

Synchrony in relationship, example with MONRADO Data

Thomas Gargot

May, 4th 2016

Contents

Fixed variables	2
Functions list	2
Import data	4
Clean dataframe	4
Presentation of the data	5
Length of the videos in minutes	6
Length of the videos in number of frames	7
Number of Available (True) and Not Available (False) data for each participant	8
Global Motion history	9
Mean Motion history by video by participant	9
Motion history box plots by frame (raw data), all videos	9
Raw data and mean of Motion History on sliding and non overlapping intervals on 00034 video	13
Raw data	13
Sliding interval	14
Non overlapping interval	14
Focus on the motion history of the first 10 seconds of the first video 00034	14
Sliding interval function on a 5 frames interval	14
Motion history of the father during 10-20 seconds of the first video 00034	16
Mean motion history by minute plots	18
Export log data in text files	38
SyncPy utilisation for creating synchrony dataframe	39
Description of SSI data frame	39

Synchrony scores for each dyad, triad and for the whole group	40
Evolution of synchrony through time, raw each second	60
Evolution of synchrony through time, mean by minute	60
Evolution of synchrony through time, mean by 10 minutes	61
Models of synchrony	61

```
# Clean Environment
rm(list = ls(all.names = TRUE))
```

Fixed variables

```
FileExtension <- ".MTS.avi_res.csv"

# working directory
setwd("/Users/0fix/Documents/Fac/internat/Recherche/projets/synchro/synchroData/Git/Monrado/Reports/")

# blue will refer to father
# red will refer to mother
# green to child
colorOrderList <- c("blue", "red", "green")

ParticipantsList <- c("father", "mother", "child")

## Create a csv files list with the directories
FullNameList <- list.files("../Data/CSV/raw", full.names=TRUE)
FullNameList

## Create a csv files list without the directories
filesList <- list.files("../Data/CSV/raw", full.names=FALSE)
filesList
```

Functions list

Import Data List

Function that import data from .csv files inside a CSV folder ##### Arguments: List FullNameList with the full name of the .csv

```
importdata <- function(FullnameList){
  data <- c()
  for (i in FullnameList){
    dataAlone <- read.csv(i)
    mydata.nas <- apply(dataAlone, 1, function(x){all(is.na(x))})
    dataAlone <- dataAlone[!mydata.nas,]
    print(i)
    data <- rbind(data, dataAlone)
  }
  return (data)
}
```

MeanMotionByTime

Function that takes raw motion history data and compute the mean on a given interval. Intervals don't overlap, so the frequency of the data change (from 25 frames by seconde to 25 frames/interval by second).

Arguments:

- subject : Subject studied (patient, mother, father or therapist)
- indexOfvideos : List of videos studied (element eg 3 or list eg 1:3 or c(1,2,4))
- interval : number of frames in the studied interval
- data : data frame where there is data

```
## Revoir nom des variables : pas clair, faire un schéma
MeanMotionByTime <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data){
  x <- c()
  for (fam in families[indexOfvideos]){
    dataVector <- data[which(data$family==fam), subject]
    ## with ceiling : superior limit of the round
    IntervalNumbersVideo <- ceiling(length(dataVector)/interval)
    for (i in 1:IntervalNumbersVideo){
      borneinf<- 1+(i-1)*interval
      bornesup <- i*interval

      dataVectorInterval <- dataVector[borneinf:bornesup]
      mean <- mean(dataVectorInterval, na.rm=TRUE)
      x <- c(x, mean)}}
  return (x)}
```

Slidinginterval

Function that takes raw motion history data and compute the mean on a given interval. The interval overlap, so the frequency of the data don't change. It stays at 25 frames/s.

Arguments:

- subject : subject studied (patient, mother, father or therapist)
- indexOfvideos : list of videos studied (element eg. 3 or list eg 1:3 or c(1,2,4))
- interval : number of frames in the studied interval
- data : data frame where there is data

```
## faire un schéma
SlidingInterval <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data)
{x <- c()
 for (file in families[indexOfvideos]){
  dataVector <- data[which(data$family==file), subject]
  NBofAnalysedFrames <- length(dataVector)-interval+1
  for (i in 1:NBofAnalysedFrames){
    borneinf<- (i)
    bornesup <-(interval-1+i)
    dataVectorInterval <- dataVector[borneinf:bornesup]
    mean <- mean(dataVectorInterval, na.rm=TRUE)}}
```

```

        x <- c(x, mean)})}
return (x)}

```

MeanSynchronyByTime (TODO)

Import data

```
data <- importdata(FullNameList)
```

Clean dataframe

Add new columns: compute minutes and log on data frame

```

# Detete No relevant subject here
data$therapist <- NULL

# compute time in minute
data$timeMin <- data$frame/(25*60)

## Create a list of files without the extention of the video
families <- c()
for (i in fileList){
  name <- sub(FileExtension, "", i)
  families <- c(families, name)
}
families

## [1] "00034"   "00037"   "00041"   "00048"   "0206"    "1106"    "1606"
## [8] "BAJE059"  "BALE050"  "BALU062"  "BEAL036"  "BEAM031"  "BICA"    "BRL0041"
## [15] "COL0022"   "DIPE004"  "DOMA"     "DRNE"     "FOMA057"  "GROP039"  "HAJA052"
## [22] "HUMA058"   "JAEM046"  "JEE0040"  "JOCE014"  "LACL"     "MAEL048"  "MAME20"
## [29] "MAPA029"   "MIPH043"  "MOSA065"  "RAEM049"  "RAKA008"  "RIEMO"   "SEEM035"
## [36] "SHANO42"   "SOGA061"  "TIUG032"  "VINO"

Number0fvideos <- length(families)
Number0fvideos

## [1] 39

# create a list with the simplified dname (whitout extension), make a data frmae of it and merge 2 data
a <- data.frame(family = families, unique(data$file))
data <- merge(data, a, by.x="file", by.y="unique.data.file.")

data$fatherShifted <- data$father + min(data$father[which (data$father >0)])/2
data$logFather <- log(data$fatherShifted)

data$motherShifted <- data$mother + min(data$mother[which (data$mother >0)])/2
data$logMother <- log(data$motherShifted)

```

```

data$childShifted <- data$child + min(data$child[which (data$child >0)])/2
data$logChild <- log(data$childShifted)

# Add date TODO
data$file <- NULL

data <- data[,c("family", "frame", "timeMin", "child", "childShifted", "logChild", "father", "fatherShifted")]

```

Presentation of the data

```
str(data)
```

```

## 'data.frame': 914498 obs. of 12 variables:
## $ family      : Factor w/ 39 levels "00034","00037",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ frame       : int 1 2 3 4 5 6 7 8 9 10 ...
## $ timeMin     : num 0.000667 0.001333 0.002 0.002667 0.003333 ...
## $ child        : num 1.62e-04 1.89e-04 7.50e-05 5.36e-05 8.04e-05 ...
## $ childShifted: num 1.62e-04 1.89e-04 7.55e-05 5.40e-05 8.08e-05 ...
## $ logChild     : num -8.73 -8.57 -9.49 -9.83 -9.42 ...
## $ father       : num NA NA NA NA NA NA NA NA NA ...
## $ fatherShifted: num NA NA NA NA NA NA NA NA NA ...
## $ logFather    : num NA NA NA NA NA NA NA NA NA ...
## $ mother       : num 4.00e-04 4.56e-04 2.23e-04 8.85e-05 9.58e-05 ...
## $ motherShifted: num 4.01e-04 4.57e-04 2.24e-04 8.89e-05 9.61e-05 ...
## $ logMother    : num -7.82 -7.69 -8.41 -9.33 -9.25 ...

```

```
summary(data)
```

```

##      family          frame        timeMin         child
## MOSA065: 26985   Min.   : 1   Min.   : 0.000667   Min.   :0.0000
## 00037 : 25641   1st Qu.: 5863  1st Qu.: 3.908667  1st Qu.:0.0006
## HUMA058: 25641   Median :11752   Median : 7.834667  Median :0.0034
## BAJE059: 25305   Mean   :11836   Mean   : 7.890467  Mean   :0.0093
## 1606   : 24957   3rd Qu.:17769   3rd Qu.:11.846000 3rd Qu.:0.0117
## 1106   : 24873   Max.   :26985   Max.   :17.990000  Max.   :0.9270
## (Other):761096                    NA's   :390
##      childShifted      logChild        father      fatherShifted
## Min.   :0.0000   Min.   :-14.6662   Min.   :0.0   Min.   :0.0
## 1st Qu.:0.0006   1st Qu.: -7.3412   1st Qu.:0.0   1st Qu.:0.0
## Median :0.0034   Median : -5.6720   Median :0.0   Median :0.0
## Mean   :0.0093   Mean   : -6.1928   Mean   :0.0   Mean   :0.0
## 3rd Qu.:0.0117   3rd Qu.: -4.4470   3rd Qu.:0.0   3rd Qu.:0.0
## Max.   :0.9270   Max.   : -0.0758   Max.   :0.1   Max.   :0.1
## NA's   :390      NA's   :390      NA's   :743181  NA's   :743181
##      logFather        mother      motherShifted      logMother
## Min.   :-14.6   Min.   :0.00   Min.   :0.00   Min.   :-14.78
## 1st Qu.: -9.2  1st Qu.:0.00   1st Qu.:0.00   1st Qu.: -8.41
## Median : -6.8  Median :0.00   Median :0.00   Median : -6.44
## Mean   : -7.3  Mean   :0.01   Mean   :0.01   Mean   : -7.03

```

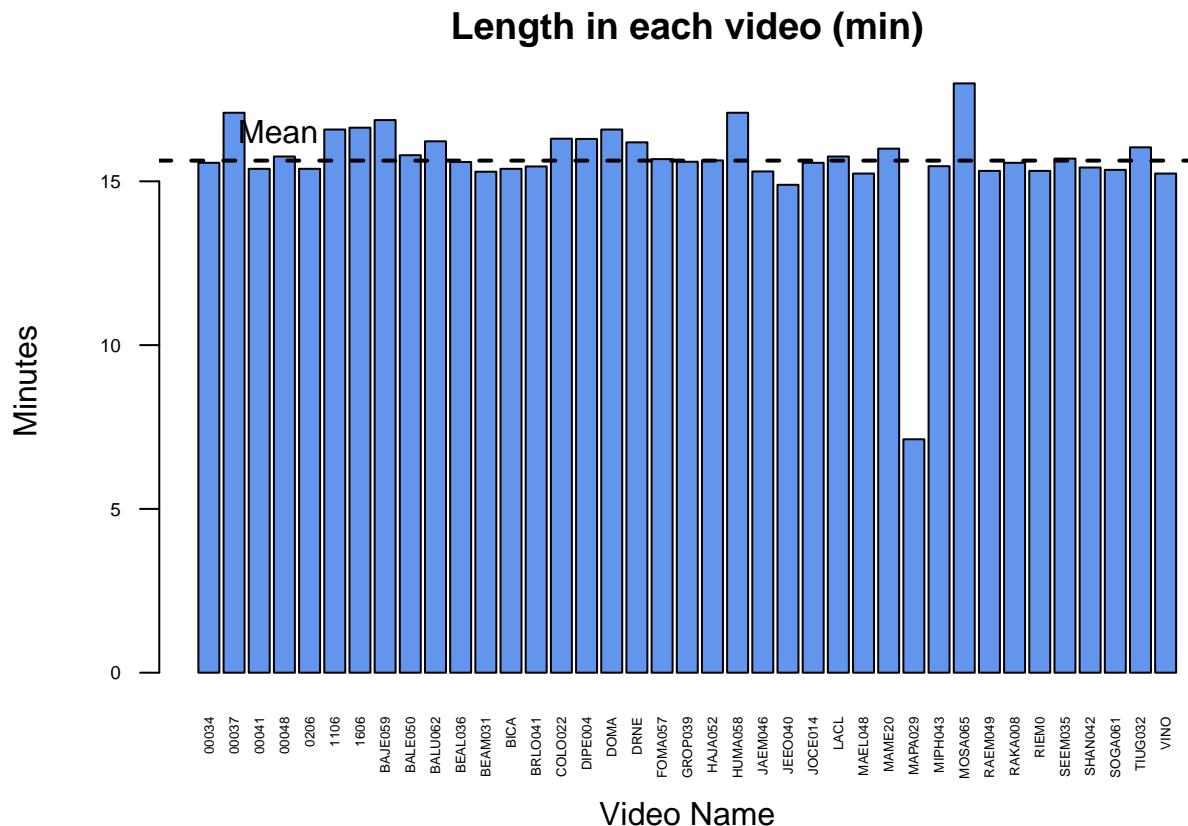
```

##   3rd Qu.: -5.1      3rd Qu.:0.01      3rd Qu.:0.01      3rd Qu.: -5.09
##   Max.    : -2.0      Max.    :0.96      Max.    :0.96      Max.    : -0.04
##   NA's    :743181     NA's    :148368     NA's    :148368     NA's    :148368

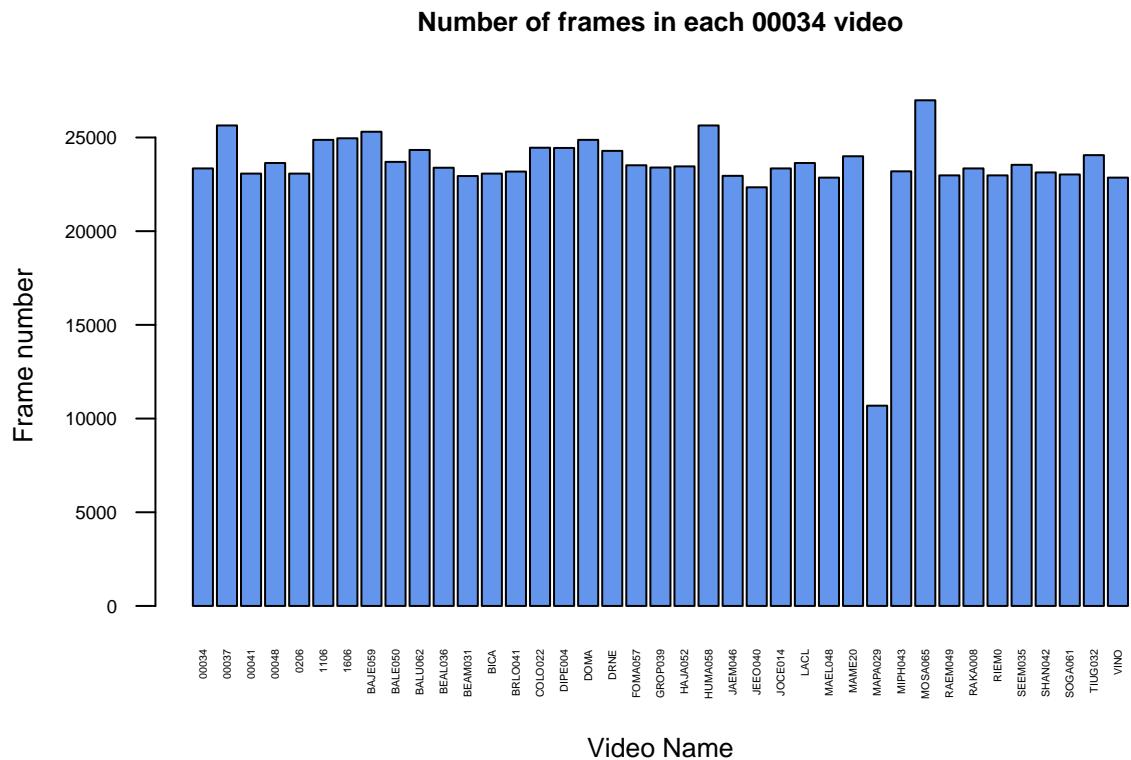
```

The timeMin is calculated with a frame rate of 25/sec.

Length of the videos in minutes

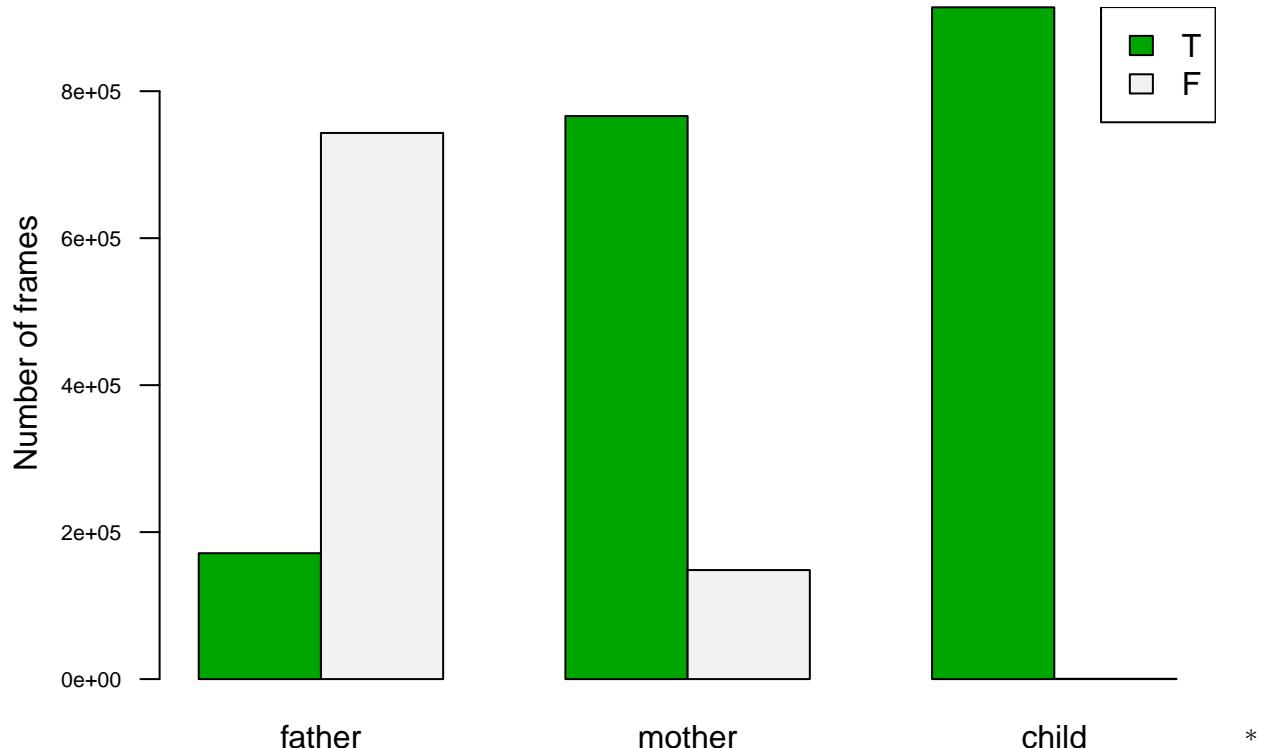


Length of the videos in number of frames



Number of Available (True) and Not Available (False) data for each participant

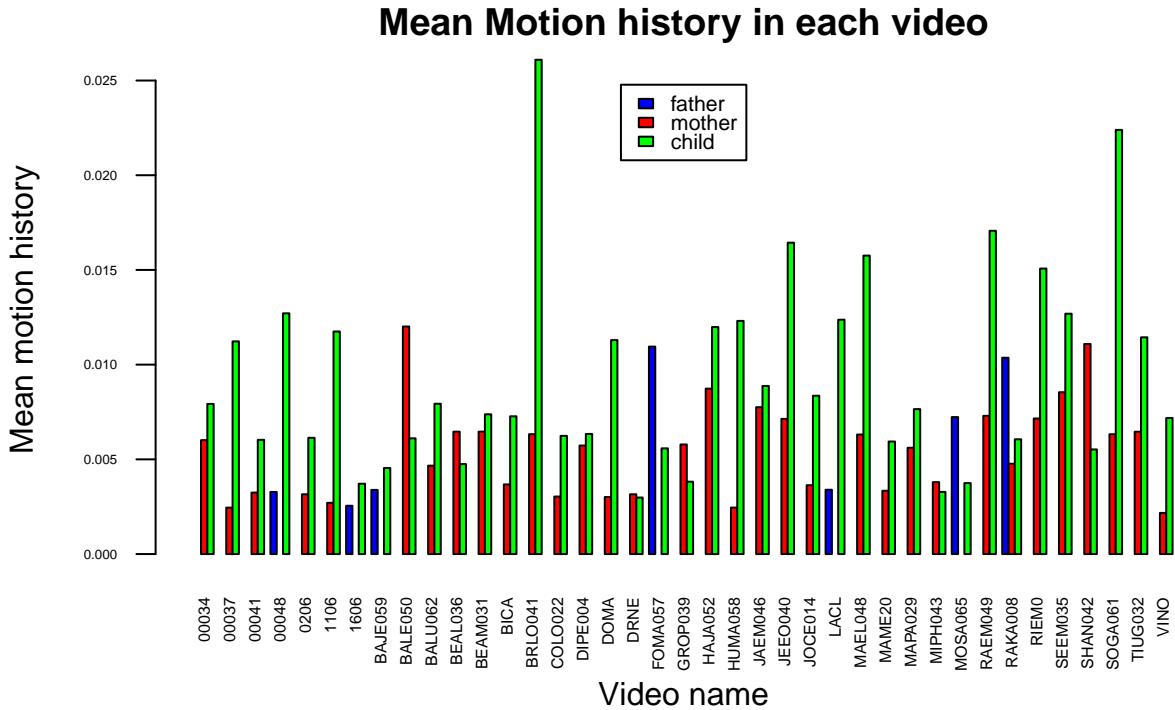
Number of available data by participant



All the participants involved are filmed. * All the children are filmed and we have data for each. * More often there is the other with him/her sometimes, it is the father * In some videos for instance RAKA008, there are 3 subjects

Global Motion history

Mean Motion history by video by participant



We can see that configurations of subjects are very similar (with always 2 subjects, except RAKA008 with 3 subjects). More often the child is with his mother. Consequently, it makes the comparaisons of the videos quite easy

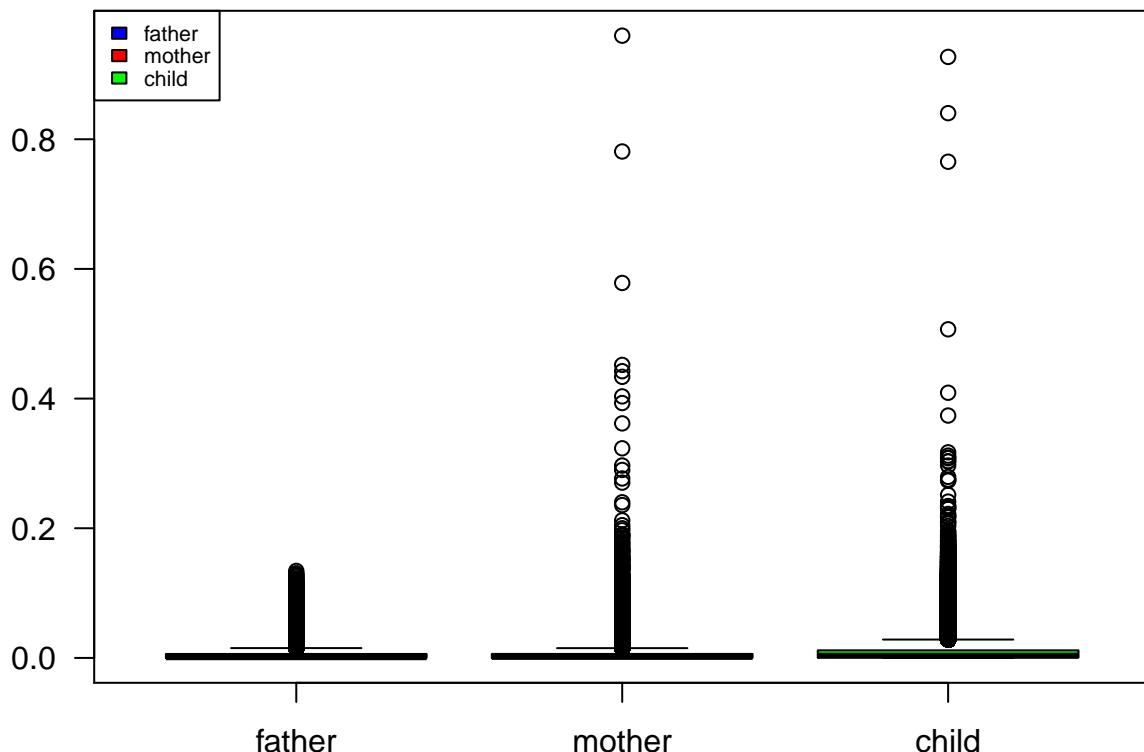
Motion history box plots by frame (raw data), all videos

The motion history is not normalized at all. Most of motions are very small but some of them are much big (long tail). This is very usual with this algorithm extraction motion history.

The subjects data are very similar.

```
par(mar=c(3,3,2,2))
boxplot(data$father, data$mother, data$child,
        col=colOrderList,
        names=ParticipantsList,
        main= "Motion history by frame box plots (raw data), all videos", las=1)
par(mar=c(1,0.5,0.5,1))
legend("topleft", ParticipantsList, fill=colOrderList, cex=0.7)
```

Motion history by frame box plots (raw data), all videos

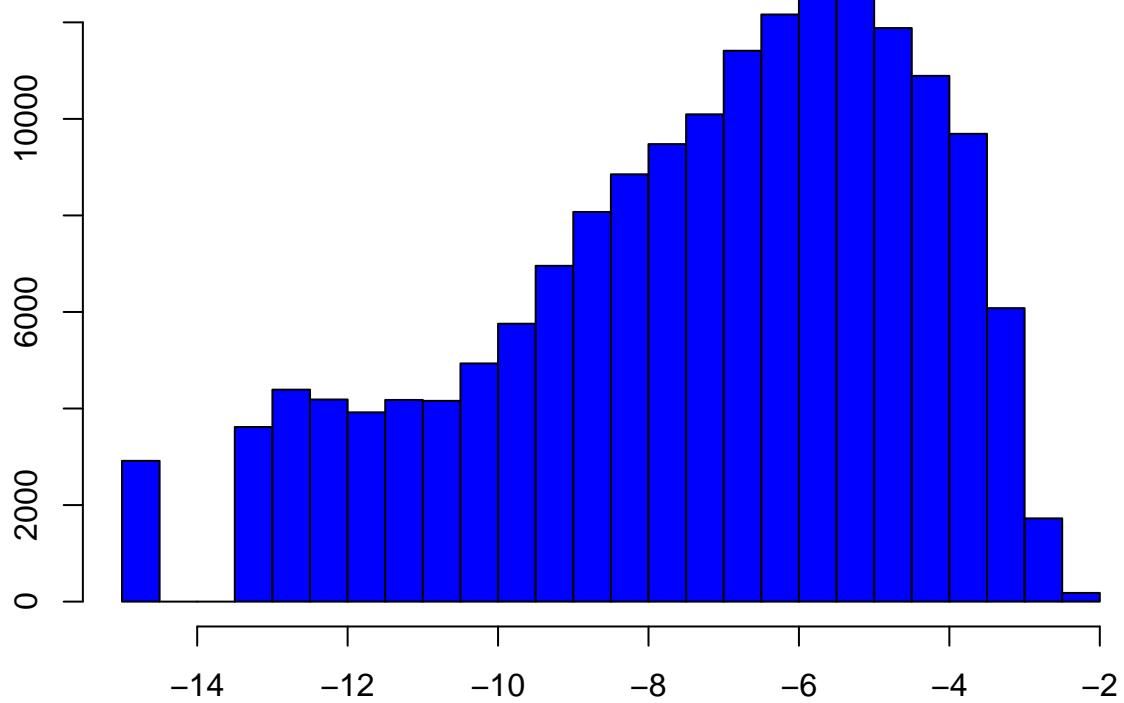


jects data are very similar.

The sub-

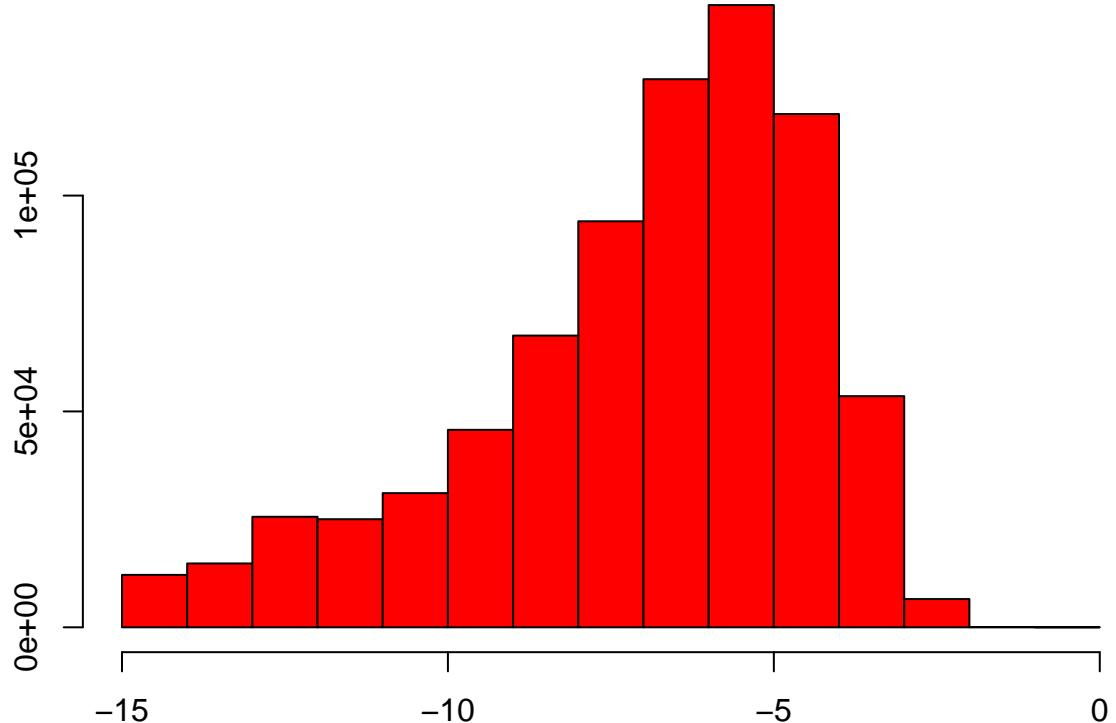
```
par(mar=c(3,3,2,2))
hist(data$logFather , col="blue")
```

Histogram of data\$logFather

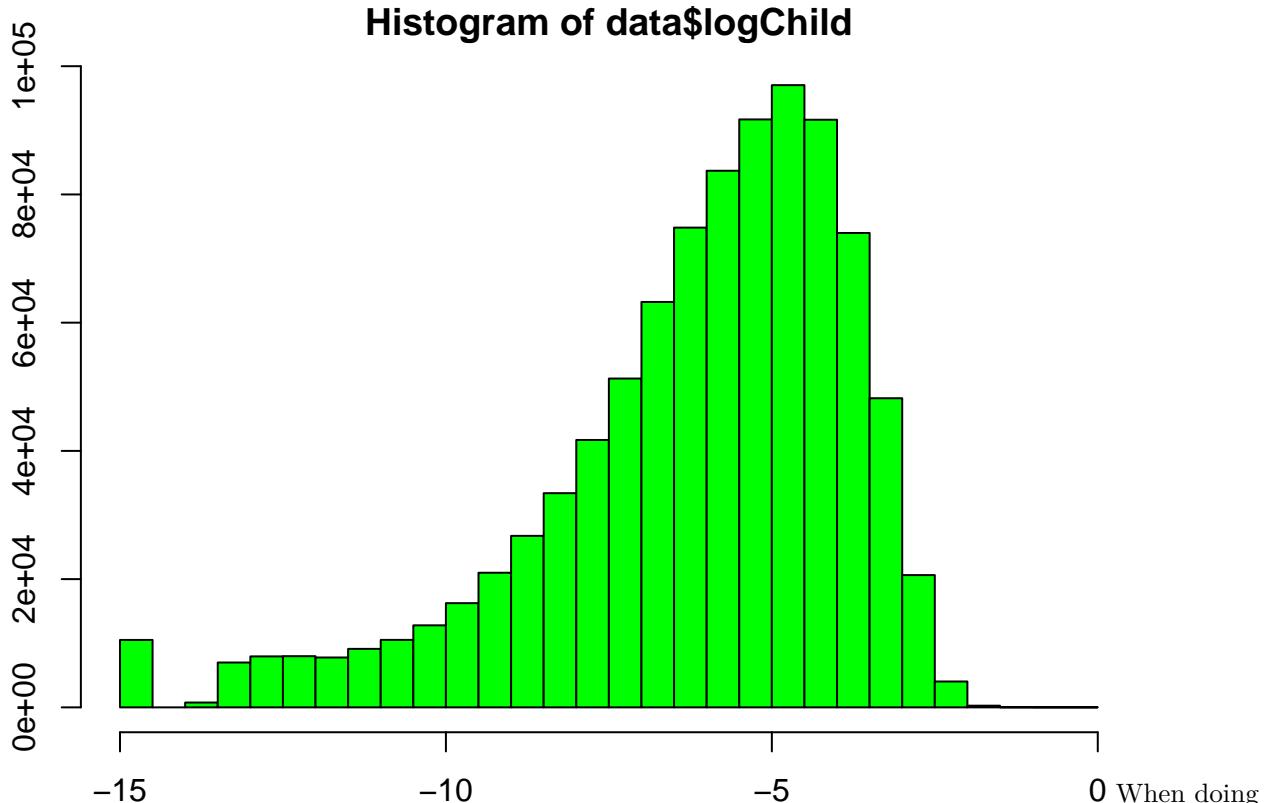


```
hist(data$logMother, col="red")
```

Histogram of data\$logMother



```
hist(data$logChild, col="green")
```

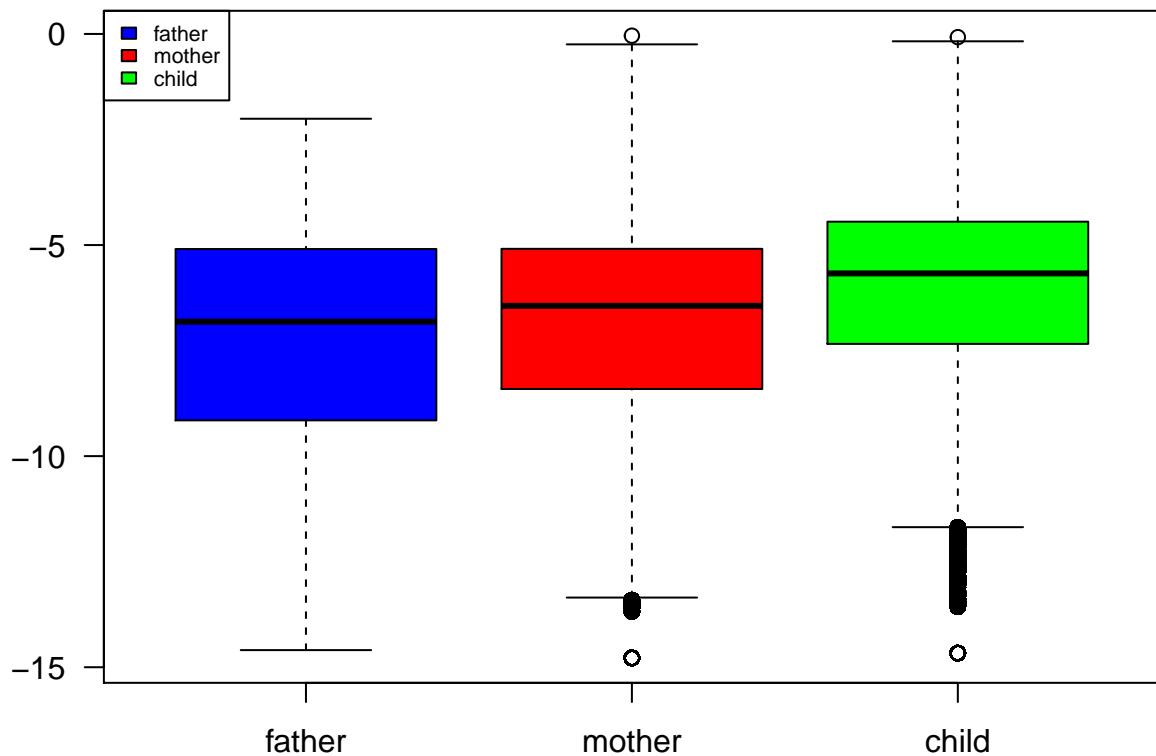


the log, we almost normalized the distribution. We couldn't do the log on 0. The result would give -Inf. We shifted all the distribution to the right by adding the half of the minimum after 0 of the distribution.

```
data$childShifted <- data$child + min(data$child[which(data$child > 0)]) / 2
```

```
par(mar=c(3,3,2,2))
boxplot(data$logFather, data$logMother, data$logChild,
        col=colOrderList,
        names=ParticipantsList,
        main= "Motion history by frame box plots (raw data), all videos", las=1)
par(mar=c(1,0.5,0.5,1))
legend("topleft", ParticipantsList, fill=colOrderList, cex=0.7)
```

Motion history by frame box plots (raw data), all videos



Raw data and mean of Motion History on sliding and non overlapping intervals on 00034 video

It is the first video of 00034.

Raw data

```
rawdataMother <- data[which(data$family=="00034"),]$mother
rawdataChild <- data[which(data$family=="00034"),]$child
```

```
summary(rawdataMother)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.    NA's
## 0.000000 0.000445 0.002666 0.006014 0.008546 0.173500      10
```

```
summary(rawdataChild)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.    NA's
## 0.000000 0.000537 0.002285 0.007926 0.008199 0.179000      10
```

Sliding interval

```
## REMINDER:  
# SlidingInterval <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data) with :  
# subject : subject studied (patient, mother, father or therapist)  
# indexOfvideos : list of videos studied (element eg. 3 or list eg 1:3 or c(1,2,4))  
# interval : number of frames in the studied interval  
# data : data frame where there is data  
  
slidedMother <- SlidingInterval("mother", 1 , 5, data)  
slidedChild <- SlidingInterval("child", 1 , 5, data)  
  
summary(slidedMother)  
  
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.    NA's  
## 0.000001 0.000536 0.002971 0.006014 0.008495 0.161600          6  
  
summary(slidedChild)  
  
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.    NA's  
## 0.000000 0.000636 0.002439 0.007926 0.008279 0.168400          6
```

Non overlapping interval

```
motherFive <- MeanMotionByTime("mother", indexOfvideos=1, interval=5, data)  
  
childFive <- MeanMotionByTime("child", indexOfvideos=1, interval=5, data)  
  
summary(childFive)  
  
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.    NA's  
## 0.0000002 0.0006356 0.0024170 0.0079250 0.0082800 0.1646000          2  
  
summary(motherFive)  
  
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.    NA's  
## 0.0000005 0.0005269 0.0030150 0.0060130 0.0085040 0.1529000          2
```

Focus on the motion history of the first 10 seconds of the first video 00034

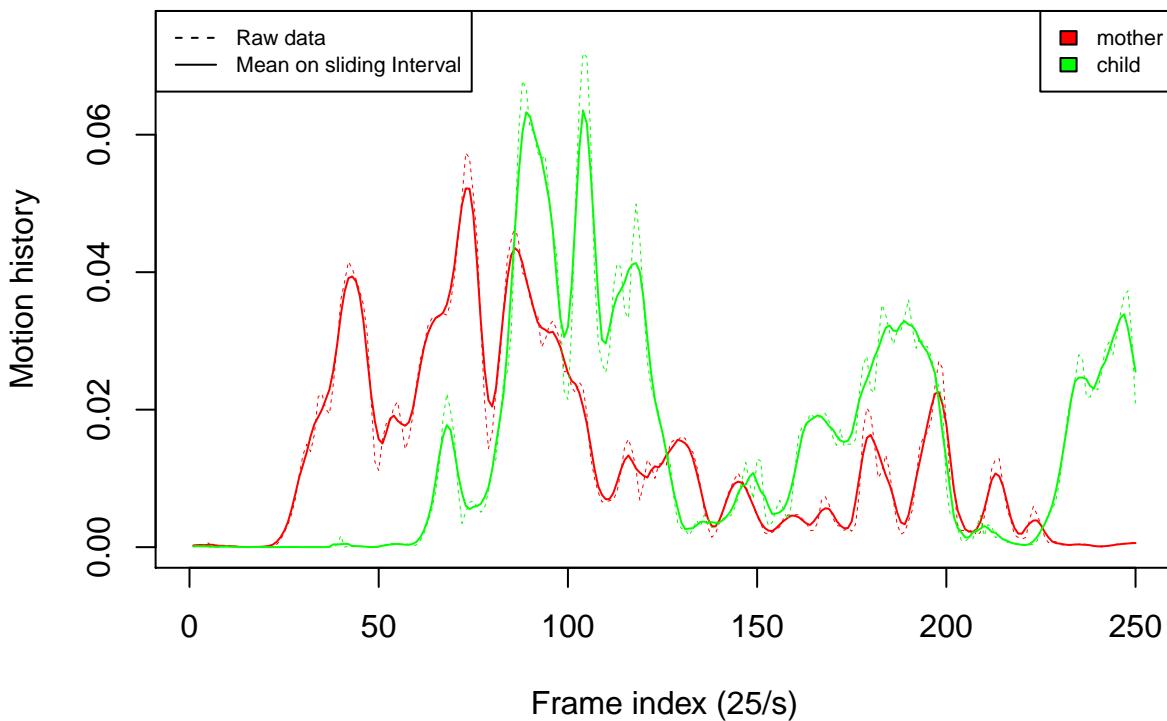
Sliding interval function on a 5 frames interval

```

par(mar=c(4,4,4,2))
plot(1:250, data$mother[3:252], main="Mean motion history (Sliding 5 frames interval)
on 00034 video, first 10 seconds ", xlab="Frame index (25/s)",
ylab="Motion history",
col="red", type="l", lty=2, lwd=0.5, ylim=c(0, 0.075))
lines(slidedMother[1:250], col="red", lty=1)
lines(slidedChild[1:250], col="green", lty=1)
lines(data$child[3:252], col="green", lty=2, lwd=0.5)
legend("topleft", c("Raw data", "Mean on sliding Interval"), lty=c(2, 1), cex=0.7)
legend("topright", ParticipantsList[c(2,3)], fill=colOrderList[c(2,3)], cex=0.7)

```

Mean motion history (Sliding 5 frames interval) on 00034 video, first 10 seconds



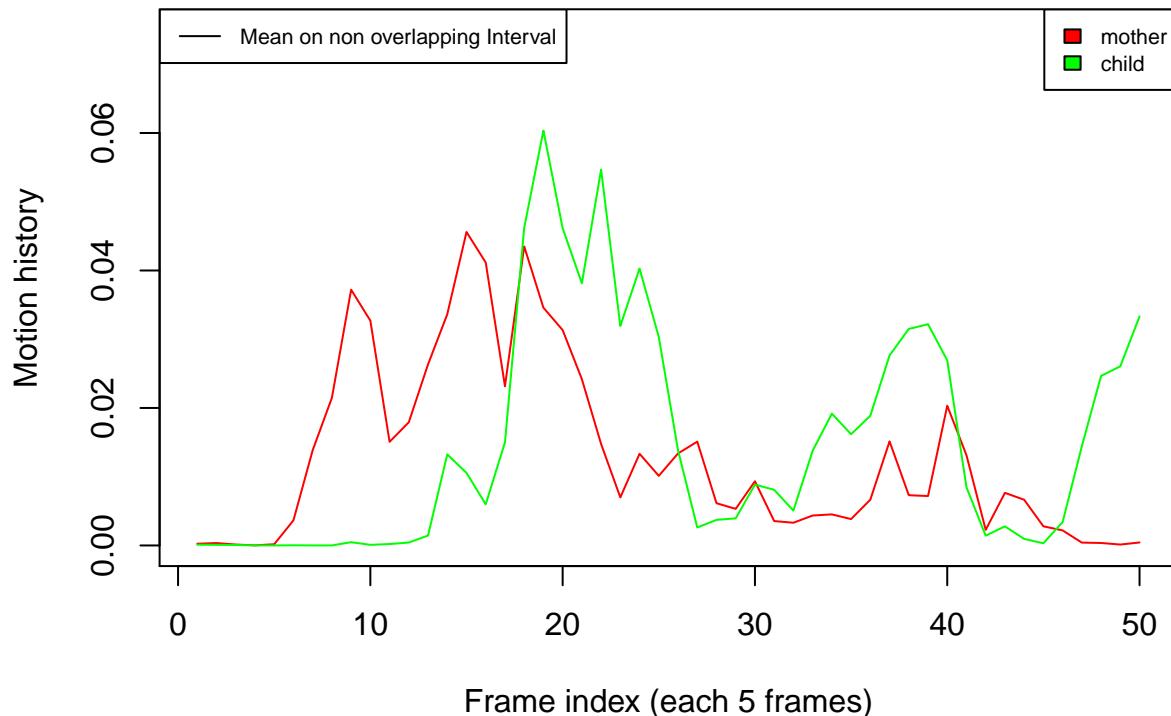
Non overlapping interval function on a 5 frames interval

```

par(mar=c(4,4,4,2))
plot (1:50, motherFive[1:50], type="l", col="red",
main="Mean Motion history (non overlapping 5 frames
intervals) for father on 00034 video, first 10 seconds",
ylab="Motion history", xlab="Frame index (each 5 frames)", ylim=c(0, 0.075))
lines(childFive[1:50], col="green", lty=1)
legend("topleft", "Mean on non overlapping Interval" , lty=1, cex=0.7)
legend("topright", ParticipantsList[c(2,3)], fill=colOrderList[2:3], cex=0.7)

```

Mean Motion history (non overlapping 5 frames intervals) for father on 00034 video, first 10 seconds

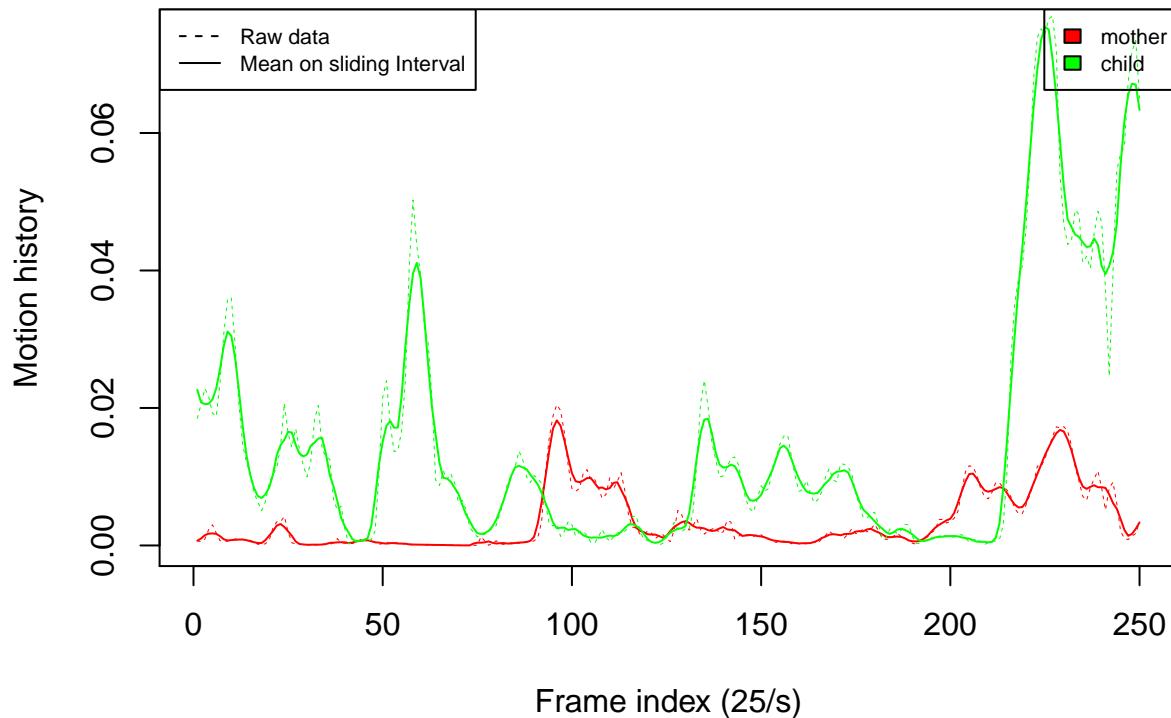


Motion history of the father during 10-20 seconds of the first video 00034

Non overlapping interval function on a 5 frames interval

```
par(mar=c(4,4,4,2))
plot(1:250, data$mother[253:502], main="Mean motion history (Sliding 5 frames
interval) for father on 00034 video, 10-20 seconds", xlab="Frame index (25/s)",
ylab="Motion history", col="red", type="l", lty=2, lwd=0.5, ylim=c(0, 0.075))
lines(slidedMother[251:500], col="red", lty=1)
lines(data$child[253:502], col="green", lty=2, lwd=0.5)
lines(slidedChild[251:500], col="green", lty=1)
legend("topleft", c("Raw data", "Mean on sliding Interval"), lty=c(2, 1), cex=0.7)
legend("topright", ParticipantsList[c(2,3)], fill=colOrderList[c(2,3)], cex=0.7)
```

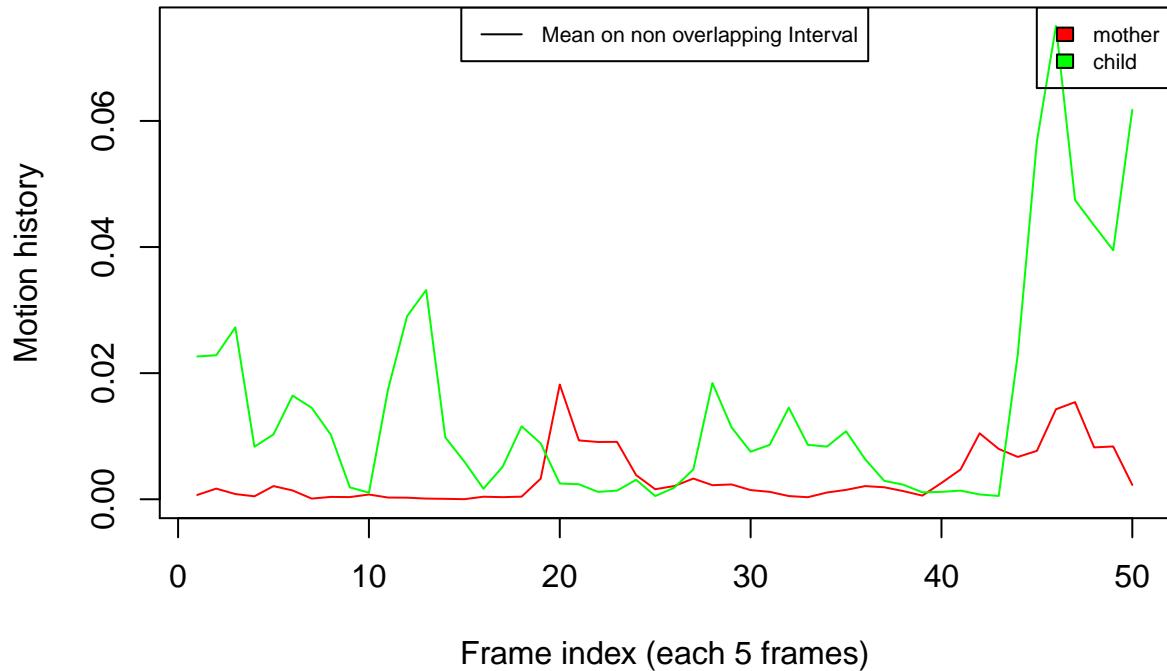
Mean motion history (Sliding 5 frames interval) for father on 00034 video, 10–20 seconds



Non overlapping interval function on a 5 frames interval

```
plot (1:50, motherFive[51:100], type="l", col="red",
main="Mean motion history (non overlapping 5 frames intervals) on
00034 video, between 10–20 seconds",
ylab="Motion history", xlab="Frame index (each 5 frames)", ylim=c(0, 0.075))
lines(childFive[51:100], col="green", lty=1)
legend("top", "Mean on non overlapping Interval" , lty=1, cex=0.7)
legend("topright", ParticipantsList[c(2,3)], fill=colOrderList[c(2,3)], cex=0.7)
```

Mean motion history (non overlapping 5 frames intervals) on 00034 video, between 10–20 seconds



Mean motion history by minute plots

```

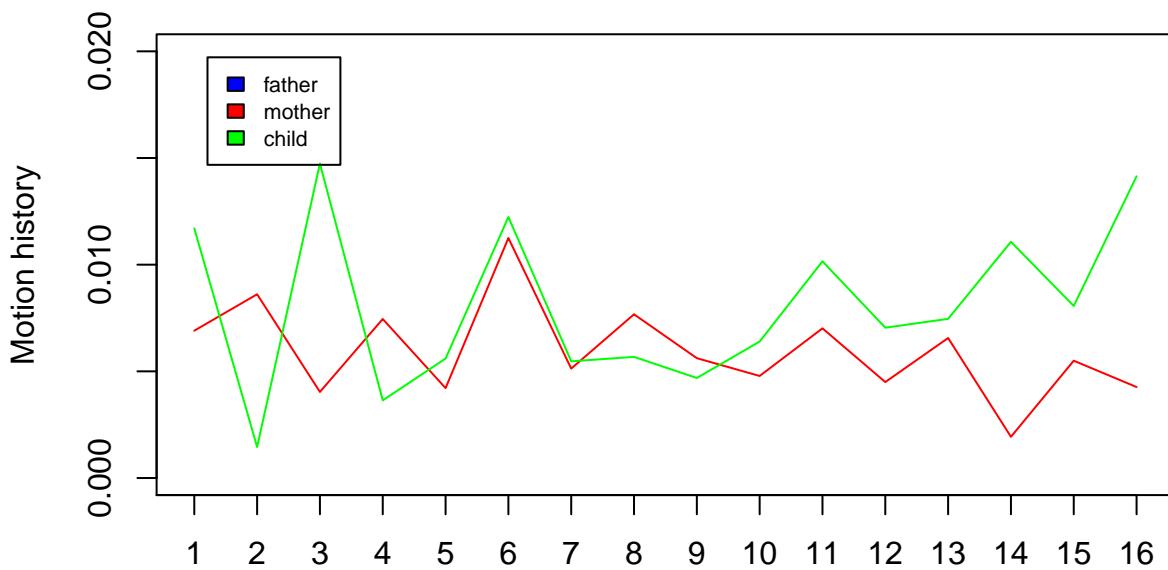
for (i in 1:Number0fvideos){
  fatherMinute<- MeanMotionByTime("father", index0fvideos=i, interval=1500, data)

  motherMinute<- MeanMotionByTime("mother", index0fvideos=i, interval=1500, data)

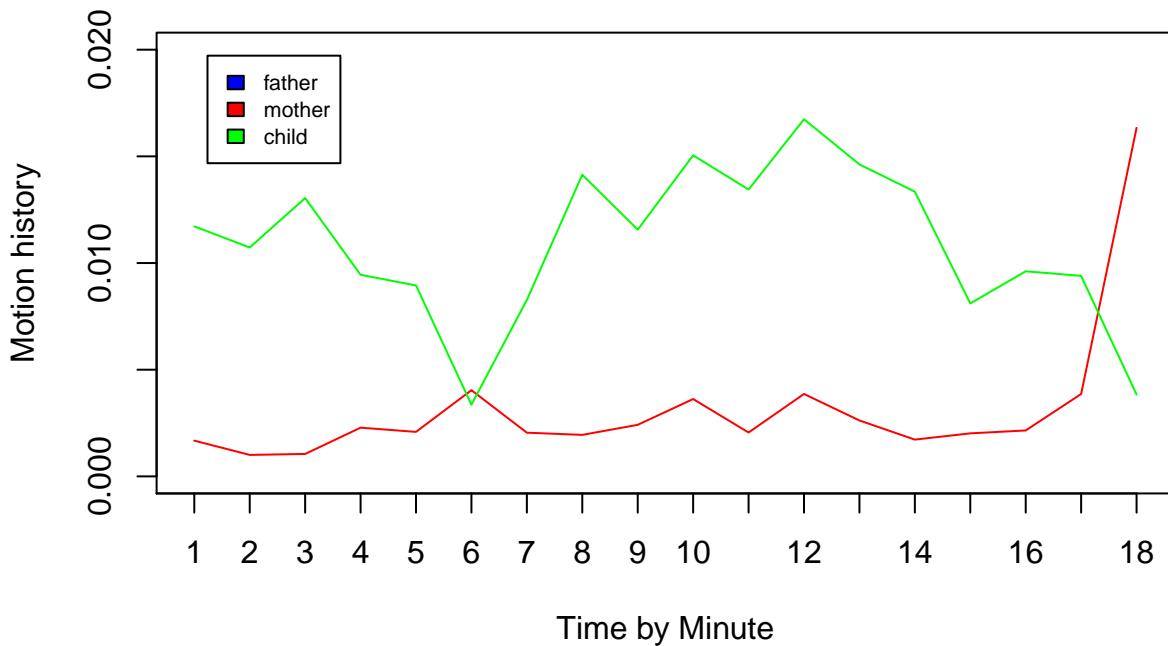
  childMinute<- MeanMotionByTime("child", index0fvideos=i, interval=1500, data)

  par(mar=c(4,4,4,2))
    plot (1:length(fatherMinute), fatherMinute, type="l", col="blue",
    main=paste("Mean motion history (non overlaping minute intervals)
    on ", families[i], " video" , sep=""),
    ylab="Motion history", xlab="Time by Minute", ylim=c(0, 20E-03),
    xaxp=c(0, length(fatherMinute), length(fatherMinute)))
    lines(motherMinute, col="red")
    lines(childMinute, col="green")
    legend("topleft", inset=.05, ParticipantsList[1:3],
    fill=colOrderList[1:3], cex=0.7)}
  
```

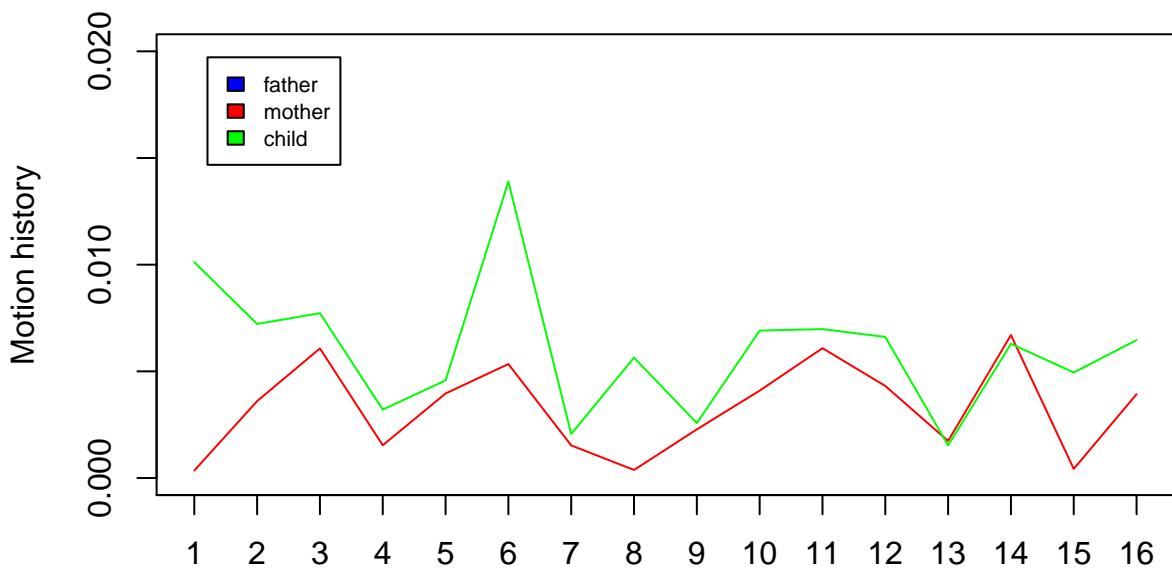
**Mean motion history (non overlapping minute intervals)
on 00034 video**



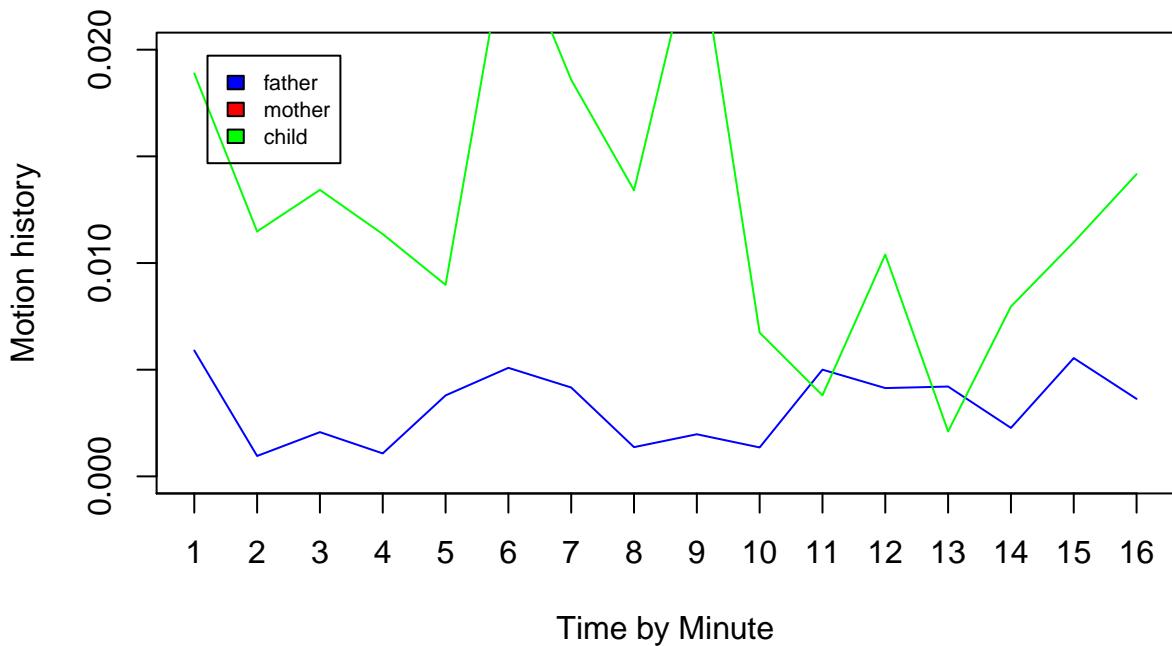
**Mean motion history (non overlapping minute intervals)
on 00037 video**



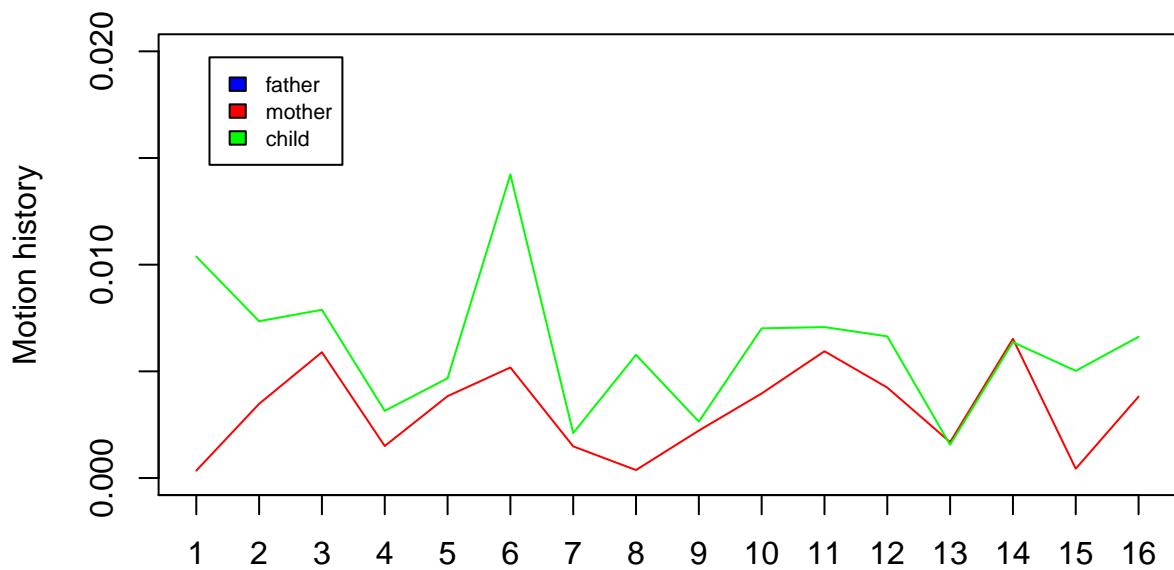
**Mean motion history (non overlapping minute intervals)
on 00041 video**



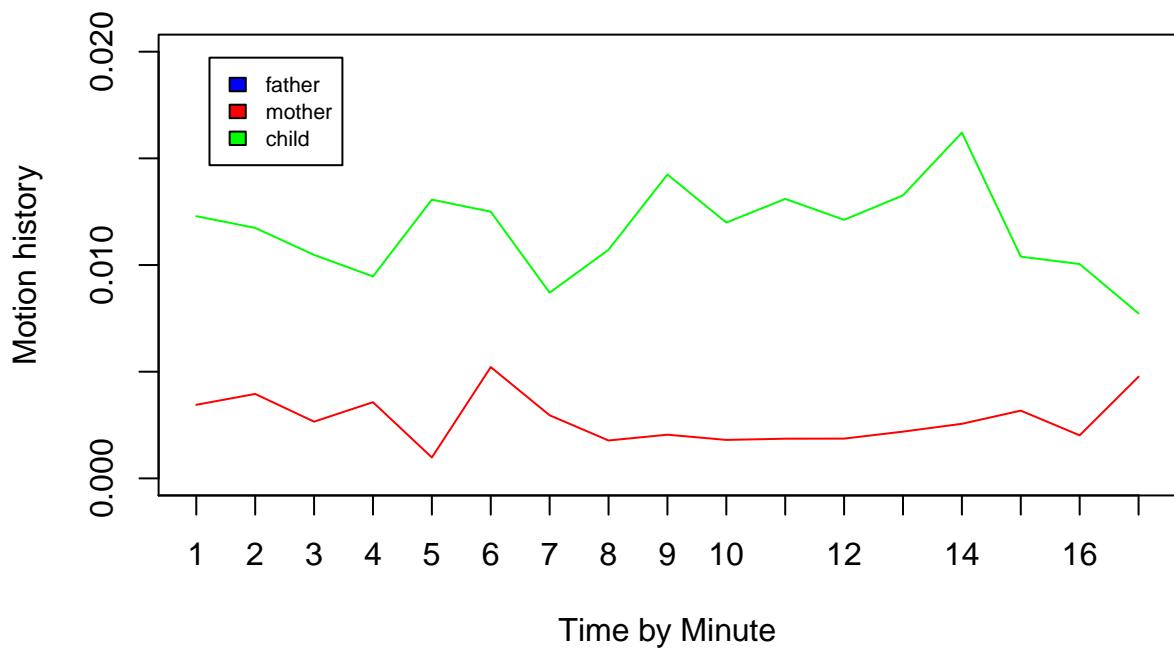
**Time by Minute
Mean motion history (non overlapping minute intervals)
on 00048 video**



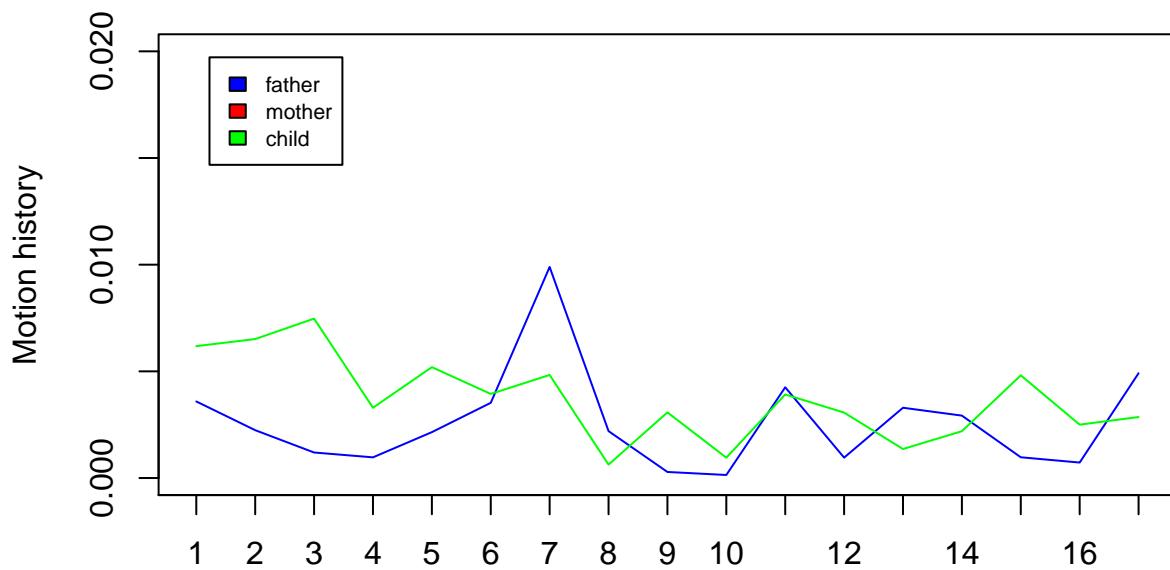
**Mean motion history (non overlapping minute intervals)
on 0206 video**



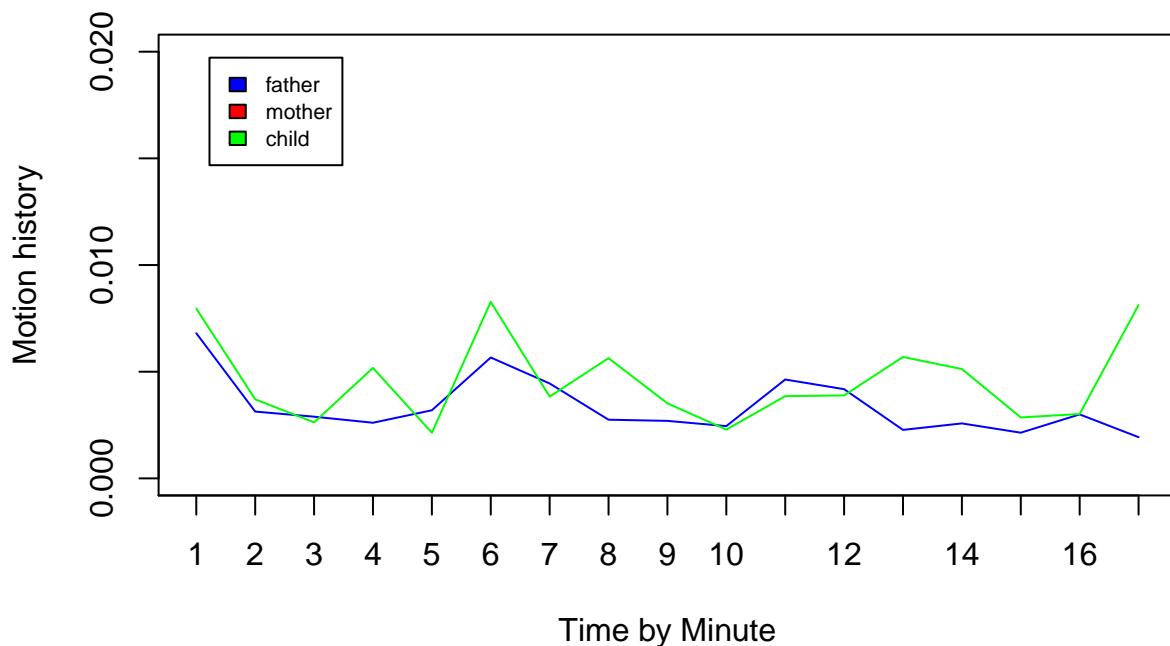
**Time by Minute
Mean motion history (non overlapping minute intervals)
on 1106 video**



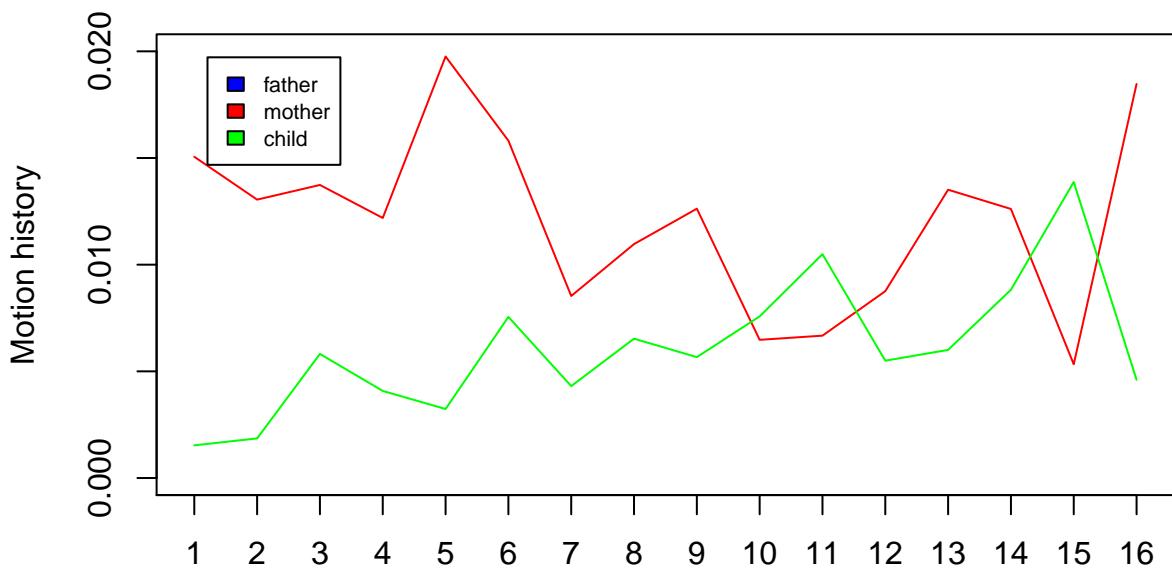
**Mean motion history (non overlapping minute intervals)
on 1606 video**



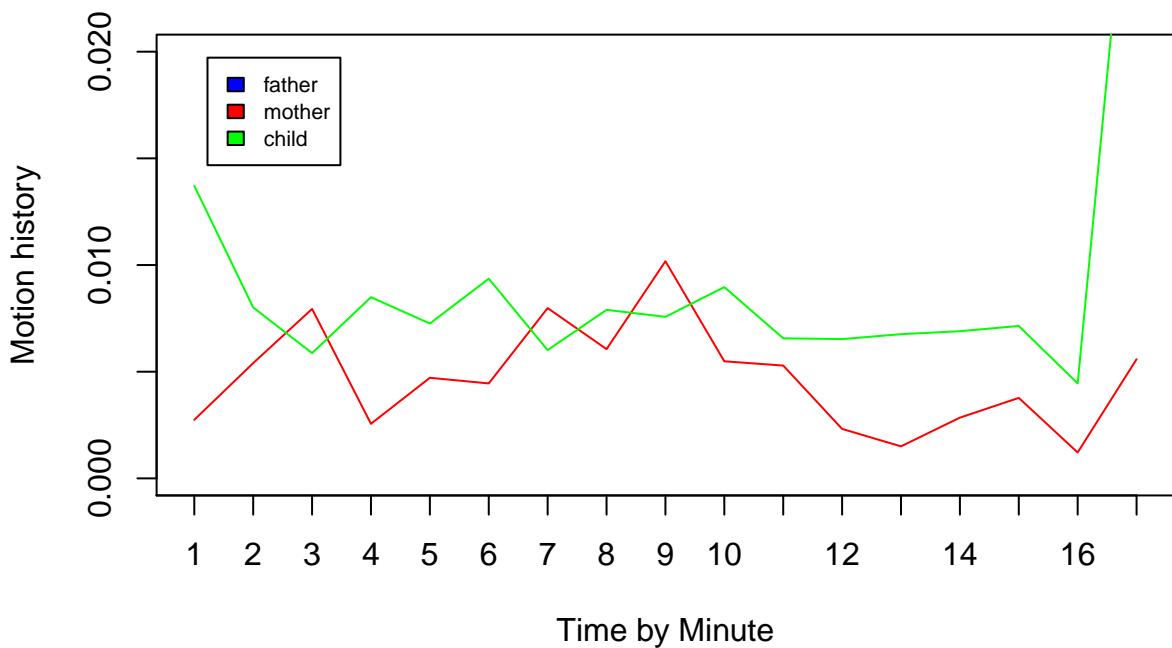
**Time by Minute
Mean motion history (non overlapping minute intervals)
on BAJE059 video**



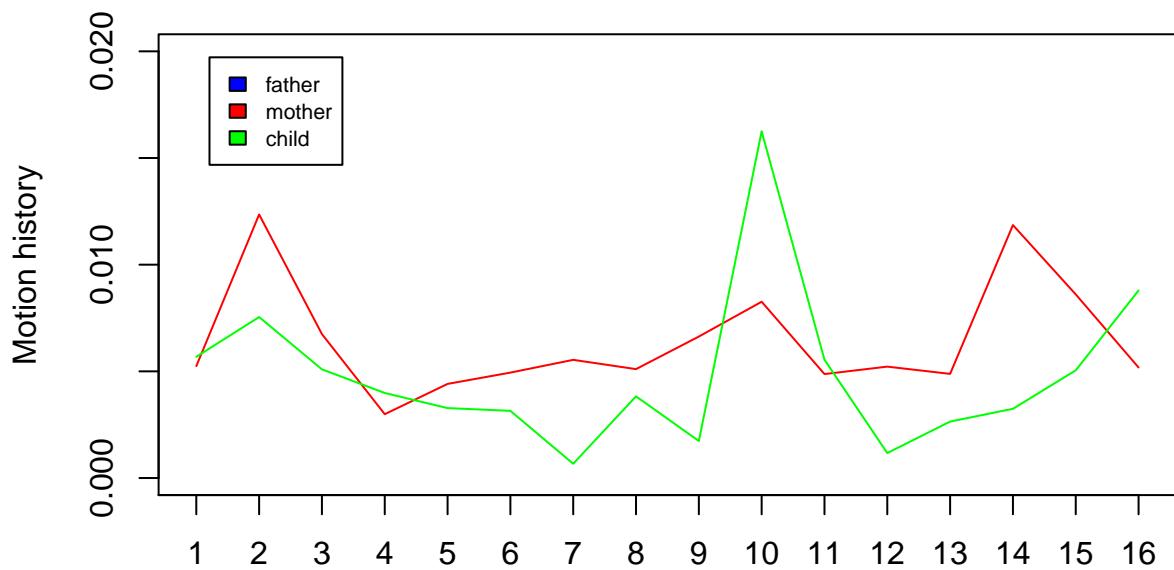
**Mean motion history (non overlapping minute intervals)
on BALE050 video**



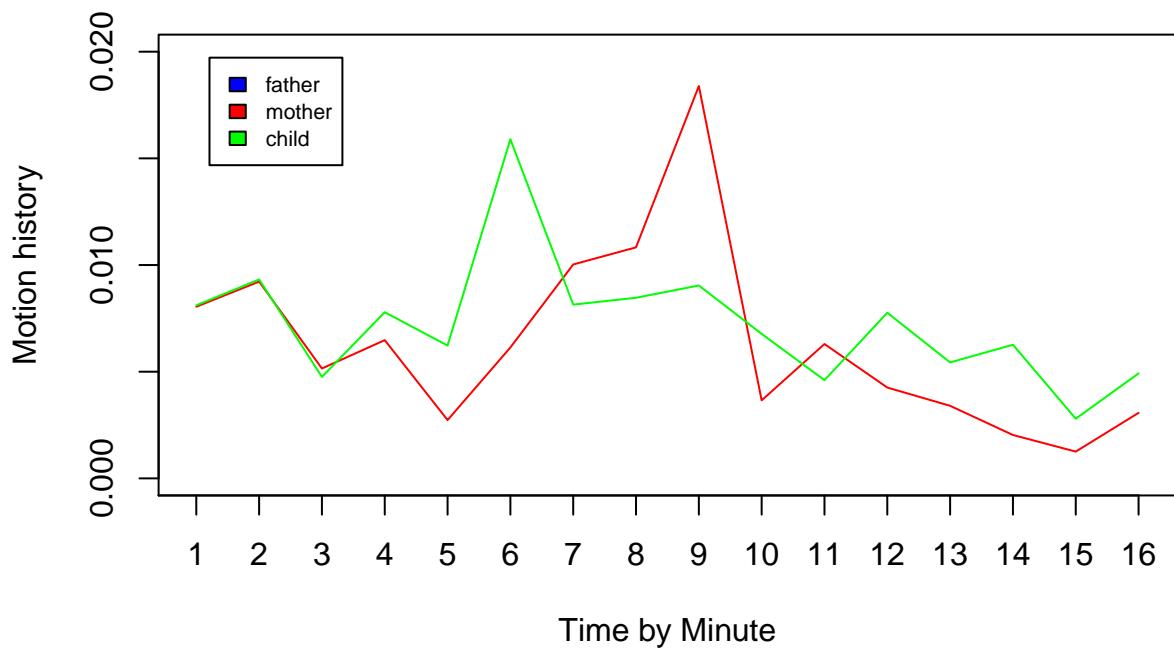
**Time by Minute
Mean motion history (non overlapping minute intervals)
on BALU062 video**



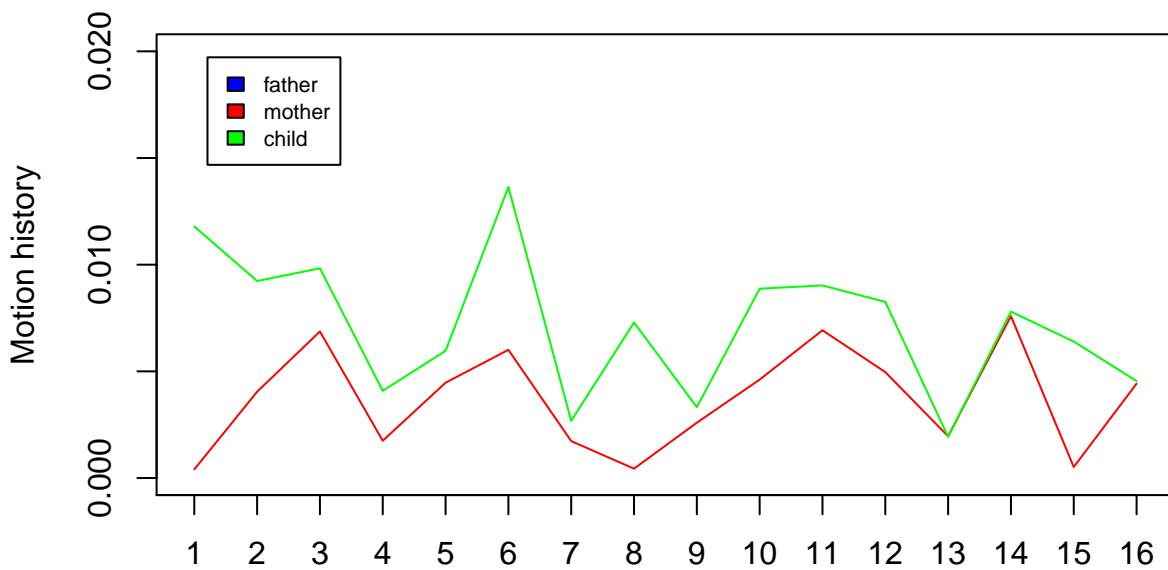
**Mean motion history (non overlapping minute intervals)
on BEAL036 video**



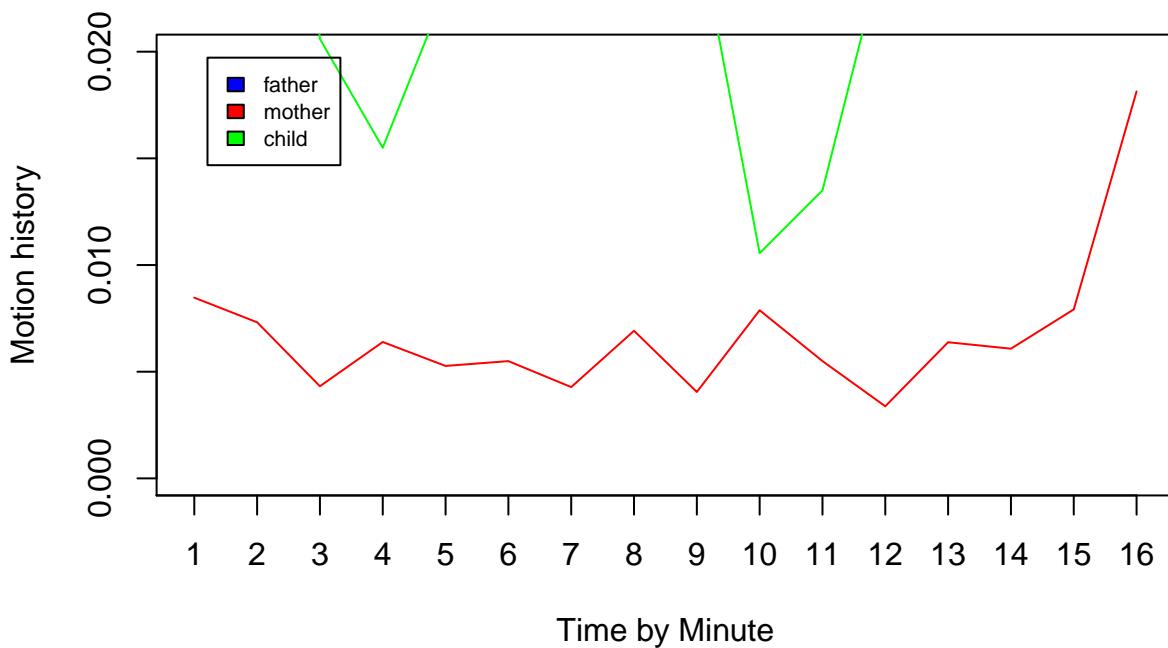
**Time by Minute
Mean motion history (non overlapping minute intervals)
on BEAM031 video**



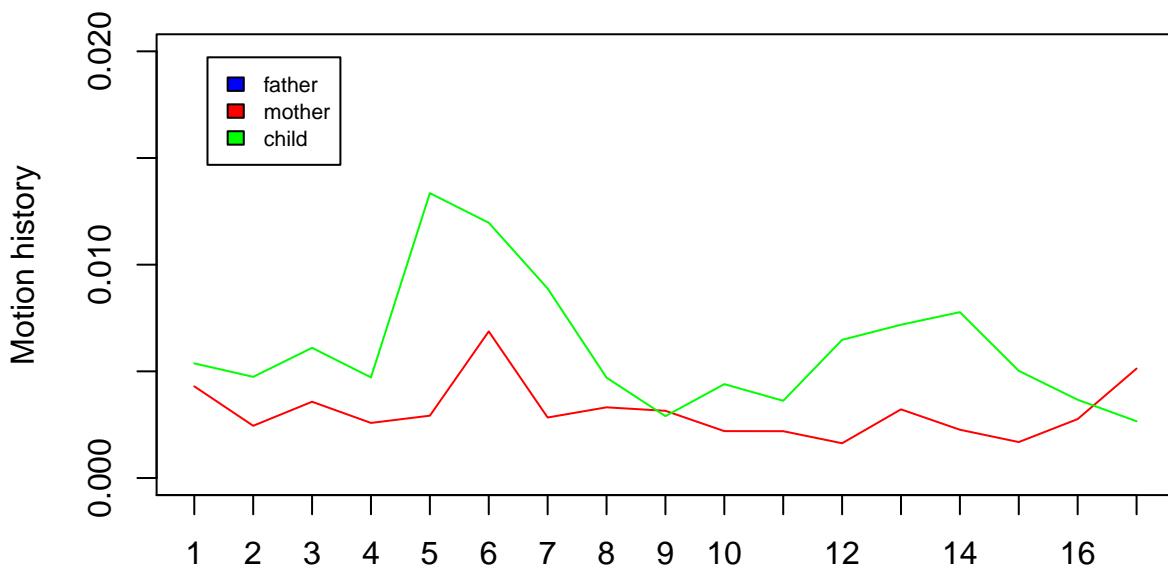
**Mean motion history (non overlapping minute intervals)
on BICA video**



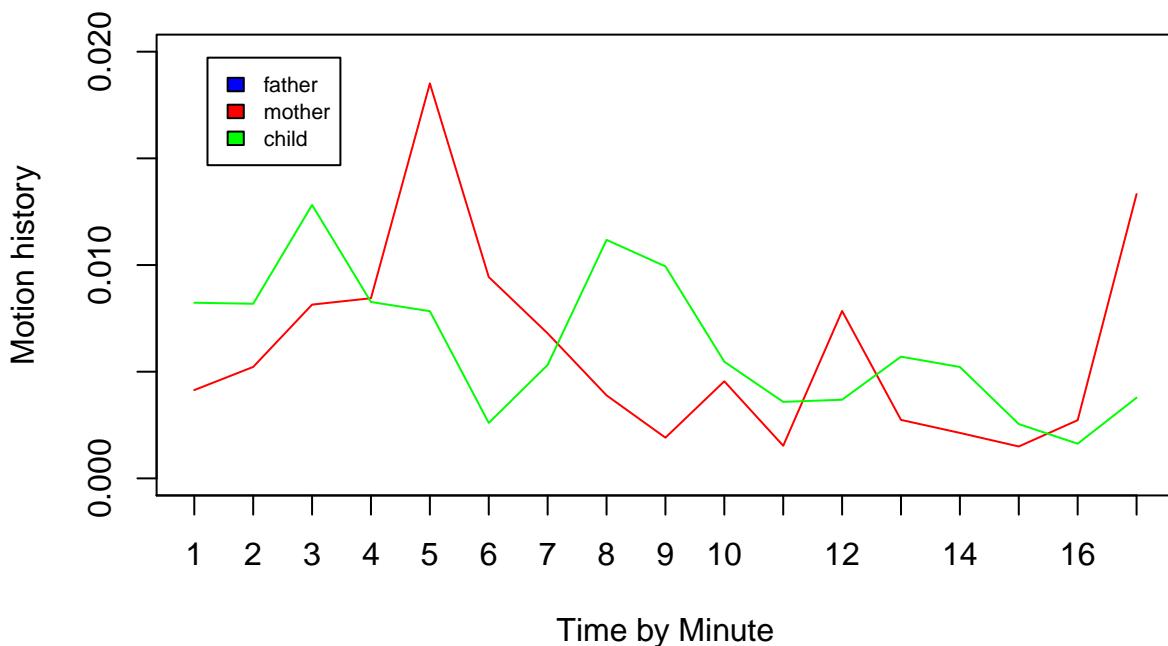
**Mean motion history (non overlapping minute intervals)
on BRLO041 video**



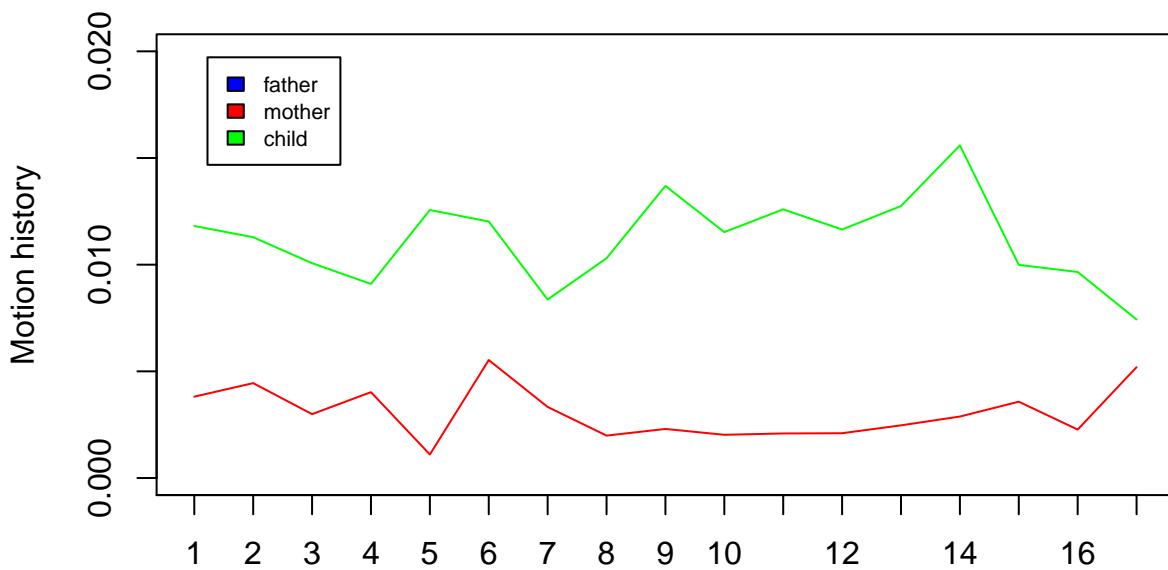
**Mean motion history (non overlapping minute intervals)
on COLO022 video**



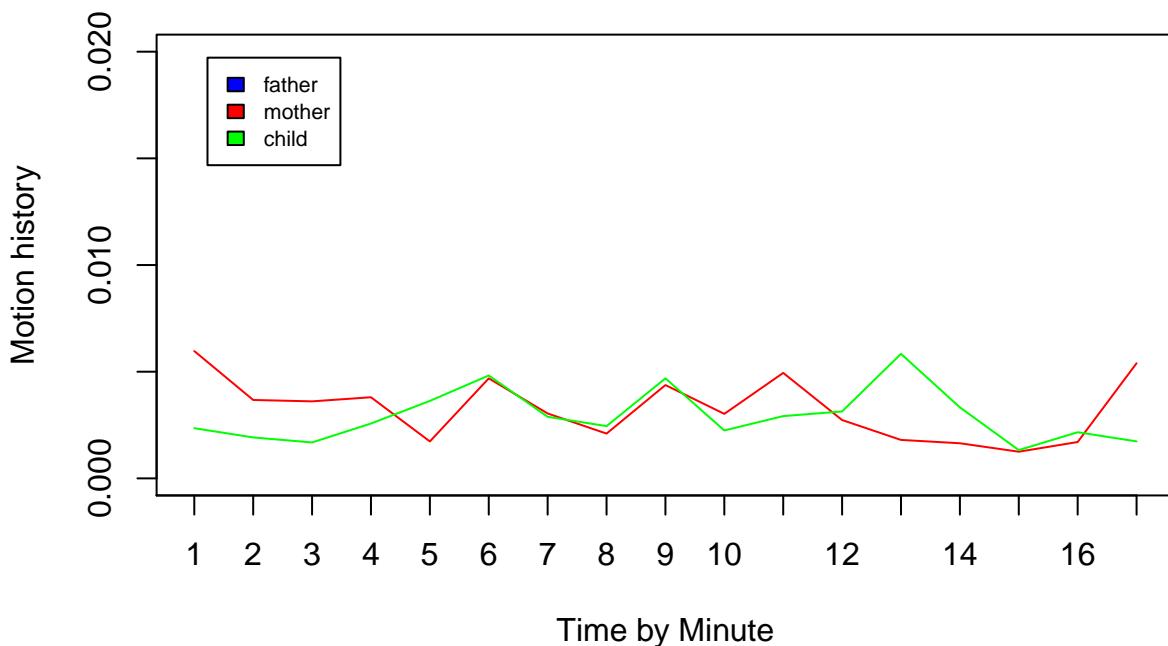
**Time by Minute
Mean motion history (non overlapping minute intervals)
on DIPE004 video**



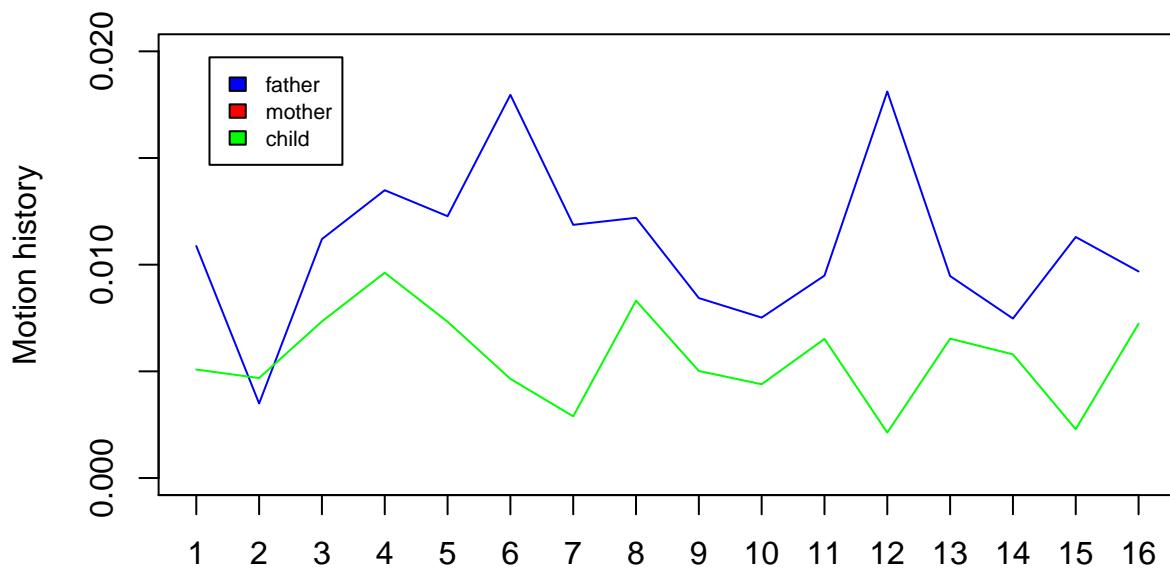
**Mean motion history (non overlapping minute intervals)
on DOMA video**



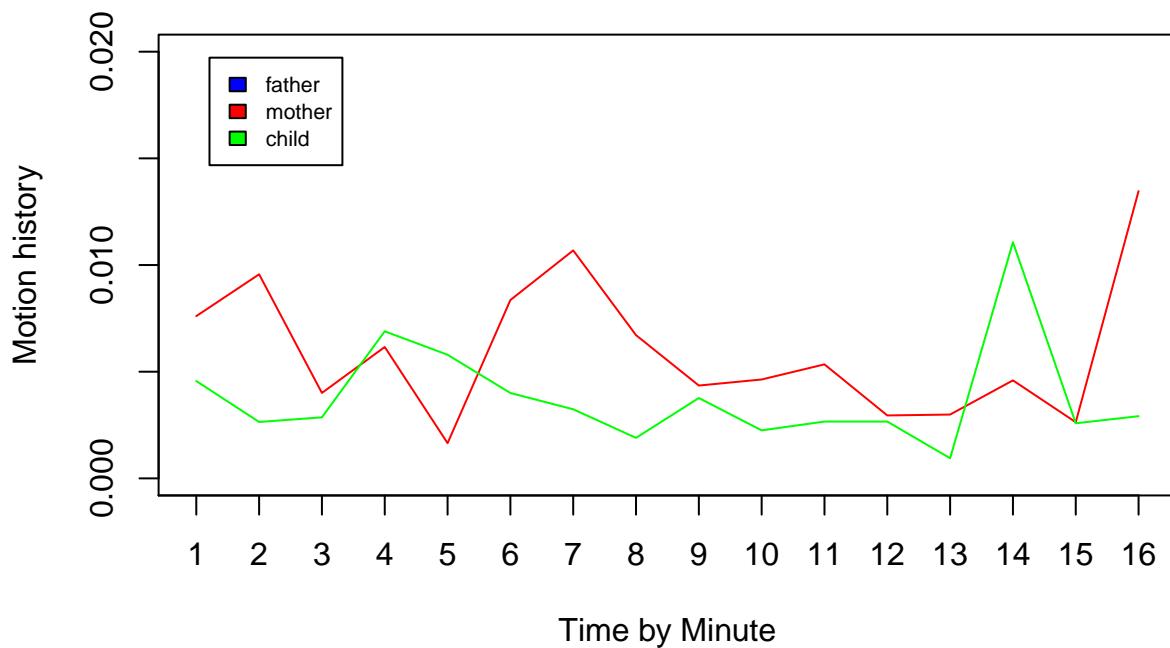
**Time by Minute
Mean motion history (non overlapping minute intervals)
on DRNE video**



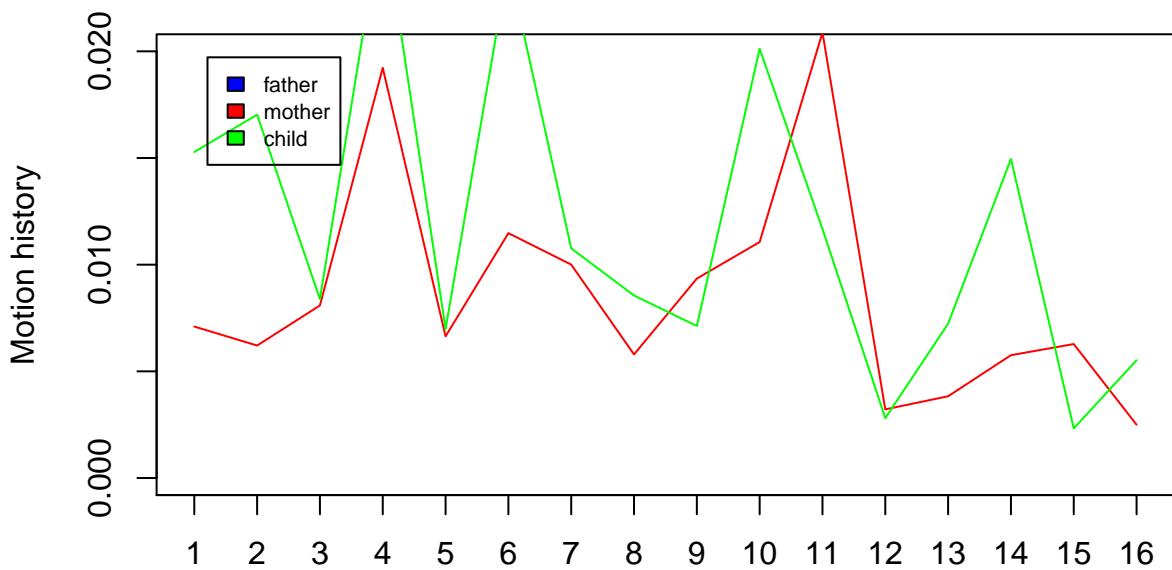
**Mean motion history (non overlapping minute intervals)
on FOMA057 video**



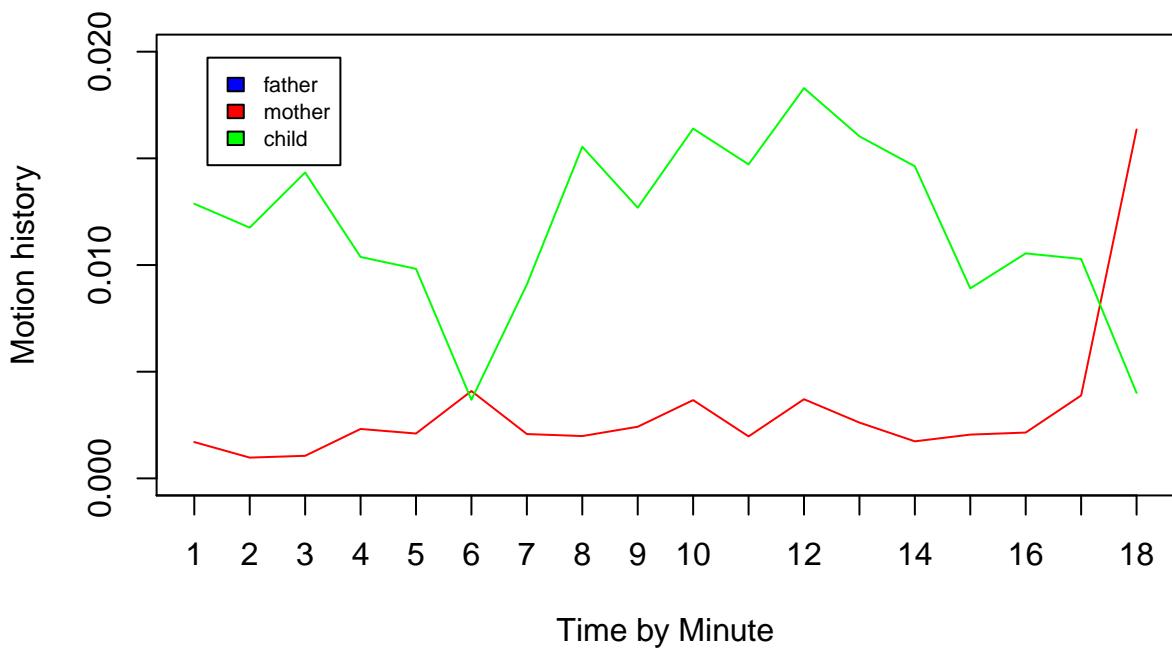
**Mean motion history (non overlapping minute intervals)
on GROP039 video**



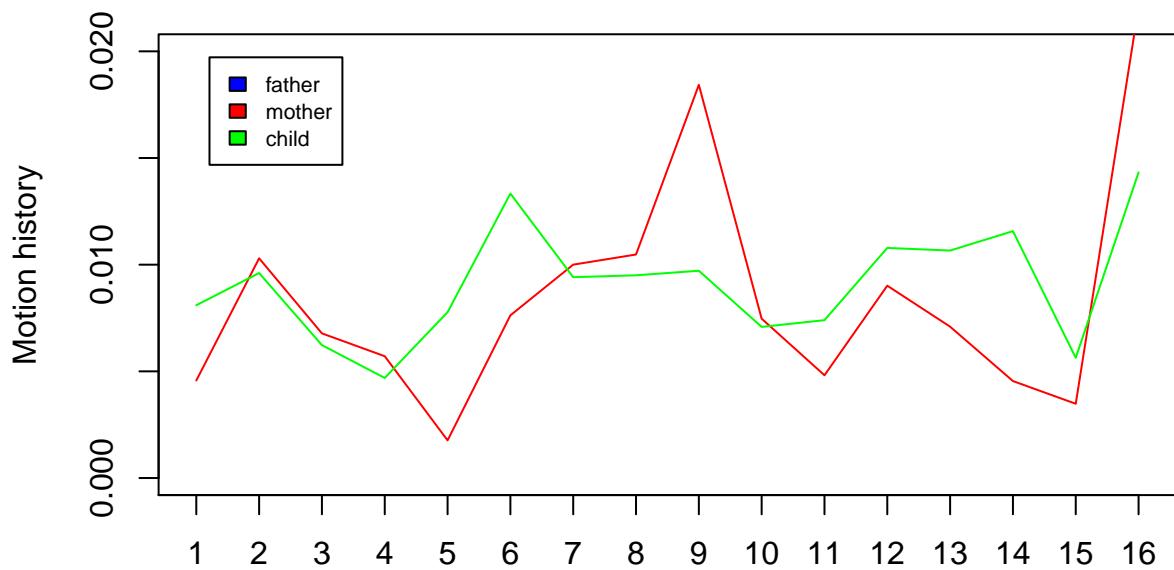
**Mean motion history (non overlapping minute intervals)
on HAJA052 video**



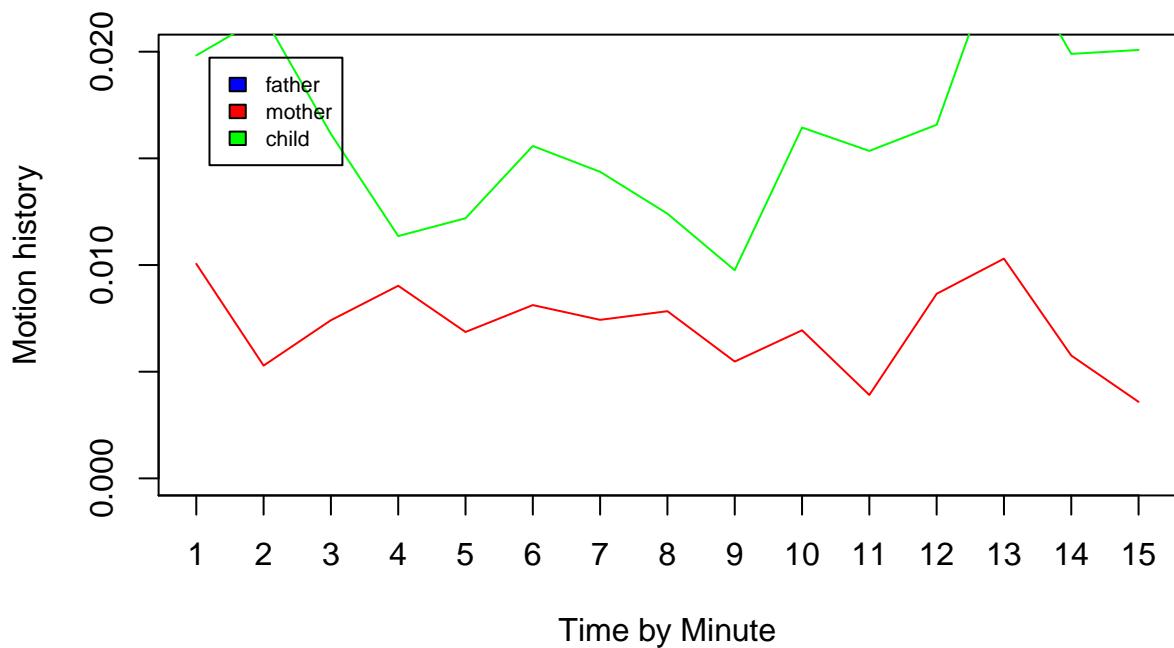
**Mean motion history (non overlapping minute intervals)
on HUMA058 video**



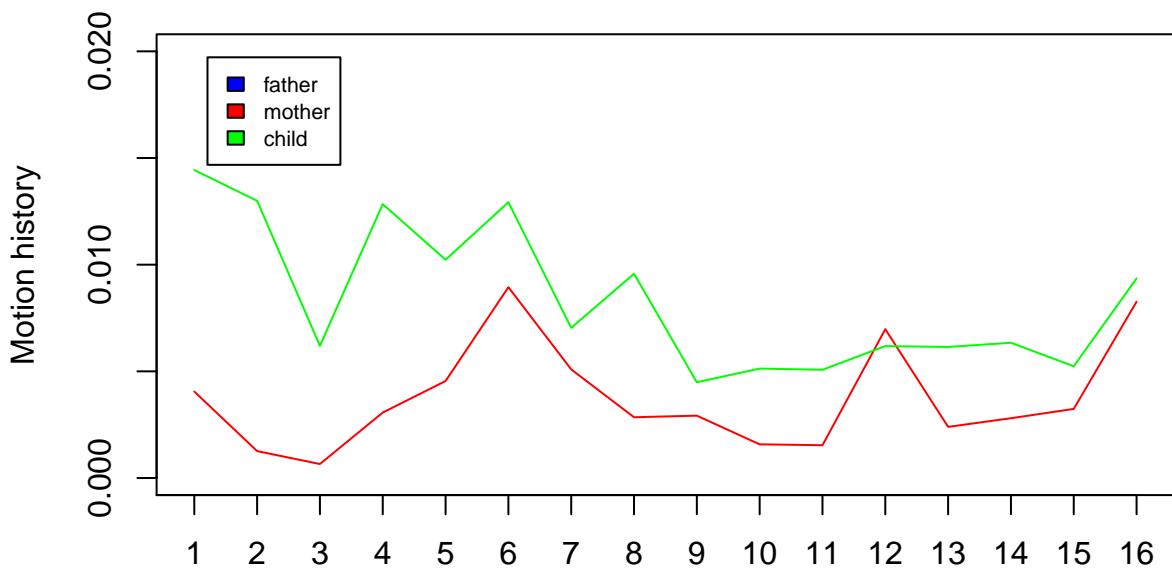
**Mean motion history (non overlapping minute intervals)
on JAEM046 video**



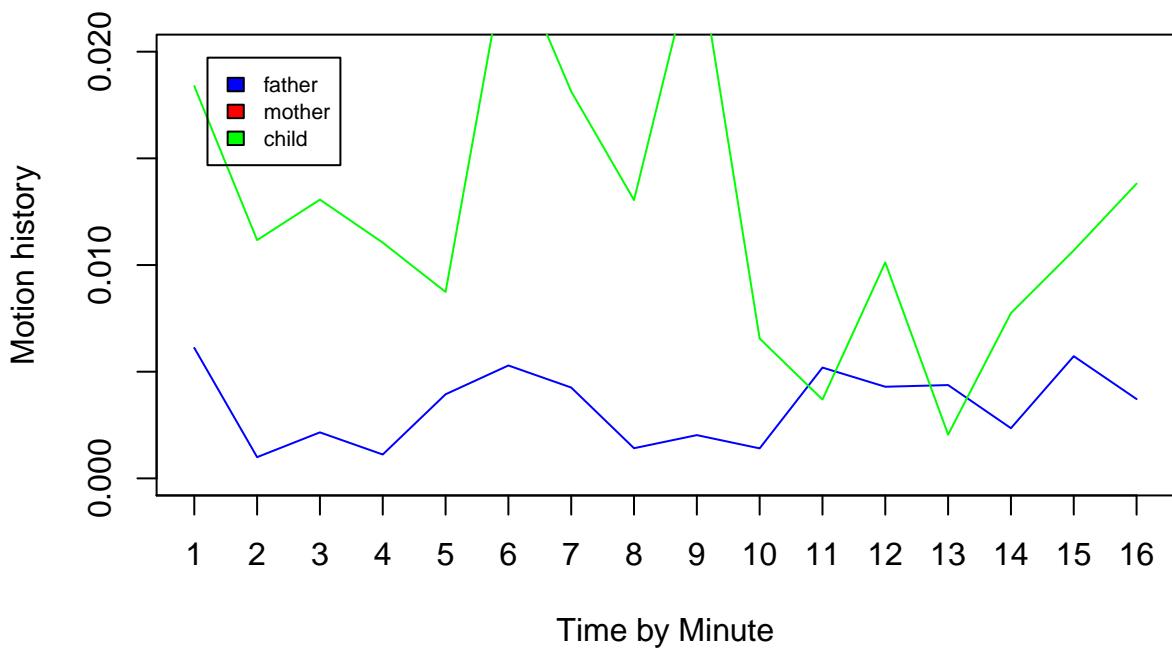
**Time by Minute
Mean motion history (non overlapping minute intervals)
on JEEO040 video**



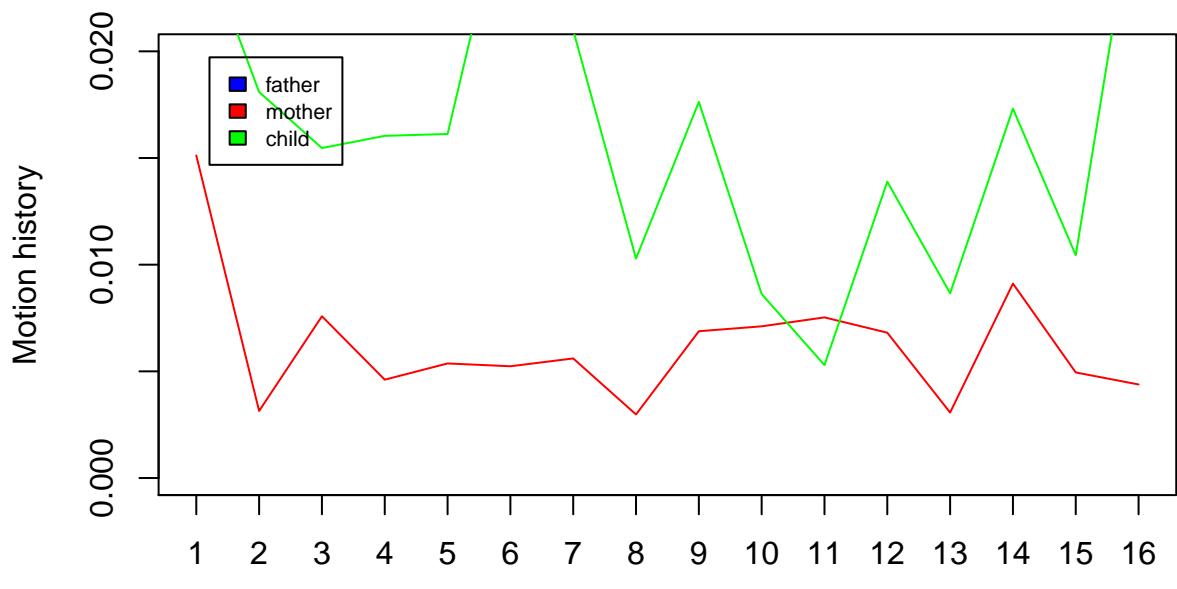
**Mean motion history (non overlapping minute intervals)
on JOCE014 video**



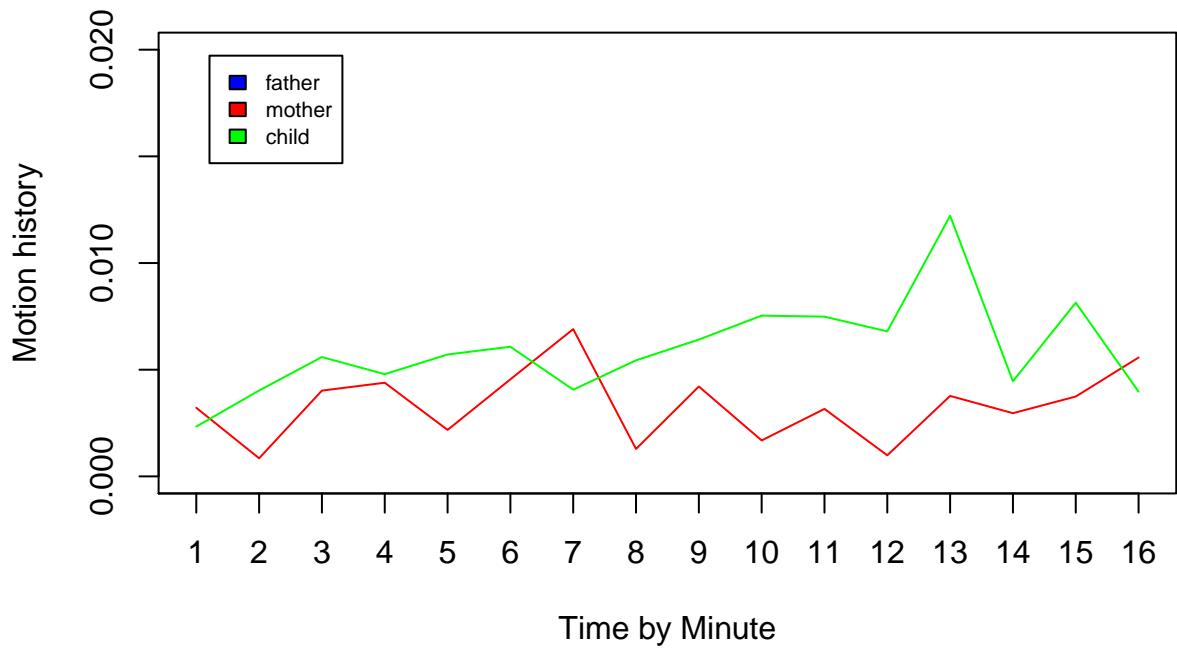
Time by Minute
**Mean motion history (non overlapping minute intervals)
on LACL video**



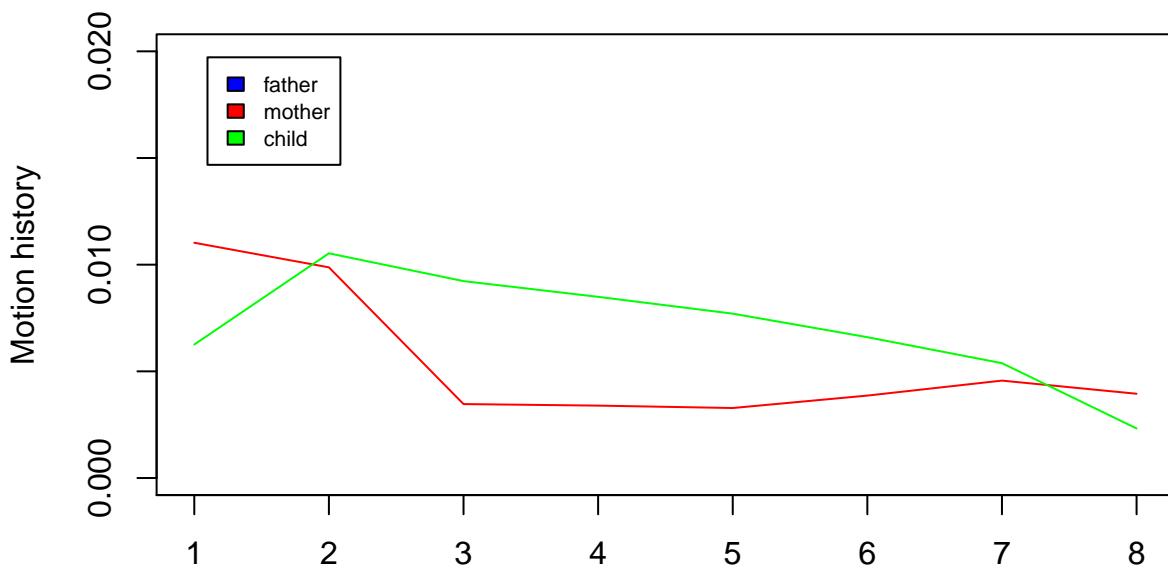
**Mean motion history (non overlapping minute intervals)
on MAEL048 video**



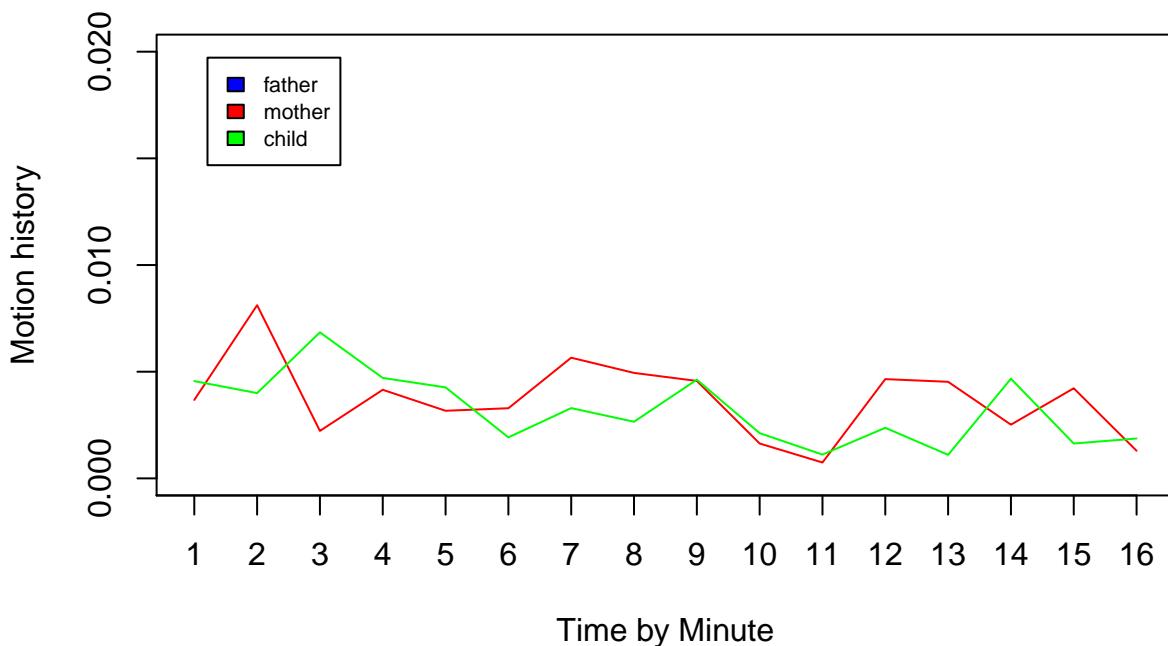
**Time by Minute
Mean motion history (non overlapping minute intervals)
on MAME20 video**



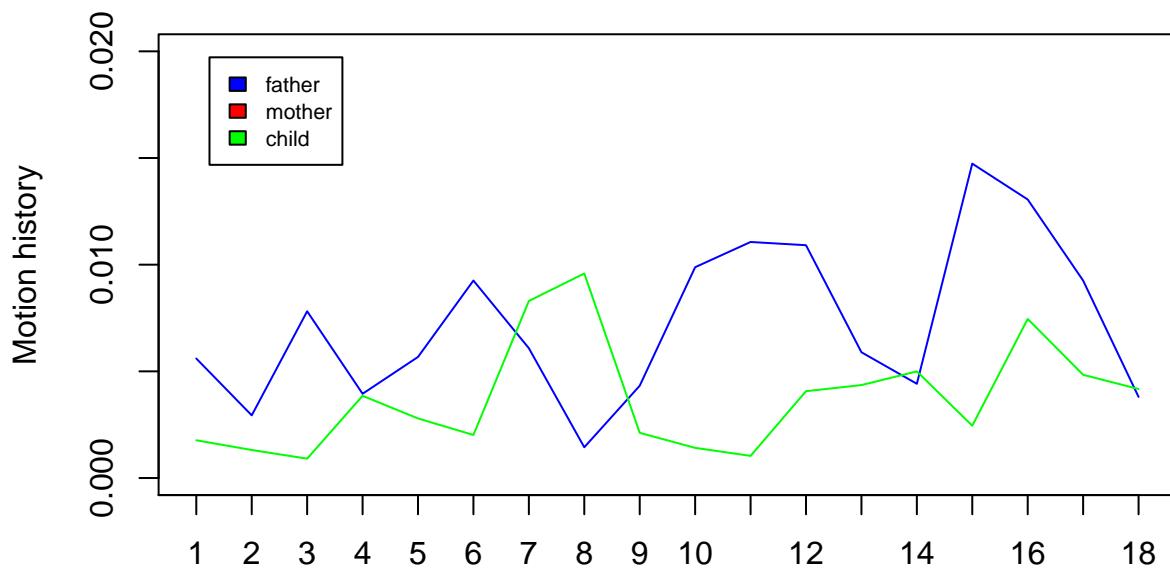
**Mean motion history (non overlapping minute intervals)
on MAPA029 video**



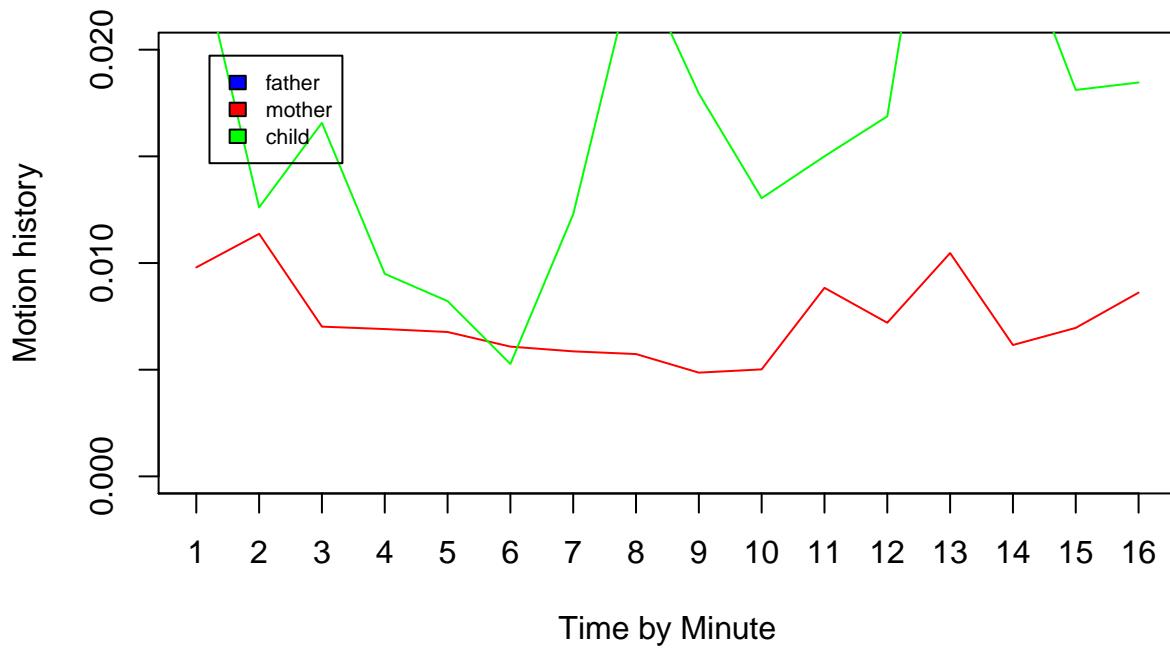
**Mean motion history (non overlapping minute intervals)
on MIPH043 video**



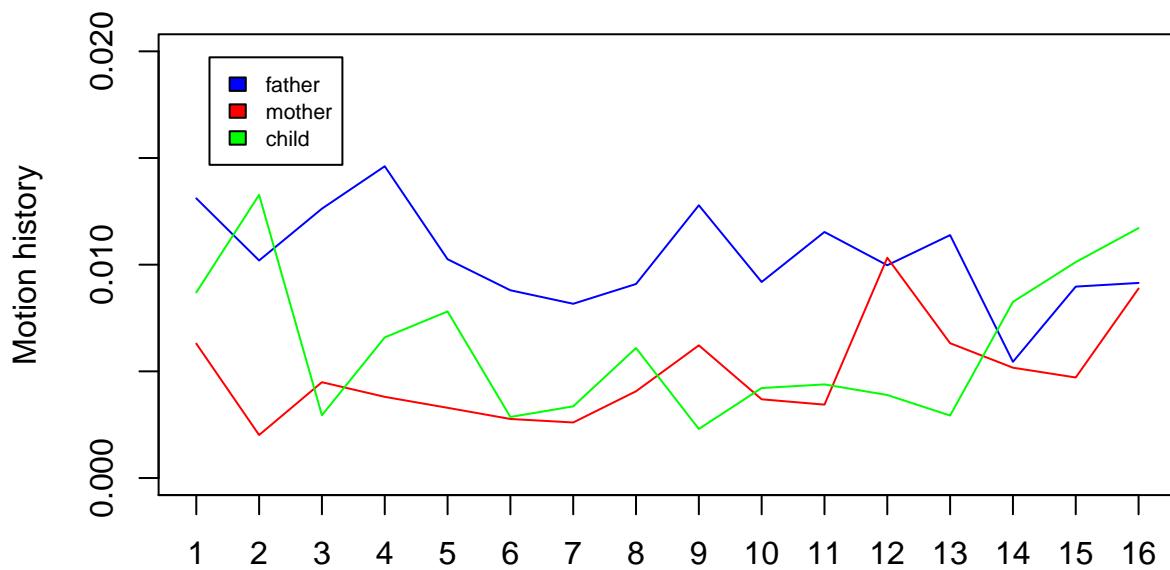
**Mean motion history (non overlapping minute intervals)
on MOSA065 video**



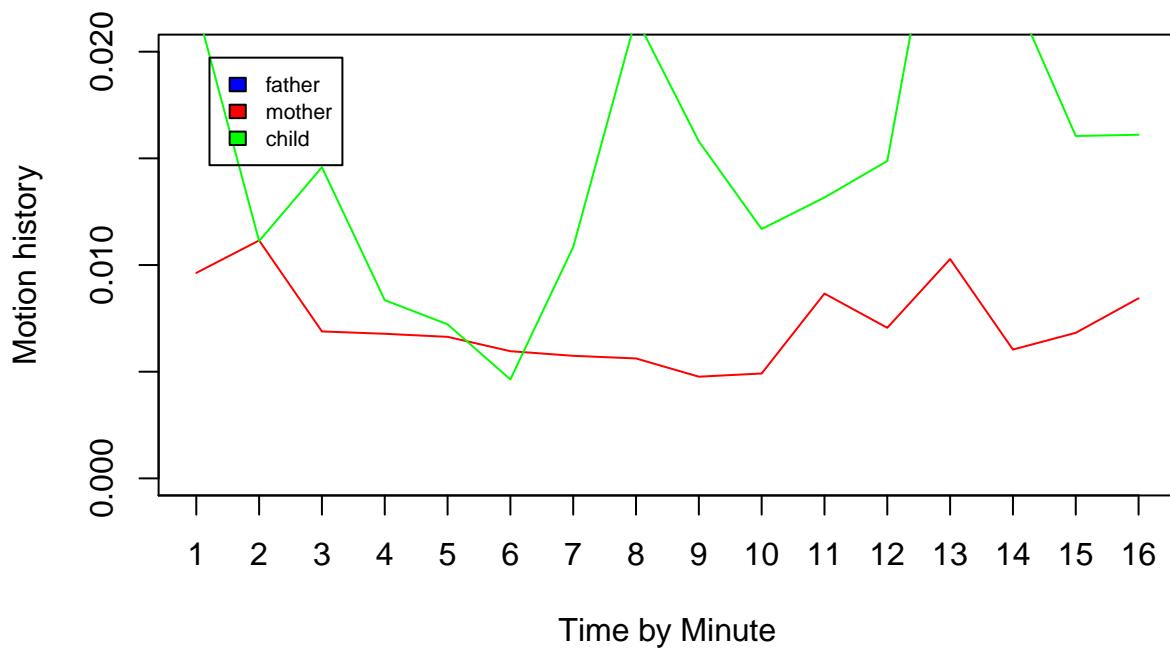
**Time by Minute
Mean motion history (non overlapping minute intervals)
on RAEM049 video**



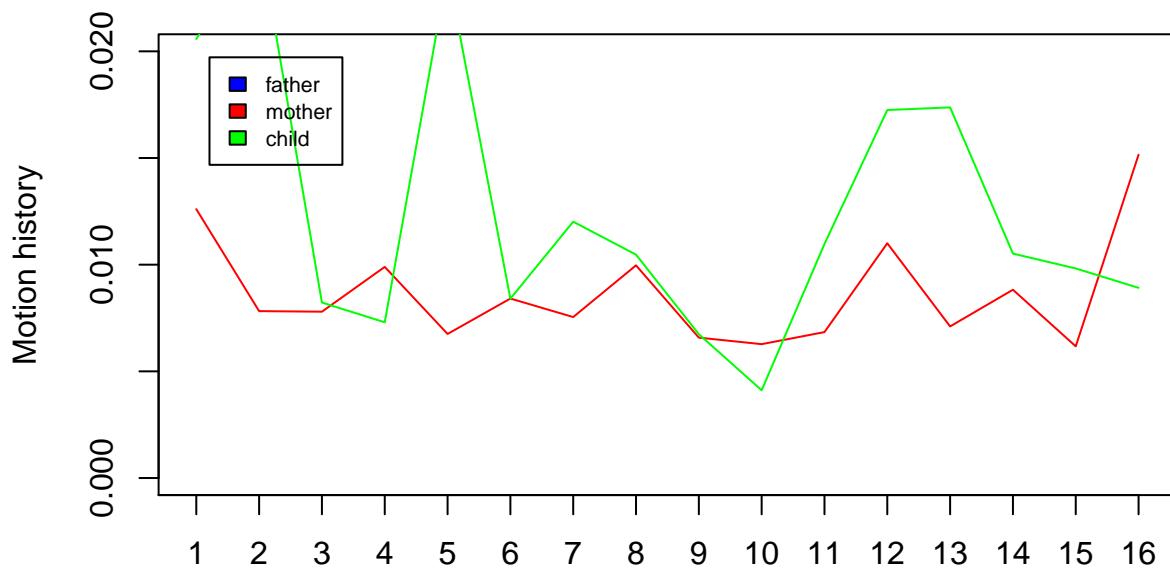
**Mean motion history (non overlapping minute intervals)
on RAKA008 video**



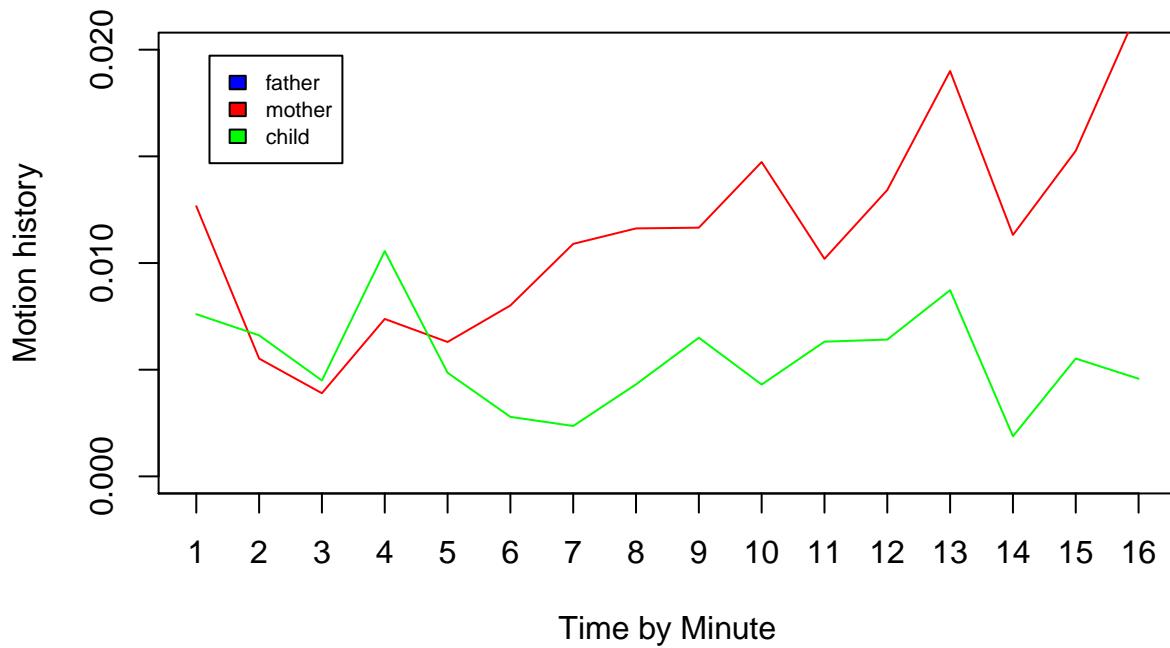
**Time by Minute
Mean motion history (non overlapping minute intervals)
on RIEM0 video**



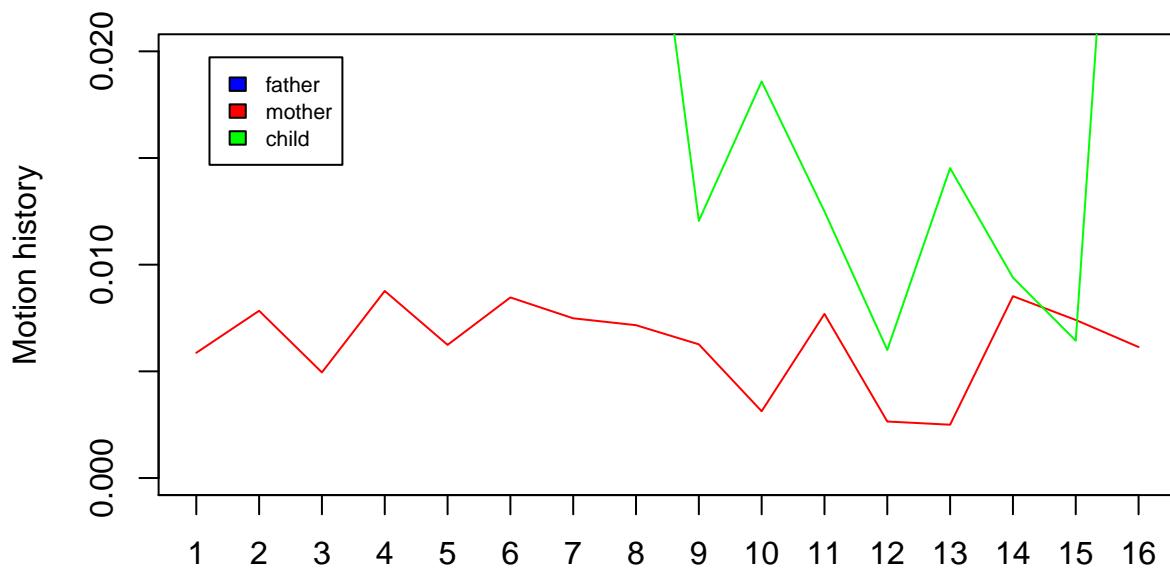
**Mean motion history (non overlapping minute intervals)
on SEEM035 video**



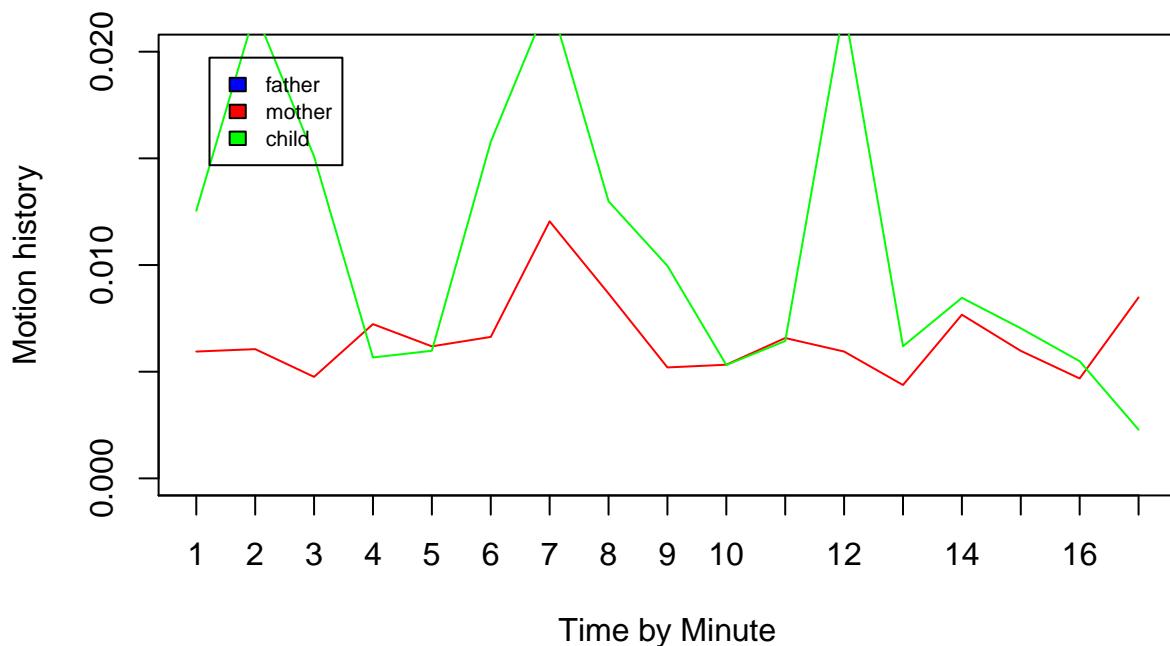
**Time by Minute
Mean motion history (non overlapping minute intervals)
on SHAN042 video**



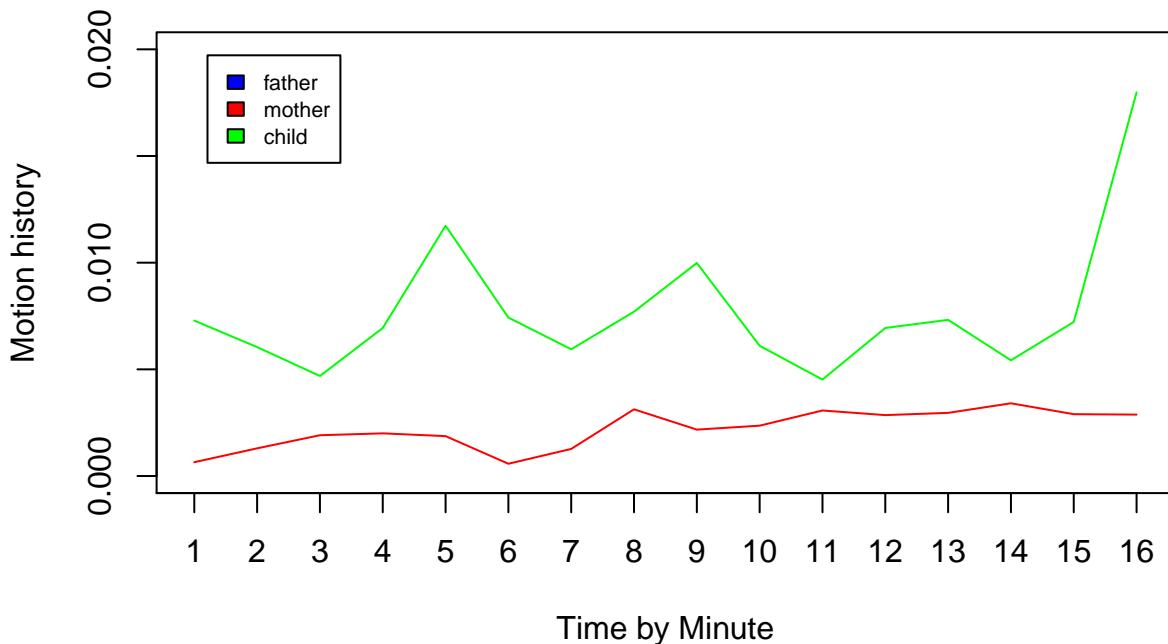
**Mean motion history (non overlapping minute intervals)
on SOGA061 video**



**Time by Minute
Mean motion history (non overlapping minute intervals)
on TIUG032 video**



Mean motion history (non overlapping minute intervals) on VINO video



Export log data in text files

```

videoIndex <- 1
# videoName est le nom de la video actuelle
for (videoName in unique(data$family)){
# Compute sliding interval for each participant
  print(paste("Computing滑动父亲", videoName))
  slidedFather <- SlidingInterval("logFather", videoIndex, 5, data)
  print(paste("Computing滑动母亲", videoName))
  slidedMother <- SlidingInterval("logMother", videoIndex, 5, data)
  print(paste("Computing滑动孩子", videoName))
  slidedChild <- SlidingInterval("logChild", videoIndex, 5, data)

  slidedVideo <- data.frame(
    slidedFather, slidedMother, slidedChild,
    "video"=rep(families[videoIndex], length(slidedFather)),
    frame_index = 1:length(slidedFather))

  write.csv(slidedVideo, paste("../Data/CSV/filtered/log/",videoName, ".log.slidedata.csv", sep=""))
  videoIndex <-(videoIndex+1)
}

```

SyncPy utilisation for creating synchrony dataframe

After extracting filtered motion history with mean on sliding interval (overlapping interval) of 5 frames

And after putting this data on a CSV file slideddata.csv

We import this data on python Script with panda module Call_S_Estimator.py

This script will compute the synchrony between each dyad of the interaction and of the whole group

It will return a csv file for each video SSIXXXX.csv with XXXX the name of the video (F1044C, F1044D1, etc) that we can import with R with

this following function

```
SSI <- data.frame()
for (file in SSIFilesList){
  SSIalone <- read.csv(file, )
  SSI<- rbind.fill(SSI, SSIalone)}
```

Description of SSI data frame

```
str(SSI)
```

```
## 'data.frame': 1807 obs. of 8 variables:
## $ X : int 0 1 2 3 4 5 6 7 8 9 ...
## $ Interval : chr "1" "2" "3" "4" ...
## $ Time_min : num 0 0.167 0.333 0.5 0.667 ...
## $ video : chr "A34" "A34" "A34" "A34" ...
## $ SSI_mo_ch : num 0.000087 0.002983 0.000295 0.005316 0.00278 ...
## $ SSI_fa_ch : num NA NA NA NA NA NA NA NA NA ...
## $ SSI_fa_mo : num NA NA NA NA NA NA NA NA NA ...
## $ SSI_fa_mo_ch: num NA NA NA NA NA NA NA NA NA ...
```

```
View(SSI)
unique(SSI$video)
```

```
## [1] "A34"      "37"       "41"       "48"       "206"      "1106"     "1606"
## [8] "BAJE059"   "BALE050"   "BALU062"   "BEAL036"   "BEAM031"   "BICA"      "BRL0041"
## [15] "COL0022"   "DIPE004"   "DOMA"      "DRNE"      "FOMA057"   "GROP039"   "HAJA052"
## [22] "HUMA058"   "JAEM046"   "JEE0040"   "JOCE014"   "LACL"      "MAEL048"   "MAME20"
## [29] "MAPA029"   "MIPH043"   "MOSA065"   "RAEM049"   "RAKA008"   "RIEMO"     "SEEM035"
## [36] "SHAN042"   "SOGA061"   "TIUG032"   "VINO"
```

```
SSI$video <- as.factor(SSI$video)
unique(SSI$video)
```

```

## [1] A34      37       41       48       206      1106     1606     BAJE059
## [9] BALE050  BALU062  BEAL036  BEAM031  BICA      BRL0041  COLO022  DIPE004
## [17] DOMA     DRNE     FOMA057  GROP039  HAJA052  HUMA058  JAEM046  JEE0040
## [25] JOCE014  LACL     MAEL048  MAME20    MAPA029  MIPH043  MOSA065  RAEM049
## [33] RAKA008  RIEMO    SEEM035  SHAN042  SOGA061  TIUG032  VINO
## 39 Levels: 1106 1606 206 37 41 48 A34 BAJE059 BALE050 BALU062 ... VINO

```

```
#SSI <- SSI[-which(SSI$video=="") , ]
```

Synchrony scores for each dyad, triad and for the whole group

```

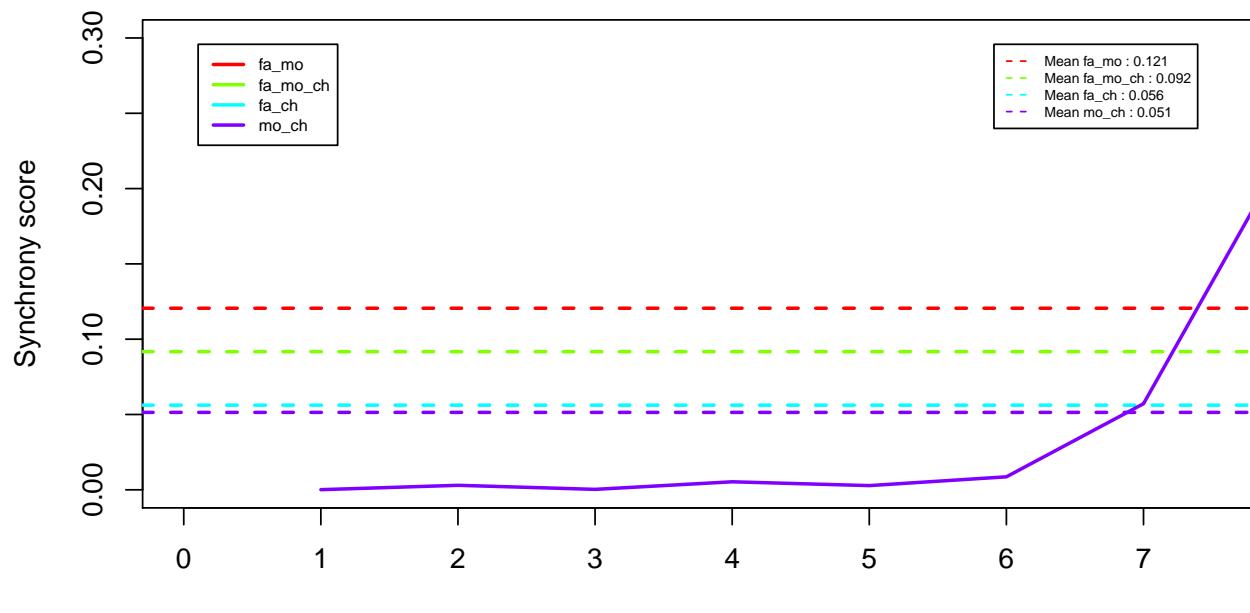
for (i in unique(SSI$video))
  {par(mar=c(4,4,4,3), mfrow=c(1,1))
   plot(SSI[which(SSI$video==i),]$Time_min,
        SSI[which(SSI$video==i),]$SSI_fa_mo,
        type="l", ylim=c(0, 0.3), col=rainbow(4)[1],
        main=paste("Synchrony scores for each dyad and for \n the whole group in", i, "video"),
        xlab = "Time (minute)", ylab="Synchrony score", lwd=2,
        xaxp=c(0,length(SSI$Time_min), length(SSI$Time_min)))
   abline(h=mean(SSI$SSI_fa_mo, na.rm=TRUE), col=rainbow(4)[1], lwd=2, lty=2)
   lines(SSI[which(SSI$video==i),]$SSI_fa_mo_ch, col=rainbow(4)[2], lwd=2)
   abline(h= mean(SSI$SSI_fa_mo_ch, na.rm=TRUE), col=rainbow(4)[2], lwd=2, lty=2)
   lines(SSI[which(SSI$video==i),]$SSI_fa_ch, col=rainbow(4)[3], lwd=2)
   abline(h= mean(SSI$SSI_fa_ch, na.rm=TRUE), col=rainbow(4)[3], lwd=2, lty=2)
   lines(SSI[which(SSI$video==i),]$SSI_mo_ch, col=rainbow(4)[4], lwd=2)
   abline(h= mean(SSI$SSI_mo_ch, na.rm=TRUE), col=rainbow(4)[4], lwd=2, lty=2)

  legend("topleft", inset=.05, c("fa_mo", "fa_mo_ch", "fa_ch",
  "mo_ch"),
  col=rainbow(4), cex=0.6, lwd=2)

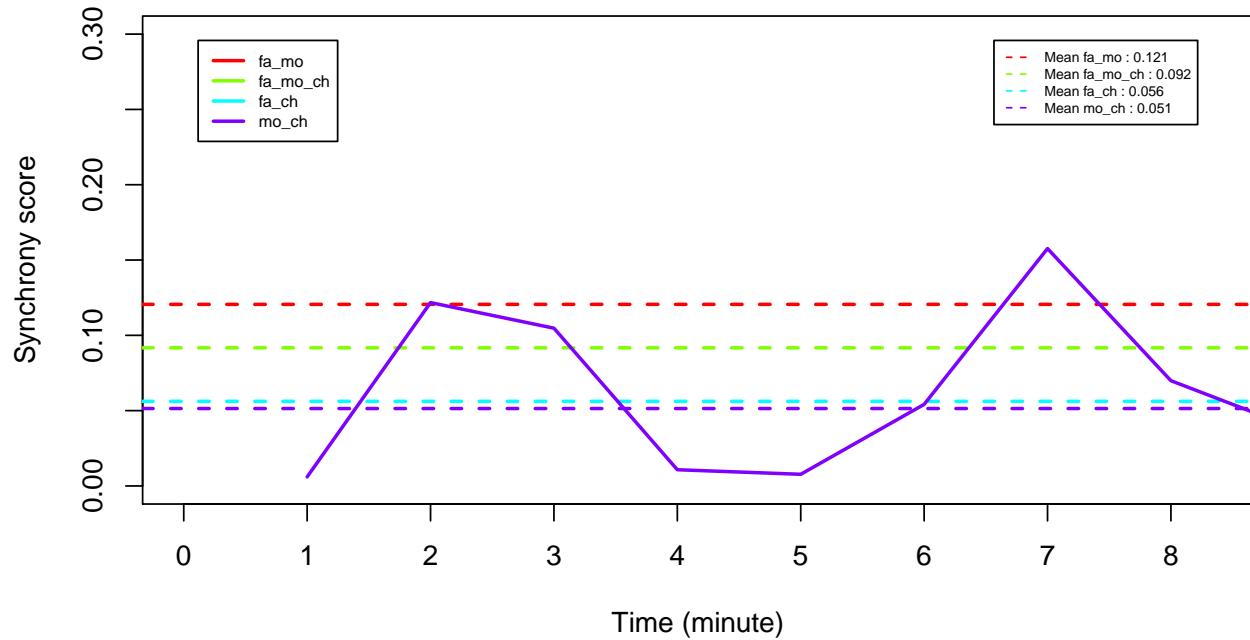
  legend ("topright", inset=.05, c(
    paste ("Mean fa_mo :", round(mean(SSI$SSI_fa_mo, na.rm=TRUE),3)),
    paste ("Mean fa_mo_ch :", round(mean(SSI$SSI_fa_mo_ch,na.rm=TRUE),3)),
    paste ("Mean fa_ch :", round(mean(SSI$SSI_fa_ch, na.rm=TRUE),3)),
    paste ("Mean mo_ch :", round(mean(SSI$SSI_mo_ch,na.rm=TRUE),3))),
  col=rainbow(4), cex=0.5, lty=2, lwd=1)}

```

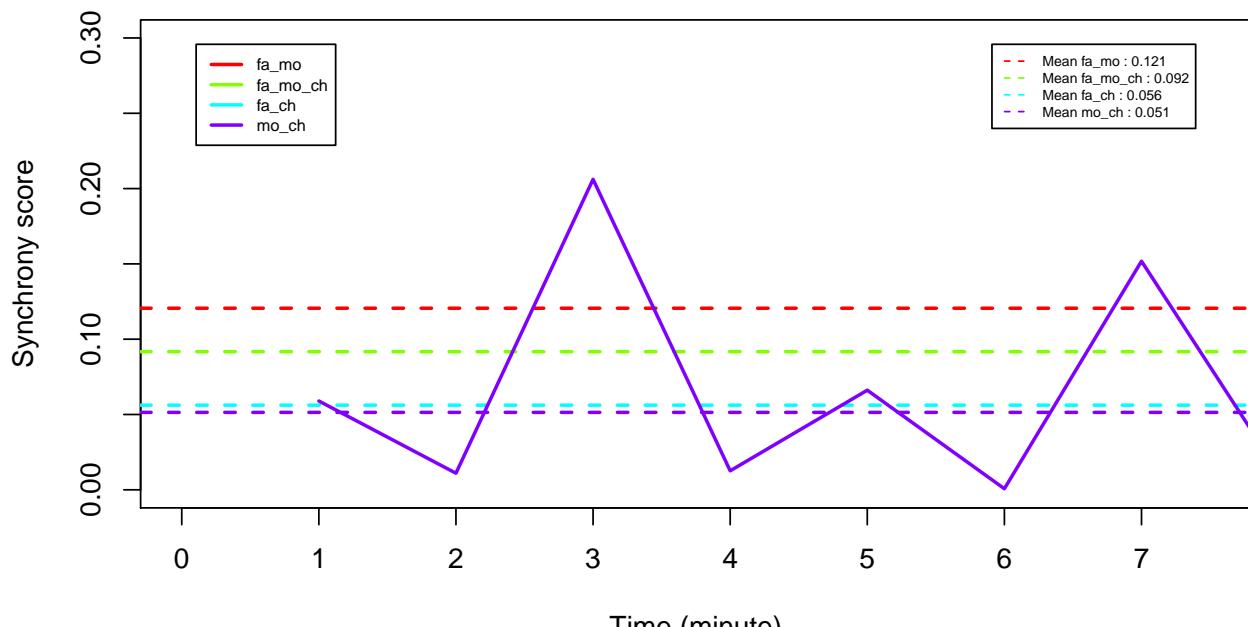
Synchrony scores for each dyad and for the whole group in A34 video



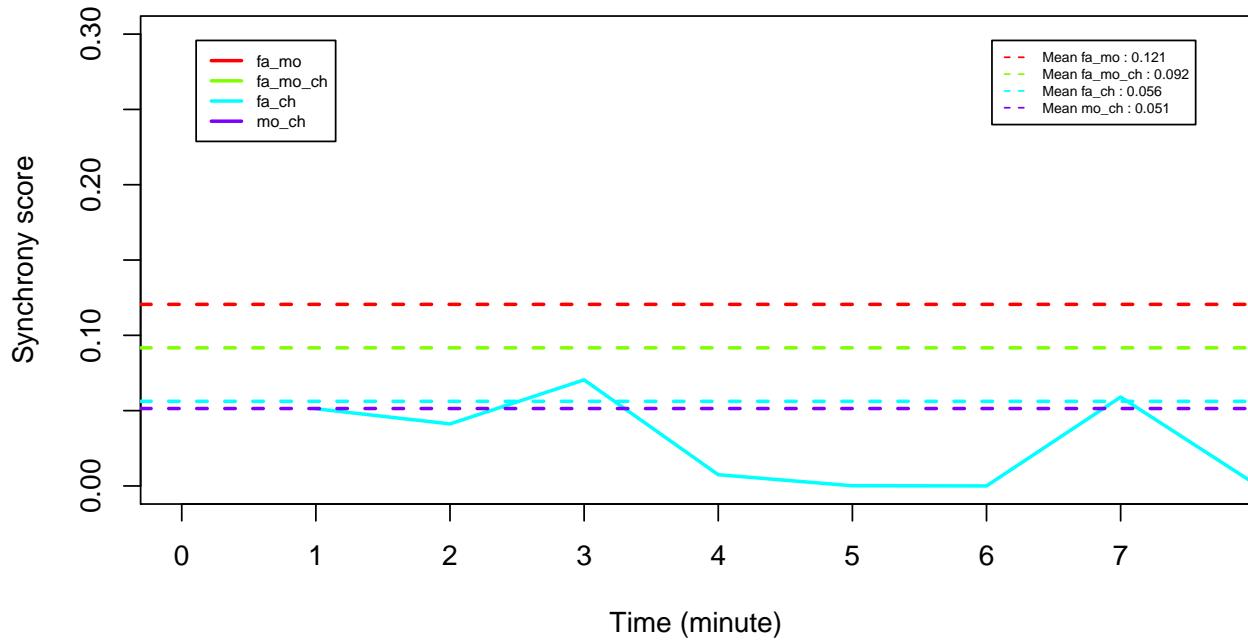
Synchrony scores for each dyad and for the whole group in 37 video



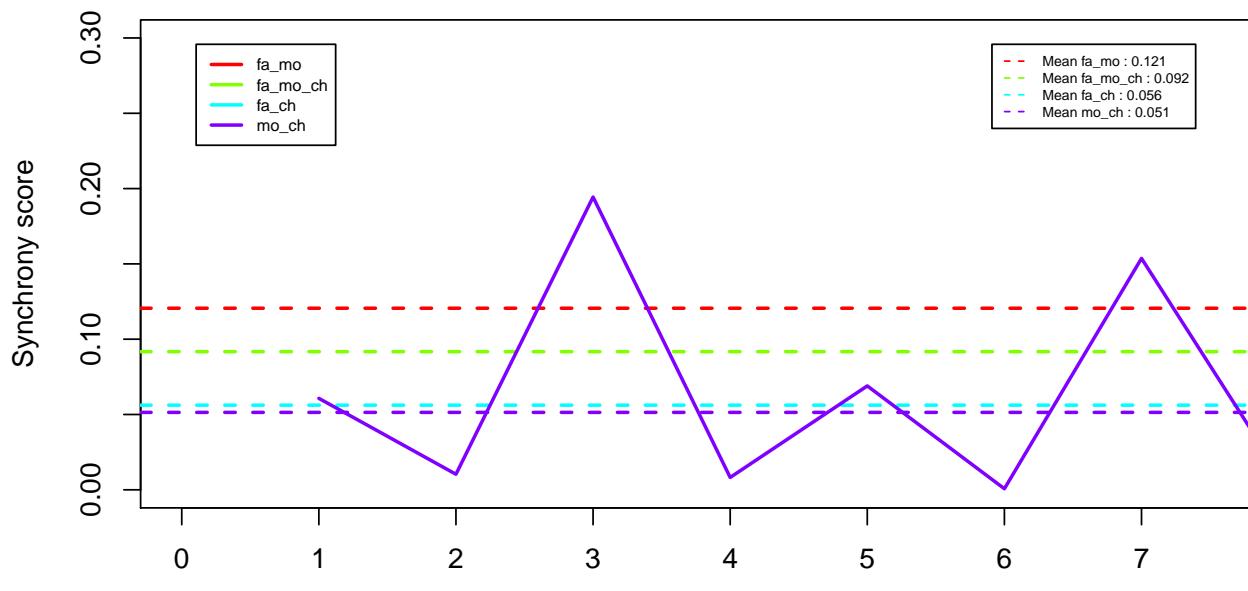
Synchrony scores for each dyad and for the whole group in 41 video



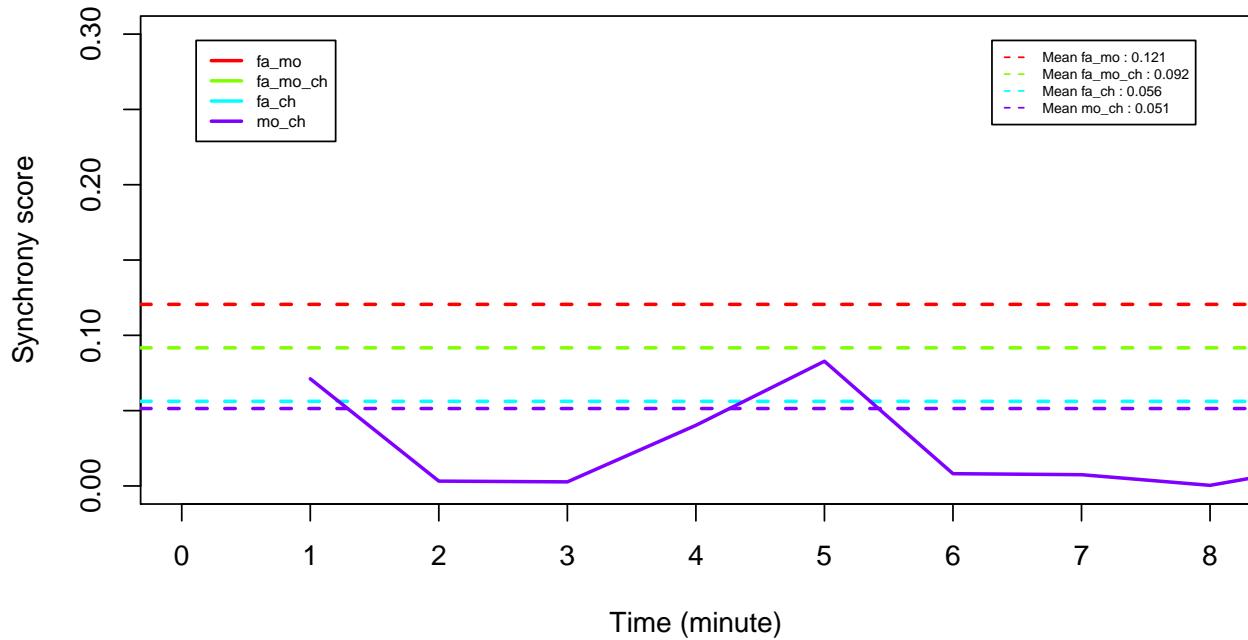
Synchrony scores for each dyad and for the whole group in 48 video



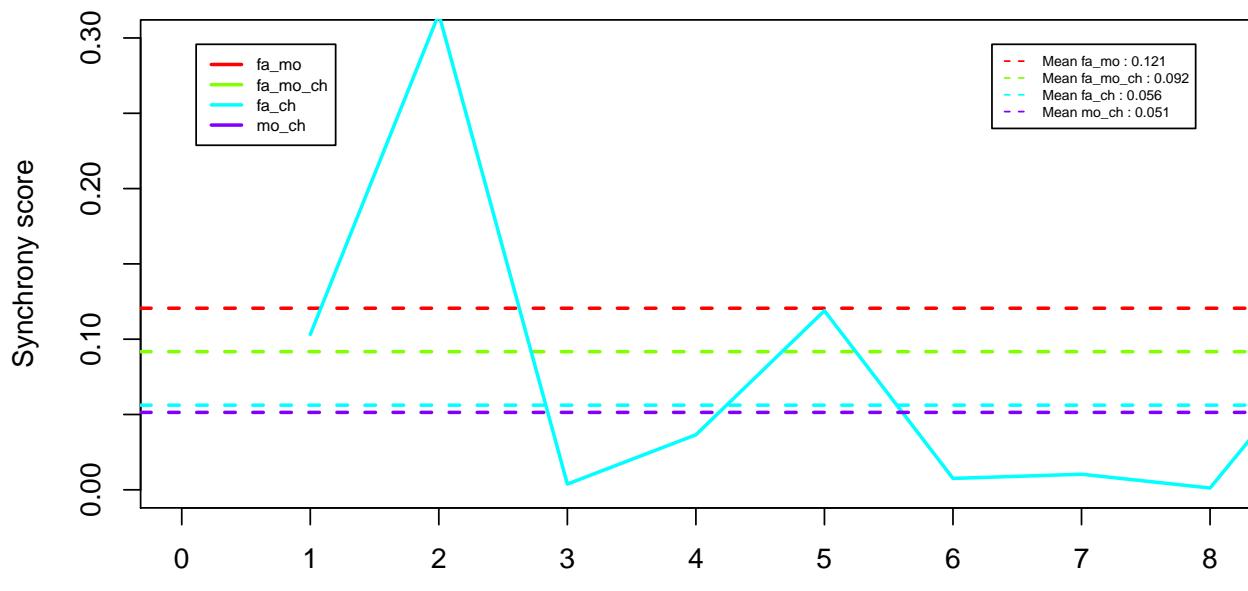
**Synchrony scores for each dyad and for
the whole group in 206 video**



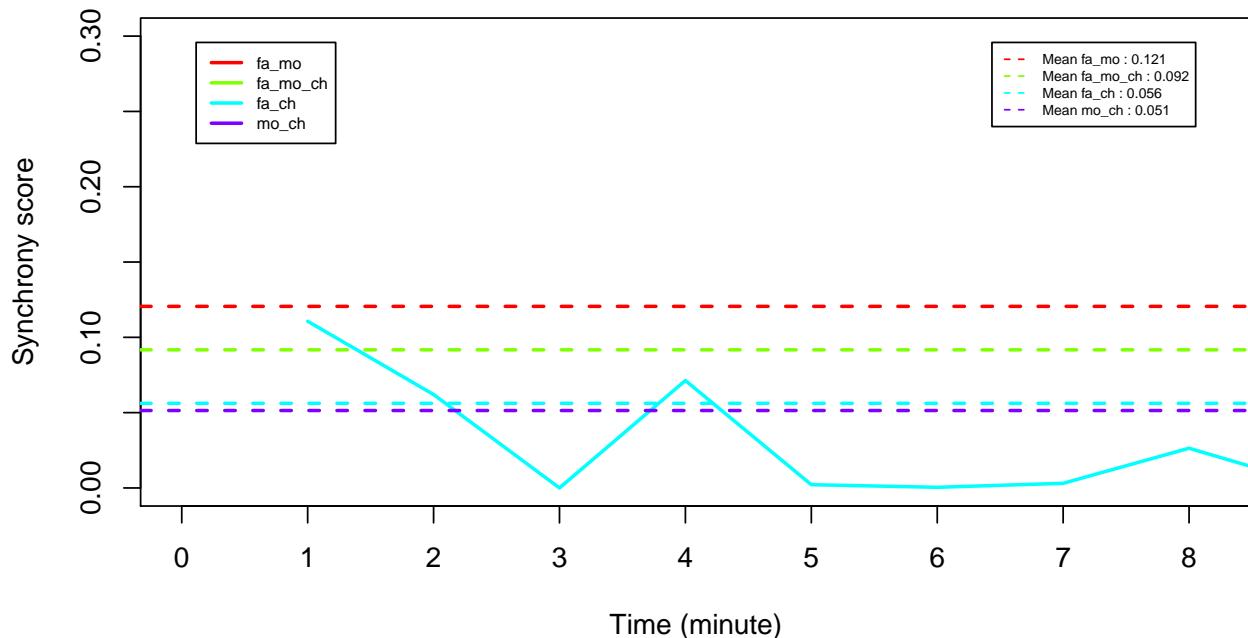
**Synchrony scores for each dyad and for
the whole group in 1106 video**



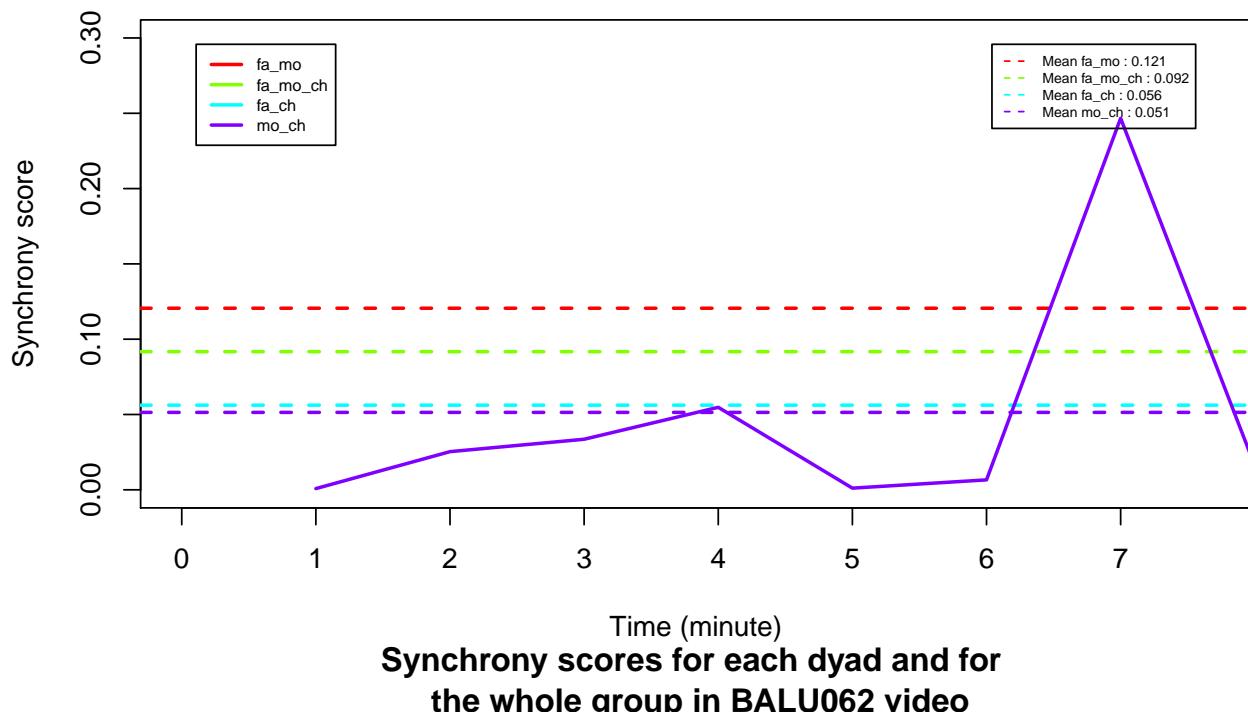
**Synchrony scores for each dyad and for
the whole group in 1606 video**



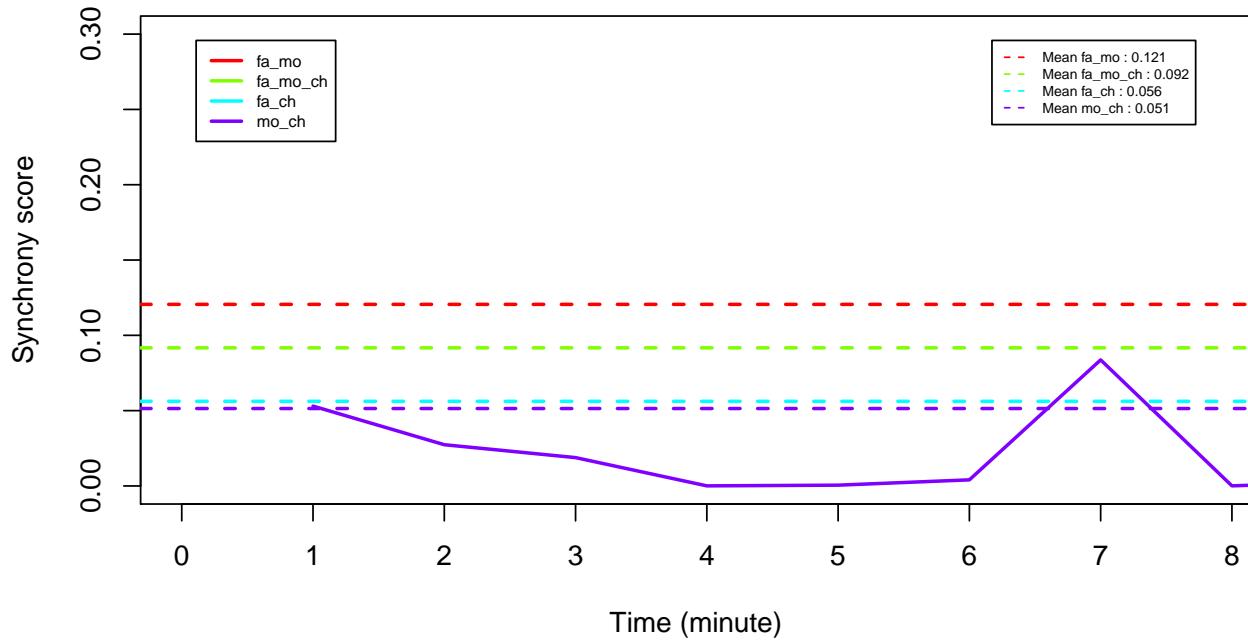
**Synchrony scores for each dyad and for
the whole group in BAJE059 video**



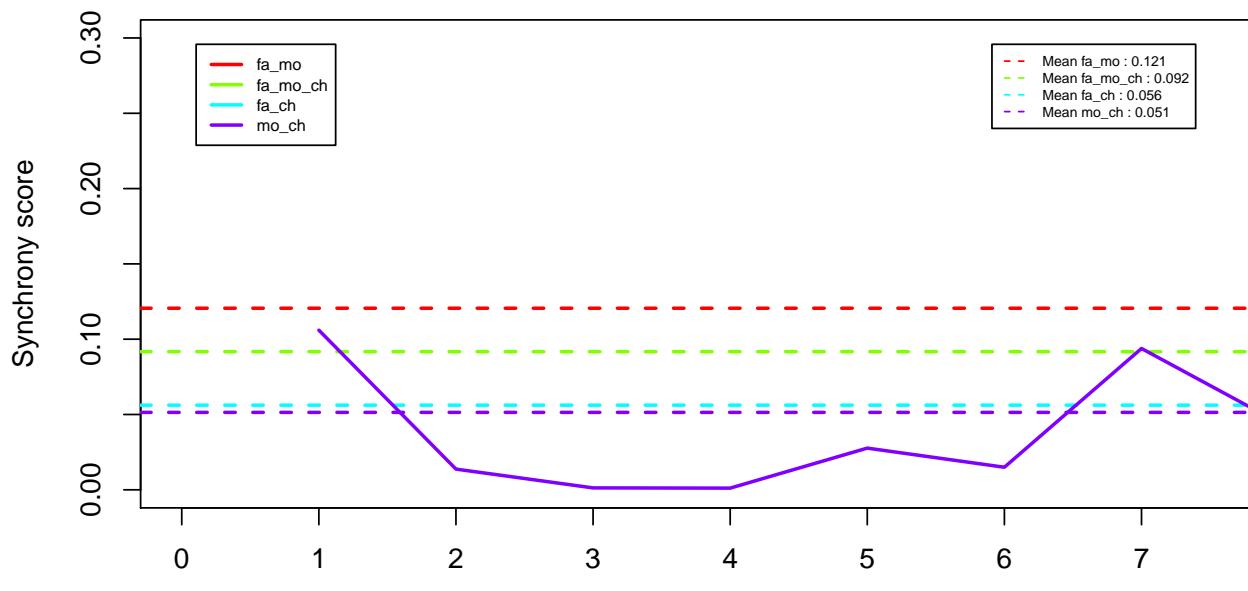
**Synchrony scores for each dyad and for
the whole group in BALE050 video**



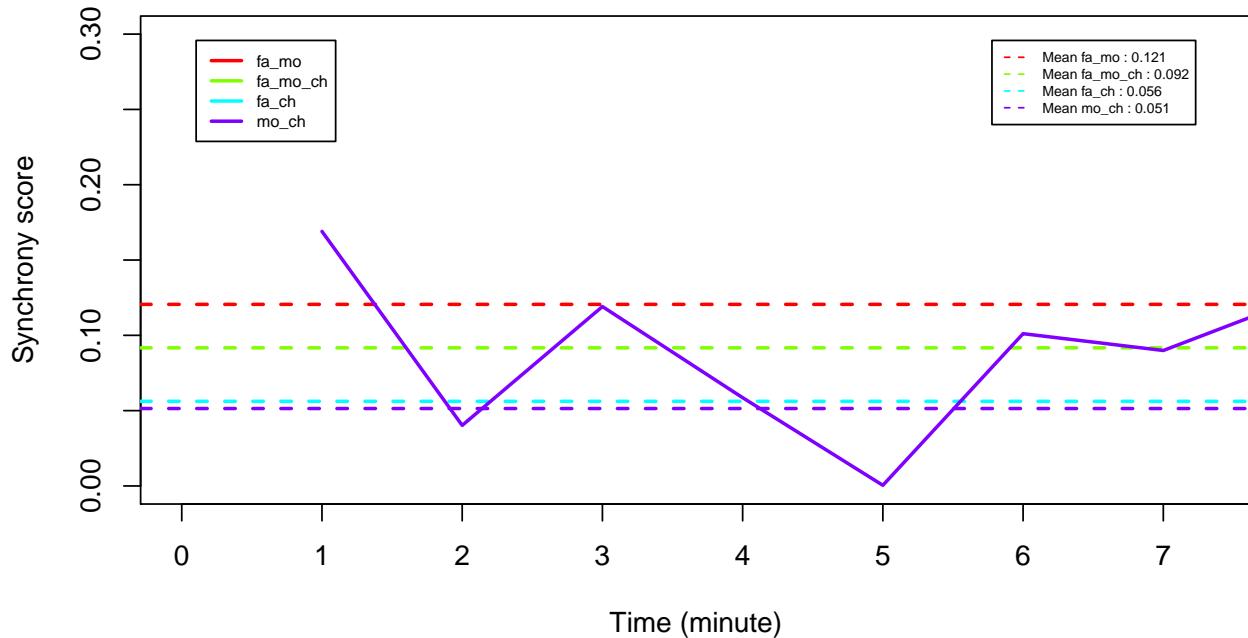
**Synchrony scores for each dyad and for
the whole group in BALU062 video**



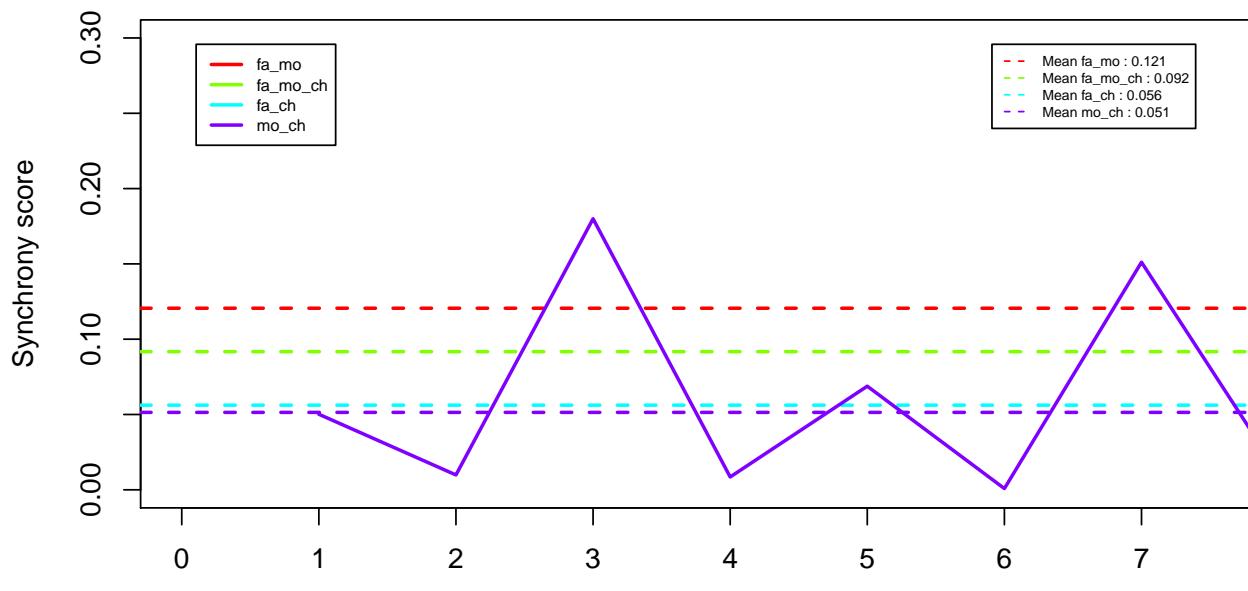
**Synchrony scores for each dyad and for
the whole group in BEAL036 video**



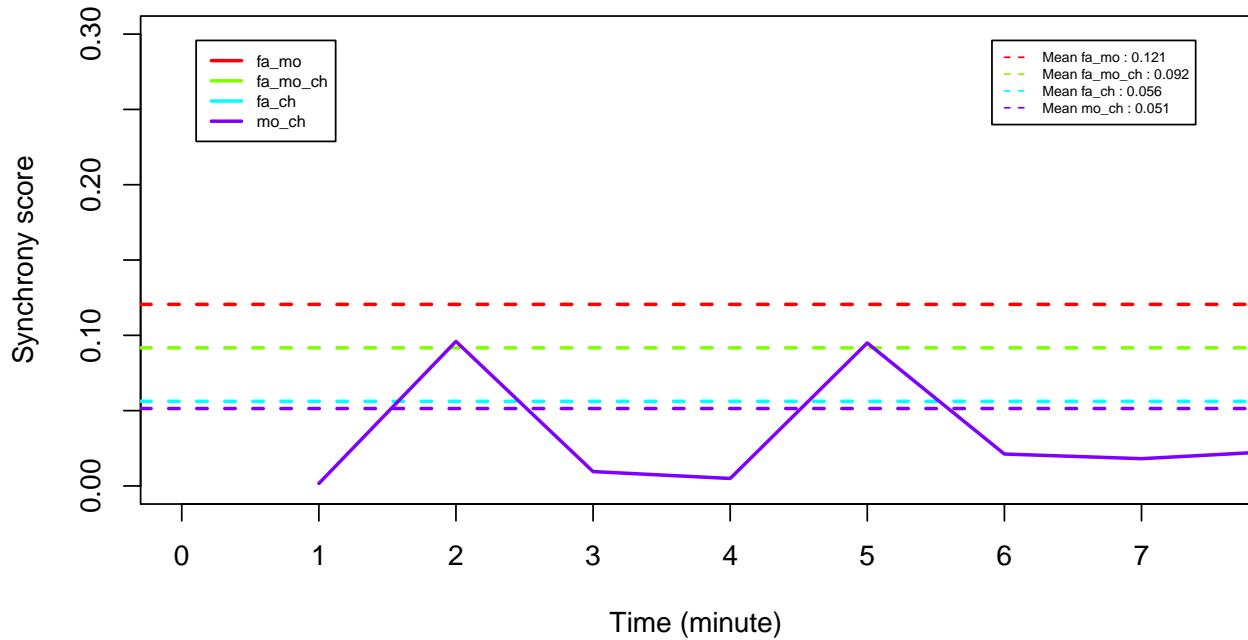
**Synchrony scores for each dyad and for
the whole group in BEAM031 video**



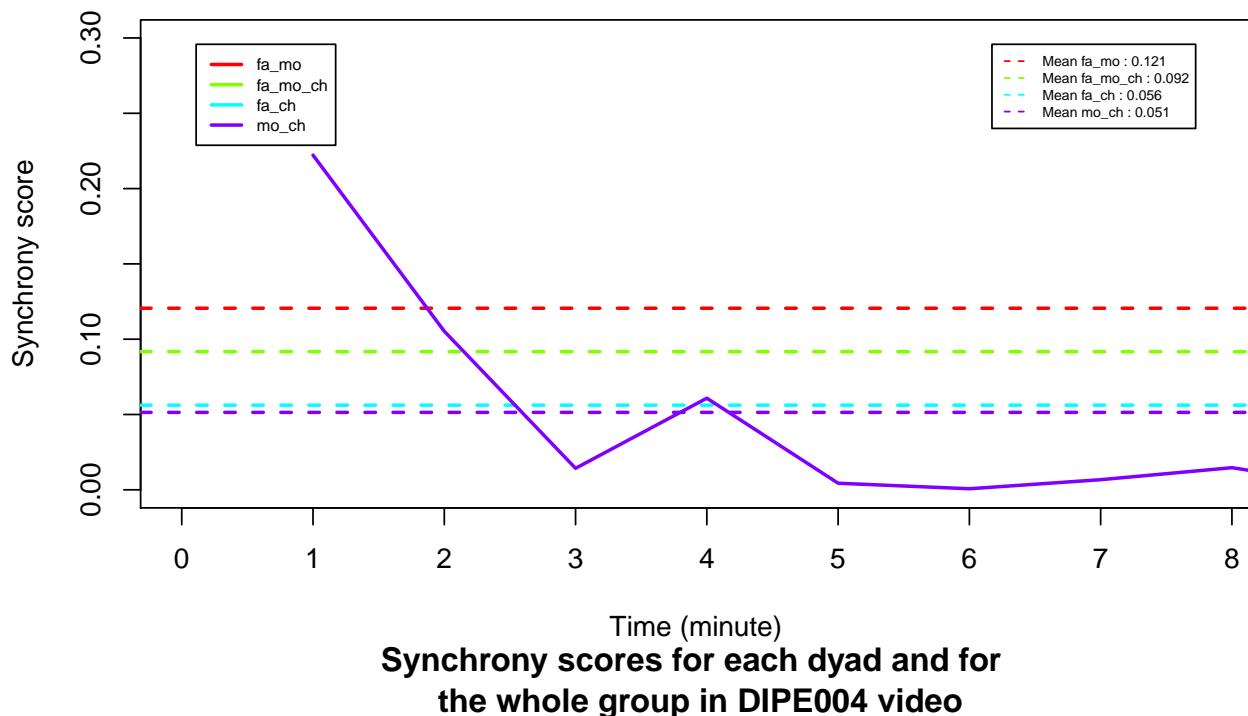
**Synchrony scores for each dyad and for
the whole group in BICA video**



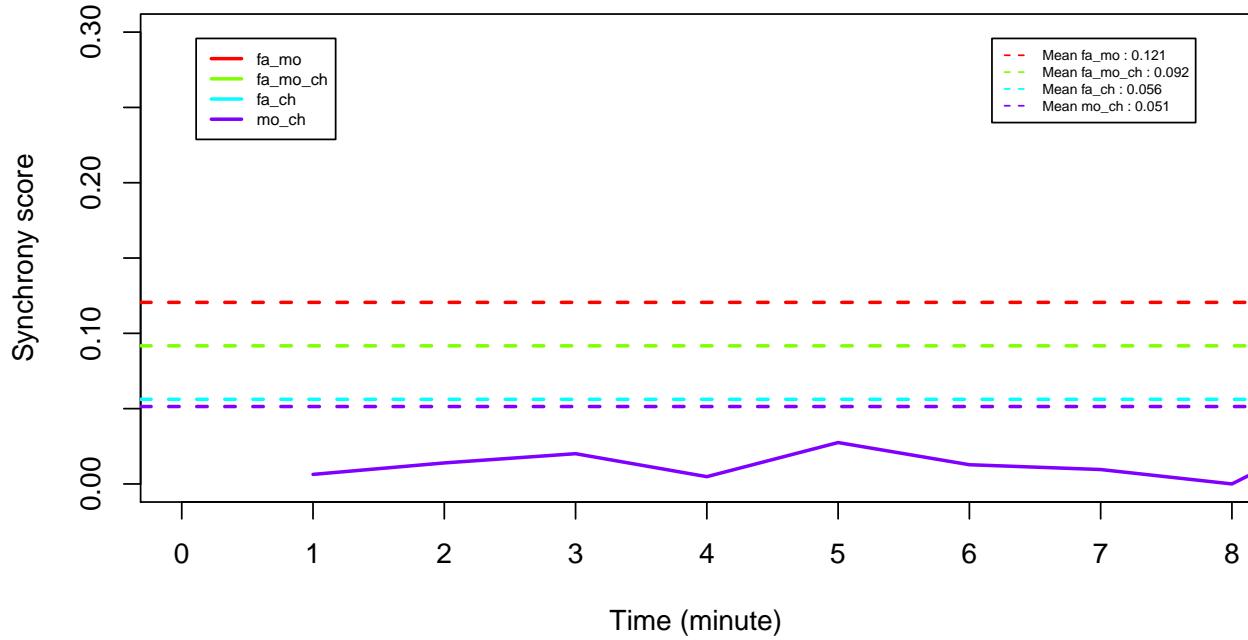
**Synchrony scores for each dyad and for
the whole group in BRLO041 video**



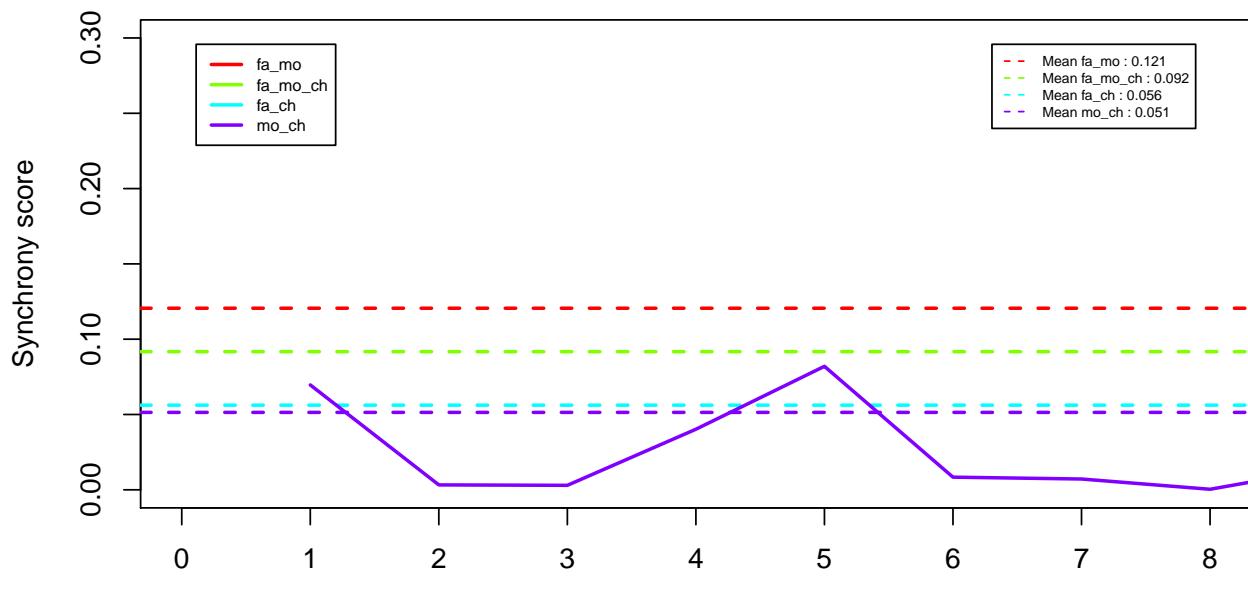
**Synchrony scores for each dyad and for
the whole group in COLO022 video**



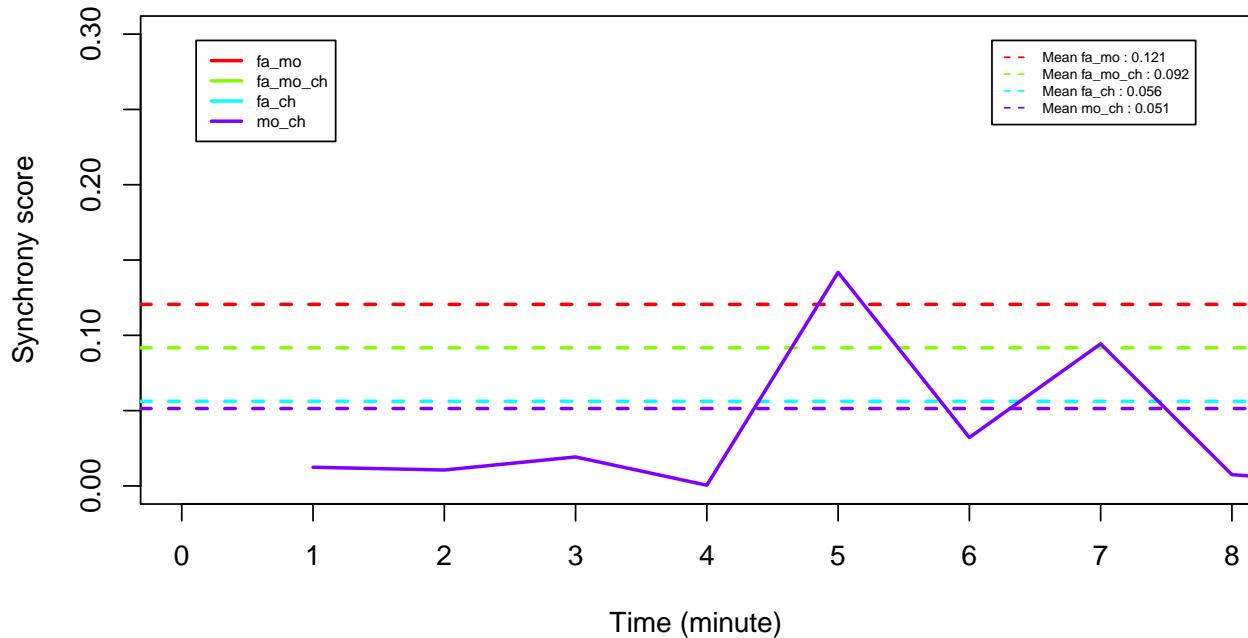
**Synchrony scores for each dyad and for
the whole group in DIPE004 video**



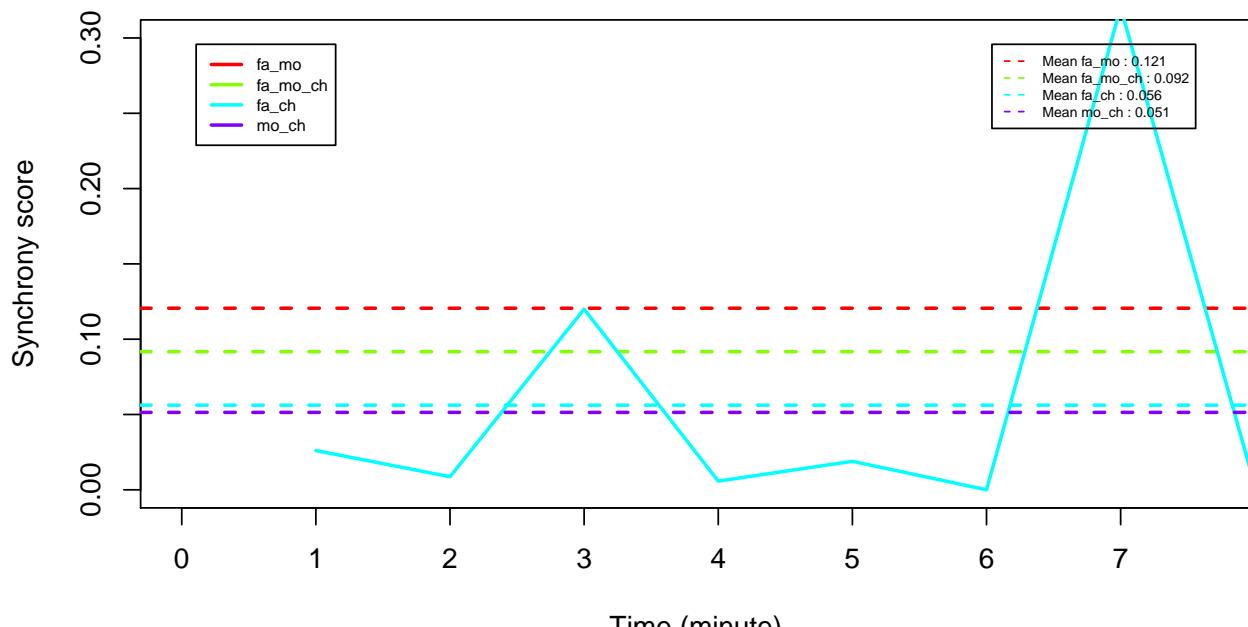
Synchrony scores for each dyad and for the whole group in DOMA video



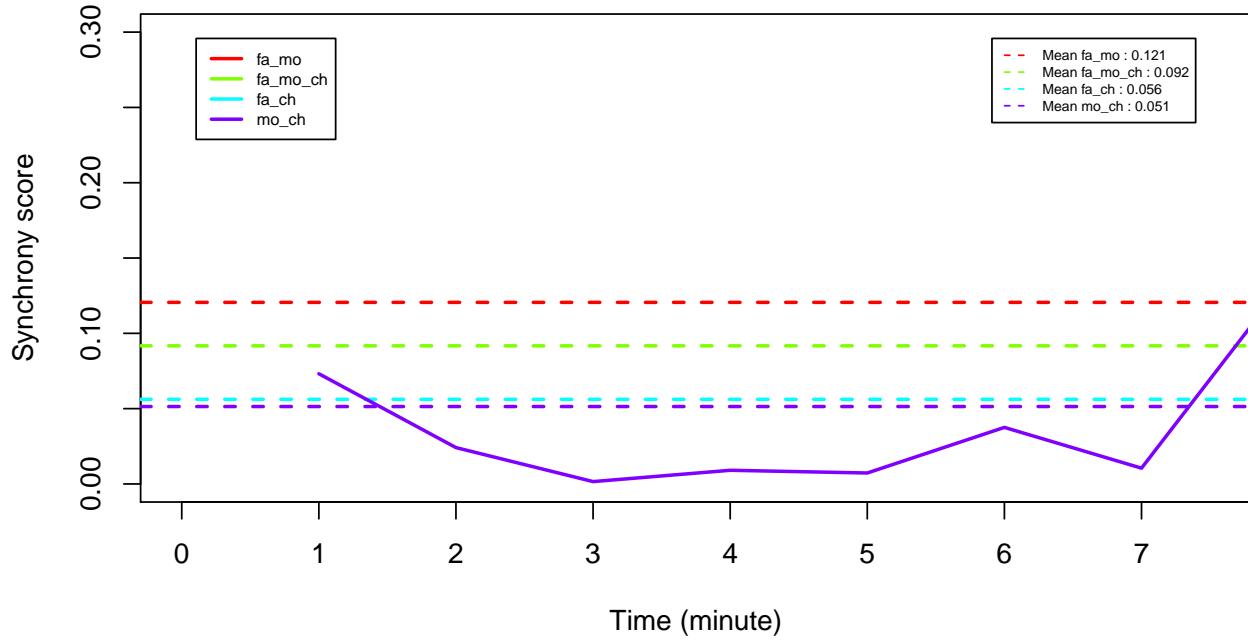
Synchrony scores for each dyad and for the whole group in DRNE video



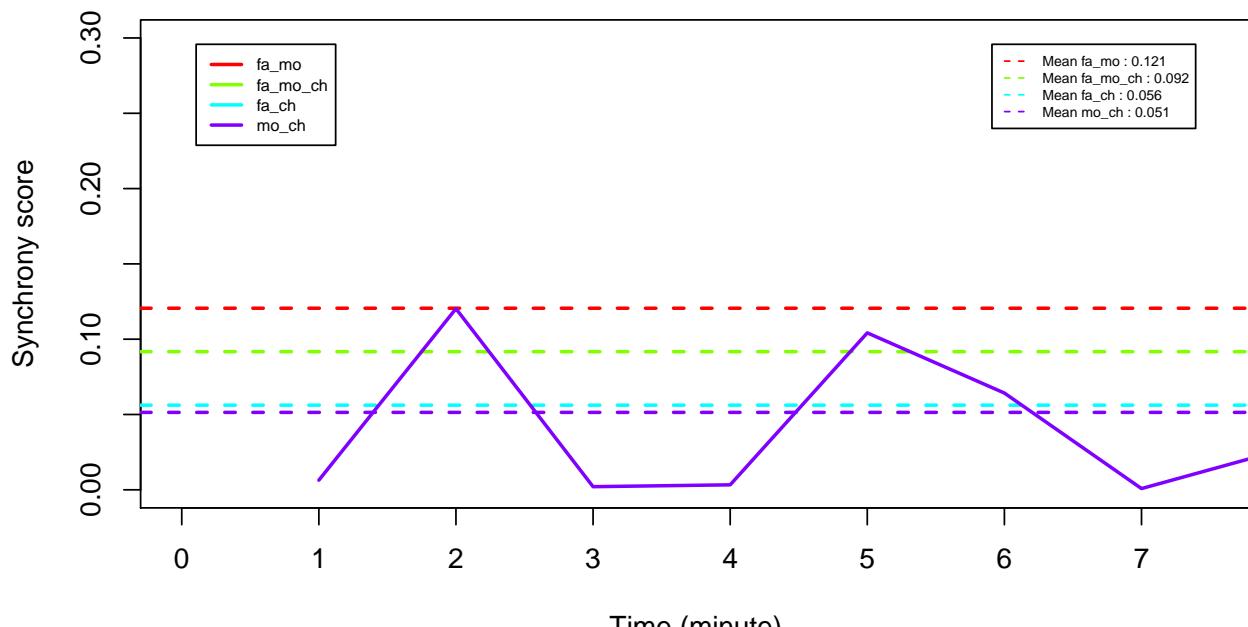
**Synchrony scores for each dyad and for
the whole group in FOMA057 video**



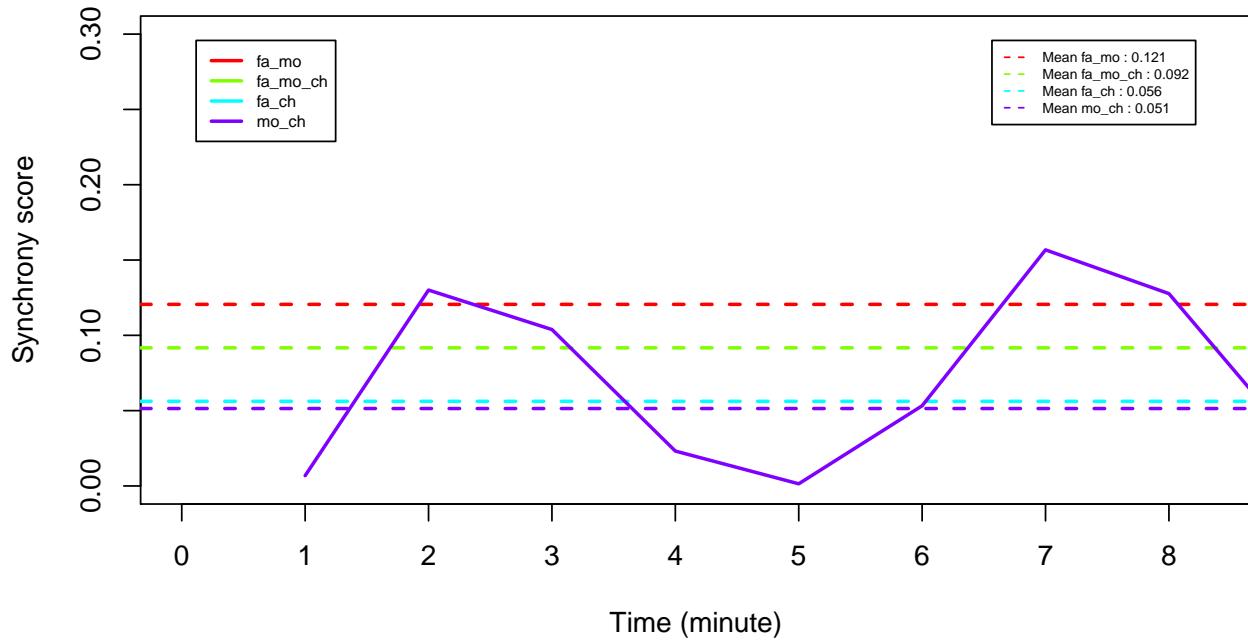
**Synchrony scores for each dyad and for
the whole group in GROP039 video**



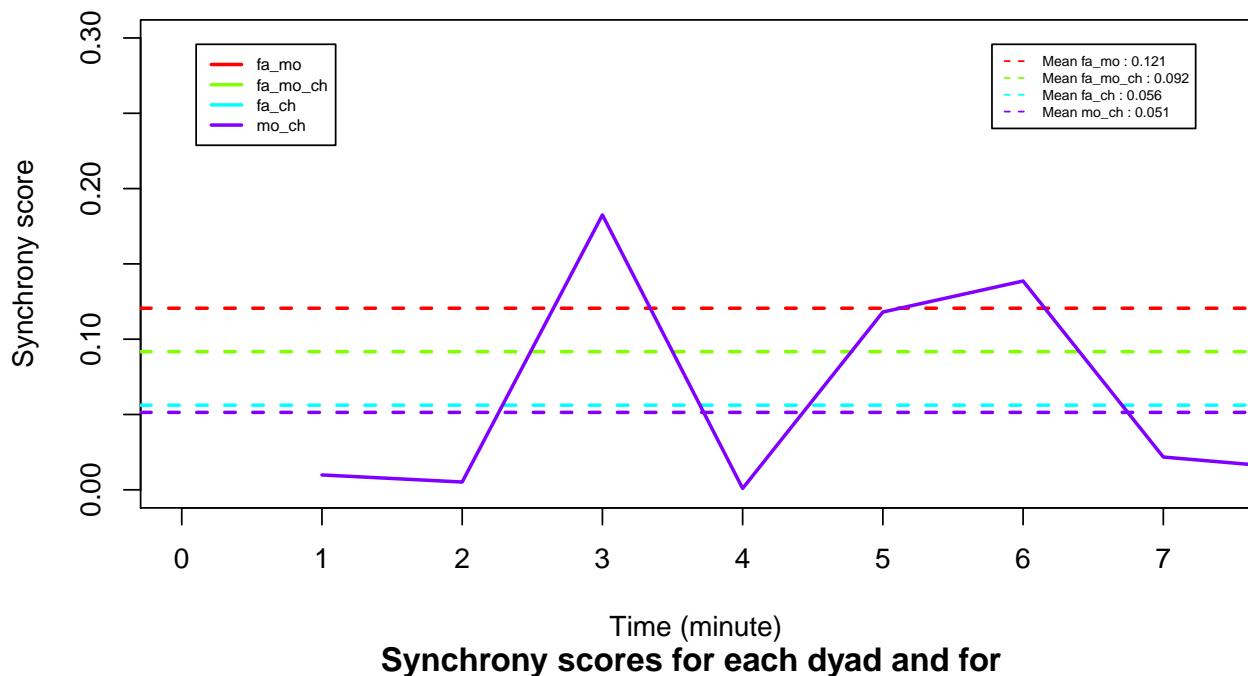
**Synchrony scores for each dyad and for
the whole group in HAJA052 video**



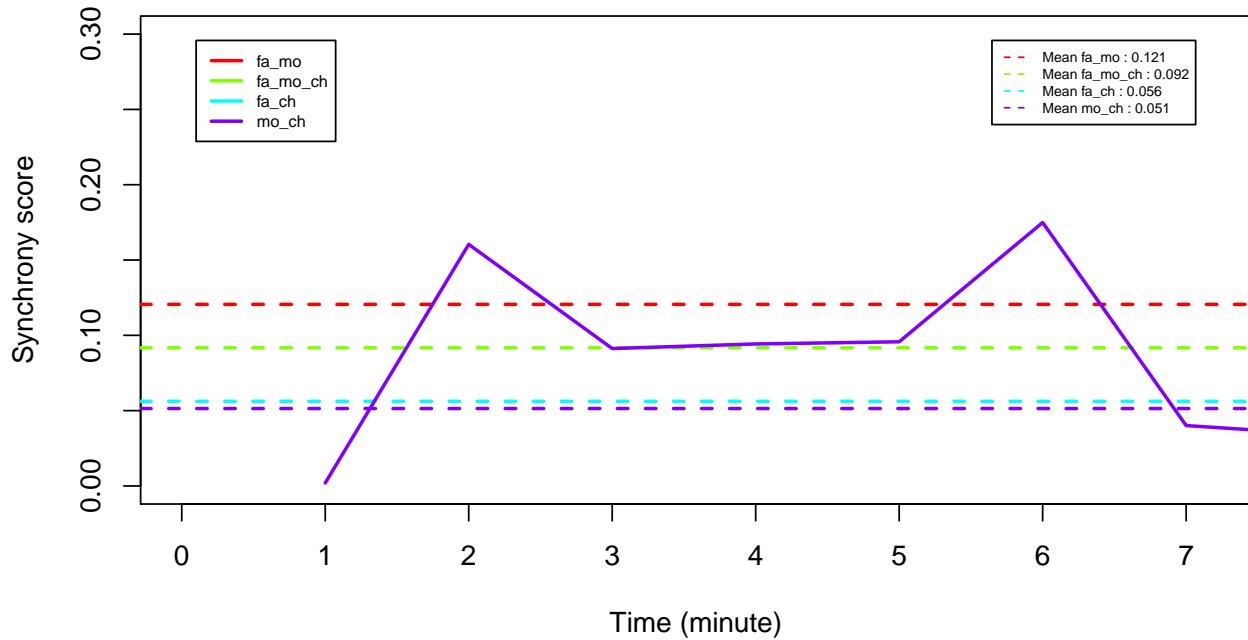
**Synchrony scores for each dyad and for
the whole group in HUMA058 video**



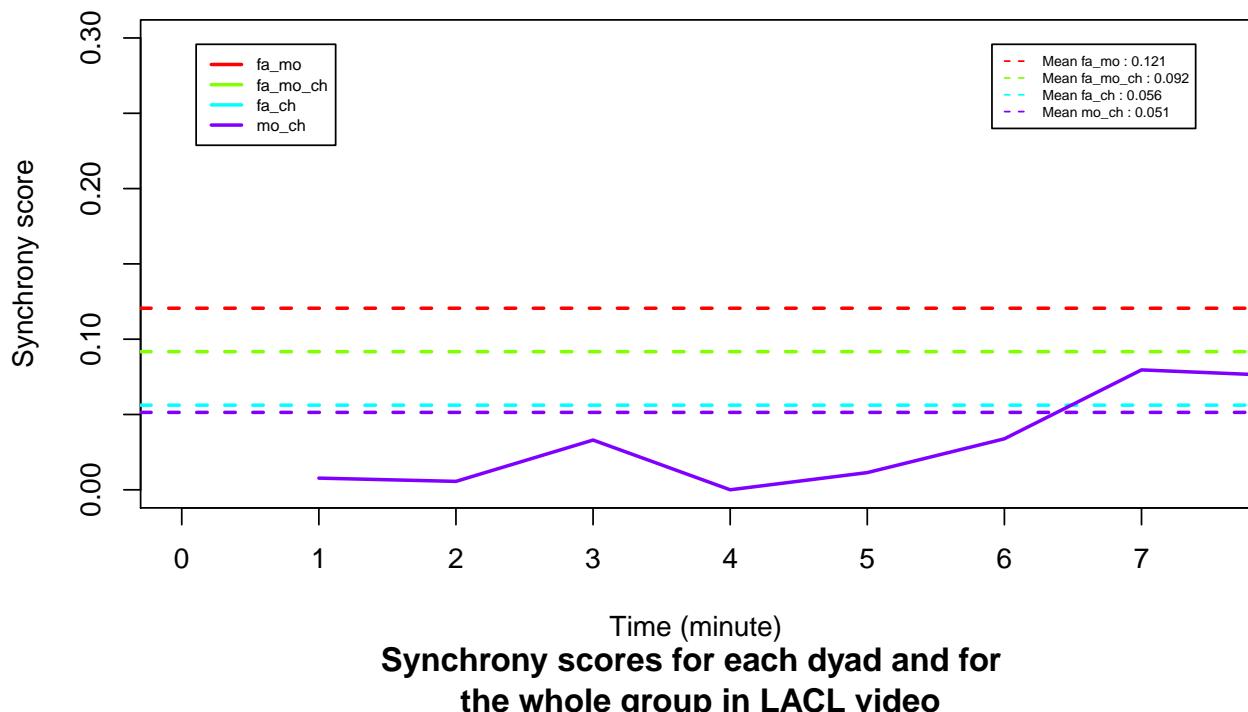
Synchrony scores for each dyad and for the whole group in JAEM046 video



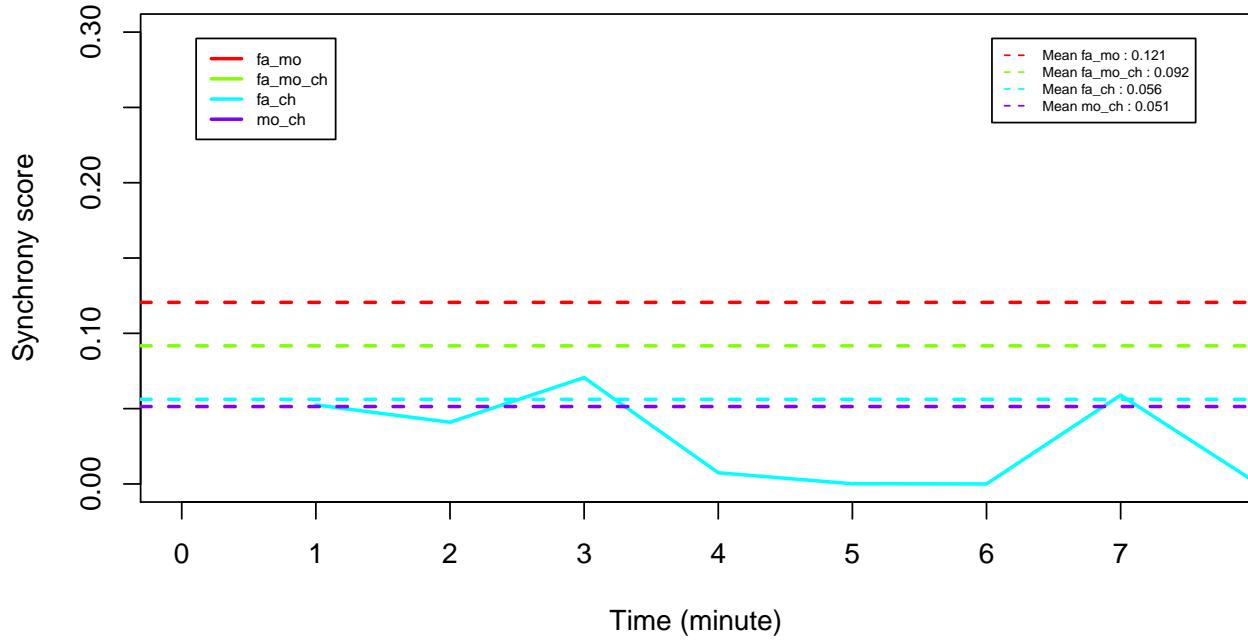
Synchrony scores for each dyad and for the whole group in JEEO040 video



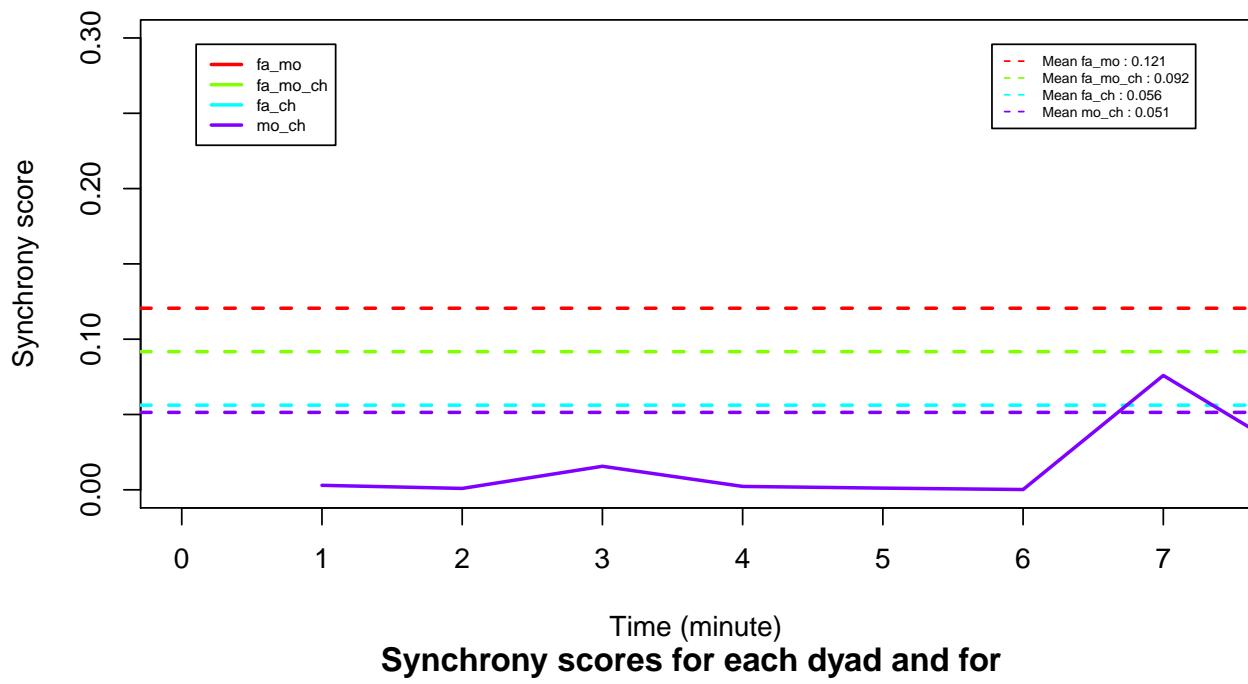
**Synchrony scores for each dyad and for
the whole group in JOCE014 video**



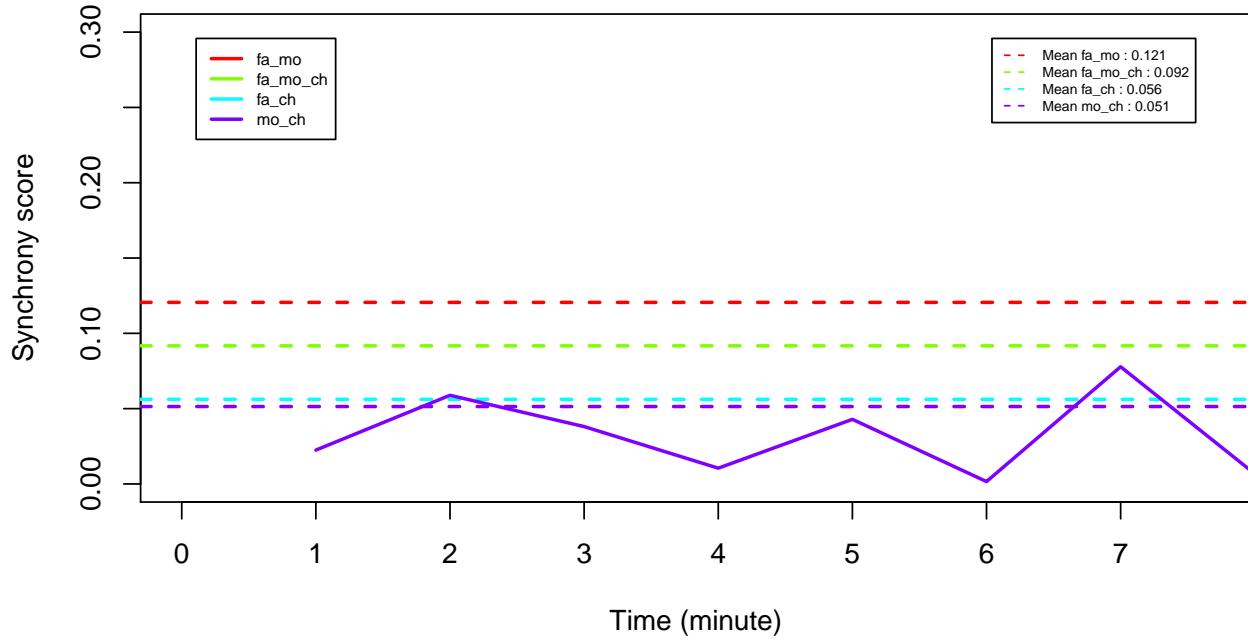
**Synchrony scores for each dyad and for
the whole group in LACL video**



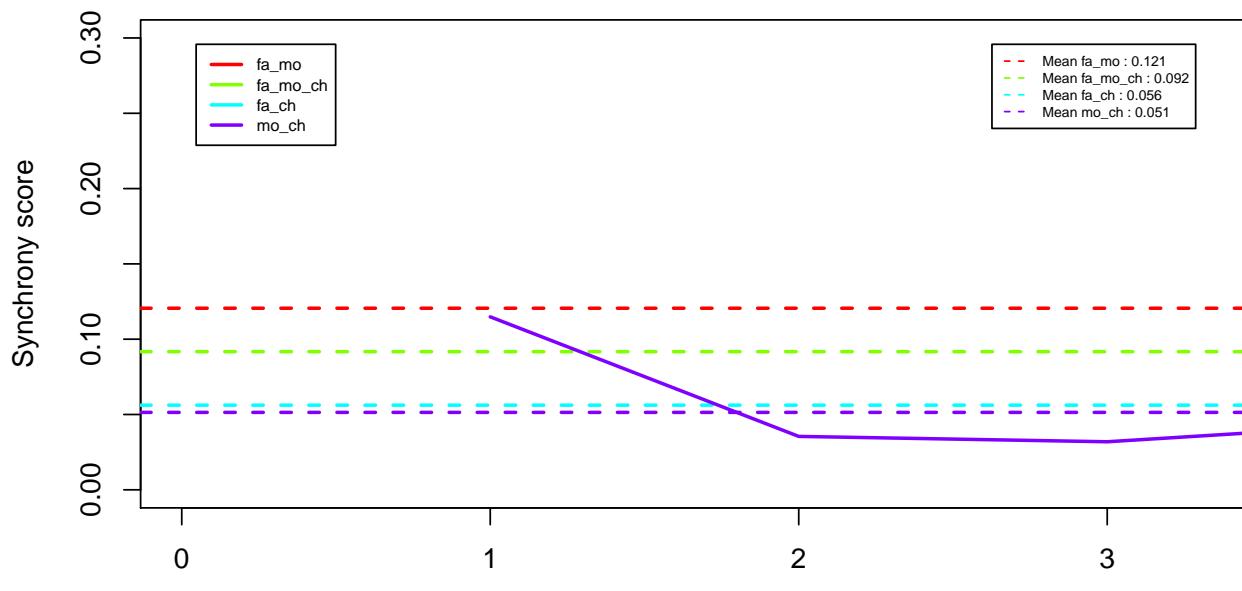
**Synchrony scores for each dyad and for
the whole group in MAEL048 video**



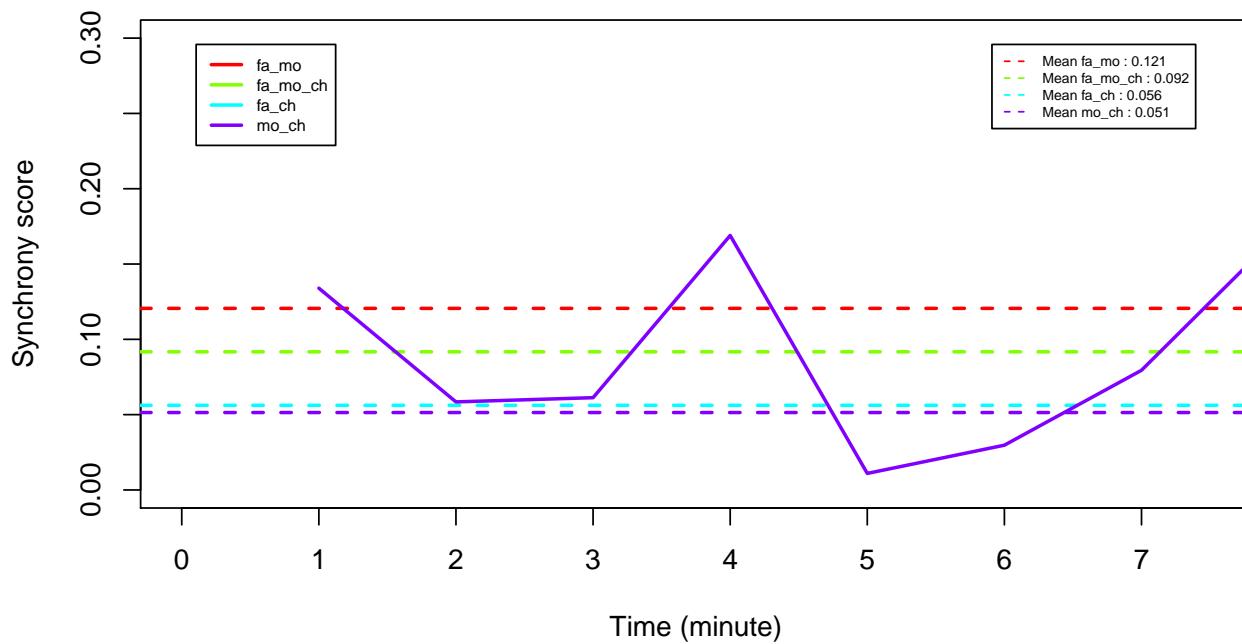
**Synchrony scores for each dyad and for
the whole group in MAME20 video**



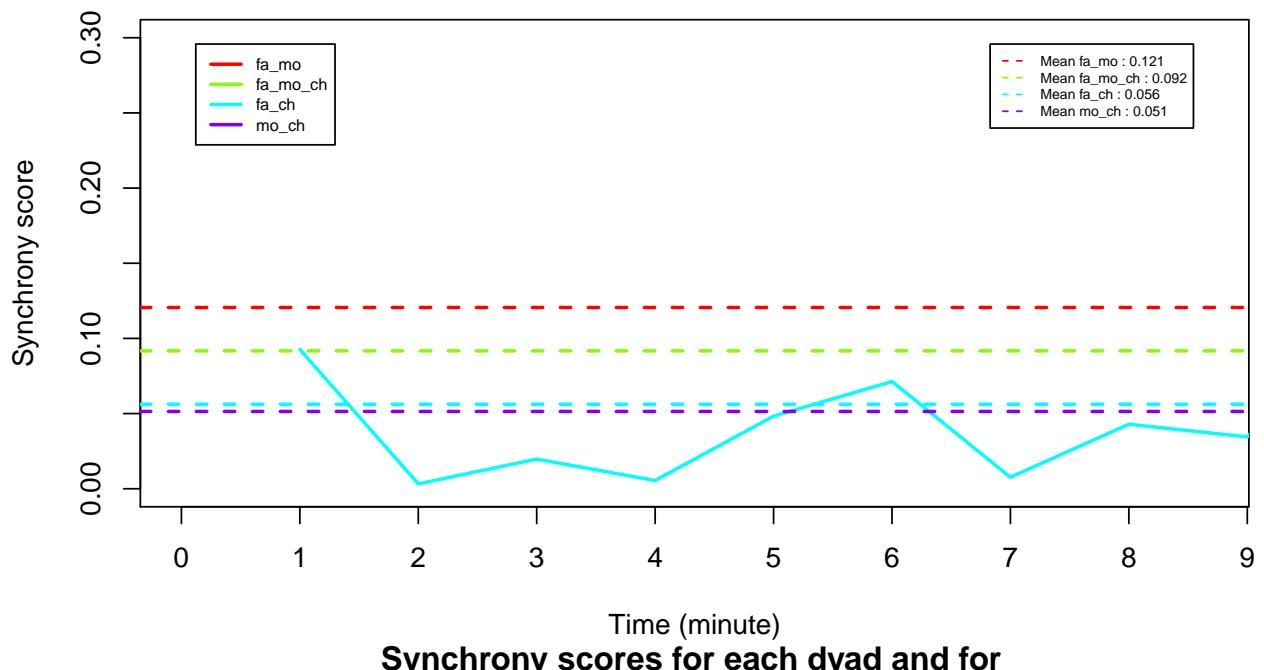
**Synchrony scores for each dyad and for
the whole group in MAPA029 video**



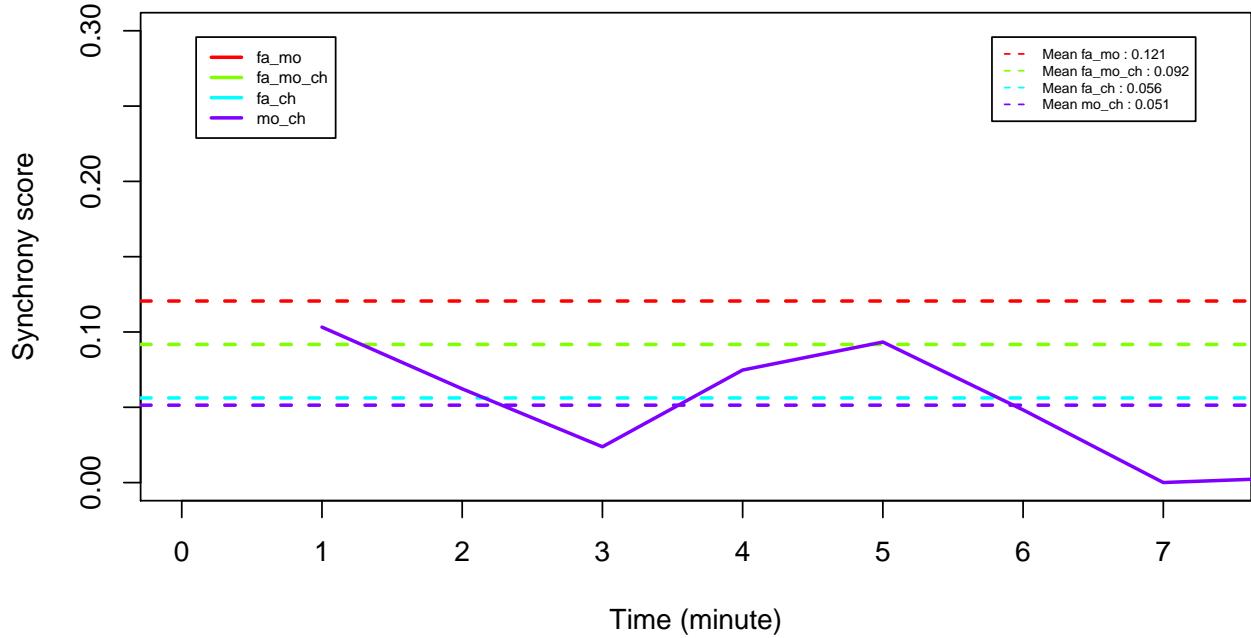
**Synchrony scores for each dyad and for
the whole group in MIPH043 video**



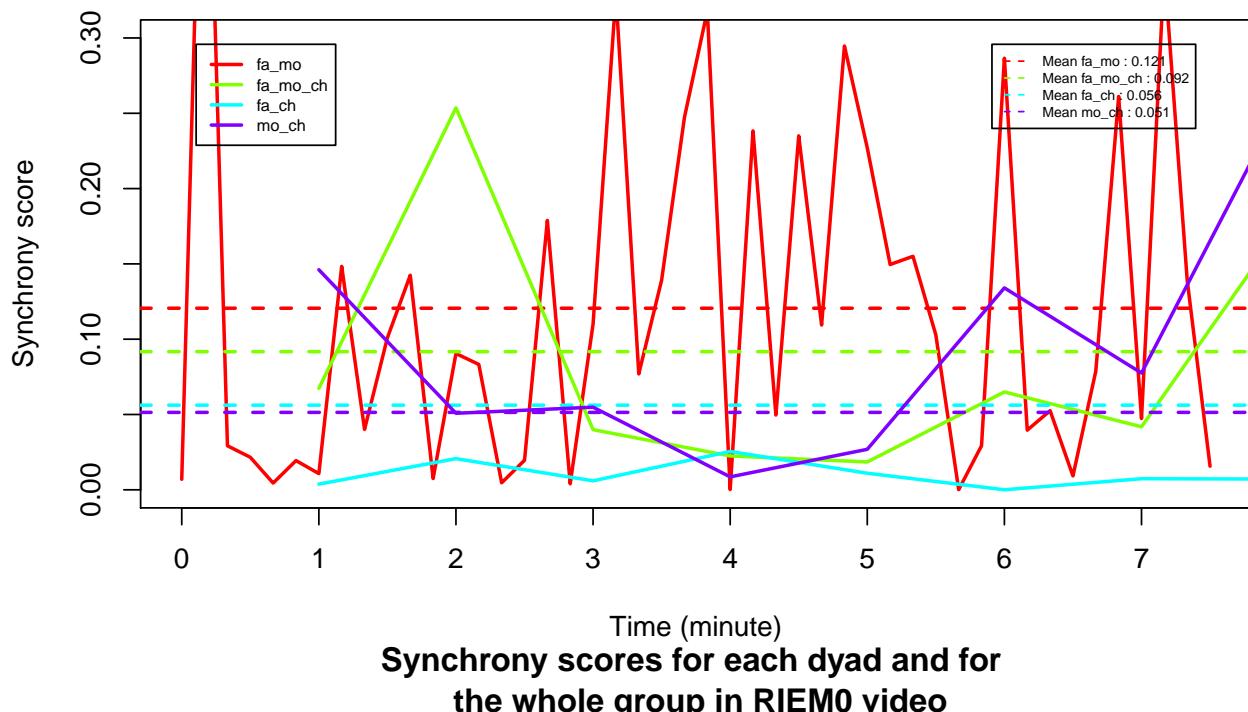
**Synchrony scores for each dyad and for
the whole group in MOSA065 video**



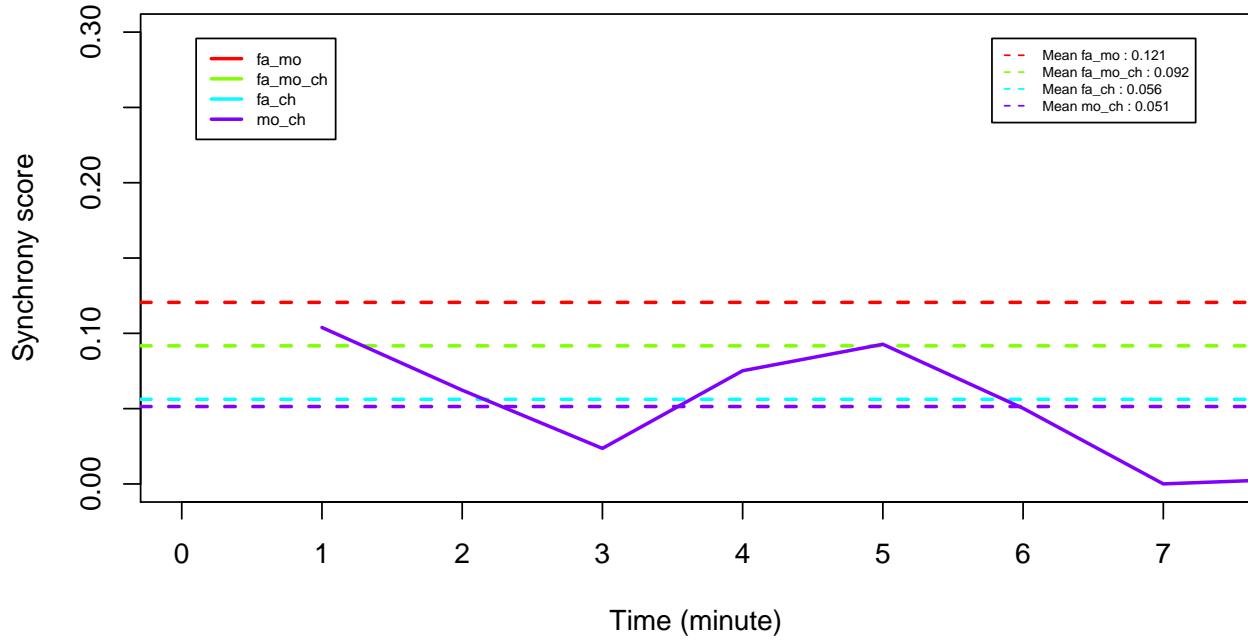
**Synchrony scores for each dyad and for
the whole group in RAEM049 video**



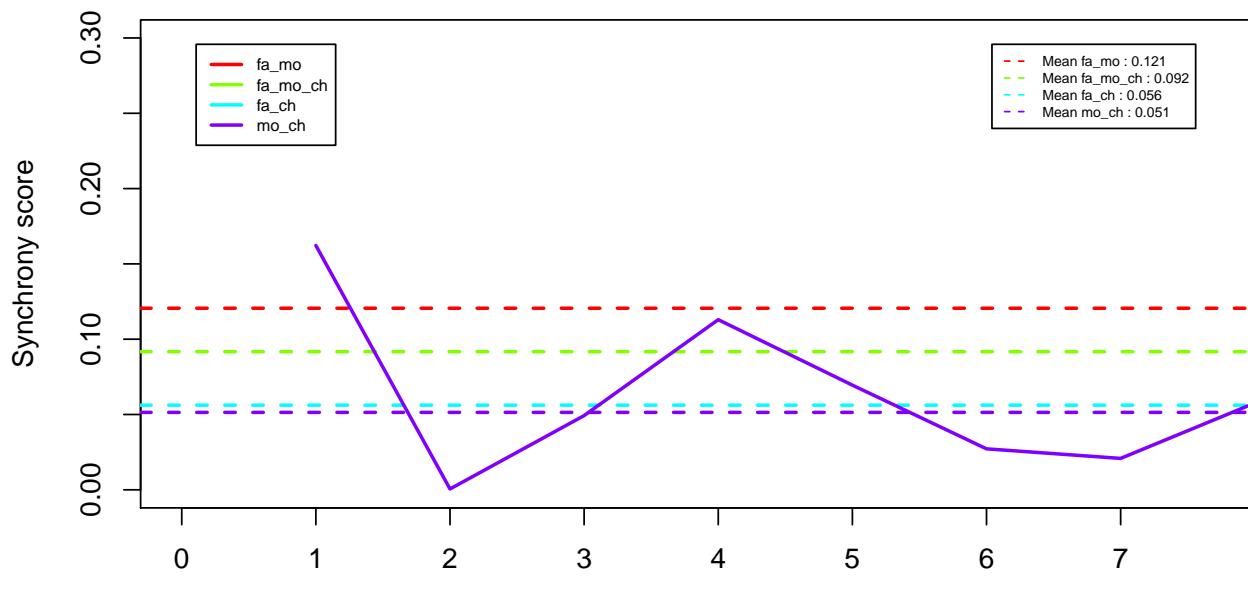
Synchrony scores for each dyad and for the whole group in RAKA008 video



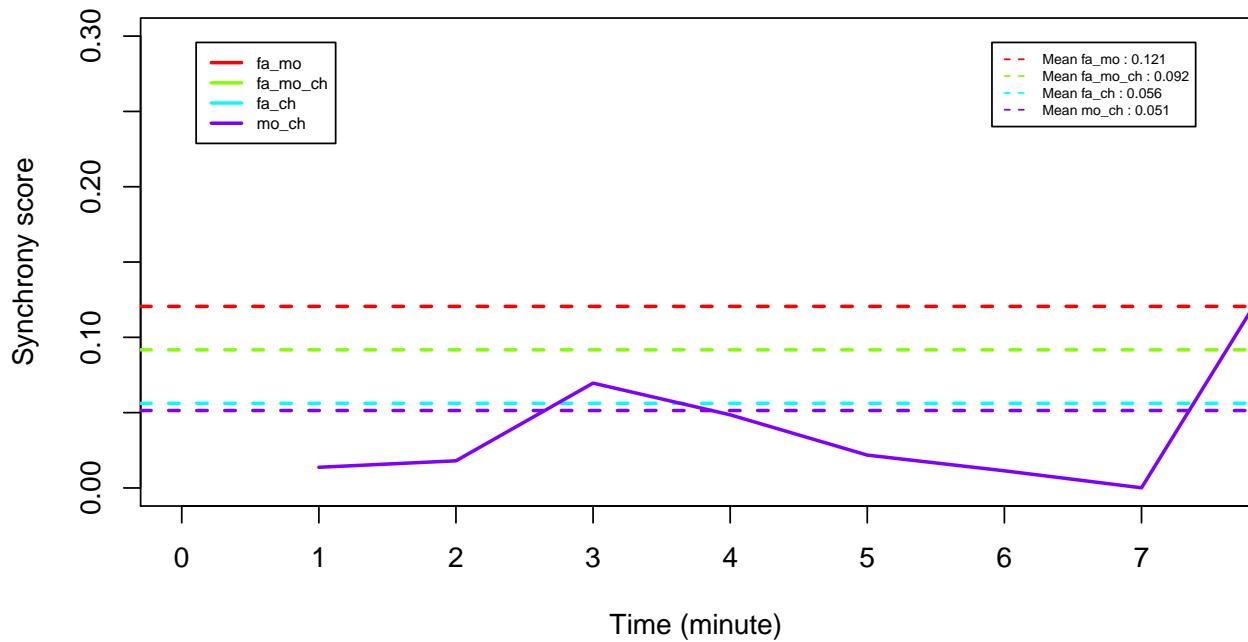
Synchrony scores for each dyad and for the whole group in RIEM0 video



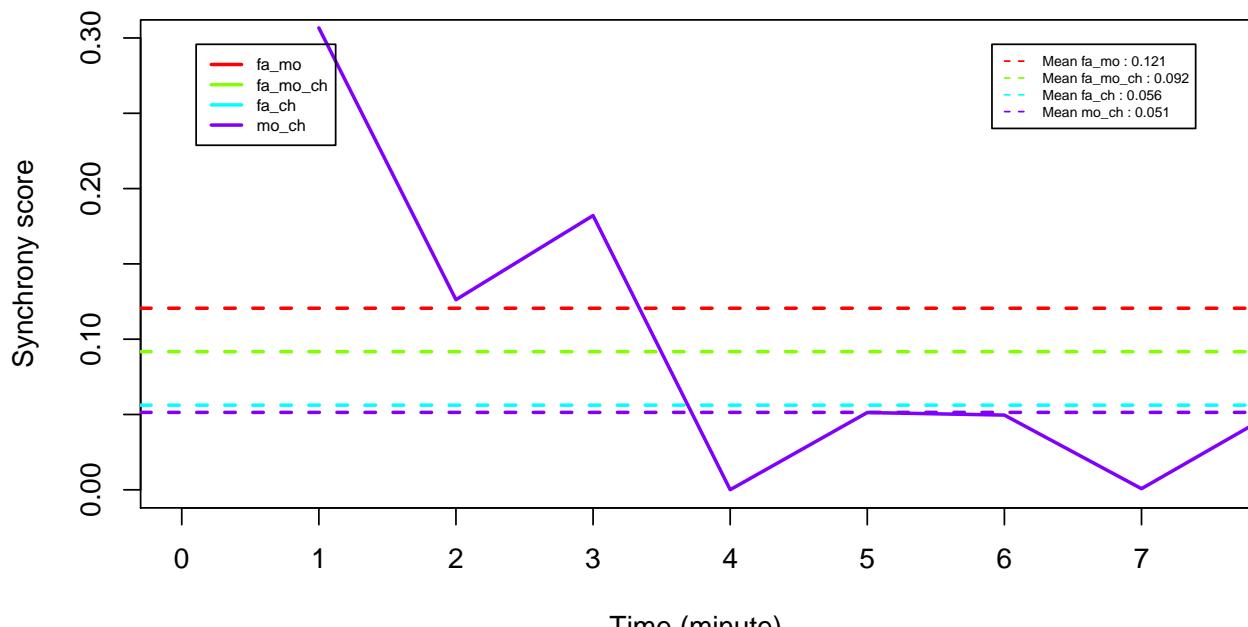
**Synchrony scores for each dyad and for
the whole group in SEEM035 video**



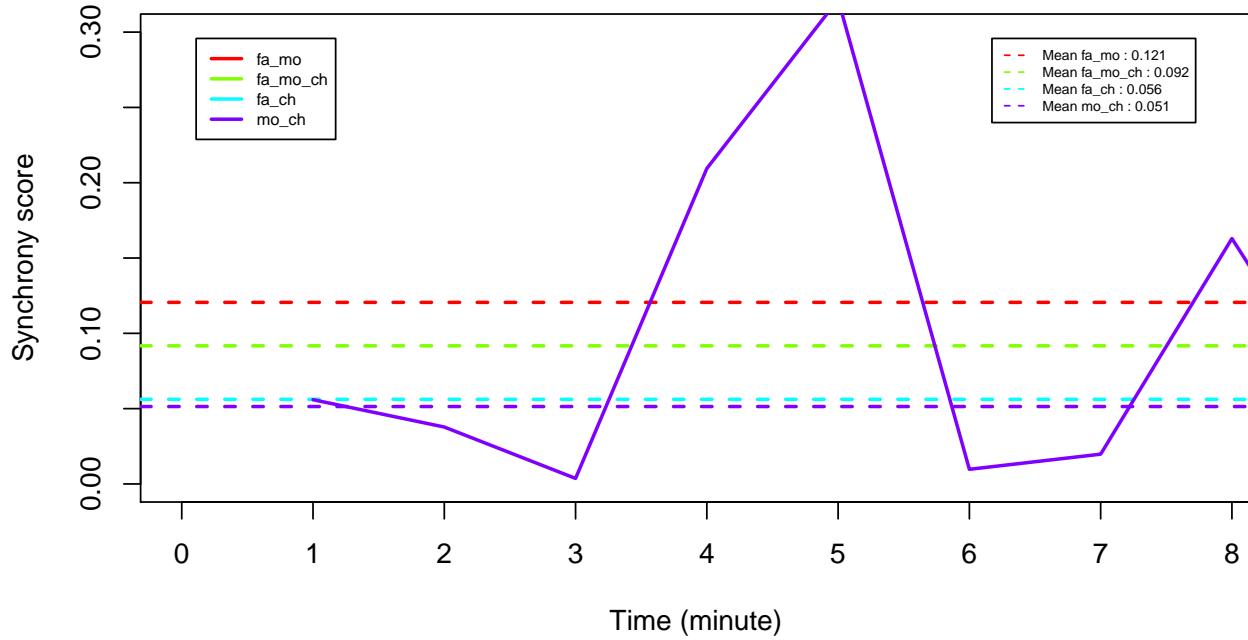
**Synchrony scores for each dyad and for
the whole group in SHAN042 video**



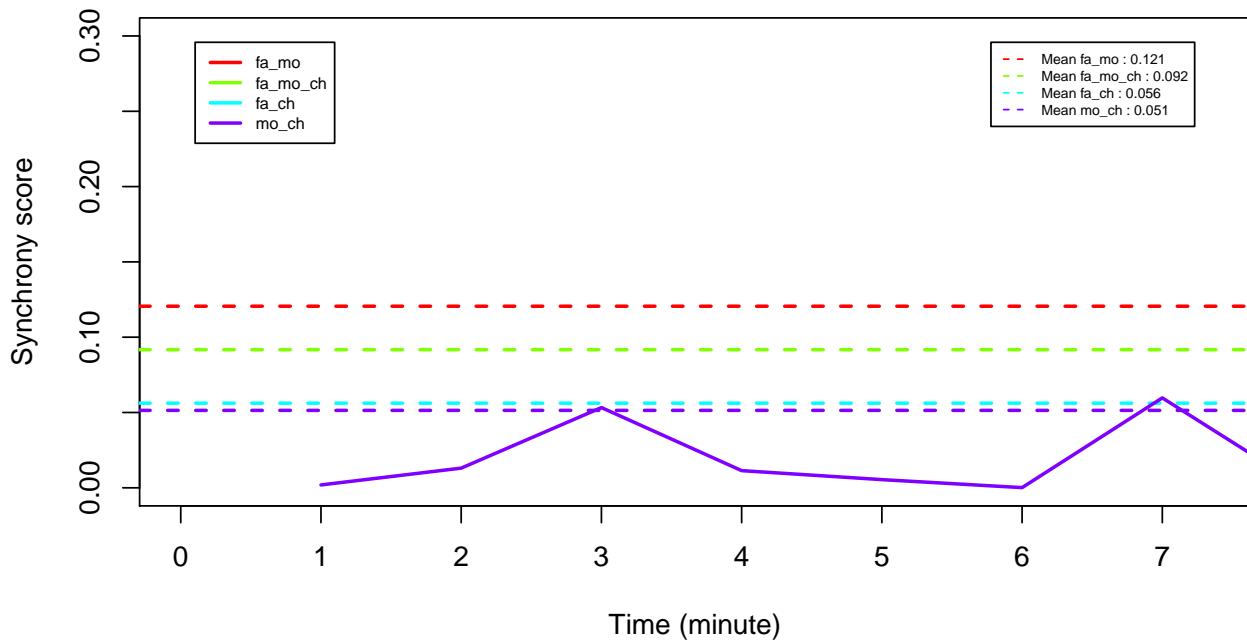
**Synchrony scores for each dyad and for
the whole group in SOGA061 video**



**Synchrony scores for each dyad and for
the whole group in TIUG032 video**



Synchrony scores for each dyad and for the whole group in VINO video



Evolution of synchrony through time, raw each second

```
par(mar=c(4,4,4,4))
col <- 1
for (i in 5:length(SSI)){
  plot(1:length(SSI[,i]), SSI[,i], type="l",
  col=rainbow(4)[col], main = names(SSI)[i])
  col <- col+1}
```

Evolution of synchrony through time, mean by minute

```
par(mar=c(4,4,4,4))
col = 1
for (indexSSI in 5:length(SSI)){
  IntervalNumbersVideo <- ceiling(length(SSI[,indexSSI])/6)
  SSIColumn <- SSI[,indexSSI]
  SSIMinute <- c()
  for (i in 1:IntervalNumbersVideo){
    borneInf <- 1+(i-1)*6
    borneSup <- i * 6
    SSIVectorInterval <- SSIColumn[borneInf:borneSup]
    mean <- mean(SSIVectorInterval, na.rm=TRUE)
    SSIMinute <- c(SSIMinute, mean)}
  plot(1:length(SSIMinute), SSIMinute, type="l", col=rainbow(11)[col], main = names(SSI)[indexSSI])
  col <- col+1}
```

Evolution of synchrony through time, mean by 10 minutes

```
par(mar=c(4,4,4,4))
  col = 1
for (indexSSI in 5:length(SSI)){
  IntervalNumbersVideo <- ceiling(length(SSI[,indexSSI])/60)
  SSIColumn <- SSI[,indexSSI]
  SSITenMinute <- c()
  for (i in 1:IntervalNumbersVideo){
    borneInf <- 1+(i-1)*60
    borneSup <- i * 60
    SSIVectorInterval <- SSIColumn[borneInf:borneSup]
    mean <- mean(SSIVectorInterval, na.rm=TRUE)
    SSITenMinute <- c(SSITenMinute, mean)}
  plot(1:length(SSITenMinute), SSITenMinute, type="l", col=rainbow(4)[col], main = names(SSI)[indexSSI])
  col <- col+1}
```

Models of synchrony

```
SSI_fa_th_lme <- lmer(SSI_fa_th ~ Time_min + (1|video), data=SSI)
summary(SSI_fa_th_lme)
#plot(SSI_fa_th_lme)
res <- residuals(SSI_fa_th_lme)
hist(SSI$SSI_fa_th)
qqnorm(res)
SSI_fa_th_List <- c()
for (i in families){
  SSI_fa_th_List <- c(SSI_fa_th_List, mean(SSI[which(SSI$video==i),]$SSI_fa_th, na.rm=TRUE))
}
print(SSI_fa_th_List)
#plot(SSI_fa_th_List, type="b")

# log of the data
log_SSI_fa_th <- hist(log(SSI$SSI_fa_th))
SSI_fa_th_log_lme <- lmer(log(SSI_fa_th) ~ Time_min + (1|video), data=SSI)
res_log <- residuals(SSI_fa_th_log_lme)
qqnorm(res_log)
summary(SSI_fa_th_log_lme)

# root square of the data
sq_SSI_fa_th <- hist(sqrt(SSI$SSI_fa_th))
SSI_fa_th_sq_lme <- lmer(sqrt(SSI_fa_th) ~ Time_min + (1|video), data=SSI)
res_sq <- residuals(SSI_fa_th_sq_lme)
qqnorm(res_sq)
summary(SSI_fa_th_sq_lme)
```