Package 'tdsc'

September 30, 2018

Title Time Domain Signal Coding

Version 1.0.0.1

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 (>= 3.4.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0.9000

Imports data.table, methods

Suggests tuneR, alluvial

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 2018-09-30 14:00:02 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))
```

chesmore2011 *Coding Matrix from Chesmore (2011)*

Description

Coding matrix used for Orthoptera.

Usage

```
data(chesmore2011)
```

Format

Matrix

Source

QTL Archive

References

Chesmore, E David (2001). "Application of time domain signal coding and archecktificial 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(chesmore2011)
t <- tdsc(wave, coding_matrix=chesmore2011)
```

`emptyBands` *Empty Bands Discovery*

Description

Identifies unused codes across multiple S-matricies. 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,
  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
plot	If "alluvial" plots the found sequences in a river plot

Examples

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

`sample_waveform` *Visualise sampled waveforms*

Description

Function to genertate 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 neagtive 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 = 1, coding_matrix = NULL, plot = FALSE, max_D = 25)
```

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)
```

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

Description

An S4 class to hold resuts 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

Index