R CODE

library(emuR)

emuR::convert\_TextGridCollection(dir="C:\\rec\\KH",dbName="KH",targetDir="C:/rec")

KH <- load\_emuDB("C:\\rec\\KH\_emuDB")

#to know the

list\_levelDefinitions(KH)

list\_linkDefinitions(KH)

add\_linkDefinition(KH,type = "ONE\_TO\_MANY",superlevelName = "ORT",sublevelName = "KAN")

add\_linkDefinition(KH,type = "ONE\_TO\_MANY",superlevelName = "KAN",sublevelName = "MAU")

autobuild\_linkFromTimes(KH,"ORT","KAN")

autobuild\_linkFromTimes(KH,"KAN","MAU")

serve(KH)

add\_labelGroup(KH,"shortvowels",c("a","e","i","o","u"))

q1\_KH <- emuR::query(KH,query="[MAU==shortvowels]")

q1\_KH.trackdata <- get\_trackdata(KH,seglist = q1\_KH,ssffTrackName = "F0")

serve(KH)

#list\_ssffTrackDefinitions(KH)

plot(q1\_KH.trackdata)

plot(q1\_KH.trackdata[3],type="l")

summary(q1\_KH)

summary(q1\_KH.trackdata)

head(q1\_KH)

head(q1\_KH.trackdata)

names(q1\_KH.trackdata)

table(q1\_KH$bundle)

#create a query for vowels

q1\_KH <- emuR::query(KH,query="[MAU==shortvowels]")

#creating DFT spectrum for the trackdata

q1\_KH\_DFT <- get\_trackdata(KH,seglist = q1\_KH,ssffTrackName= "dft")

#cut the spectrum

q1\_KH\_DFT\_cut <- dcut(q1\_KH\_DFT, .5,prop=TRUE)

View(q1\_KH\_DFT\_cut)

q1\_KH\_labs <- label(q1\_KH)

plot(q1\_KH\_DFT\_cut[1,],q1\_KH\_labs[1],xlab="Frequency",ylab="Amplitude")

fapply(q1\_KH\_DFT\_cut[,0:8000],mean)

#to find the high frequency energy ratio

s1000=fapply(q1\_KH\_DFT\_cut[,0:1000],sum,power=T)

stotal=fapply(q1\_KH\_DFT\_cut,sum,power=T)

#function to calculate the high frequency energy ratio

ratio<- function(tot,lower)

{

s=tot-lower

ratio1=s/tot

return(ratio1)

}

o <- ratio(stotal,s1000)

write.csv(o,file="C:\\rec\\KH\\power.csv",append = T,sep = ',',row.names= F,col.names = F)

#function to find out the peak values

peaks <- function(series, span = 3, do.pad = TRUE) {

if((span <- as.integer(span)) %% 2 != 1) stop("'span' must be odd")

s1 <- 1:1 + (s <- span %/% 2)

if(span == 1) return(rep.int(TRUE, length(series)))

z <- embed(series, span)

v <- apply(z[,s1] > z[, -s1, drop=FALSE], 1, all)

if(do.pad) {

pad <- rep.int(FALSE, s)

c(pad, v, pad)

} else v

}

#applying the peak function to all segments

for(i in 1:828){

df <- q1\_KH\_DFT\_cut[i,]

temp <- as.vector(df)

temp1 <- peaks(temp,11,TRUE)

which(temp1==TRUE)

t <- which(temp1==TRUE)

s <- t[1:3]

f <- (s-1)\*31.25

p <- as.vector(df)

p1 <- p[s]

h1 <- slope1(p1[1],p1[2],f[1],f[2])

h2 <- slope2(p1[2],p1[3],f[2],f[3])

hh[i] <- as.array(h1)

hhh2[i] <- as.array(h2)

}

write.csv(hh,file="C:\\rec\\KH\\test.csv",append = T,sep = ',',row.names= F,col.names = F)

write.csv(hhh2,file="C:\\rec\\KH\\test1.csv",append = T,sep = ',',row.names= F,col.names = F)

#function to calculate slope1 value

slope1<- function(p,q,r,s)

{

h1slope=((p-q)/(r-s))

return(h1slope)

}

#function to calculate slope2 values

slope2<-function(p1,q1,r1,s1)

{

h2slope=((p1-q1)/(r1-s1))

return(h2slope)

}

utt(q1\_KH[1,])

speakers <- substring(utt(q1\_KH),1,2)

p\_types <- substring(utt(q1\_KH),7,8)

table(substring(utt(q1\_KH),1,2))

(substring(utt(q1\_KH),7,8))

table(substring(utt(q1\_KH),4,8))

write.csv(speakers,file="C:\\rec\\KH\\speakers.csv",append = T,sep = ',',row.names= F,col.names = F)

write.csv(p\_types,file="C:\\rec\\KH\\p\_types.csv",append = T,sep = ',',row.names= F,col.names = F)

kh\_all<-read.csv("C:\\rec\\KH\\kapahaka\_combined.csv")

#plotting the Mean and standard Deviation

#aggregating by mean

agg <- aggregate(list(kh\_all$h1,kh\_all$h2,kh\_all$energy), by=list(kh\_all$Speakers, kh\_all$p\_types), mean)

colnames(agg) <- c("speaker","performance\_types","H1","H2","energy\_ratio")

#aggregating by standard deviation

agg\_sd <- aggregate(list(kh\_all$h1,kh\_all$h2,kh\_all$energy), by=list(kh\_all$Speakers, kh\_all$p\_types),sd)

colnames(agg\_sd) <- c("speaker","performance\_types","H1\_sd","H2\_sd","energy\_ratio\_sd")

#aggregating for vowel count

agg1 <- aggregate(list(kh\_all$vowels), by=list(id11 = kh\_all$Speakers, id2= kh\_all$p\_types, kh\_all$vowels), length)

colnames(agg1) <- c("speaker","performance\_types","vowel","count")

#rename this

#barplot for count

ggplot(data=agg1, aes(x=vowel, y=count, fill=(performance\_types))) +

geom\_bar(stat="identity", position="dodge") + ggtitle("count of each vowels vs performance types")

#barplot vowels vs performance types for female

f1<- agg1[ which(agg1$speaker=='f1'),]

ggplot(data=f1, aes(x=vowel, y=count, fill=(performance\_types))) +

geom\_bar(stat="identity", position="dodge") + ggtitle("Barplot vowels vs performance types for f1")

#barplot vowels vs different performance types for male2

m2 <- agg1[ which(agg1$speaker=='m2'),]

ggplot(data=m2, aes(x=vowel, y=count, fill=(performance\_types))) +

geom\_bar(stat="identity", position="dodge") + ggtitle("Barplot vowels vs performance types for m2")

#barplot vowels vs different performance types for male3

m3 <- agg1[ which(agg1$speaker=='m3'),]

ggplot(data=m3, aes(x=vowel, y=count, fill=(performance\_types))) +

geom\_bar(stat="identity", position="dodge") + ggtitle("Barplot vowels vs performance types for m3")

#barplot vowels vs different performance types for male4

m4 <- agg1[ which(agg1$speaker=='m4'),]

ggplot(data=m4, aes(x=vowel, y=count, fill=(performance\_types))) +

geom\_bar(stat="identity", position="dodge") + ggtitle("Barplot vowels vs performance types for m4")

library(ggplot2)

#barplot for mean h1 for all speakers

ggplot(data=agg, aes(y=agg$H1, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot Mean H2 vs performance types for all speakers")

ggplot(data=agg, aes(y=abs(agg$H1), x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot Absolute Mean H2 vs performance types for all speakers")

#barplot for mean h2 for all speakers

ggplot(data=agg, aes(y=agg$H2, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot Mean H2 vs performance types for all speakers")

ggplot(data=agg, aes(y=abs(agg$H2), x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot Absolute Mean H2 vs performance types for all speakers")

#barplot for mean energy ratio for all speakers

ggplot(data=agg, aes(y=agg$energy\_ratio, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot Mean energy\_ratio vs performance types for all speakers")

#barplot for sd h1 for all speakers

ggplot(data=agg\_sd, aes(y=agg$H1\_sd, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot standard deviation H1 vs performance types for all speakers")

#barplot for sd h2 for all speakers

ggplot(data=agg\_sd, aes(y=agg$H2\_sd, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot standard deviation H2 vs performance types for all speakers")

#barplot for sd energy\_ratio for all speakers

ggplot(data=agg\_sd, aes(y=agg$energy\_ratio\_sd, x=agg$performance\_types, fill=agg$speaker)) +

geom\_bar(stat="identity", position="dodge")+ ggtitle("Barplot standard deviation energy ratio vs performance types for all speakers")

#boxplot for H1 for the performance types of all speakers

ggplot(data=kh\_all, aes(y=h1, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H1 for different performance types - all speakers")

ggplot(data=kh\_all, aes(y=h2, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H2 for different performance types - all speakers")

p<- kh\_all[ which(kh\_all$energy<"0.05"),]

ggplot(data=p, aes(y=energy, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("HFE Ratio for different performance types - all speakers")

#subset data

f1\_all<- kh\_all[ which(kh\_all$Speakers=='f1'),]

ggplot(data=f1\_all, aes(y=h1, x=p\_types)) +

geom\_boxplot()+ ggtitle("H1 for different performance types - f1")

#boxplot for H1 for the performance types of m2 speakers

m2\_all<- kh\_all[ which(kh\_all$Speakers=='m2'),]

ggplot(data=m2\_all, aes(y=h1, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H1 for different performance types - m2 speakers")

# boxplot for H1 for the performance types of m3 speakers

m3\_all<- kh\_all[ which(kh\_all$Speakers=='m3'),]

ggplot(data=m3\_all, aes(y=h1, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H1 for different performance types - m3 speakers")

# boxplot for H1 for the performance types of m4 speakers

m4\_all<- kh\_all[ which(kh\_all$Speakers=='m4'),]

ggplot(data=m4\_all, aes(y=h1, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H1 for different performance types - m4 speakers")

# boxplot for H2 for the performance types of m3 speakers

f11\_all<- kh\_all[ which(kh\_all$Speakers=='f1'),]

ggplot(data=f11\_all, aes(y=h2, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H2 for different performance types - f1 speakers")

# boxplot for H2 for the performance types of m3 speakers

m22\_all<- kh\_all[ which(kh\_all$Speakers=='m2'),]

ggplot(data=m22\_all, aes(y=h2, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H2 for different performance types - m2 speakers")

# boxplot for H2 for the performance types of m3 speakers

m33\_all<- kh\_all[ which(kh\_all$Speakers=='m3'),]

ggplot(data=m33\_all, aes(y=h2, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H2 for different performance types - m3 speakers")

# boxplot for H2 for the performance types of m3 speakers

m44\_all<- kh\_all[ which(kh\_all$Speakers=='m4'),]

ggplot(data=m44\_all, aes(y=h2, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("H2 for different performance types - m4 speakers")

# boxplot for energy ratio for the performance types of m1 speakers

f111\_all<- kh\_all[ which(kh\_all$Speakers=='f1'),]

ggplot(data=f111\_all, aes(y=energy, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("Energy Ratio for different performance types - f1 speakers")

# boxplot for energy ratio for the performance types of m2 speakers

m222\_all<- kh\_all[ which(kh\_all$Speakers=='m2'),]

ggplot(data=m222\_all, aes(y=energy, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("Energy Ratio for different performance types - m2 speakers")

# boxplot for energy ratio for the performance types of m3 speakers

m333\_all<- kh\_all[ which(kh\_all$Speakers=='m3'),]

ggplot(data=f111\_all, aes(y=energy, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("Energy Ratio for different performance types - m3 speakers")

# boxplot for energy ratio for the performance types of m3 speakers

m444\_all<- kh\_all[ which(kh\_all$Speakers=='m4'),]

ggplot(data=m444\_all, aes(y=energy, x=Speakers, fill=p\_types)) +

geom\_boxplot()+ ggtitle("Energy Ratio for different performance types - m4 speakers")