```
library(emu)
library(lattice)
library(latticeExtra)
library(RCurl)
source(file.path(pfadu, "readtrackfromurl.R"))
source(file.path(pfadu, "ellipsefun.R"))

source(file.path(pfadu, "tracktodf.R"))

# Segmentlisten und Label-Dateien
# Eventuell "ERCPolish" statt "ERCPolish13"
fric.s = emu.query("ERCPolish13", "*", "Phoneme = s | sz | si & Start(Word, Phoneme)=
1")
#
# oder
fric.s = read.emusegs(file.path(paste(pfadu, "polnischphys", sep="/"), "fric.s.txt"))
# following segment
fric.l = label(fric.s)
vow.s = emu.requery(fric.s, "Phoneme", "Phoneme", seq=1)
# oder
vow.s = read.emusegs(file.path(paste(pfadu, "polnischphys", sep="/"), "vow.s.txt"))
vow.l = label(vow.s)
# Promptliste einlesen
p.df = read.table(file.path(paste(pfadu, "polnischphys", sep="/"), "p.df.txt"))
all(utt(fric.s) == p.df$U)
# [1] TRUE

# Gaumendaten einlesen
gaumen = read.table(file.path(paste(pfadu, "polnischphys", sep="/"), "gaumen.txt"))

# Physiologische Daten
tty = emu.track(fric.s, "tt_posz")
ttx = emu.track(fric.s, "tt_posy")
tby = emu.track(fric.s, "tb_posz")
tbx = emu.track(fric.s, "tb_posy")

# oder
tty = readtrackfromurl("tty.txt",file.path(paste(pfadu, "polnischphys", sep="/")) ,
pfad)
ttx = readtrackfromurl("ttx.txt",file.path(paste(pfadu, "polnischphys", sep="/")) ,
pfad)
tby = readtrackfromurl("tby.txt",file.path(paste(pfadu, "polnischphys", sep="/")) ,
pfad)
tbx = readtrackfromurl("tbx.txt",file.path(paste(pfadu, "polnischphys", sep="/")) ,
pfad)

# In eine Trackdatei einbinden
phys = cbind(tty, ttx, tby, tbx)

# Trackdatei in ein Data-Frame umsetzen
phys.df = tracktodf(phys)
names(phys.df)[1:4] = c("TTY", "TTX", "TBY", "TBX")
```

```
# Labels parallel dazu bauen
fric.df.l = rep(fric.l, table(phys.df$segno))
vow.df.l = rep(vow.l, table(phys.df$segno))
acc.df.l = rep(p.df$Acc, table(phys.df$segno))
phys.df = cbind(phys.df, K = factor(fric.df.l), V = factor(vow.df.l), Acc = factor
(acc.df.l))

# Dies ist für die Farbkodierung in xyplot()
fric.col = as.numeric(factor(fric.l))
fric.leg = list(
text = list(unique(fric.l)),
lines = list(col = unique(fric.col)))

vow.col = as.numeric(factor(vow.l))
vow.leg = list(
text = list(unique(vow.l)),
lines = list(col = unique(vow.col)))


xyplot(TTY ~ times | V, group = segno, data = phys.df, type=c("l", "g"),
key = fric.leg,
par.settings = list(
superpose.line = list(col = fric.col)
))

xyplot(TTY ~ times | V * Acc, group = segno, data = phys.df, type=c("l", "g"),
key = fric.leg, main = "TTY",
par.settings = list(
superpose.line = list(col = fric.col)
))

xyplot(TTY ~ times | V * Acc, group = segno, data = phys.df, type=c("l", "g"),
key = fric.leg, main = "TTY",
par.settings = list(
superpose.line = list(col = fric.col)
))

xyplot(TTY ~ TTX | V * Acc, group = segno, data = phys.df, type=c("l", "g"),
key = fric.leg, main = "TTX x TTY",
par.settings = list(
superpose.line = list(col = fric.col)
))

xyplot(TTY ~ TTX | K * Acc, group = segno, data = phys.df, type=c("l", "g"),
key = vow.leg, main = "TTX x TTY",
par.settings = list(
superpose.line = list(col = vow.col)
))


#####
# Mit überlagertem Gaumen
xlim = c(0, 60); ylim = c(-10, 22)
```

```
p = function(x,y,...) {
panel.xyplot(x, y, ...)
panel.points(gaumen[,2], gaumen[,3])
}

xyplot(TTY ~ TTX | V * Acc, group = segno, data = phys.df, type=c("l", "g"),
key = fric.leg, main = "TTX x TTY", panel = p, xlim=xlim, ylim = ylim,
par.settings = list(
superpose.line = list(col = fric.col)
))


### Mittelwerte
phys.dfm = with(phys.df, aggregate(phys.df[,1:4], list(times, K, V, Acc), mean))
names(phys.dfm)[1:4] = c("times", "K", "V", "Acc")
xyplot(TTY ~ times | V * Acc, groups = K, data = phys.dfm, type = c("l", "g"), auto.
key=T)

xyplot(TTY ~ TTX | V * Acc, groups = K, data = phys.dfm, type = c("l", "g"),
auto.key=T, panel=p, xlim=xlim, ylim=ylim)


pneu = function(x,y, ...) {
panel.xyplot(x, y, ...)
if(panel.number()==1)
{
temp = phys.dfm$times == 0 & phys.dfm$V == "a" & phys.dfm$Acc == "ACC"
panel.points(gaumen[,2], gaumen[,3])
panel.points(phys.dfm$TTX[temp], phys.dfm$TTY[temp], col = "black")
}
if(panel.number()==2)
{
temp = phys.dfm$times == 0 & phys.dfm$V == "e" & phys.dfm$Acc == "ACC"
panel.points(gaumen[,2], gaumen[,3])
panel.points(phys.dfm$TTX[temp], phys.dfm$TTY[temp], col = "black")
}
if(panel.number()==3)
{
temp = phys.dfm$times == 0 & phys.dfm$V == "o" & phys.dfm$Acc == "ACC"
panel.points(gaumen[,2], gaumen[,3])
panel.points(phys.dfm$TTX[temp], phys.dfm$TTY[temp], col = "black")
}
}
xyplot(TTY ~ TTX | V * Acc, groups = K, data = phys.dfm, type = c("l", "g"),
auto.key=T, panel=pneu, xlim=xlim, ylim=ylim)
```