

```
library(emu)
library(lattice)
# Um Trackdateien aus einer Webseite einzulesen
library(RCurl)
source(file.path(pfadu, "readtrackfromurl.R"))
# Für Ellipsen
library(latticeExtra)
source(file.path(pfadu, "ellipsefun.R"))

##### Segmentliste der
deutschen Frikative
# Entweder timetable Sprachdatenbank und spektrale Daten berechnen...
dor.s = emu.query("timetable", "*", "Phonetic = C | x")
dor.l = label(dor.s)
# Nach Berechnung von spektralen Daten in Emu
dor.dft = emu.track(dor.s, "dft")

# Oder
dor.s = read.emusegs(file.path(paste(pfadu, "dorsal", sep="/"), "dor.s.txt"))
dor.l = scan(file.path(paste(pfadu, "dorsal", sep="/"), "dor.lab.txt"), what="")
dor.dft = readtrackfromurl("dor.dft.txt", file.path(paste(pfadu, "dorsal", sep="/"))) ,
pfad)

#####
Spektrale Daten zum zeitlichen Mittelpunkt

#####
Spektrale Objekte
# class(), is.spectral(), as.spectral()
# trackfreq() die Frequenzen, zu denen die Spektralkomponente vorkommen

#####
Spektrale Eigenschaften
# fnum: Anzahl der Spektralkomponente

# N (FFT-Length)

# fmax: Nyquist-Frequenz

# fs: die Abtastrate des Zeitsignals

# fres: Frequenz-Auflösung

# Dauer des FFT-Fensters

##### Spektrale Abbildung und Indizierung von
spektralen Daten
# plot(); spec[,1000:2000] (1000-2000 Hz)
```

```
#####  
#Die Mel- und Bark-Skalen  
# mel(), bark()  
  
#####  
polnische Frikative  
# Entweder:  
pfric.s = emu.query("pol13b", "*", "[#phon = s | si | sz & Start ( word,phon ) = 1 ->  
phon = e | o | a ]")  
pvow.s = emu.query("pol13b", "*", "[phon = s | si | sz & Start ( word,phon ) = 1 -> #  
phon = e | o | a ]")  
pfric.dft = emu.track(pfric.s, "dft")  
  
# Oder:  
pfric.s = read.emusegs(file.path(paste(pfadu, "polb", sep="/"), "pfric.s.txt"))  
pvow.s = read.emusegs(file.path(paste(pfadu, "polb", sep="/"), "pvow.s.txt"))  
# Dauert eventuell 2-3 Minuten  
pfric.dft = readtrackfromurl("pfric.dft.txt",file.path(paste(pfadu, "polb", sep="/"))  
, pfad)
```