Package: dydea (via r-universe)

November 4, 2024

Type Package

Title Detection of Chaotic and Regular Intervals in the Data

Version 0.1.0

Description Finds regular and chaotic intervals in the data using the 0-1 test for chaos proposed by Gottwald and Melbourne (2004) <DOI:10.1137/080718851>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

Imports Chaos01

RoxygenNote 6.1.1

Author Radek Halfar [aut, cre]

Maintainer Radek Halfar <radek.halfar@vsb.cz>

Date/Publication 2019-03-25 11:00:03 UTC

Repository https://radekhalfar.r-universe.dev

RemoteUrl https://github.com/cran/dydea

RemoteRef HEAD

RemoteSha 2a941ed90404eab55c18aafde25f0f58fbde5613

Contents

find_chaos *Find chaotic motions in the data.*

Description

Find chaotic motions in the data.

Usage

```
find_chaos(data, window_length, skip_window, skip_test01 = 1,
  test01_thresh = 0.05, find_thresh = 20)
```

Arguments

data	Analyzed data.
window_length	Length of the window for in which the 0-1 test for chaos will be computed.
skip_window	Length of the skip of the window moving in the data.
skip_test01	Length of the skip to take data for calculation the 0-1 test for chaos in the window.
test01_thresh	The threshold to decide about motion.
find_thresh	Precision of found intervals.

Value

The list of optimized chaotic motion borders.

Examples

```
# Calculate the logistic map.
cons <- 0.5
data.len <- 17000
chaos.start <- c(5536, 9768)
vec.x <- matrix(cons, data.len, 1)

vec.x[1] <- (2^0.5)/2
for (i in 2:data.len){
  # x_n+1 = r*x_n(1-x_n)
  vec.x[i] <- 3.7*vec.x[i-1]*(1-vec.x[i-1])
}
vec.x[1:(chaos.start[1]-1)] <-cons
vec.x[(chaos.start[2]+1):data.len] <-cons
tr1 <- seq(from = cons, to = vec.x[chaos.start[1]], length.out = 2001)
tr2 <- seq(from = vec.x[chaos.start[2]], to = cons, length.out = 2001)
vec.x[(chaos.start[1]-2000):chaos.start[1]] <- tr1
vec.x[chaos.start[2]:(chaos.start[2]+2000)] <- tr2

# Find chaotic intervals in vec.x and plot results.
chaotic_borders <- find_chaos(vec.x, "skip_window" = 1000,
  "window_length" = 3000, "find_thresh" = 300)
```

find_motions *Find regular and chaotic motions in the data and plots the results.*

Description

Find regular and chaotic motions in the data and plots the results.

Usage

```
find_motions(data, window_length, skip_window, skip_test01 = 1,
  test01_thresh = 0.05, find_thresh = 20)
```

Arguments

data	Analyzed data.
window_length	Length of the window for in which the 0-1 test for chaos will be computed
skip_window	Length of the skip of the window moving in the data.
skip_test01	Length of the skip to take data for calculation the 0-1 test for chaos in the window.
test01_thresh	The threshold to decide about motion.
find_thresh	Precision of found intervals.

Value

The list of optimized regular and chaotic motion borders.

Examples

```
# Calculate the logistic map.
cons <- 0.5
data.len <- 17000
chaos.start <- c(5536, 9768)
vec.x <- matrix(cons, data.len, 1)

vec.x[1] <- (2^0.5)/2
for (i in 2:data.len){
  # x_n+1 = r*x_n(1-x_n)
  vec.x[i] <- 3.7*vec.x[i-1]*(1-vec.x[i-1])
}
vec.x[1:(chaos.start[1]-1)] <-cons
vec.x[(chaos.start[2]+1):data.len] <-cons
tr1 <- seq(from = cons, to = vec.x[chaos.start[1]], length.out = 2001)
tr2 <- seq(from = vec.x[chaos.start[2]], to = cons, length.out = 2001)
vec.x[(chaos.start[1]-2000):chaos.start[1]] <- tr1
vec.x[chaos.start[2]:(chaos.start[2]+2000)] <- tr2

# Find chaotic and regular intervals in vec.x and plot results.
find_motions(vec.x, "skip_window" = 1000, "window_length" = 3000, "find_thresh" = 300)
```

`find_regularity` *Find regular motions in the data.*

Description

Find regular motions in the data.

Usage

```
find_regularity(data, window_length, skip_window, skip_test01 = 1,
  test01_thresh = 0.05, find_thresh = 20)
```

Arguments

`data`	Analyzed data.
`window_length`	Length of the window for in which the 0-1 test for chaos will be computed.
`skip_window`	Length of the skip of the window moving in the data.
`skip_test01`	Length of the skip to take data for calculation the 0-1 test for chaos in the window.
`test01_thresh`	The threshold to decide about motion.
`find_thresh`	Precision of found intervals.

Value

The list of optimized regular and chaotic motion borders.

Examples

```
# Calculate the logistic map.
cons <- 0.5
data.len <- 17000
chaos.start <- c(5536, 9768)
vec.x <- matrix(cons, data.len, 1)

vec.x[1] <- (2^0.5)/2
for (i in 2:data.len){
  # x_n+1 = r*x_n(1-x_n)
  vec.x[i] <- 3.7*vec.x[i-1]*(1-vec.x[i-1])
}
vec.x[1:(chaos.start[1]-1)] <-cons
vec.x[(chaos.start[2]+1):data.len] <-cons
tr1 <- seq(from = cons, to = vec.x[chaos.start[1]], length.out = 2001)
tr2 <- seq(from = vec.x[chaos.start[2]], to = cons, length.out = 2001)
vec.x[(chaos.start[1]-2000):chaos.start[1]] <- tr1
vec.x[chaos.start[2]:(chaos.start[2]+2000)] <- tr2

# Find regular intervals in vec.x and plot results.
regular_borders <- find_regularity(vec.x, "skip_window" = 1000,
  "window_length" = 3000, "find_thresh" = 300)
```

Index