Package 'tdsc'

October 14, 2022

Title Time Domain Signal Coding

Version 1.0.3

Description Functions for performing time domain signal coding as used in Chesmore (2001) <doi:10.1016/S0003-682X(01)00009-3>, and related tasks. This package creates the standard S-matrix and A-matrix (with variable lag), has tools to convert coding matrices into distributed matrices, provides published codebooks and allows for extraction of code sequences.

Depends R (>= 3.5.0)

License GPL-3

Language en-GB

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports data.table, methods, moments

Suggests tuneR, alluvial, covr, testthat, devtools, GA, sonicscrewdriver

NeedsCompilation no

Author Ed Baker [aut, cre] (<https://orcid.org/0000-0002-5887-9543>)

Maintainer Ed Baker <ed@ebaker.me.uk>

Repository CRAN

Date/Publication 2019-12-19 18:50:02 UTC

R topics documented:

`c2dmatrix` *Convert a coding matrix to a distributed matrix*

Description

This function converts a coding matrix of any kind into a distributed matrix as described in Farr (2007).

Usage

```
c2dmatrix(t, sf = 100)
```

Arguments

`t`	A tdsc object or a matrix
`sf`	The scaling factor

References

Farr (2007) "Automated Bioacoustic Identification of Statutory Quarantined Insect Pests". PhD thesis. University of York.

Examples

```
c2dmatrix(as.matrix(c(1,2,3,4), nrow=2))
```

`chesmore2001` *Coding Matrix from Chesmore (2001)*

Description

Coding matrix used for Orthoptera.

Usage

```
data(chesmore2001)
```

Format

Matrix

Source

QTL Archive

References

Chesmore, E David (2001). "Application of time domain signal coding and artificial neural networks to passive acoustical identification of animals". In: Applied Acoustics 62.12, pp. 1359–1374.

Examples

```
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
data(chesmore2001)
t <- tdsc(wave, coding_matrix=chesmore2001)
```

`emptyBands` *Empty Bands Discovery*

Description

Identifies unused codes across multiple S-matrices. Unused bands can be used to reduce the codebook as in Stammers (2011).

Usage

```
emptyBands(...)
```

Arguments

...	Two or more TDSC objects

References

Stammers (2011) "Audio Event Classification for Urban Soundscape Analysis". PhD thesis. University of York.

Examples

```
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
t <- tdsc(wave)
emptyBands(t,t)
```

farr2007 *Coding Matrix from Farr (2007)*

Description

Coding matrix used by Farr (2007).

Usage

```
data(farr2007)
```

Format

Matrix

References

Farr (2007) "Automated Bioacoustic Identification of Statutory Quarantined Insect Pests". PhD thesis. University of Hull.

Examples

```
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
data(farr2007)
t <- tdsc(wave, coding_matrix=farr2007)
```

followingCodes *Following Codes*

Description

Identifies sequences of codes that follow each other from time domain signal analysis, and optionally plots them as a Sankey diagram.

Usage

```
followingCodes(tdsc, depth = 2, min_code = 0, max_code = 10,
  colourCode = 1, plot = F, ...)
```

Arguments

tdsc	A TDSC object
depth	The length of the sequence of codes to search for
min_code	The minimum value of code to include in sequence
max_code	The maximum value of code to include in sequence
colourCode	If plot is alluvial, colour all codes following this code
plot	If "alluvial" plots the found sequences in a river plot
...	Arguments to pass to the plotting function

Examples

```
## Not run:
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
t <- tdsc(wave)
followingCodes(t)
followingCodes(t, colourCode=2,plot="alluvial")

## End(Not run)
```

---

normalise.a.matrix *Normalise an A Matrix*

Description

Normalises the A Matrix of a tdsc object either by scale or by codewords.

Usage

```
normalise.a.matrix(td, method = "scale")
```

Arguments

td	A tdsc object
method	Either scale (default) or codewords

Examples

```
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
t <- tdsc(wave)
t <- normalise.a.matrix(t)
t <- normalise.a.matrix(t, method="codewords")
```

sample_waveform *Visualise sampled waveforms*

Description

Function to generate images of sampled waveforms with shapes analysed by Time Domain Signal Coding.

Usage

```
sample_waveform(samples = 3, fig_max_samples = NULL,
  start_zero = TRUE, invert = FALSE, tdsc_shapes = FALSE,
  limit_y = TRUE, ...)
```

Arguments

samples	The number of samples
fig_max_samples	When constructing multiple figures this parameter can be used to ensure the plots are of the same size and are aligned
start_zero	If TRUE the waveform starts at zero, if FALSE the zero crossings are between samples
invert	If TRUE the shapes are positive with positive minima, if FALSE shapes are negative with negative maxima
tdsc_shapes	If TRUE the shapes correspond to TDSC shapes, if FALSE they resemble sampled sine waves
limit_y	If TRUE the shape fills the plot, if FALSE the complete range of the y axis is plotted (-1 to 1).
...	Further arguments to pass to plot.

Examples

```
sample_waveform()
```

tdsc *Time Domain Signal Coding*

Description

Performs Time Domain Signal Coding on a Wave object calculating the S-matrix and A-matrix.

Usage

```
tdsc(wave, lag = 1L, coding_matrix = NULL, plot = FALSE,
  max_D = 25L)
```

Arguments

wave	A Wave object
lag	The lag used to create the A-matrix
coding_matrix	A matrix used to code the Duration-Shape pairs
plot	If TRUE plots the workings of the coding algorithm
max_D	The maximum Duration to code

Examples

```
library(tuneR)
wave <- readWave(system.file("extdata", "1.wav", package="tdsc"))
t <- tdsc(wave)
t <- tdsc(wave, lag=2, max_D=10)
```

---

tdsc-class *An S4 class to hold results from TDSC*

Description

An S4 class to hold results from TDSC

Slots

`raw` Two column vector of Durations and Shapes

`codelist` Vector of sequential epoch codings

`b_matrix` The basic matrix

`c_matrix` The coding matrix

`s_matrix` The S-matrix

`a_matrix` The A-matrix

`sample_count` The number of samples in the waveform

`epoch_count` The number of identified epochs

`stdsc` Statistical TDSC feature vector

tdsc.plot *Plot the A matrix or S Matrix*

Description

Plot the A matrix or S Matrix from a tdsc analysis.

Usage

```
tdsc.plot(td, plotter = "persp", ...)
```

Arguments

td	A TDSC object
plotter	Function used to plot the A matrix (persp or perp3D) or S matrix (hist)
...	Parameters to pass to plotting function

Examples

```
## Not run:
tdsc.plot(td)
tdsc.plot(td, plotter="persp3D")
tdsc.plot(td, plotter="hist")

## End(Not run)
```

tdsc.w *TDSC analysis compatible with windowing*

Description

Wrapper function for TDSC analysis that is compatible with the windowing function of the package sonicscrewdriver.

Usage

```
tdsc.w(start, wave = NULL, window.length = NULL)

tdsc.w(start, wave = NULL, window.length = NULL)
```

Arguments

start	Start position in samples
wave	Wave object to analyse
window.length	Length of window

Examples

```
## Not run:
windowing(wave, window.length=1000, window.overlap=0, FUN=tdsc.w)

## End(Not run)
```

Index