Code and steps to replicate the analysis

1. **Load video files to MATLAB and extract all frames.**

Code (MATLAB):

```
ef=VideoReader('name.of.video.mp4')
for img= 1:ef.NumberOfFrames;
   filename=strcat('frame',num2str(img),'.jpg');
   b=read(ef,img);
   imwrite(b,filename);
end
```

2. **Load frames in MATLAB to measure mouth displacement.**

Code (MATLAB); example for a video recorded at 25 fps with 20 frames:

```
imtool('frame20.jpg')
imtool('frame19.jpg')
imtool('frame18.jpg')
imtool('frame17.jpg')
imtool('frame16.jpg')
imtool('frame15.jpg')
imtool('frame14.jpg')
imtool('frame13.jpg')
imtool('frame12.jpg')
imtool('frame11.jpg')
imtool('frame10.jpg')
imtool('frame9.jpg')
imtool('frame8.jpg')
imtool('frame7.jpg')
imtool('frame6.jpg')
imtool('frame5.jpg')
imtool('frame4.jpg')
```

```
imtool('frame3.jpg')
imtool('frame2.jpg')
imtool('frame1.jpg')
```

3. **Extract power spectral density of each bout and calculate its dominant frequency.**

```
% Code (MATLAB); example for a video recorded at 25 fps with 20 frames:

Fs = 25; % Sampling frequency = 25 Hz (i.e., 25 Frames per Second)
t = 0:0.04:(19*0.04); % Time scale vector of lip-smack bout: To each mouth displacement measurement (i.e., each frame) there is a corresponding time stamp, which is 0 seconds for the first frame and (n-1)*0.04 seconds (i.e., 1/(sampling frequency)) for the nth frame
x = [88.50233704, 90.52310448, 93.74177839, 93.76416916, 92.68941196, 90.39155868, 84.94220381, 83.42522881, 92.76498083, 98.37666881, 88.72064709, 85.52996166, 84.16412438, 91.24240813, 98.56139271, 92.34235495, 89.16566375, 88.4939405, 94.23437544, 100]; % example time series (i.e., mouth displacement measures of the lip-smack bout)
x = x - mean(x); % avoid getting 0 as the domain frequency (D-C offset)
plot(t,x), axis('tight'), grid('on'), title('Time series'), figure % plot the time series
nfft = 1024; % large enough power of 2 which provides good resolution of the file without compromising computational time
y = fft(x,nfft); % Fast Fourier Transform
y = abs(y.^2); % raw power spectrum density
y = y(1:1+nfft/2); % half-spectrum, for plotting
[v,k] = max(y); % find maximum
f_scale = (0:nfft/2)* Fs/nfft; % frequency scale
plot(f_scale, y),axis('tight'),grid('on'),title('Dominant Frequency') % plot the power spectrum of the time series
fest = f_scale(k); % dominant frequency estimate
fprintf('Dominant freq.: %f Hz\n', fest)
```

4. **Prepare .csv file where column “Frequency”= f_scale values obtained from step 3; column “Power” = y values obtained from step 3; column “Individual” = name of the emitter, and column “Origin” = origin of the emitter (i.e., “Edinburgh (captive)”; “Leipzig (captive)”; “Waibira (wild)”; “Kanyawara (wild)”).**

5. **Obtain curves of the mean ± 95% confidence interval of the power spectrum density of lip-smack bouts for *a)* all bouts; *b)* bouts discriminated by emitter (coloured with the emitters’ origin); *c)* bouts discriminated by origin (one graph for each pair of Origins).**

 # Code (R):

```
xdata<-read.csv2("chimpanzee_lipsmack_standardized.csv",h=T, sep=",")
library("ggplot2")
str (xdata)
xdata$Power<-as.numeric(as.character(xdata$Power))
xdata$Frequency<-as.numeric(as.character(xdata$Frequency))
```

 # Code for curve a)

```
curve= ggplot(xdata, aes(x=Frequency, y=Power)) + stat_summary(fun.data =
"mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10, colour="black") +
labs(x="Frequency (Hz)", y="Standardized spectral power") + theme_classic(base_size = 33,
base_family = "", base_line_size=1.1, base_rect_size=1.1) + geom_vline(data=xdata,
aes(xintercept=2), linetype="dashed", size=1.1) + geom_vline(data=xdata, aes(xintercept=7),
linetype="dashed", size=1.1) + scale_x_continuous (breaks=seq(0,12.5,1)) +
geom_vline(data=xdata, aes(xintercept= 4.150391), colour= "black", size=1.1)

curve
```

 # Code for curve b)

 # b.1) Edinburgh (captive)

```
xdata2 <- xdata[xdata$Origin=="Edinburgh (captive)", ]

curve = ggplot(xdata2, aes(x=Frequency, y=Power, colour=Origin, ID=Individual,
fill=Origin)) + stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE,
alpha=0.10) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12.5,1)) + scale_colour_manual(values= c("Edinburgh
(captive)" = "coral1")) + scale_fill_manual(values= c("Edinburgh (captive)" = "coral1"))
curve
```

 # b.2) Leipzig (captive)

```
xdata2 <- xdata[xdata$Origin=="Leipzig (captive)", ]

curve = ggplot(xdata2, aes(x=Frequency, y=Power, colour=Origin, ID=Individual,
fill=Origin)) + stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE,
alpha=0.10) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12.5,1)) + scale_colour_manual(values= c("Leipzig
(captive)" = "cadetblue2")) + scale_fill_manual(values= c("Leipzig (captive)" =
"cadetblue2"))
curve
```

 # b.2) Kanyawara (wild)

```
xdata2 <- xdata[xdata$Origin=="Kanyawara (wild)", ]
```

curve = ggplot(xdata2, aes(x=Frequency, y=Power, colour=Origin, ID=Individual, fill=Origin)) + stat_summary(fun.data = "mean_cl_boot", geom= "smooth", se= TRUE, alpha=0.10) + labs(x="Frequency (Hz)", y="Standardized spectral power") + theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) + geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) + geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) + scale_x_continuous (breaks=seq(0,12.5,1)) + scale_colour_manual(values= c("Kanyawara (wild)" = "chartreuse3")) + scale_fill_manual(values= c("Kanyawara (wild)" = "chartreuse3"))

curve

 # b.2) Waibira (wild)

xdata2 <- xdata[xdata$Origin=="Waibira (wild)",]

curve = ggplot(xdata2, aes(x=Frequency, y=Power, colour=Origin, ID=Individual, fill=Origin)) + stat_summary(fun.data = "mean_cl_boot", geom= "smooth", se= TRUE, alpha=0.10) + labs(x="Frequency (Hz)", y="Standardized spectral power") + theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) + geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) + geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) + scale_x_continuous (breaks=seq(0,12.5,1)) + scale_colour_manual(values= c("Waibira (wild)" = "blueviolet")) + scale_fill_manual(values= c("Waibira (wild)" = "blueviolet"))

curve

 # Code for curves c)

 # c.1) Edinburgh (captive) and Leipzig (captive)

xdata2 <- xdata[xdata$Origin=="Edinburgh (captive)" | xdata$Origin=="Leipzig (captive)",]

curve=ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) + stat_summary(fun.data = "mean_cl_boot", geom= "smooth", se= TRUE, alpha=0.10, size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +

```
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=4.199219), colour= "coral1", size=1.1) + geom_vline(data=xdata,
aes(xintercept=4.101562), colour= "cadetblue2", size=1.1) + scale_colour_manual(values=
c("Edinburgh (captive)" = "coral1", "Leipzig (captive)" = "cadetblue2")) +
scale_fill_manual(values= c("Edinburgh (captive)" = "coral1", "Leipzig (captive)" =
"cadetblue2"))
curve

# c.2) Edinburgh (captive) and Kanyawara (wild)

xdata2 <- xdata[xdata$Origin=="Edinburgh (captive)" |  xdata$Origin=="Kanyawara (wild)",
]

curve=ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) +
stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10,
size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=4.199219), colour= "coral1", size=1.1) + geom_vline(data=xdata,
aes(xintercept=2.856445), colour= "chartreuse3", size=1.1) + scale_colour_manual(values=
c("Edinburgh (captive)" = "coral1", "Kanyawara (wild)" = "chartreuse3")) +
scale_fill_manual(values= c("Edinburgh (captive)" = "coral1", "Kanyawara (wild)" =
"chartreuse3"))
curve

# c.3) Edinburgh (captive) and Waibira (wild)

xdata2 <- xdata[xdata$Origin=="Edinburgh (captive)" |  xdata$Origin=="Waibira (wild)", ]

curve=ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) +
stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10,
```

```
size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=4.199219), colour= "coral1", size=1.1) + geom_vline(data=xdata,
aes(xintercept=1.953125), colour= "blueviolet", size=1.1)+ scale_colour_manual(values=
c("Edinburgh (captive)" = "coral1", "Waibira (wild)" = "blueviolet")) +
scale_fill_manual(values= c("Edinburgh (captive)" = "coral1", "Waibira (wild)" =
"blueviolet"))
curve
```

 # c.4) Leipzig (captive) and Kanyawara (wild)

```
xdata2 <- xdata[xdata$Origin=="Leipzig (captive)" |  xdata$Origin=="Kanyawara (wild)", ]

curve=ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) +
stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10,
size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=4.101562), colour= "cadetblue2", size=1.1) + geom_vline(data=xdata,
aes(xintercept=2.856445), colour= "chartreuse3", size=1.1) + scale_colour_manual(values=
c("Leipzig (captive)" = "cadetblue2", "Kanyawara (wild)" = "chartreuse3")) +
scale_fill_manual(values= c("Leipzig (captive)" = "cadetblue2", "Kanyawara (wild)" =
"chartreuse3"))
curve
```

 # c.5) Leipzig (captive) and Waibira (wild)

```
xdata2 <- xdata[xdata$Origin=="Leipzig (captive)" |  xdata$Origin=="Waibira (wild)", ]

curve=ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) +
stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10,
```

```
size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=4.101562), colour= "cadetblue2", size=1.1) + geom_vline(data=xdata,
aes(xintercept=1.953125), colour= "blueviolet", size=1.1) + scale_colour_manual(values=
c("Leipzig (captive)" = "cadetblue2", "Waibira (wild)" = "blueviolet")) +
scale_fill_manual(values= c("Leipzig (captive)" = "cadetblue2", "Waibira (wild)" =
"blueviolet"))
curve
```

 # c.6) Kanyawara (wild) and Waibira (wild)

```
xdata2 <- xdata[xdata$Origin=="Kanyawara (wild)" |  xdata$Origin=="Waibira (wild)", ]

curve= ggplot(xdata2, aes(x=Frequency, y=Power, colour= Origin, fill=Origin)) +
stat_summary(fun.data = "mean_cl_boot",  geom= "smooth", se= TRUE, alpha=0.10,
size=1.1) + labs(x="Frequency (Hz)", y="Standardized spectral power") +
theme_classic(base_size = 33, base_family = "", base_line_size=1.1, base_rect_size=1.1) +
geom_vline(data=xdata, aes(xintercept=2), linetype="dashed", size=1.1) +
geom_vline(data=xdata, aes(xintercept=7), linetype="dashed", size=1.1) +
scale_x_continuous (breaks=seq(0,12,1)) + geom_vline(data=xdata,
aes(xintercept=2.856445), colour= "chartreuse3", size=1.1) + geom_vline(data=xdata,
aes(xintercept=1.953125), colour= "blueviolet", size=1.1) + scale_colour_manual(values=
c("Kanyawara (wild)" = "chartreuse3", "Waibira (wild)" = "blueviolet")) +
scale_fill_manual(values= c("Kanyawara (wild)" = "chartreuse3", "Waibira (wild)" =
"blueviolet"))
curve
```

6. **Find the peak of the relevant curves.**

 # Code (R):

```
gb <- ggplot_build(curve)
```

```
exact_x_value_of_the_curve_maximum <- gb$data[[1]]$x[which.max(gb$data[[1]]$y)]
curve + geom_vline(xintercept=exact_x_value_of_the_curve_maximum)
exact_x_value_of_the_curve_maximum
```

7. **Prepare .csv file where column “Peak”= peak (Hz) of each individual bout; column “Individual” = ID of the chimpanzee who produced the bout; column “Origin” = origin of the emitter (i.e., “Edinburgh (captive)”; “Leipzig (captive)”; “Waibira (wild)”; “Kanyawara (wild)”, in this order as to allow for correct contrast set up). The peak of each individual bout is the result obtained from step 3. For the bouts whose dominant frequency = ~1Hz, we selected the peak corresponding to the rate of open close mouth cycles as indicated by the bout’s time-series (S1 supplementary material); to do this we found the frequency within the area of the peak to which the largest value of Power (i.e., the y values obtained from step 3) corresponded to. In these cases, when the largest value of Power was shared by two consecutive values of frequency, we considered the dominant frequency of the bout to be the mean between the two frequency values.**

8. **Build generalized linear mixed model with contrasts to test for differences between lip-smack rhythm of captive vs wild populations.**

```
# Code (R):

# Load dataset

boutpeak<-read.csv2("chimpanzee_lipsmack_individual bout peak.csv",h=T, sep=",")
str (boutpeak)
boutpeak$Peak<-as.numeric(as.character(boutpeak$Peak))
library(lme4)
library(MASS)

# Set up generalized linear mixed model with contrasts

mat<-rbind(c(0.795,0.205,-0.467,-0.533)) #set up contrasts
        #Edinburgh contributes 79.5% of the number of Peak measures of captive populations
        #Leipzig contributes 20.5% of the number of Peak measures of captive populations
```

```
    #Kanyawara contributes 46.7% of the number of Peak measures of wild populations
    #Waibira contributes 53.3% of the number of Peak measures of wild populations
cMat<-ginv(mat) #set up contrasts: create inverted contrast matrix
colnames(cMat) <- c("Edinburgh, Leipzig, vs Kanyawara, Waibira")
model <-glmer(Peak~Origin+(1|Individual), contrasts = list(Origin = cMat), data=boutpeak,
family=Gamma(link = "inverse"))
summary(model)
```