

```

library(lattice)

#read in beta, gamma and EMG waves from the input dataset
dir <- getwd()
ourdata <- read.csv(paste(dir,"/pt7-21_combined_ave_%_-4s_
+2s_Rinput.csv",sep=""),head=T,sep=",")

F3data <- ourdata[1:17,]
F3beta <- colSums(as.matrix(F3data[2:5, 3:603]))/4
F3gamma <- colSums(as.matrix(F3data[6:17,3:603]))/12

C3data <- ourdata[18:34,]
C3beta <- colSums(as.matrix(C3data[2:5 ,3:603]))/4
C3gamma <- colSums(as.matrix(C3data[6:17,3:603]))/12

P3data <- ourdata[35:51,]
P3beta <- colSums(as.matrix(P3data[2:5 ,3:603]))/4
P3gamma <- colSums(as.matrix(P3data[6:17,3:603]))/12

O1data <- ourdata[52:68,]
O1beta <- colSums(as.matrix(O1data[2:5 ,3:603]))/4
O1gamma <- colSums(as.matrix(O1data[6:17,3:603]))/12

F4data <- ourdata[69:85,]
F4beta <- colSums(as.matrix(F4data[2:5, 3:603]))/4
F4gamma <- colSums(as.matrix(F4data[6:17,3:603]))/12

C4data <- ourdata[86:102,]
C4beta <- colSums(as.matrix(C4data[2:5 ,3:603]))/4
C4gamma <- colSums(as.matrix(C4data[6:17,3:603]))/12

P4data <- ourdata[103:119,]
P4beta <- colSums(as.matrix(P4data[2:5 ,3:603]))/4
P4gamma <- colSums(as.matrix(P4data[6:17,3:603]))/12

O2data <- ourdata[120:136,]
O2beta <- colSums(as.matrix(O2data[2:5 ,3:603]))/4
O2gamma <- colSums(as.matrix(O2data[6:17,3:603]))/12

EMG1data <- ourdata[137:153,]
EMG1avg <- colSums(as.matrix(EMG1data[3:17,3:603]))/15
EMG2data <- ourdata[154:170,]
EMG2avg <- colSums(as.matrix(EMG2data[3:17,3:603]))/15

#set a matrix to save results;
results <- matrix(rep(NA,60), ncol=6,nrow=10,byrow=T,

dimnames=list(c("F3beta","C3beta","P3beta","O1beta","EMG2",

"F4beta","C4beta","P4beta","O2beta","EMG1"),

```

```

c("baseline_mean","baseline_SD","criterion25SD","OnsetLatency","PeakLa
tency","PeakAmp"))
results

wavelist <- rbind(F3beta,C3beta,P3beta,O1beta,EMG2avg,
                 F4beta,C4beta,P4beta,O2beta,EMG1avg)

for (i in 1:10){
  tempWv <- wavelist[i,]
  #peak amplitude
  results[i,6]<-max(tempWv)
  #peak latency
  peakColIndex<-which(tempWv==max(tempWv)) #the column index;
  results[i,5]<-colnames(wavelist)[peakColIndex]

  #onset latency;
  mean0 <- mean(tempWv[1:201])
  results[i,1] <- round(mean0,3)
  sd0 <- sd(tempWv[1:201])
  results[i,2] <- round(sd0,3)
  thresh25SD <- mean0+2.5*sd0
  results[i,3] <- round(thresh25SD,3)

  #Onset latency was defined as the time point at which
  #[1] the waveform's amplitude exceeded a criterion,
  #[2] the amplitude sustained over the criterion for at least 100ms
  after this time point, and

  threshC<-thresh25SD
  allpoints<-NULL
  allpoints<-which(tempWv[1:peakColIndex]>threshC)
  allpointsFlag <- rep(NA,length(allpoints))
  names(allpointsFlag) <- names(allpoints)

  for(j in 1:length(allpoints)){
    #check if the next 100ms are above the threshold;
    flag=T;
    for(k in 1:10){
      flag <- flag & (tempWv[allpoints[j]+k]>threshC)
    }
    allpointsFlag[j]=flag;
  }

  #Onset Latency satisfying [1] and [2]
  results[i,4]<-names(which(allpointsFlag==T))[1]
}

results

```

