

Package ‘tdsc’

July 22, 2025

Title Time Domain Signal Coding

Version 1.0.4

Description Functions for performing time domain signal coding as used in Chesmore (2001) <[doi:10.1016/S0003-682X\(01\)00009-3](https://doi.org/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 7.2.3

Imports data.table, methods, moments

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

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-06-17 13:10:02 UTC

Contents

c2dmatrix	2
chesmore2001	2
emptyBands	3
farr2007	4
followingCodes	4
normalise.a.matrix	5
sample_waveform	6

tdsc	7
tdsc-class	7
tdsc_plot	8

Index	9
--------------	----------

c2dmatrix	<i>Convert a coding matrix to a distributed matrix</i>
-----------	--

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	<i>Coding Matrix from Chesmore (2001)</i>
--------------	---

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 code-book 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	<i>Coding Matrix from Farr (2007)</i>
----------	---------------------------------------

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	<i>Following Codes</i>
----------------	------------------------

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	<i>Normalise an A Matrix</i>
--------------------	------------------------------

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

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

Examples

```
sample_waveform()
```

tdsc	<i>Time Domain Signal Coding</i>
------	----------------------------------

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	<i>An S4 class to hold results from TDSC</i>
------------	--

Description

An S4 class to hold results from TDSC

Slots

raw	Two column vector of Durations and Shapes
positive	Identify non-negative sections
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

<code>td</code>	A TDSC object
<code>plotter</code>	Function used to plot the A matrix (persp or persp3D) or S matrix (hist)
<code>...</code>	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)
```


Index

* **CodingMatrix**

chesmore2001, [2](#)

farr2007, [4](#)

* **TDSC**

c2dmatrix, [2](#)

chesmore2001, [2](#)

emptyBands, [3](#)

farr2007, [4](#)

followingCodes, [4](#)

sample_waveform, [6](#)

tdsc, [7](#)

c2dmatrix, [2](#)

chesmore2001, [2](#)

emptyBands, [3](#)

farr2007, [4](#)

followingCodes, [4](#)

normalise.a.matrix, [5](#)

sample_waveform, [6](#)

tdsc, [7](#)

tdsc-class, [7](#)

tdsc_plot, [8](#)