Package 'retimer'

May 9, 2026

Title Retime and Analyse Speech Signals

Version 0.1.3

Description Retime speech signals with a native Waveform Similarity Overlap-Add (WSOLA) implementation translated from the 'TSM toolbox' by Driedger & Müller (2014) <https://www.audiolabs-erlangen.de/content/resources/MIR/TSMtoolbox/2014_DriedgerMueller_TSM-Toolbox_DAFX.pdf>. Design retimings and pitch (f0) transformations with tidy data and apply them via 'Praat' interface. Produce spectrograms, spectra, and amplitude envelopes. Includes implementation of vocalic speech envelope analysis (fft_spectrum) technique and example data (mm1) from Tilsen, S., & Johnson, K. (2008) <doi:10.1121/1.2947626>.

Depends R (>= 4.1)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports rPraat, purrr, dplyr, tibble, tidyr, tuneR, ggplot2, stringr, signal, gsignal, methods, seewave, phonTools

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Alistair Beith [aut, cre, cph]

Maintainer Alistair Beith <alistair.beith@gmail.com>

Repository CRAN

Date/Publication 2025-01-22 17:20:02 UTC

Contents

`extractPitchTier` *extractPitchTier*

Description

Extracts 'Praat' PitchTier from wav object.

Usage

```
extractPitchTier(wav, res = 0.1, fmin = 50, fmax = 250, output = "PitchTier")
```

Arguments

`wav`	path to a wav file or a tuneR WAVE object
`res`	resolution of PitchTier
`fmin`	minimum frequency of PitchTier
`fmax`	maximum frequency of PitchTier
`output`	can be "PitchTier" or "file"

Value

Returns a PitchTier object or the temporary path to the generated PitchTier file

extractWord *extractWord*

Description

Extract from a wav file with reference to a TextGrid.

Usage

```
extractWord(
  x,
  word,
  tier = "Word",
  ignore_case = TRUE,
  instance = "random",
  wd = getwd()
)
```

Arguments

x	path to a TextGrid
word	word to search for
tier	name of word tier in TextGrid
ignore_case	default is 'TRUE'
instance	instance of word in TextGrid to extract. Default extracts a random instance. Can also be numeric (row number)
wd	working directory for Praat to use. Accepts relative paths.

Value

Extracts section of wav file corresponding to word and saves in format name_wordi.wav where name is the original name, word is the word and x is the numeric instance.

See Also

density

extract_env *extract_env*

Description

Extract amplitude envelope of filtered speech signal. Adapted from Tilson & Johnson (2008). Procedure:

Usage

```
extract_env(
  x,
  fs,
  low_pass = 80,
  fs_out = 80,
  win = c(700, 1300),
  mean_centre = FALSE,
  replace_init = FALSE
)
```

Arguments

x	a speech signal
fs	sampling frequency of signal
low_pass	frequency of lowpass filter used for smoothing
fs_out	output sampling frequency
win	lower and upper frequencies for initial bypass filter. Default is 700Hz-1300Hz as in Tilson & Johnson (2008)
mean_centre	if TRUE signal will be scaled between 0 and 1 and then mean centred. Default is FALSE
replace_init	if TRUE (default is FALSE) first sample of result will be replaced with second sample to deal with initialisation issue in resampling

Details

1. Signal is bypass filtered to extract desired frequency range 2. Absolute signal is then lowpass filtered 3. Signal is downsampled and mean centred if desired

Value

A matrix with time and amplitude

References

Tilsen, S., & Johnson, K. (2008). Low-frequency Fourier analysis of speech rhythm. The Journal of the Acoustical Society of America, 124(2), EL34–EL39. doi:10.1121/1.2947626

See Also

fft_spectro

`fft_spectro` *fft_spectro*

Description

Calculates low frequency power spectrogram of vocalic interval of speech signal. Following method of Tilsen & Johnson (2008)

Usage

```
fft_spectro(x, f_out = 80, window_size = 256, padding = 2048, plot = TRUE)
```

Arguments

`x`	a ‘tuneR‘ "Wave" object or the path to a .wav file.
`f_out`	the sample frequency for the output
`window_size`	number of samples to calculate each spectrum over
`padding`	length to zero pad signal to. If signal is longer than padding, this will be increased.
`plot`	if true a spectrogram will be plotted

Value

Returns a tibble with frequency (Hz), time (s) and power

References

Tilsen, S., & Johnson, K. (2008). Low-frequency Fourier analysis of speech rhythm. The Journal of the Acoustical Society of America, 124(2), EL34–EL39. doi:10.1121/1.2947626

See Also

fft_spectrum

`fft_spectrum` *fft_spectrum*

Description

Calculates low frequency power spectrum of vocalic interval of speech signal. Following method of Tilsen & Johnson (2008)

Usage

```
fft_spectrum(signal, f, f_out = 80, padding = 512)
```

Arguments

`signal`	a speech signal
`f`	sampling frequency
`f_out`	output sampling frequency. Signal will be lowpass filtered at f_out/2
`padding`	length to zero pad signal to. If signal is longer than padding, this will be increased.

Value

Returns a matrix with columns ‘freq‘ (frequency in Hz) and ‘pwr‘ (spectral power).

References

Tilsen, S., & Johnson, K. (2008). Low-frequency Fourier analysis of speech rhythm. The Journal of the Acoustical Society of America, 124(2), EL34–EL39. doi:10.1121/1.2947626

See Also

fft_spectro

`findPeak` *findPeak*

Description

Find the mode of numeric vector using the peak of its density distribution.

Usage

```
findPeak(x, ...)
```

Arguments

x	a numeric vector
...	further arguements to be passed to ‘density‘

Value

Returns the value of ‘x‘ that corresponds to the peak of the density curve.

See Also

density

`findWord` *findWord*

Description

Find a word in a TextGrid

Usage

```
findWord(x, word = "speech", tier = "Word", ignore_case = TRUE)
```

Arguments

x	path to a TextGrid
word	word to search for
tier	name of word tier in TextGrid
ignore_case	default is ‘TRUE‘

Value

Returns a tibble with onset (t1) and offset (t2) of each occurance of the word in the TextGrid

See Also

extractWord

`flatF0` *flatF0*

Description

Flatten fundamental frequency contour using 'Praat'

Usage

```
flatF0(wav, .f = findPeak, ...)
```

Arguments

`wav`	path to a wav file or a tuneR WAVE object
`.f`	function to use to determine pitch. Default is findPeak which finds the mode of the existing pitch contour.
`...`	Additional arguments passed to extractPitchTier

Value

Returns a tuneR WAVE object of the input with a flat F0 contour

See Also

extractPitchTier

`get_serial_anchors` *get_serial_anchors*

Description

Convert a set of point anchors to a set of anchors that prevent overlaps while fixing the retiming factor within words.

Usage

```
get_serial_anchors(
  anc_in,
  anc_out,
  w_onsets,
  w_offsets,
  fs = NULL,
  retime_f = NULL,
  dry_run = FALSE,
  smudge = 0
)
```

Arguments

anc_in	a vector of time points in the input signal
anc_out	a vector of the times anc_in should be mapped to in the output signal
w_onsets	a vector of time points for the onsets of words. Should be same length as anc_in.
w_offsets	a vector of time points for the offsets of words. Should be same length as anc_in.
fs	Sample rate of signal. If provided, returned anchor points with be expressed in samples. If NULL result will be expressed in seconds.
retime_f	The desired factor that words should be sped by. If NULL the minimum change in rate that will prevent overlaps will be calculated.
dry_run	If TRUE function will exit early with the minimum factor that will prevent overlaps.
smudge	If > 0 this applies a crude adjustment to the calculated anchors to ensure monotonicity. Not necessary unless w_onsets are same as previous w_offsets.

Value

A list that can be used to perform retiming with the wsola function of this package.

See Also

wsola

---

length,Wave-method *S4 generic for length*

Description

S4 generic for length.

Usage

```
## S4 method for signature 'Wave'
length(x)
```

Arguments

x	a 'tuneR' WAVE object

Value

The length of the left channel of the WAVE object

See Also

length

mm1 *mm1*

Description

Example speech from Tilsen & Johnson (2008)

Usage

```
mm1
```

Format

A tuneR "Wave" object:

References

Tilsen, S., & Johnson, K. (2008). Low-frequency Fourier analysis of speech rhythm. The Journal of the Acoustical Society of America, 124(2), EL34–EL39. doi:10.1121/1.2947626

praatRetime *praatRetime*

Description

praatRetime

Usage

```
praatRetime(wav, tg)
```

Arguments

wav	path to a wav file or a tuneR WAVE object
tg	a 'Praat' TextGrid object with 2 tiers: First tier should be intervals in the input audio file and second tier should be the same intervals with the desired onsets (t1) and offsets (t2).

Value

A wav file with the timing of the second tier of the TextGrid will be saved to the outfile location.

See Also

[read_tg()] for reading an existing TextGrid and [write_tg()] for saving a tibble as a TextGrid.

Examples

```
set.seed(42)
data(mm1)
dur <- length(mm1)/mm1@samp.rate

x <- runif(10)
t2_out <- dur*cumsum(x)/sum(x)
t1_out <- c(0, t2_out[-length(t2_out)])
t2_in <- dur*seq_len(10)/10
t1_in <- c(0, t2_in[-length(t2_in)])

tg <- dplyr::tibble(
               name = rep(c("old", "new"), each = 10),
               type = "interval",
               t1 = c(t1_in, t1_out),
               t2 = c(t2_in, t2_out),
               label = rep(letters[1:10], times = 2)
             ) |>
  tidyr::nest(data = c(t1, t2, label))
if (Sys.which("praat") != "") {
 wav_retimed <- praatRetime(mm1, tg)
} else {
 message("Skipping example because Praat is not installed.")
}
```

---

praatScript *praatScript*

Description

Executes a Praat script using the R system function.

Usage

```
praatScript(args, script = "reTimeWin.praat", wd = getwd(), praat = NULL)
```

Arguments

args	arguements to pass to Praat script ("–run" not required)
script	name of script if using a script from this package, or path to script for other scripts
wd	working directory for Praat to use
praat	path to Praat. If null will search for Praat in C:/Program Files (for Windows) or attempt to use "praat" for Unix based systems.

Value

Runs script in Praat and prints stdout to console.

praatSys *praatSys*

Description

Call 'Praat' via system2()

Usage

```
praatSys(args = "--version", praat = NULL, ...)
```

Arguments

args	arguements to pass to 'Praat'
praat	path to 'Praat'. If null will search for 'Praat' in C:/Program Files (for Windows) or attempt to use "praat" for Unix based systems.
...	arguements to pass to internal get_praat_path() function. This can be used to change the folder to look for R in for Windows (default is appDir = "C:/Program Files")

Value

Prints stdout to console

See Also

system2

read_tg *read_tg*

Description

Reads a 'Praat' TextGrid as a nested tibble

Usage

```
read_tg(file, encoding = "auto")
```

Arguments

file	path to TextGrid file
encoding	Passed to rPraat::tg.read: 'auto' (default) will detect encoding, or can be set to 'UTF-8' (rPraat default)

Value

Returns a nested tibble with ‘name‘, ‘type‘ and ‘data‘. ‘data‘ has the variables ‘t1‘, ‘t2‘ and ‘label‘

`spectrogram` *spectrogram*

Description

Universal spectrogram function.

Usage

```
spectrogram(
  x,
  fs = NULL,
  method = NULL,
  output = "tibble",
  wintime = 25,
  steptime = 10
)
```

Arguments

`x`	a signal, ‘tuneR‘ WAVE object, or the path to an .wav or .mp3 file.
`fs`	sample rate if supplying the signal as a vector
`method`	spectrogram implementation to use. Available options are ’phonTools’, ’tuneR’, ’gsignal’, and ’seewave’. Default is to select the first of these methods that is available.
`output`	format of output
`wintime`	length of analysis window in ms
`steptime`	interval between steps in ms

Value

Returns a spectrogram in the desired format

write_tg *write_tg*

Description

Writes a nested tibble to a 'Praat' TextGrid file

Usage

```
write_tg(x, file)
```

Arguments

x	Nested tibble. Must contain the columns 'name', 'type' and 'data'. 'data' must have the columns 't1', 't2' and 'label'
file	File name to save TextGrid as

Value

Returns path of saved TextGrid file

wsola *wsola*

Description

Waveform Similarity Overlap-add. Translated from 'TSM Toolbox'.

Usage

```
wsola(x, s, win = "hann", winLen = 1024, synHop = 512, tol = 512)
```

Arguments

x	an audio signal
s	a scaling factor or a list of two vector with anchor points
win	window function. Default is 'hann' for hanning window. Can also be a custom window supplied as a vector
winLen	window length
synHop	synthesis window hop size
tol	tolerance for overlap delta

Value

retimed audio signal as vector

References

Driedger, J., Müller, M. (2014). TSM Toolbox: MATLAB Implementations of Time-Scale Modification Algorithms. In Proceedings of the International Conference on Digital Audio Effects (DAFx): 249–256.

See Also

fft_spectrum, get_serials_anchors

Examples

```
set.seed(42)
data(mm1)
dur <- length(mm1)
n <- 10
x <- runif(n)
anchors <- list(anc_in = c(0, dur*seq_len(n)/n),
                anc_out = c(0, dur*cumsum(x)/sum(x)))
sig <- wsola(mm1@left, anchors)
```

Index