

```

library(emu)
source(file.path(pfadu, "cendiff.R"))

# 0. Funktion für Anwendung auf Datenbanken:
# trackinfo()

# 1. Segmentlisten
#####
# Wort-initiale /k/s, Sprachdatenbank ema5, alle Sprecher
k.s = emu.query("ema5", "*", "Segment=k & Start(Word, Segment)=1")
# Wenn nicht zugänglich, dann:
# k.s = read.emusegs(file.path(paste(pfadu, "psych5", sep="/"), "k.s.txt"))

# Die danachkommende Aspiration
h.s = emu.requery(k.s, "Segment", "Segment", seq=1)
# h.s = read.emusegs(file.path(paste(pfadu, "psych5", sep="/"), "h.s.txt"))

# Reihenfolge von raise-lower der TT-Ebene
tip.s = emu.requery(k.s, "Segment", "TT")
# tip.s = read.emusegs(file.path(paste(pfadu, "psych5", sep="/"), "tip.s.txt"))

# Wie oben der TB-Ebene
body.s = emu.requery(k.s, "Segment", "TB")
# body.s = read.emusegs(file.path(paste(pfadu, "psych5", sep="/"), "body.s.txt"))

# Nur 'raise' der TB-Ebene
tbraise.s = emu.query("ema5", "*", "[TB=raise ^ Start(Word, Segment)=1]")
# tbraise.s = read.emusegs(file.path(paste(pfadu, "psych5", sep="/"), "tbraise.s.
txt"))

# Etikettierungen: 'n' oder 'l'
son.lab = emu.requery(k.s, "Segment", "Segment", seq=2, j=T)
# son.lab = scan(file.path(paste(pfadu, "psych5", sep="/"), "son.lab.txt"), what="")

# Die Wortetikettierungen von /k/
word.lab = emu.requery(k.s, "Segment", "Word", j=T)
# word.lab = scan(file.path(paste(pfadu, "psych5", sep="/"), "word.lab.txt"), what=
"")

# 2. Funktionen für Segmentlisten
#####
# Indizierung von Segmentlisten: wie für Matrizen

# Allgemeine Funktionen in R mit Anwendung auf Segmentlisten: class(), summary(),
nrow(), ncol(), dim(), rbind()
# Funktionen für Segmentlisten und Trackdateien: start(), end(), dur()
# Funktionen nur für Segmenlisten: is.seglist(), label(), utt()

# 3. Trackdateien
#####
# Zungenspitzen-Y-Werte (von raise-lower in TT)
tip.tt = emu.track(tip.s, "tt_posz")

```

```
# Alternativ
# library(RCurl)
# ggf. zuerst install.packages("RCurl")
# source(file.path(pfadu, "readtrackfromurl.R"))
# tip.tt = readtrackfromurl("tip.tt.txt",file.path(paste(pfadu, "psych5", sep="/"))
, pfad)

# Zungendorsum-Y-Werte (von raise-lower in TT)
tip.tb = emu.track(tip.s, "tb_posz")
# tip.tb = readtrackfromurl("tip.tb.txt",file.path(paste(pfadu, "psych5", sep="/"))
, pfad)

# Zungendorsum-Y-Werte (von raise-lower in TB)
body.tb = emu.track(body.s, "tb_posz")
# body.tb = readtrackfromurl("body.tb.txt",file.path(paste(pfadu, "psych5", sep="/"))
) , pfad)

# Kiefer-Y-Werte von raise-lower in TT
jaw = emu.track(tip.s, "jw_posz")
# jaw = readtrackfromurl("jaw.txt",file.path(paste(pfadu, "psych5", sep="/")) , pfad
)

# Zungendorsum-Y-Werte (von raise in TB)
tbraise.tb = emu.track(tbraise.s, "tb_posz")
# tbraise.tb = readtrackfromurl("tbraise.tb.txt",file.path(paste(pfadu, "psych5",
sep="/")) , pfad)

# 4. Funktionen für Trackdateien
#####
# Indizierung von Trackdateien: wie für Matrizen
# Funktionen in R mit Anwendung auf Trackdateien: class(), summary(),
# nrow(), ncol(), dim(), rbind(), cbind(), plot(); auch Arith, Compare, Math, Math2
Funktionen in help(Ops)
# Funktionen für Segmentlisten und Trackdateien: start(), end(), dur()
# Funktionen nur für Trackdateien: is.trackdata(), frames(), tracktimes(), dcut(),
dplot(), trapply()
```