of all animals absolute BL variation (in pixels)
- a compound plot of %BL variation of each animal during the recording.

```
##      Individuals      DeltaCS      PreCSretraction MvtCategory Video
## 1 individual1 0.0226443978783744 -16.4105492511826      <NA> Vid5
## 2 individual2 -2.17884817087702 -3.15562104745417    immobile Vid5
## 3 individual3 -0.138785679141666 -2.92445921273047    immobile Vid5
## 4 individual4 -10.4707598297533 -5.13887902881334      retract Vid5
## 5 individual5 -9.18831805514962 -2.5967559559085    immobile Vid5
## 6 individual6 -3.54868158045646 -3.18564270407933    immobile Vid5
## 7 individual7 -6.45166485460924 -3.39479423953073    immobile Vid5
## 8 individual8 -15.4790889268213 -3.72978189950274      retract Vid5
## 9 individual9 -20.8988728796198 -6.80359047332174      retract Vid5
## 10 individual10 -3.53880654983412 -0.135738237231949    immobile Vid5
## 11      Mean -7.18711821283842 -4.74758120497556      3 <NA>
##      Condition Group
## 1 unpaired up1
## 2 unpaired up1
## 3 unpaired up1
## 4 unpaired up1
## 5 unpaired up1
## 6 unpaired up1
## 7 unpaired up1
## 8 unpaired up1
## 9 unpaired up1
## 10 unpaired up1
## 11      <NA> <NA>
## [1] "The mean bodylength difference of the 10 animals between the
beginning and the end of the video is:"
## [1] -7.187118
## [1] "The number of animals retracting is:"
## [1] 3
```

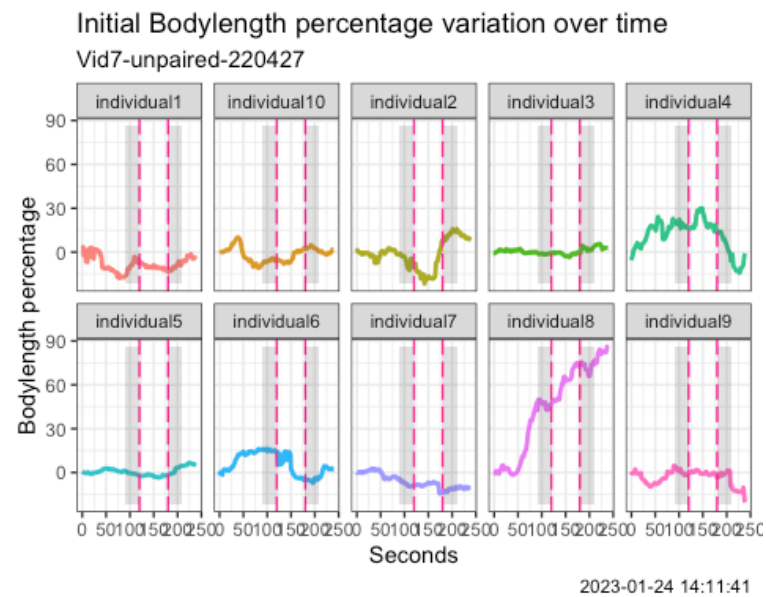
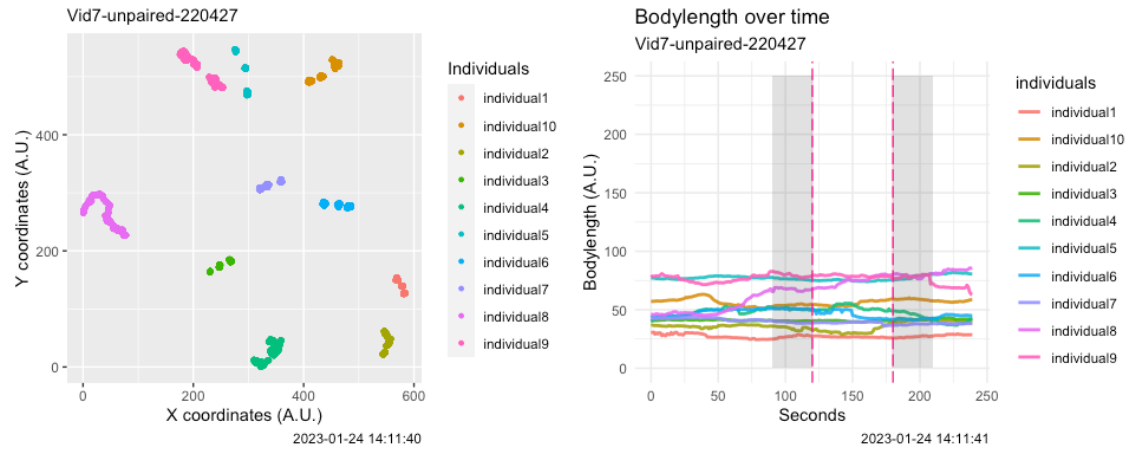


```
##      Individuals      DeltaCS      PreCSretraction MvtCategory Video
## 1 individual1 -2.84474948848824 -6.38556823059955    immobile Vid6
## 2 individual2 -11.6969630426253 0.0255187566243293    retract Vid6
## 3 individual3 -5.60352303295888 -3.91025295822773    immobile Vid6
## 4 individual4 -0.221240718395429 -0.531348116165011    immobile Vid6
## 5 individual5 -74.4186798296179 -7.64684269168578    retract Vid6
## 6 individual6 -0.325099436130977 -3.24963320776256    immobile Vid6
## 7 individual7 -10.5511662151406 -2.5519105101164    retract Vid6
## 8 individual8 -0.291937289983903 -1.05047521748484    immobile Vid6
## 9 individual9 -4.42309232510478 -0.502666810986342    immobile Vid6
## 10 individual10 -0.431731095128552 -12.7904714426346    <NA> Vid6
## 11      Mean -11.0808182473575 -3.85936504290384      3 <NA>
##      Condition Group
## 1      paired      p1
## 2      paired      p1
## 3      paired      p1
## 4      paired      p1
## 5      paired      p1
## 6      paired      p1
## 7      paired      p1
## 8      paired      p1
## 9      paired      p1
## 10     paired      p1
## 11     <NA> <NA>
## [1] "The mean bodylength difference of the 10 animals between the beginnin
g and the end of the video is:"
## [1] -11.08082
## [1] "The number of animals retracting is:"
## [1] 3
```

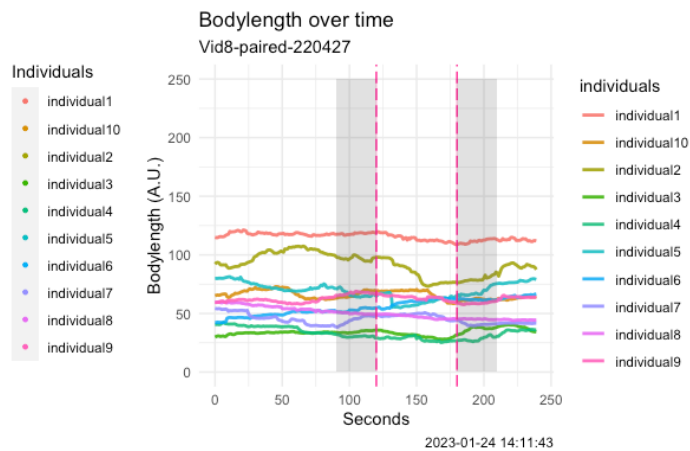
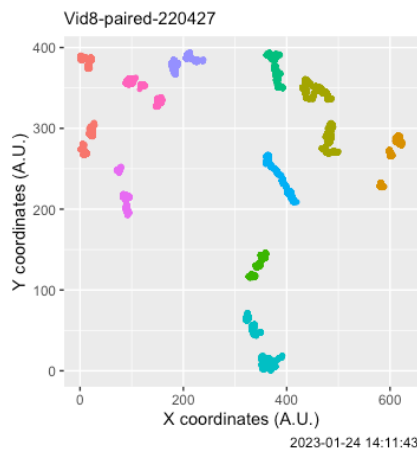


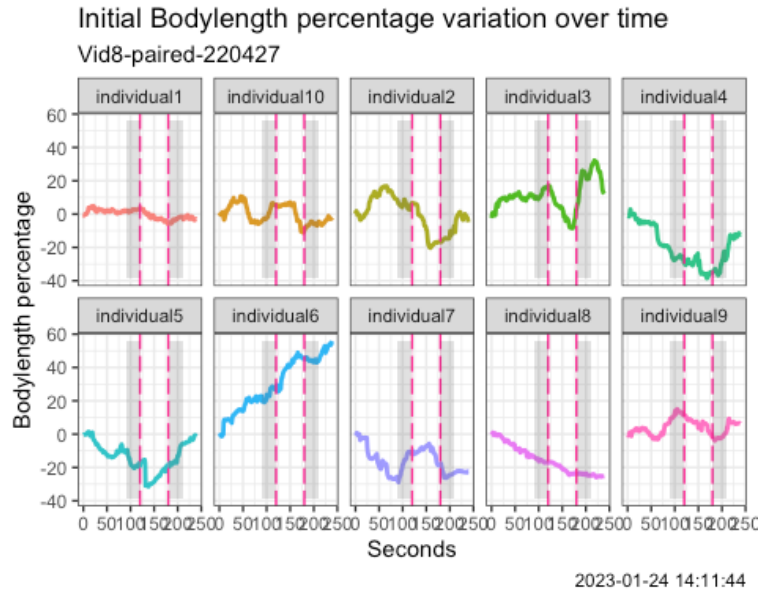


```
##      Individuals      DeltaCS      PreCSretraction MvtCategory Video
## 1  individual1  -5.41392173818563 -4.33927399986249    immobile Vid7
## 2  individual2 -13.1266277421497  -7.5468827701212      retract Vid7
## 3  individual3 -2.23582027978963  -1.40122077723276    immobile Vid7
## 4  individual4 -2.28545918467857  -6.06176726125086    immobile Vid7
## 5  individual5 -1.6082410011643   -3.26349223165229    immobile Vid7
## 6  individual6 -19.9248681997765  -1.46257300523937      retract Vid7
## 7  individual7 -5.29970518034909  -4.97316600070618    immobile Vid7
## 8  individual8 -0.473079460110419  -3.80986183646417    immobile Vid7
## 9  individual9 -2.01629024599527  -5.18672611188916    immobile Vid7
## 10 individual10 -3.1072997115665   -0.831957270453122    immobile Vid7
## 11      Mean  -5.54913127437656  -3.88769212648716          2  <NA>
##      Condition Group
## 1  unpaired  up2
## 2  unpaired  up2
## 3  unpaired  up2
## 4  unpaired  up2
## 5  unpaired  up2
## 6  unpaired  up2
## 7  unpaired  up2
## 8  unpaired  up2
## 9  unpaired  up2
## 10 unpaired  up2
## 11      <NA>  <NA>
## [1] "The mean bodylength difference of the 10 animals between the beginnin
g and the end of the video is:"
## [1] -5.549131
## [1] "The number of animals retracting is:"
## [1] 2
```



```
##      Individuals      DeltaCS      PreCSretraction MvtCategory Video
## 1 individual1 -8.89240337897981 -0.208654250051868    immobile Vid8
## 2 individual2 -26.716935161908 -2.37114772178013    retract Vid8
## 3 individual3 -25.1952344739777 -0.0131058274833933    retract Vid8
## 4 individual4 -10.6964642852394 -6.5323768851862    retract Vid8
## 5 individual5 -13.3005107937794 -8.77928217479642    retract Vid8
## 6 individual6 -2.10897379837778 -1.02636740787326    immobile Vid8
## 7 individual7 -11.099175261341 -0.659195790491765    retract Vid8
## 8 individual8 -7.61702212684665 -3.27094692963502    immobile Vid8
## 9 individual9 -12.4949323019222 -3.61726746235538    retract Vid8
## 10 individual10 -16.6021473658685 -1.11442099208162    retract Vid8
## 11      Mean -13.472379894824 -2.7592765441735      7 <NA>
##      Condition Group
## 1      paired      p2
## 2      paired      p2
## 3      paired      p2
## 4      paired      p2
## 5      paired      p2
## 6      paired      p2
## 7      paired      p2
## 8      paired      p2
## 9      paired      p2
## 10     paired      p2
## 11      <NA> <NA>
## [1] "The mean bodylength difference of the 10 animals between the beginnin
g and the end of the video is:"
## [1] -13.47238
## [1] "The number of animals retracting is:"
## [1] 7
```





```
# convert the output list into a dataframe
Joined <- data.table::rbindlist(OutputData)

# convert values into numbers and factors
Joined$DeltaCS<-as.numeric(Joined$DeltaCS)
Joined$MvtCategory<-as.factor(Joined$MvtCategory)
Joined$Individuals<-as.factor(Joined$Individuals)

# generate another data frame, without the summary row at the end of each group of rows for all the individuals per video
Joined_noNA<-na.omit(Joined)

# export Joined result data frames
fileName1<-paste(getwd(), '/', date, '_', cutoffBL, 'percent_CS1min_preCS10perc_full.csv', sep="")
write.csv(Joined,fileName1, row.names = F)
```

Summary plots and analysis

```
#Count the categories of movements for each condition -----

JoinedCount<- Joined_noNA %>% count(Video, Condition, MvtCategory)

Condition<-c("unpaired", "paired")
level_order <- c("unpaired", "paired")

# reshape data in a contingency table

library(dplyr)

ContingencyT<- JoinedCount %>%
  group_by(Condition, MvtCategory) %>%
  summarise(Freq = sum(n))

library(tidyr)

ContingencyT <- ContingencyT %>%
  # select only the columns we're interested in
  select(Condition, MvtCategory, Freq) %>%
  # use pivot_wider to go from long to wide format
  pivot_wider(names_from = "MvtCategory",
              names_prefix = "",
              values_from = "Freq")
ContingencyT<-as.data.frame(ContingencyT)

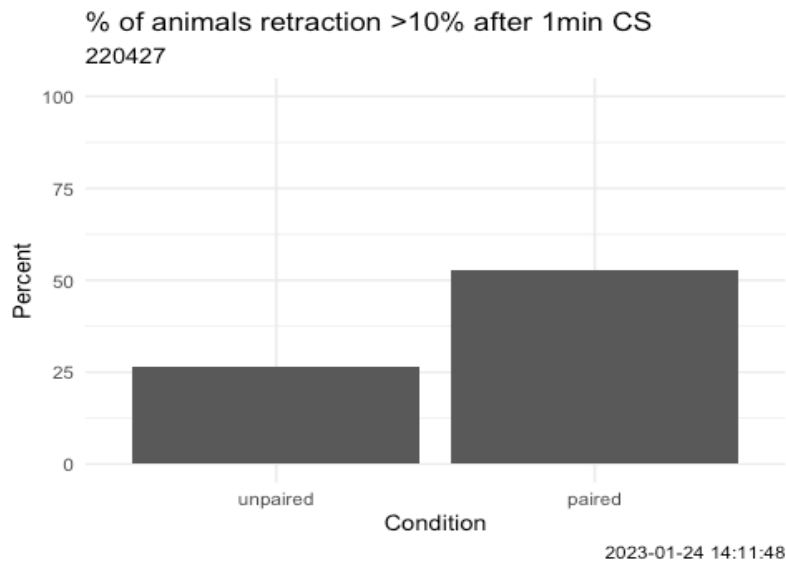
rownames(ContingencyT)<-ContingencyT[,1]
ContingencyT<-ContingencyT[,-1]

Percent<-c((ContingencyT[1, 2]/(ContingencyT[1,1]+ContingencyT[1,2])*100),(Co
ntingencyT[2, 2]/(ContingencyT[2,1]+ContingencyT[2,2])*100))
Prop<-data.frame(Condition, Percent)
Prop$Condition<-factor(Prop$Condition, levels=level_order)
```

```
# Plot the percentage of animals retracting in each condition -----
```

```
library(ggplot2)

ggplot(Prop, aes(x=Condition, y=Percent)) +
  geom_bar(stat = "identity")+
  theme_minimal()+
  ylim(0, 100)+
  labs(title = "% of animals retraction >10% after 1min CS",
        subtitle = paste(date), caption = Sys.time())
```



```
# Calculates summary statistics for the DeltaCS values: mean, sd, se and IC -  
---
```

```
library(dplyr)
my_sum <- Joined_noNA %>%
  group_by(Condition) %>%
  summarise(
    n=n(),
    mean=mean(DeltaCS),
    sd=sd(DeltaCS)
  ) %>%
  mutate( se=sd/sqrt(n)) %>%
  mutate( ic=se * qt((1-0.05)/2 + .5, n-1))

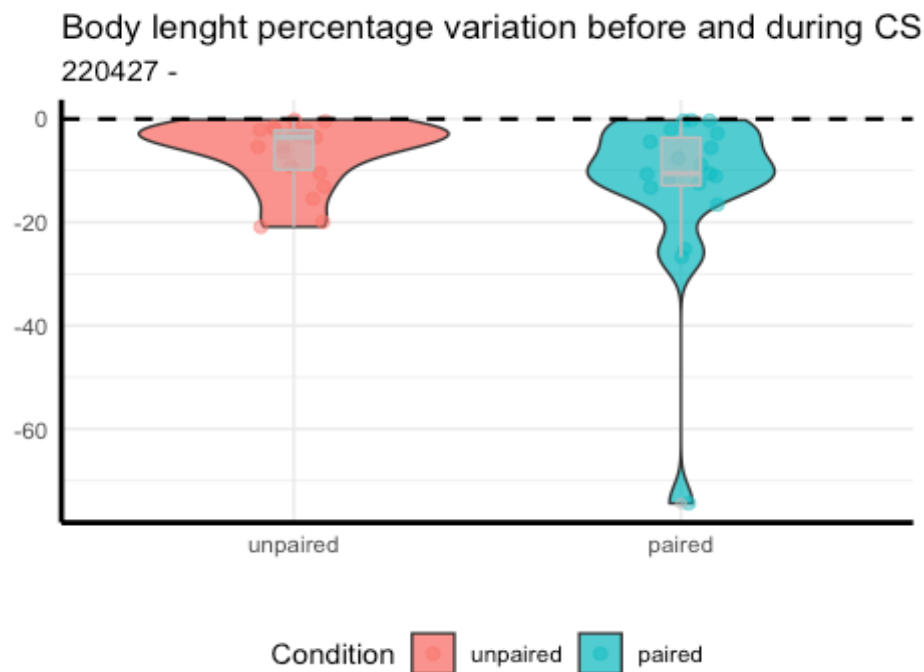
print(my_sum)

## # A tibble: 2 × 6
##   Condition     n  mean    sd    se    ic
##   <fct>    <int> <dbl> <dbl> <dbl> <dbl>
## 1 unpaired    19  -6.70  6.42  1.47  3.09
## 2 paired     19 -12.9  16.7  3.83  8.05
```

```
# Plot DeltaCS - Violin plot and boxplot -----
```

```
plot5<- ggplot(Joined_noNA, aes(x=Condition, y=DeltaCS, fill=Condition)) +  
  geom_violin(width=0.8, alpha=0.8)+  
  geom_point(aes(color=Condition), position=position_jitterdodge(0.4), size=2  
, alpha=0.5)+  
  geom_boxplot(notch=FALSE, width=0.1, color="grey", fill="grey", alpha=0.5)+  
  geom_hline(yintercept=0, linetype="dashed", color = "black", size=0.8)+  
  labs(title = "Body lenght percentage variation before and during CS",  
       subtitle = paste(date, "", sep=" - "), caption = Sys.time())+  
  theme_minimal()+  
  theme(legend.position = 'bottom', axis.line = element_line(colour = "black"  
, size = 1, linetype = "solid"))+  
  xlab('')+  
  ylab('')
```

```
print(plot5)
```



2023-01-24 14:11:48