Package ‘cyphid’

February 19, 2015

Type Package
Title Cycle and Phase Identification for mastication data
Version 1.1
Date 2013-04-04
Author Elizabeth Crane
Maintainer Elizabeth Crane <bcrane@umich.edu>
Depends fda
Description This library contains a primary function that divides
chewing sequences in cycles and cycles into phases. See
get.all.breaks for an example.
License GPL (>= 2)
LazyLoad yes
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2013-05-22 07:52:43

R topics documented:

cyphid-package ................................................. 2
cyphid.function.help ........................................... 2
get.all.breaks .................................................. 3
get.peaks ...................................................... 4
jaw ............................................................. 5

Index 6
cyphid-package  

Identifies chewing cycles and phases

Description

The primary function in this library automates the process of dividing chewing sequences into cycles and cycles into phases.

Details

Package: cyphid  
Type: Package  
Version: 1.1  
Date: 2013-04-04  
License: GPL-2  
LazyLoad: yes

The primary function in this package is get.all.breaks (See example below).

Author(s)

Maintainer: Elizabeth Crane <bcrane@umich.edu>

Examples

# Run primary function for dividing sequences into cycles and cycles into phases  
JawBreaks40 <- get.all.breaks(jaw, window=40)

# Check window based on cycle durations.  
cycledurs <- get.cycle.durations(JawBreaks40$cyclemat)  
win <- get.window(cycledurs)  
win

# Rerun with modified window  
JawBreaks27 <- get.all.breaks(jaw, window=27)

# Plot the output  
plot(jaw[,1])  
abline(v=JawBreaks27$openbreaks[,1])

Description

This function is a subroutine called by the function get.all.breaks in the cyphid package.

Subroutine
get.all.breaks

Identify all breakpoints.

Description

This function divides chewing sequences into individual cycles and identifies the phase transitions within each cycle.

Usage

get.all.breaks(dataset, CycleBreaks = NULL, window = NULL)

Arguments

dataset A matrix of chewing sequences. Each column represents a single sequence.
CycleBreaks A matrix of frame numbers used to force a sequence to break at predetermined locations. Each column represents the break points for a single sequence.
window The frame limit for which no two cycle breaks can occur.

Value

This function returns 8 objects.

openbreaks, closebreaks, FCSC, and SOFO are matrices with one column for each sequence. Columns contain the frame numbers associated with either the open, close, FCSC, or SOFO locations for each sequence.

closebreaks see openbreaks

FCSC see openbreaks

SOFO see openbreaks
cyclemat cyclemat is a matrix with one column for each cycle. Columns contain the displacement values for each cycle.
close.cycle close.cycle, FCSC.cycle, SOFO.cycle is a vector with a value for each cycle. The values represent either the location of close, FCSC, or SOFO relative to the cycle.

FCSC.cycle see close.cycle

SOFO.cycle see close.cycle

Examples

# Run primary function for dividing sequences into cycles and cycles into phases
JawBreaks40 <- get.all.breaks(jaw, window=40)

# Check window based on cycle durations.
cycledurs <- get.cycle.durations(JawBreaks40$cyclemat)
win <- get.window(cycledurs)
get.peaks

win

# Rerun with modified window
JawBreaks27 <- get.all.breaks(jaw, window=27)

# Plot the output
plot(jaw[,1])
abline(v=JawBreaks27$openbreaks[,1])

---

get.peaks  

A function to identify peaks within a dataset.

Description

A subroutine called by functions within the cyphid package. get.peaks is originally from the package msProcess version 1.0.5. The package was archived on 2012-09-21 and is unavailable as a dependency for cyphid. The archived package and original code can be obtained from http://cran.r-project.org/src/contrib/Archive/msProcess/

Usage

get.peaks(x, span = 40)

Arguments

x
span

Examples

# Should be DIRECTLY executable !! ----
# >>> == Define data, use random,
# >>> or do help(data=index) for the standard data sets.

# The function is currently defined as
function (x, span = 40)
{
  z <- embed(rev(as.vector(x)), dim = span)
  z <- z[rev(seq(nrow(z))), ]
  s <- span/2
  v <- max.col(z, ties.method = "first") == 1 + s
  z <- c(rep(FALSE, s), v)
  ans <- c(z[1:(length(z) - s)], rep(FALSE, span - 1))
  nx <- NCOL(x)
  if (nx > 1)
    matrix(ans, ncol = nx)
  else ans
}
Description
This data set provides displacement data during gum chewing. Motion capture was used to collect the kinematic data at a 60 Hz sampling frequency. Two chewing sequences are provided.

Usage
data(jaw)

Format
The format is: num [1:1200, 1:2] -0.462 -0.48 -0.506 -0.534 -0.539 -0.537 -0.548 -0.562 -0.574 -0.582 ... - attr(*, "dimnames")=List of 2 ..$ : NULL ..$ : chr [1:2] "vecY" "vecY.1"

References

Examples
data(jaw)
plot(jaw[,1], xlab="frame number", ylab="displacement")
Index

+ Topic datasets
  jaw, 5
+ Topic package
  cyphid-package, 2

add.col (cyphid.function.help), 2
BWFiter (cyphid.function.help), 2
check.for.sequence.endpoint.breaks (cyphid.function.help), 2
create.transition.location.matrix (cyphid.function.help), 2
cyphid (cyphid-package), 2
cyphid-package, 2
cyphid.function.help, 2

do.transitions.exist (cyphid.function.help), 2

g et.all.breaks, 3
g et.all.breaks.traditional (cyphid.function.help), 2
g et.breaks.vector (cyphid.function.help), 2
g et.cycle.breaks (cyphid.function.help), 2
g et.cycle.counts (cyphid.function.help), 2
g et.cycle.durations (cyphid.function.help), 2
g et.cycle.matrix (cyphid.function.help), 2
g et.cycle.minimums (cyphid.function.help), 2
g et.peaks, 4
g et.trans (cyphid.function.help), 2
g et.transition.locations (cyphid.function.help), 2
g et.window (cyphid.function.help), 2

is.dividing.valid (cyphid.function.help), 2

jaw, 5

r m.end.NAs (cyphid.function.help), 2