Package 'baRulho'

July 24, 2026

Type Package

Title Quantifying (Animal) Sound Degradation

Version 2.1.7

Maintainer Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

Description
Intended to facilitate acoustic analysis of (animal) sound propagation experiments, which typically aim to quantify changes in signal structure when transmitted in a given habitat by broadcasting and re-recording animal sounds at increasing distances. The package offers a workflow with functions to prepare the data set for analysis as well as to calculate and visualize several degradation metrics, including blur ratio, signal-to-noise ratio, excess attenuation and envelope correlation among others (Dabelsteen et al 1993 <doi:10.1121/1.406682>).

License GPL (>= 2)

Imports utils, stats, seewave, tuneR, fftw, methods, viridis, Sim.DiffProc, png, checkmate, cli, rlang

Depends R (>= 3.2.1), warbleR (>= 1.1.32), ohun (>= 1.0.2)

LazyData TRUE

URL https://github.com/ropensci/baRulho, https://docs.ropensci.org/baRulho/

BugReports https://github.com/ropensci/baRulho/issues

Suggests rmarkdown, ggplot2, knitr, kableExtra, testthat (>= 3.0.0), covr, formatR, Rraven, monitoR

VignetteBuilder knitr

Repository CRAN

Language en-US

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Marcelo Araya-Salas [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3594-619X>),
Michael Mahoney [rev] (ORCID: <https://orcid.org/0000-0003-2402-304X>),
Dena Clink [rev] (ORCID: <https://orcid.org/0000-0003-0363-5581>)

Date/Publication 2026-07-24 21:50:02 UTC

Contents

add_noise . . . 2
align_test_files . . . 5
attenuation . . . 7
auto_realign . . . 8
blur_ratio . . . 10
detection_distance . . . 13
envelope_correlation . . . 16
excess_attenuation . . . 19
find_markers . . . 22
manual_realign . . . 24
master_est . . . 28
master_sound_file . . . 29
noise_profile . . . 31
plot_aligned_sounds . . . 34
plot_blur_ratio . . . 37
plot_degradation . . . 40
set_reference_sounds . . . 44
signal_to_noise_ratio . . . 47
spcc . . . 50
spectrum_blur_ratio . . . 52
spectrum_correlation . . . 55
spot_ambient_noise . . . 58
synth_sounds . . . 60
tail_to_signal_ratio . . . 63
test_sounds_est . . . 65

Index **67**

`add_noise` *Add synthetic noise*

Description

`add_noise` adds synthetic noise to annotations in extended selection tables

Usage

```
add_noise(
  X,
  mar = NULL,
  target.snr = 2,
  precision = 0.1,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  max.iterations = 1000,
  kind = c("pink", "white", "brown", "red", "power"),
  alpha = 1,
  seed = 123,
  ...
)
```

Arguments

X	Object of class 'extended_selection_table' (created by the function selection_table from the warbleR package), generated 'by element' (see 'https://marce10.github.io/warbleR/articles/b_an element-vs-by-song-extended-selection-tables'), with the test sound files' annotations (which in baRulho is typically the output of align_test_files). Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass. If the "sound.id" column is supplied noise is only added to those sounds with a 'sound.id' different from "ambient", "start_marker" or "end_marker".
mar	numeric vector of length 1. Specifies the margins adjacent to the start point of the annotation over which to measure ambient noise.
target.snr	numeric vector of length 1. Specifies the desired signal-to-noise ratio. Must be lower that the current signal-to-noise ratio. Annotations showing a signal-to-noise ratio higher than 'target.snr' will remain unchanged. Must be supplied.
precision	numeric vector of length 1. Specifies the precision of the adjusted signal-to-noise ratio (in dB).
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
max.iterations	Numeric vector of length 1. Specifies the maximum number of iterations that the internal signal-to-noise adjusting routine will run before stopping. Note that in most cases the default maximum number of iterations (1000) is not reached.
kind	Character vector of length 1 indicating the kind of noise, "white", "pink", "power", "brown", or "red". Noise is synthesized with a modified version of the function noise. Default is "pink" which is similar to background noise in natural environments.

alpha	Numeric vector of length 1. The power for the power law noise (defaults are 1 for pink and 1.5 for red noise). Only used when kind = "power".
seed	Numeric vector of length 1. Seed for random number generation. Default is 123. If NULL, the seed is not set.
...	Additional arguments to be passed internally to signal_to_noise_ratio. Note that "custom" noise reference (argument 'noise.ref' in signal_to_noise_ratio) is currently not supported.

Details

The function adds synthetic noise to sounds referenced in an extended selection table (class created by the function selection_table from the warbleR package) to decrease the signal-to-noise ratio. This can be useful, for instance, for evaluating the effect of background noise on signal structure. Note that the implementation is slow.

Value

Object 'X' in which the wave objects have been modified to match the target signal-to-noise ratio. It also includes an additional column, 'adjusted.snr', with the new signal-to-noise ratio values.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Timmer. J and M. König (1995): On generating power law noise. Astron. Astrophys. 300, 707-710.

See Also

signal_to_noise_ratio

Other miscellaneous: attenuation(), noise_profile()

Examples

```
## Not run:
# load example data
data("test_sounds_est")

# make it a 'by element' extended selection table
X <- warbleR::by_element_est(X = test_sounds_est)

# add noise to the first five rows
X_noise <- add_noise(X = X[1:5, ], mar = 0.2, target.snr = 3)

## End(Not run)
```

`align_test_files` *Align test sound files*

Description

`align_test_files` aligns test (re-recorded) sound files.

Usage

```
align_test_files(
  X,
  Y,
  path = getOption("sound.files.path", "."),
  by.song = TRUE,
  marker = NULL,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  ...
)
```

Arguments

`X`	object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function `selection_table` from the warbleR package) with the master sound file annotations. This should be the same data than that was used for finding the position of markers in `find_markers`. It should also contain a 'sound.id' column that will be used to label re-recorded sounds according to their counterpart in the master sound file.
`Y`	object of class 'data.frame' with the output of `find_markers`. This object contains the position of markers in the re-recorded sound files. If more than one marker is supplied for a sound file only the one with the highest correlation score ('scores' column in 'X') is used.
`path`	Character string containing the directory path where test (re-recorded) sound files are found.
`by.song`	Logical argument to indicate if the extended selection table should be created by song (see 'by.song' `selection_table` argument). Default is `TRUE`.
`marker`	Character string to define whether a "start" or "end" marker would be used for aligning re-recorded sound files. Default is `NULL`. DEPRECATED.
`cores`	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see `options`).
`pb`	Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).
`...`	Additional arguments to be passed to `selection_table` for customizing extended selection table.

Details

The function aligns sounds found in re-recorded sound files (referenced in 'Y') according to a master sound file (referenced in 'X'). If more than one marker is supplied for a sound file only the one with the highest correlation score ('scores' column in 'X') is used. The function outputs an 'extended selection table' by default.

Value

An object of the same class than 'X' with the aligned sounds from test (re-recorded) sound files.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

manual_realign; find_markers; plot_aligned_sounds

Other test sound alignment: auto_realign(), find_markers(), manual_realign(), plot_aligned_sounds()

Examples

```
{
  # load example data
  data(list = c("master_est", "test_sounds_est"))

  # save example files in working director to recreate a case in which working
  # with sound files instead of extended selection tables.
  # This doesn't have to be done with your own data as you will
  # have them as sound files already.
  for (i in unique(test_sounds_est$sound.files)[1:2]) {
    writeWave(object = attr(test_sounds_est, "wave.objects")[[i]],
              file.path(tempdir(), i))
  }

  # save master file
  writeWave(object = attr(master_est, "wave.objects")[[1]],
        file.path(tempdir(), "master.wav"))

  # get marker position for the first test file
    markers <- find_markers(X = master_est,
    test.files = unique(test_sounds_est$sound.files)[1],
    path = tempdir())

  # align all test sounds
  alg.tests <- align_test_files(X = master_est, Y = markers,
```

```
  path = tempdir())
}
```

`attenuation` *Estimate attenuation of sound pressure level*

Description

`attenuation` estimates atmospheric attenuation and atmospheric absorption.

Usage

```
attenuation(
  frequency,
  dist0,
  dist,
  temp = 20,
  rh = 60,
  pa = 101325,
  hab.att.coef = 0.02
)
```

Arguments

`frequency`	Numeric vector of length 1 with frequency (in Hertz).
`dist0`	Numeric vector of length 1 with distance (m) for the reference SPL.
`dist`	Numeric vector of length 1 with distance (m) over which a sound propagates.
`temp`	Numeric vector of length 1 with frequency (in Celsius). Default is 20.
`rh`	Numeric vector of length 1 with relative humidity (in percentage). Default is 60.
`pa`	Numeric vector of length 1 with atmospheric (barometric) pressure in Pa (standard: 101325, default). Used for atmospheric attenuation.
`hab.att.coef`	Attenuation coefficient of the habitat (in dB/kHz/m).

Details

Calculate the geometric, atmospheric and habitat attenuation and the overall expected attenuation (the sum of the other three) based on temperature, relative humidity, atmospheric pressure and sound frequency. Attenuation values are given in dB. The function is modified from http://www.sengpielaudio.com

Value

Returns the geometric, atmospheric and habitat attenuation (in dB) as well as the combined attenuation.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

Other miscellaneous: `add_noise()`, `noise_profile()`

Examples

```
{
  # measure attenuation
  attenuation(frequency = 2000, dist = 50, dist0 = 1)
}
```

`auto_realign` *Fix small misalignments in the time position test sounds*

Description

`auto_realign` fixes small misalignments in the time position of test sounds in an extended selection table using spectrographic cross-correlation

Usage

```
auto_realign(
  X,
  Y,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 90),
  wn = c("hanning", "hamming", "bartlett", "blackman", "flattop", "rectangle"),
  bp = NULL
)
```

Arguments

`X`	object of class 'extended_selection_table' (created by the function `selection_table` from the warbleR package) with the test sound files' annotations to be aligned. Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass and 7) "sound.id": ID of sounds used to identify counterparts across distances. Each sound must have a unique

ID within a given distance.. The object must include the following additional columns: 'sound.id', 'bottom.freq' and 'top.freq'.

`Y` object of class 'extended_selection_table' (a class created by the function `selection_table` from the warbleR package) with the master sound file annotations. This should be the same data than that was used for finding the position of markers in `find_markers`. It should also contain a 'sound.id' column.

`cores` Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see `options`).

`pb` Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).

`hop.size` A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see `options`).

`wl` a vector with a single even integer number specifying the window length of the spectrogram, default is `NULL`. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see `options`).

`ovlp` Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in `spectro`. Default is 90. High values slow down the function but produce more accurate results. Can be set globally for the current R session via the "ovlp" option (see `options`).

`wn` A character vector of length 1 specifying the window name as in `ftwindow`.

`bp` Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Default is `NULL`.

Details

Precise alignment is crucial for downstream measures of sound degradation. This function uses spectrogram cross-correlation to improve the time position alignment of test sounds. The master sound file is used as reference. The function calls warbleR's `cross_correlation` internally to align sounds using cross-correlation. The output extended selection table contains the new start and end values after alignment. **Note that 1) this function only works to further improve alignments if the estimated position of the test sound is already close to the actual position and 2) both 'X' and 'Y' must be extended selection tables sensu** `selection_table`. The function might not work properly with annotations with a small frequency range (e.g. pure tones).

Value

Object 'X' in which time parameters (columns 'start' and 'end') have been tailored to more closely match the start and end of the reference sound.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

Clark, C.W., Marler, P. & Beeman K. (1987). Quantitative analysis of animal vocal phonology: an application to Swamp Sparrow song. Ethology. 76:101-115.

See Also

blur_ratio, cross_correlation

Other test sound alignment: align_test_files(), find_markers(), manual_realign(), plot_aligned_sounds()

Examples

```
{
  # load example data
  data("test_sounds_est")
  data("master_est")

  # create "unaligned_test_sounds_est" by
  # adding error to "test_sounds_est" start and end
  unaligned_test_sounds_est <- test_sounds_est
  set.seed(123)
  noise_time <- sample(c(0.009, -0.01, 0.03, -0.03, 0, 0.07, -0.007),
  nrow(unaligned_test_sounds_est),
  replace = TRUE)

  attr(unaligned_test_sounds_est, "check.res")$start <-
  unaligned_test_sounds_est$start <-
  unaligned_test_sounds_est$start + noise_time
  attr(unaligned_test_sounds_est, "check.res")$end <-
  unaligned_test_sounds_est$end  <-
  unaligned_test_sounds_est$end + noise_time

# re align
realigned_est <- auto_realign(X = unaligned_test_sounds_est, Y = master_est)
}
```

---

blur_ratio *Measure blur ratio in the time domain*

Description

blur_ratio measures blur ratio in sounds referenced in an extended selection table.

Usage

```
blur_ratio(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  env.smooth = getOption("env.smooth", 200),
  envelopes = FALSE,
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
  n.samples = 100,
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
env.smooth	Numeric vector of length 1 determining the length of the sliding window (in amplitude samples) used for a sum smooth for amplitude envelope calculation (used internally by env). Default is 200. Can be set globally for the current R session via the "env.smooth" option (see options).
envelopes	Logical to control if envelopes are returned (as attributes, 'attributes(X)$envelopes'). Default is FALSE.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).

ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 70. Used for applying bandpass filtering. Can be set globally for the current R session via the "ovlp" option (see options).
n.samples	Numeric vector of length 1 specifying the number of amplitude samples to use for representing amplitude envelopes. Default is 100. If null the raw amplitude envelope is used (note that this can result in high RAM memory usage for large data sets). Amplitude envelope values are interpolated using approx.
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

Details

Blur ratio measures the degradation of sound as a change in sound power in the time domain as described by Dabelsteen et al (1993). Low values indicate low degradation of sounds. The function measures the blur ratio on sounds in which a reference playback has been re-recorded at different distances. Blur ratio is measured as the mismatch between amplitude envelopes (expressed as probability mass functions) of the reference sound and the re-recorded sound. By converting envelopes to probability mass functions the effect of energy attenuation is removed, focusing the analysis on the modification of the envelope shape. The function compares each sound to the corresponding reference sound within the supplied frequency range (e.g. bandpass) of the reference sound ('bottom.freq' and 'top.freq' columns in 'X'). The 'sound.id' column must be used to tell the function to only compare sounds belonging to the same category (e.g. song-types). Two methods for setting the experimental design are provided. All wave objects in the extended selection table must have the same sampling rate so the length of envelopes is comparable.

Value

Object 'X' with an additional column, 'blur.ratio', containing the computed blur ratio values. If envelopes = TRUE the output would include amplitude envelopes for all sounds as attributes ('attributes(X)$envelopes').

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Dabelsteen, T., Larsen, O. N., & Pedersen, S. B. (1993). Habitat-induced degradation of sound signals: Quantifying the effects of communication sounds and bird location on blur ratio, excess attenuation, and signal-to-noise ratio in blackbird song. The Journal of the Acoustical Society of America, 93(4), 2206.

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

envelope_correlation, spectrum_blur_ratio

Other quantify degradation: detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

 # add reference to X
  X <- set_reference_sounds(X = test_sounds_est)
  blur_ratio(X = X)

  # using method 2
X <- set_reference_sounds(X = test_sounds_est, method = 2)
  blur_ratio(X = X)

  # get envelopes
  br <- blur_ratio(X = X, envelopes = TRUE)
  envs <- attributes(br)$envelopes

  # make distance a factor for plotting
  envs$distance <- as.factor(envs$distance)


  # plot
  rlang::check_installed("ggplot2")
  library(ggplot2)

  ggplot(envs, aes(x= time, y = amp, col = distance)) +
  geom_line() + facet_wrap(~ sound.id) +
  scale_color_viridis_d() +
  labs(x = "Time (s)", y = "Amplitude (PMF)") +
  theme_classic()

}
```

---

detection_distance *Measure detection distance of sound*

Description

detection_distance detection distance of sounds.

Usage

```
detection_distance(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  path = getOption("sound.files.path", "."),
  spl = NULL,
  spl.cutoff = NULL,
  temp = 20,
  rh = 60,
  pa = 101325,
  hab.att.coef = 0.02,
  max.distance = 1000,
  resolution = 0.1,
  subtract.bgn = TRUE,
  envelope = c("abs", "hil"),
  mar = NULL
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).

path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
spl	A numeric vector of length 1 specifying the sound pressure level of sounds. If not supplied then it will be measured from the sounds themselves.
spl.cutoff	A numeric vector of length 1 specifying the sound pressure level cutoff to define if the sound is no longer detected. Ideally it should be estimated based on the sound detection threshold of the species.
temp	Numeric vector of length 1 with frequency (in Celsius). Default is 20.
rh	Numeric vector of length 1 with relative humidity (in percentage). Default is 60.
pa	Numeric vector of length 1 with ambient pressure in Pa (standard: 101325, default). Used for Atmospheric attenuation.
hab.att.coef	Attenuation coefficient of the habitat (in dB/kHz/m).
max.distance	Numeric vector of length 1 with the maximum distance (in m) at which detection would be evaluated. Note that the function calculates the expected sound pressure level values along a vector of distances to find the distance at which the expected sound pressure level equates 'spl.cutoff'. Default is 1000 m.
resolution	Numeric vector of length 1 with the distance resolution (in m) for estimated detection distance. Higher resolutions take longer to estimate. Default is 0.1 m.
subtract.bgn	Logical argument to control if SPL from background noise is excluded from the measured signal SPL. Default is FALSE.
envelope	Character string vector with the method to calculate amplitude envelopes (in which SPL is measured, only used if 'spl' is not supplied), as in env. Must be either 'abs' (absolute envelope, default) or 'hil' (Hilbert transformation).
mar	numeric vector of length 1. Specifies the margins adjacent to the start and end points of selection over which to measure background noise. This is required to subtract background noise sound pressure level (so only needed when 'subtract.bgn = TRUE').

Details

The function computes the maximum distance at which a sound would be detected, which is calculated as the distance in which the sound pressure level (SPL) goes below the specified SPL cutoff ('spl.cutoff')). This is returned as an additional column 'detection.distance' (in m). The function uses internally attenuation to estimate SPL at increasing values until it reaches the defined cutoff. The peak frequency (calculated on the power spectrum of the reference sound) of the reference sound for each sound ID is used as the carrier frequency for distance estimation. The sound recorded at the lowest distance is used as reference. **This function assumes that all recordings have been made at the same recording volume**.

Value

Object 'X' with an additional column, 'detection.distance', containing the computed detection distances (in m).

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

Clark, C.W., Marler, P. & Beeman K. (1987). Quantitative analysis of animal vocal phonology: an application to Swamp Sparrow song. Ethology. 76:101-115.

See Also

attenuation

Other quantify degradation: blur_ratio(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
## Not run:
# load example data
data("test_sounds_est")

# add reference to X
X <- set_reference_sounds(X = test_sounds_est)

detection_distance(X = X[X$distance %in% c(1, 10), ], spl.cutoff = 5, mar = 0.05)

## End(Not run)
```

---

envelope_correlation *Measure amplitude envelope correlation*

Description

envelope_correlation measures amplitude envelope correlation of sounds referenced in an extended selection table.

Usage

```
envelope_correlation(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  cor.method = c("pearson", "spearman", "kendall"),
```

```
  env.smooth = getOption("env.smooth", 200),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
cor.method	Character string indicating the correlation coefficient to be applied ("pearson", "spearman", or "kendall", see cor).
env.smooth	Numeric vector of length 1 to determine the length of the sliding window used for a sum smooth for amplitude envelope calculation (used internally by env). Can be set globally for the current R session via the "env.smooth" option (see options).
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 70. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

Details

Amplitude envelope correlation measures the similarity of two sounds in the time domain. The function measures the envelope correlation coefficients of sounds in which a reference playback has been re-recorded at increasing distances. Values close to 1 means very similar amplitude envelopes (i.e. little degradation has occurred). If envelopes have different lengths (which means sounds have different lengths) cross-correlation is used and the maximum correlation coefficient is returned. Cross-correlation is achieved by sliding the shortest sound along the largest one and computing the correlation at each step. The 'sound.id' column must be used to indicate the function to only compare sounds belonging to the same category (e.g. song-types). The function compares each sound to the corresponding reference sound within the supplied frequency range (e.g. bandpass) of the reference sound ('bottom.freq' and 'top.freq' columns in 'X'). Two methods for computing envelope correlation are provided (see 'method' argument). Use `blur_ratio` to create envelopes graphs.

Value

Object 'X' with an additional column, 'envelope.correlation', containing the computed envelope correlation coefficients.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

Apol, C.A., Sturdy, C.B. & Proppe, D.S. (2017). Seasonal variability in habitat structure may have shaped acoustic signals and repertoires in the black-capped and boreal chickadees. Evol Ecol. 32:57-74.

See Also

`blur_ratio`, `spectrum_blur_ratio`

Other quantify degradation: `blur_ratio()`, `detection_distance()`, `plot_blur_ratio()`, `plot_degradation()`, `set_reference_sounds()`, `signal_to_noise_ratio()`, `spcc()`, `spectrum_blur_ratio()`, `spectrum_correlation()`, `tail_to_signal_ratio()`

Examples

```
{
  # load example data
  data("test_sounds_est")

  # add reference to X
  X <- set_reference_sounds(X = test_sounds_est)

  envelope_correlation(X = X)
```

```
  # method 2
  # add reference to X
  X <- set_reference_sounds(X = test_sounds_est, method = 2)
  envelope_correlation(X = X)
}
```

---

excess_attenuation *Measure excess attenuation*

Description

excess_attenuation measures excess attenuation in sounds referenced in an extended selection table.

Usage

```
excess_attenuation(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  hop.size = getOption("hop.size", 1),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 50),
  bp = "freq.range",
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).

`hop.size` A numeric vector of length 1 specifying the time window duration (in ms). Default is 1 ms, which is equivalent to ~45 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see `options`).

`wl` A numeric vector of length 1 specifying the window length of the spectrogram, default is `NULL`. If supplied, 'hop.size' is ignored. Note that lower values will increase time resolution, which is more important for amplitude calculations.

`ovlp` Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in `spectro`. Default is 50. Only used for bandpass filtering. Can be set globally for the current R session via the "ovlp" option (see `options`).

`bp` Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Alternatively, when set to 'freq.range' (default), the function will use the 'bottom.freq' and 'top.freq' for each sound as the bandpass range.

`path` Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see `options`).

Details

Excess attenuation is the amplitude loss of a sound in excess due to spherical spreading (observed attenuation - expected attenuation). With every doubling of distance, sounds attenuate with a 6 dB loss of amplitude (Morton, 1975; Marten & Marler, 1977). Any additional loss of amplitude results in energy loss in excess of that expected to occur with distance via spherical spreading. So it represents power loss due to additional factors like vegetation or atmospheric conditions (Wiley & Richards, 1978). It accounts for attenuation from additional factors such as:

- `Ground absorption`: sound energy can be absorbed by the ground, especially in environments like forests with soft or uneven terrain.
- `Vegetation and obstacles`: trees, shrubs, and other obstacles can absorb or scatter sound energy, reducing the sound level more than geometric spreading alone would predict.
- `Air absorption`: as sound travels through air, it loses energy due to air molecules absorbing the sound waves, and this effect becomes more pronounced over longer distances.
- `Wind and temperature gradients`: These environmental factors can cause sound waves to bend or refract.

Excess attenuation is computed as `-20 * log10(rms("test signal") / rms("reference signal"))) - (20 * log10(1 / "distance")` in which 'rms(..)' represents the root mean square of an amplitude envelope. Low values indicate little additional attenuation. The goal of the function is to measure the excess attenuation on sounds in which a reference playback has been re-recorded at increasing distances. The 'sound.id' column must be used to indicate which sounds belonging to the same category (e.g. song-types). The function will then compare each sound type to the corresponding reference sound. Two approaches for computing excess attenuation are provided (see 'type' argument). NAs will be returned if one of the envelopes is completely flat (e.g. no variation in amplitude).

Value

Object 'X' with an additional column, 'excess.attenuation', containing the computed excess attenuation values (in dB).

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

Dabelsteen, T., Larsen, O. N., & Pedersen, S. B. (1993). Habitat-induced degradation of sound signals: Quantifying the effects of communication sounds and bird location on blur ratio, excess attenuation, and signal-to-noise ratio in blackbird song. The Journal of the Acoustical Society of America, 93(4), 2206.

Dabelsteen, T., & Mathevon, N. (2002). Why do songbirds sing intensively at dawn?. Acta ethologica, 4(2), 65-72.

Darden, SK, Pedersen SB, Larsen ON, & Dabelsteen T. (2008). Sound transmission at ground level in a short-grass prairie habitat and its implications for long-range communication in the swift fox *Vulpes velox*. The Journal of the Acoustical Society of America, 124(2), 758-766.

Marten K, & Marler P. (1977). Sound transmission and its significance for animal vocalization. Behavioral Ecology and Sociobiology, 2(3), 271-290.

Morton ES. (1975). Ecological sources of selection on avian sounds. The American Naturalist, 109(965), 17-34.

Wiley, R., & Richards, D. G. (1978). Physical constraints on acoustic communication in the atmosphere: implications for the evolution of animal vocalizations. Behavioral Ecology and Sociobiology, 3(1), 69-94.

See Also

spcc; envelope_correlation

Examples

```
{
  # load example data
  data("test_sounds_est")

  # using method 1
  # add reference to X
  X <- set_reference_sounds(X = test_sounds_est)
  excess_attenuation(X = X)

  # using method 2
  X <- set_reference_sounds(X = test_sounds_est, method = 2)
  # excess_attenuation(X = X)
```

```
}
```

`find_markers` *Find acoustic markers on test sound files*

Description

`find_markers` find acoustic markers on test (re-recorded) sound files using spectrographic cross-correlation.

Usage

```
find_markers(
  X,
  markers = c("start_marker", "end_marker"),
  test.files = NULL,
  path = getOption("sound.files.path", "."),
  pb = getOption("pb", TRUE),
  cores = getOption("mc.cores", 1),
  ...
)
```

Arguments

`X`	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function `selection_table` from the warbleR package) with the reference to the sounds in the master sound file. Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time, 4) "end": end time of selections and 5) "sound.id": unique identifier for each of the annotated sounds in 'X'. Columns for 'top.freq', 'bottom.freq' and 'channel' are optional. The acoustic start and end markers (added by `master_sound_file`) should be labeled as "start_marker" and "end_marker" respectively. Required.
`markers`	Character vector with the name of the annotations (as in the column 'sound.id') to be used as templates for cross-correlation. Default is `c("start_marker", "end_marker")`. Using more than one marker is recommended as the time difference between their position can be used to evaluate the precision of the detection (see 'Value' section).
`test.files`	Character vector of length 1 with the name(s) of the test (re-recorded) file(s) in which to search for the marker(s). If not supplied all sound files in 'path' are used instead.
`path`	Character string containing the directory path where test (re-recorded) sound files are found.
`pb`	Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).

cores — Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).

... — Additional arguments to be passed to template_correlator for setting cross-correlation parameters (e.g. 'wl', 'ovlp', etc).

Details

The function takes a master sound file's reference data ('X') and finds the position of acoustics markers ('markers' argument, included as selections in 'X') in the re-recorded sound files. This is used to align signals found in re-recorded sound files according to a master sound file referenced in 'X'. The position of the markers is determined as the highest spectrogram cross-correlation value for each marker using the functions template_correlator and template_detector. **Make sure the master sound file (that referred to in 'X') is found in the same folder than the re-recorded sound files**. Take a look at the package vignette for information on how to incorporate this function into a sound degradation analysis workflow. In cases in which markers are not correctly detected editing test sound files to remove audio segments with no target sounds (before the start marker and after the end marker) can improve performance. Using a low 'hop.size' or window length 'wl' (used internally by template_correlator) can help to improve precision. Other spectrogram types (argument 'type' in template_correlator) can sometimes show better performance when markers are highly degraded. If frequency range columns are included ('bottom.freq' and 'top.freq', in kHz) cross-correlation will be run on those frequency ranges. All templates must have the same sampling rate and both templates and 'files' (in which to find templates) must also have the same sampling rate.

Value

A data frame with test file names, marker id, maximum cross-correlation score for each marker and the start and end where it was detected. If two or more markers are used the function computes an additional column, 'time.mismatch', that compares the time difference between the two markers in the test-files against that in the master sound file. In a perfect detection the value must be 0.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

manual_realign; auto_realign; align_test_files; master_sound_file

Other test sound alignment: align_test_files(), auto_realign(), manual_realign(), plot_aligned_sounds()

Examples

```
{
  # set temporary directory
  td <- tempdir()

  # load example data
  data("master_est")

  # save example files in working director to recreate a case in which working
  # with sound files instead of extended selection tables.
  # This doesn't have to be done with your own data as you will
  # have them as sound files already.
  for (i in unique(test_sounds_est$sound.files)[1:2]) {
    writeWave(object = attr(test_sounds_est, "wave.objects")[[i]], file.path(td, i))
  }

  # save master file
  writeWave(object = attr(master_est, "wave.objects")[[1]], file.path(td, "master.wav"))

  # set path and no progress bar in global options
  options(sound.files.path = td, pb = FALSE)

  # get marker position
 markers <- find_markers(X = master_est, test.files = unique(test_sounds_est$sound.files)[2])
}
```

---

manual_realign *Plot spectrograms to check test sound files alignment*

Description

manual_realign plots spectrograms to visually inspect alignment precision on test sound files.

Usage

```
manual_realign(
  X,
  Y,
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 0),
  path = getOption("sound.files.path", "."),
  collevels = seq(-120, 0, 5),
  palette = viridis::viridis,
  duration = 2,
  mar = 0.2,
  step.lengths = c(5, 30),
  flim = NULL,
```

```
  label.col = "white",
  ext.window = TRUE,
  width = 10,
  height = 5,
  srt = 0,
  cex = 1,
  fast.spec = TRUE,
  marker = "start_marker",
  grid = 0.2,
  ...
)
```

Arguments

X	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations (typically the output of align_test_files) to be aligned. Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass and 7) "sound.id": ID of sounds used to identify counterparts across distances. Each sound must have a unique ID within a given distance.
Y	object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the master sound file annotations. This should be the same data than that was used for finding the position of markers in find_markers. It should also contain a 'sound.id' column.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 0. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
collevels	A numeric vector of length 3. Specifies levels to partition the amplitude range of the spectrogram (in dB). The more levels the higher the resolution of the spectrogram. Default is seq(-120, 0, 1).

palette	Color palette function for spectrogram. Default is viridis. See spectro for more palettes. Palettes as gray.2 may work better when fast.spec = TRUE (an argument that can be passed to the internal spectrogram function using "...").
duration	A numeric vector of length 1. Specifies the overall duration of the clip that will be plotted. Notice that only the initial part of the test files are plotted as this is enough to tell the precision of the alignment.
mar	numeric vector of length 1. Specifies the minimum margins adjacent (before and after) to the start of the marker used for checking alignments (see 'marker' argument). Default is 0.2.
step.lengths	Numeric vector of length 2 indicating the time length (in ms) of short (min(step.lengths)) and long steps (max(step.lengths)) for manually aligning spectrograms. Default is c(5, 30).
flim	A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrogram, as in spectro. Default is NULL which will plot spectrograms in the full frequency range (0 - nyquist frequency).
label.col	Character string controlling the color of lines and sound ID labels.
ext.window	Logical. If TRUE then and external graphic window is used.Dimensions can be set using the 'width' and 'height' arguments. Default is TRUE.
width	Numeric vector of length 1. Single value (in inches) indicating the width of the output image files. Default is 10.
height	Numeric vector of length 1. Single value (in inches) indicating the height of the output image files. Default is 5.
srt	Numeric argument of length 1. The rotation (in degrees) of the sound id labels. Default is 0.
cex	Numeric argument of length 1controlling the size of sound id text labels. Default is 1.
fast.spec	Logical. If TRUE then image function is used internally to create spectrograms, which substantially increases performance (much faster), although some options become unavailable, as collevels (amplitude scale). Default is FALSE.
marker	Character string with the name of the marker to be used as the main reference for checking/adjusting time alignments. Default is 'start_marker'. Note that this can take any of the sound IDs in 'Y$sound.id'.
grid	Numeric vector of length 1 controlling the spacing between vertical lines on the spectrogram. Default is 0.2 s. Use 0 to remove grid.
...	Additional arguments to be passed to the internal spectrogram creating function for customizing graphical output. The function is a modified version of spectro, so it takes the same arguments.

Details

This function allows the interactive adjustment of the alignment of test sound files produced by align_test_files. The function generates a multipanel graph with the spectrogram of the master sound file in top of that from test sound files, highlighting the position of correspondent test sounds on both in order to simplify assessing and adjusting their alignment. Spectrograms include the

first few seconds of the sound files (controlled by 'duration') which is usually enough to tell the precision of the alignment. The lower spectrogram shows a series of 'buttons' that users can click on to control if the test sound file spectrogram (low panel) needs to be moved to the left ("<") or right (">"). Users can also reset the spectrogram to its original position ('reset'), move on to the next sound file in 'X' (test sound file annotations) or stop the process (stop button). The function returns an object similar to the input object 'X' in which the start and end of the sounds have been adjusted.

Value

Creates a multipanel graph with spectrograms of master and test sound files in which users can interactively adjust their alignment in time. Return an object similar to the input object 'X' in which the start and end of the sounds have been adjusted.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

auto_realign; find_markers; align_test_files

Other test sound alignment: align_test_files(), auto_realign(), find_markers(), plot_aligned_sounds()

Examples

```
{
  # load example data
  data(list = c("master_est", "test_sounds_est"))

  # save example files in working director to recreate a case in which working
  # with sound files instead of extended selection tables.
  # This doesn't have to be done with your own data as you will
  # have them as sound files already.
  for (i in unique(test_sounds_est$sound.files)[1:2]) {
 writeWave(object = attr(test_sounds_est, "wave.objects")[[i]], file.path(tempdir(), i))
  }

  # save master file
 writeWave(object = attr(master_est, "wave.objects")[[1]], file.path(tempdir(), "master.wav"))

  # get marker position
 markers <- find_markers(X = master_est, test.files = unique(test_sounds_est$sound.files)[2],
  path = tempdir())

  # align all test sounds
```

```
  alg.tests <- align_test_files(X = master_est, Y = markers, path = tempdir())

  # add error to alignment
  lag <- (as.numeric(as.factor(alg.tests$sound.files)) - 2) / 30
  alg.tests$start <- alg.tests$start + lag
  alg.tests$end <- alg.tests$end + lag

  if(interactive()){
  realigned_est <- manual_realign(X = alg.tests, Y = master_est, duration = 2,
  ovlp = 50, hop.size = 14, collevels = seq(-140, 0, 5), palette = viridis::mako,
  ext.window = FALSE)
 }
}
```

`master_est` *Extended selection table of master acoustic data*

Description

Extended selection table (est) with the acoustic data and annotations of the master sound file of synthetic sounds. The synthetic sounds files are 2 s long, frequency modulated and amplitude modulated. The data was created by the function `selection_table` from the warbleR package. The re-recorded data generated with these sounds is found in the example object `test_sounds_est`.

Usage

```
data(master_est)
```

Format

Extended selection table object in the warbleR format, which contains annotations and acoustic data.

Source

Marcelo Araya-Salas

See Also

Other data sets: `test_sounds_est`

`master_sound_file` *Create a master sound file*

Description

`master_sound_file` creates a master sound file to be used in playback experiments related to sound degradation.

Usage

```
master_sound_file(
  X,
  file.name,
  dest.path = getOption("dest.path", "."),
  overwrite = FALSE,
  delay = 1,
  gap.duration = 1,
  amp.marker = 2,
  flim = c(0, 4),
  cex = 14,
  path = getOption("sound.files.path", ".")
)
```

Arguments

`X`	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function `selection_table` from the warbleR package) with the reference to the model sounds. Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass and 6) "top.freq": high frequency for bandpass. An optional 'sound.id' column can be included to use a custom label for each sound in the output. This column must contain a unique id for each sound (labels cannot repeated). If not supplied the function will make it by combining the sound file and selection columns.
`file.name`	Character string indicating the name of the sound file.
`dest.path`	Character string containing the directory path where the sound file will be saved. If `NULL` (default) then the current working directory will be used instead.
`overwrite`	Logical argument to determine if the function will overwrite any existing sound file with the same file name. Default is the current working directory.
`delay`	Numeric vector of length 1 to control the duration (in s) of a silence gap at the beginning (and at the end) of the sound file. This can be useful to allow some time at the start of the playback experiment. Default is 1.
`gap.duration`	Numeric vector of length 1 to control the duration (in s) of silence gaps to be placed in between sounds. Default is 1 s.

`amp.marker` Numeric vector of length 1 to use as a constant to amplify markers amplitude. This is useful to increase the amplitude of markers in relation to those of sounds, so it is picked up at further distances. Default is 2.

`flim` Numeric vector of length 2 to control the (approximate) frequency range in which the markers would be found. If `NULL` markers would span across the entire frequency range. Default is c(0, 4).

`cex` Numeric vector of length 1 indicating the font size for the start and end markers. Default is 14.

`path` Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see `options`).

Details

The function is intended to simplify the creation of master sound files for playback experiments in sound degradation studies. The function clips sounds from sound files (or wave objects from extended selection tables) and concatenates them in a single sound file. All clips are peak normalized and rescaled to the maximal possible dynamic range. The function also adds acoustic markers at the start and end of the playback that can be used to time-sync test (re-recorded) sounds to facilitate the streamlining of degradation quantification. There is no predefined limit to the duration of the output master sound file, although the creation of long files could be constrained by computer memory. As a reference, master sound files of up to 10 min have been created in a 16GB RAM laptop computer.

Value

A .wav file in 'path' as well as a data frame in the R environment with the annotations (i.e. time position) of sounds in the master sound file and an additional column 'sound.id' that provides a unique id for each sound in the sound file. This is useful for identifying/labeling sounds in test (re-recorded) sound files for downstream analyses.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

`exp_raven`

Other prepare acoustic data: `spot_ambient_noise()`, `synth_sounds()`

Examples

```
{
  # load example data from warbleR
  data(list = c(
    "Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4",
    "lbh_selec_table"
  ))

  # save sound files to temporary folder
  writeWave(Phae.long1, file.path(tempdir(), "Phae.long1.wav"))
  writeWave(Phae.long2, file.path(tempdir(), "Phae.long2.wav"))
  writeWave(Phae.long3, file.path(tempdir(), "Phae.long3.wav"))
  writeWave(Phae.long4, file.path(tempdir(), "Phae.long4.wav"))

  # make an extended selection table
  est <- selection_table(
    X = lbh_selec_table, extended = TRUE,
    path = tempdir()
  )

  # create master sound file
  master.sel.tab <- master_sound_file(
    X = est, file.name = "example_master",
    dest.path = tempdir(), gap.duration = 0.3
  )

## Not run:
  # the following code exports the selection table to Raven
  # using the Rraven package
  Rraven::exp_raven(master.sel.tab, path = tempdir(),
  file.name = "example_master_selection_table")

## End(Not run)
}
```

`noise_profile` *Measure full spectrum sound noise profiles*

Description

`noise_profile` Measure full spectrum sound pressure levels (i.e. noise profiles) in sound files or extended selection tables.

Usage

```
noise_profile(
  X = NULL,
  files = NULL,
```

```
  mar = NULL,
  noise.ref = c("adjacent", "custom"),
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  path = getOption("sound.files.path", "."),
  bp = NULL,
  hop.size = getOption("hop.size", 1),
  wl = getOption("wl", NULL),
  PSD = FALSE,
  norm = TRUE,
  dB = c("A", "B", "C", "D", "ITU", "max0"),
  averaged = TRUE
)
```

Arguments

X	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass and 7) "sound.id": ID of sounds used to identify counterparts across distances (needed for "custom" noise reference, see "noise.ref" argument). Default is NULL.
files	Character vector with names of wave files to be analyzed. Files must be found in 'path' supplied (or in the working directory if 'path' is not supplied). Default is NULL.
mar	numeric vector of length 1. Specifies the margins adjacent to the start and end points of selection over which to measure ambient noise. Required if 'X' is supplied and ignored if not supplied. Default is NULL.
noise.ref	Character vector of length 1 to determined which noise segment must be used for measuring ambient noise. Ignored if 'X' is not supplied. Two options are available: • adjacent: measure ambient noise right before the sound (using argument 'mar' to define duration of ambient noise segments). • custom: measure ambient noise segments referenced in the selection table (labeled as 'ambient' in the 'sound.id' column).
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the

current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

bp — Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Default is NULL.

hop.size — A numeric vector of length 1 specifying the time window duration (in ms). Default is 1 ms, which is equivalent to ~45 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).

wl — A numeric vector of length 1 specifying the window length of the spectrogram, default is NULL. Ignored if bp = NULL. If supplied, 'hop.size' is ignored. Note that lower values will increase time resolution, which is more important for amplitude ratio calculations.

PSD — Logical to control whether the Probability Mass Function (the probability distribution of frequencies). See meanspec. Default is FALSE.

norm — Logical to control whether amplitude values are normalized (divided by the maximum) so the highest value is 1. See meanspec. Default is TRUE.

dB — A character string of length 1 specifying the type dB to return: "max0" for a maximum dB value at 0, "A", "B", "C", "D", and "ITU" for common dB weights. See meanspec. Default is "A".

averaged — Logical to control if frequency spectra are averaged within a sound file. Default is TRUE.

Details

The function estimates full spectrum sound pressure levels (i.e. noise profiles) of ambient noise. This can be done on data frames/(extended) selection tables (using the segments containing no target sound or the 'ambient' sound id) or over complete sound files in the working directory (or path supplied). The function uses meanspec internally to calculate frequency spectra.

Value

A data frame containing the frequency spectra for each sound file or wave object (if 'X' is supplied and is of class 'extended.selection.table').

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481.

See Also

excess_attenuation

Other miscellaneous: add_noise(), attenuation()

Examples

```
{
  # load example data
  data("test_sounds_est")

  # measure on custom noise reference
  noise_profile(X = test_sounds_est, mar = 0.01, pb = FALSE, noise.ref = "custom")

  # remove noise selections so noise is measured right before the signals
  pe <- test_sounds_est[test_sounds_est$sound.id != "ambient", ]

  noise_profile(X = pe, mar = 0.01, pb = FALSE, noise.ref = "adjacent")
}
```

---

plot_aligned_sounds *Plot spectrograms to check test sound files alignment*

Description

plot_aligned_sounds plots spectrograms to visually inspect alignment precision on test sound files.

Usage

```
plot_aligned_sounds(
  X,
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 50),
  path = getOption("sound.files.path", "."),
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  collevels = seq(-120, 0, 5),
  palette = viridis::viridis,
  duration = 2,
  mar = 0.2,
  dest.path = getOption("dest.path", "."),
  flim = NULL,
  col = "white",
  width = 7,
  height = 4,
  res = 100,
  label = TRUE,
  fast.spec = FALSE,
  srt = 0,
  cex = 1,
  ...
)
```

Arguments

X	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass and 7) "sound.id": ID of sounds used to identify counterparts across distances. Each sound must have a unique ID within a distance.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is NULL. Ignored if bp = NULL. If supplied, 'hop.size' is ignored.
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 0. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
collevels	A numeric vector of length 3. Specifies levels to partition the amplitude range of the spectrogram (in dB). The more levels the higher the resolution of the spectrogram. Default is seq(-40, 0, 1). seq(-115, 0, 1) will produces spectrograms similar to other acoustic analysis software packages.
palette	Color palette function for spectrogram. Default is viridis. See spectro for more palettes. Palettes as gray.2 may work better when fast.spec = TRUE.
duration	A numeric vector of length 1. Specifies the overall duration of the clip that will be plotted. Notice that only the initial part of the test files are plotted as this is enough to tell the precision of the alignment.
mar	numeric vector of length 1. Specifies the margins adjacent to the start of the first annotation to be included in the plot.
dest.path	Character string containing the directory path where the image files will be saved. If not supplied the current working directory will be used instead. Can be set globally for the current R session via the "dest.path" option (see options).
flim	A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrogram, as in spectro. Default is NULL which will plot spectrograms in the full frequency range (0 - nyquist frequency).

col	Character string controlling the color of lines and sound ID labels.
width	Numeric vector of length 1. Single value (in inches) indicating the width of the output image files. Default is 7.
height	Numeric vector of length 1. Single value (in inches) indicating the height of the output image files. Default is 4.
res	Numeric argument of length 1. Controls image resolution. Default is 100 (faster) although 300 - 400 is recommended for publication/presentation quality.
label	Logical to control if labels (from 'sound.id' column in 'X') are plotted. Default is TRUE.
fast.spec	Logical. If TRUE then image function is used internally to create spectrograms, which substantially increases performance (much faster), although some options become unavailable, as collevels (amplitude scale). Default is FALSE.
srt	Numeric argument of length 1. The rotation (in degrees) of the sound id labels. Default is 0.
cex	Numeric argument of length 1controlling the size of sound id text labels. Default is 1.
...	Additional arguments to be passed to the internal spectrogram creating function for customizing graphical output. The function is a modified version of spectro, so it takes the same arguments.

Details

This functions aims to simplify the evaluation of the alignment of test sound files from align_test_files. The function creates a single spectrogram for each sound file (saved at 'dest.path'). Spectrograms include the first few seconds of the sound files (controlled by 'duration') which is usually enough to tell the precision of the alignment. The plots include vertical lines denoting the start and end of each sound as well as the sound ID ('sound.id' column in 'X'). Note that no plot is created in the R graphic device.

Value

Image files in jpeg format with spectrograms in the working directory, one for each sound file in 'X'. It also returns the file path of the images invisibly.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

manual_realign; auto_realign; find_markers; align_test_files

Other test sound alignment: align_test_files(), auto_realign(), find_markers(), manual_realign()

Examples

```
{
  # load example data
  data("test_sounds_est")

  # plot (look into temporary working directory `tempdir()`)
  plot_aligned_sounds(X = test_sounds_est, dest.path = tempdir(), duration = 3, ovlp = 0)
}
```

`plot_blur_ratio` *Plot blur ratio*

Description

`plot_blur_ratio` plots time and frequency blur ratio in sounds referenced in an extended selection table.

Usage

```
plot_blur_ratio(
  X,
  type = c("envelope", "spectrum"),
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  env.smooth = getOption("env.smooth", 200),
  spec.smooth = getOption("spec.smooth", 5),
  res = 150,
  flim = c("-1", "+1"),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
  palette = viridis::viridis,
  collevels = seq(-120, 0, 5),
  dest.path = getOption("dest.path", "."),
  path = getOption("sound.files.path", "."),
  colors = viridis::viridis(3),
  n.samples = if (type == "envelope") 100 else 500
)
```

Arguments

`X`	The output of `set_reference_sounds` which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function `selection_table` from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file),

3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See `set_reference_sounds` for more details on the structure of 'X'.

`type` Character vector of length 1 indicating the type of blur ratio to plot. The two options are 'envelope' (for regular blur ratio as in `blur_ratio`, default) and 'spectrum' (for spectrum blur ratio as in `spectrum_blur_ratio`).

`cores` Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see `options`).

`pb` Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).

`env.smooth` Numeric vector of length 1 determining the length of the sliding window (in amplitude samples) used for a sum smooth for amplitude envelope calculation (used internally by `env`). Default is 200.

`spec.smooth` Numeric vector of length 1 determining the length of the sliding window used for a sum smooth for power spectrum calculation (in kHz). Default is 5. Can be set globally for the current R session via the "spec.smooth" option (see `options`).

`res` Numeric argument of length 1. Controls image resolution. Default is 150 (faster) although 300 - 400 is recommended for publication/presentation quality.

`flim` A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrograms, as in `spectro`. Default is `NULL`. Alternatively, a character vector similar to `c("-1", "1")` in which the first number is the value to be added to the minimum bottom frequency in 'X' and the second the value to be added to the maximum top frequency in 'X'. This is computed independently for each sound id so the frequency limit better fits the frequency range of the annotated signals. This is useful when test sounds show marked differences in their frequency ranges.

`hop.size` A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see `options`).

`wl` a vector with a single even integer number specifying the window length of the spectrogram, default is `NULL`. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see `options`).

`ovlp` Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in `spectro`. Only used when plotting. Default is 70. Applied to both spectra and spectrograms on image files. Can be set globally for the current R session via the "ovlp" option (see `options`).

`palette` A color palette function to be used to assign colors in the plot, as in `spectro`. Default is `viridis`.

`collevels` Numeric vector indicating a set of levels which are used to partition the amplitude range of the spectrogram (in dB) as in `spectro`. Default is `seq(-120, 0, 5)`.

`dest.path` Character string containing the directory path where the image files will be saved. If not supplied the current working directory will be used instead. Can be set globally for the current R session via the "dest.path" option (see `options`).

`path` Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see `options`).

`colors` Character vector of length 4 containing the colors to be used for the color to identify the reference sound (element 1), the color to identify the test sound (element 2) and the color of blurred region (element 3).

`n.samples` Numeric vector of length 1 specifying the number of amplitude samples (or frequency bins if `spectrum = TRUE`) to use for representing power distributions. Default is 100 for `type = "envelope"` and 500 for `type = "spectrum"`. If null the raw power distribution is used (note that this can result in high RAM memory usage for large data sets).

Details

The function generates image files (in 'jpeg' format) for each possible blur ratio estimation in 'X'. The image files show the spectrograms of both sounds and the overlaid power distribution (either amplitude envelopes or power spectrum, see argument 'type') as probability mass functions (PMF). The output graphs highlight the mismatch between the compared distribution which represent the estimated blur ratio returned by either `blur_ratio` or `spectrum_blur_ratio`. Spectrograms are shown within the frequency range of the reference sound and also show dotted lines with the time (type = "envelope") or frequency range (type = "spectrum") in which energy distributions where computed.

Value

It returns 1 image file (in 'jpeg' format) for each blur ratio estimation, showing spectrograms of both sounds and the overlaid amplitude envelopes (or power spectra if `spectrum = TRUE`) as probability mass functions (PMF). Spectrograms are shown within the frequency range of the reference sound. It also returns the file path of the images invisibly.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Dabelsteen, T., Larsen, O. N., & Pedersen, S. B. (1993). Habitat-induced degradation of sound signals: Quantifying the effects of communication sounds and bird location on blur ratio, excess attenuation, and signal-to-noise ratio in blackbird song. The Journal of the Acoustical Society of America, 93(4), 2206.

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

envelope_correlation, spectrum_blur_ratio, blur_ratio

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

  # add reference to X
  X <- set_reference_sounds(X = test_sounds_est)

  # create plots
  plot_blur_ratio(X = X, dest.path = tempdir())
}
```

---

plot_degradation *Save multipanel plots with reference and test sounds*

Description

plot_degradation creates multipanel plots (as image files) with reference and test sounds by distance and transect.

Usage

```
plot_degradation(
  X,
  nrow = 4,
  env.smooth = getOption("env.smooth", 200),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
  path = getOption("sound.files.path", "."),
  dest.path = getOption("dest.path", "."),
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  collevels = seq(-120, 0, 5),
  palette = viridis::viridis,
```

```
  flim = c("-1", "+1"),
  envelope = TRUE,
  spectrum = TRUE,
  heights = c(4, 1),
  widths = c(5, 1),
  margins = c(2, 1),
  row.height = 2,
  col.width = 2,
  cols = viridis::mako(4, alpha = 0.3),
  res = 120,
  ...
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
nrow	Numeric vector of length 1 with the number of rows per image file. Default is 4. This would be dynamically adjusted if more rows than needed are set.
env.smooth	Numeric vector of length 1 determining the length of the sliding window (in amplitude samples) used for a sum smooth for amplitude envelope and power spectrum calculations (used internally by env). Default is 200.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Only used when plotting. Default is 70. Applied to both spectra and spectrograms on image files. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

`dest.path` Character string containing the directory path where the image files will be saved. If not supplied the current working directory will be used instead. Can be set globally for the current R session via the "dest.path" option (see `options`).

`cores` Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see `options`).

`pb` Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).

`collevels` Numeric vector indicating a set of levels which are used to partition the amplitude range of the spectrogram (in dB) as in `spectro`. Default is `seq(-120, 0, 5)`.

`palette` A color palette function to be used to assign colors in the plot, as in `spectro`. Default is `viridis`.

`flim` A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrogram, as in `spectro`. Default is `NULL`. Alternatively, a character vector similar to `c("-1", "1")` in which the first number is the value to be added to the minimum bottom frequency in 'X' and the second the value to be added to the maximum top frequency in 'X'. This is computed independently for each sound id so the frequency limit better fits the frequency range of the annotated signals. This is useful when test sounds show marked differences in their frequency ranges.

`envelope` Logical to control if envelopes are plotted. Default is `TRUE`.

`spectrum` Logical to control if power spectra are plotted. Default is `TRUE`.

`heights` Numeric vector of length 2 to control the relative heights of spectrogram (first number) and amplitude envelope (second number) when `envelope = TRUE`. Default is c(4, 1).

`widths` Numeric vector of length 2 to control the relative widths of spectrogram (first number) and power spectrum (second number) when `spectrum = TRUE`. Default is c(5, 1).

`margins` Numeric vector of length 2 to control the relative time of the test sound (first number) and adjacent margins (i.e. adjacent background noise, second number) to be included in the spectrogram `spectrum = TRUE`. Default is c(2, 1) which means that each margin next to the sound is half the duration of the sound. Note that all spectrograms will have the same time length so margins will be calculated to ensure all spectrograms match the duration of the spectrogram in the longest sound. As such, this argument controls the margin on the longest sound.

`row.height` Numeric vector of length 1 controlling the height (in inches) of sound panels in the output image file. Default is 2.

`col.width` Numeric vector of length 1 controlling the width (in inches) of sound panels in the output image file. Default is 2.

`cols` Character vector of length 4 containing the colors to be used for the background of column and row title panels (element 1), the color of amplitude envelopes (element 2), the color of power spectra (element 3), and the background color of envelopes and spectra (element 4).

res	Numeric argument of length 1. Controls image resolution. Default is 120 (faster) although 300 - 400 is recommended for publication/presentation quality.
...	Additional arguments to be passed to the internal spectrogram creating function for customizing graphical output. The function is a modified version of spectro, so it takes the same arguments.

Details

The function aims to simplify the visual inspection of sound degradation by producing multipanel figures (saved in 'dest.path') containing visualizations of each test sound and its reference. Sounds are sorted by distance (columns) and transect (if more than 1). Visualizations include spectrograms, amplitude envelopes and power spectra (the last 2 are optional). Each row includes all the copies of a sound id for a given transect (the row label includes the sound id in the first line and transect in the second line), also including its reference if it comes from another transect. Ambient noise annotations (sound.id 'ambient') are excluded. Amplitude envelopes and power spectra are computed using the functions envelope (warbleR package) and spec (seewave package) respectively. This two visualizations show the power distribution in time and frequency between the minimum and maximum power values for each sound. Therefore scales are not necessarily comparable across panels. If transect id is not supplied in 'X' (i.e. no 'transect' column) the function assumes that all sounds are from the same transect. Only a single copy of a test sound ("sound.id" label) is allowed per transect/distance combination. The function uses internally spectro to create the spectrograms.

Value

One ore more image files with a multipanel figure of spectrograms of test sound by distance, sound id and transect. It also returns the file path of the images invisibly.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

blur_ratio, plot_aligned_sounds, plot_degradation

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
# load example data
data("test_sounds_est")
```

```
# order so spectrograms from same sound id as close in the graph
test_sounds_est <- test_sounds_est[order(test_sounds_est$sound.id), ]

# set directory to save image files
options(dest.path = tempdir())

# method 1
Y <- set_reference_sounds(X = test_sounds_est)

# plot degradation spectrograms
plot_degradation(
  X = Y, nrow = 3, ovlp = 95
)

# using other color palettes
plot_degradation(
  X = Y, nrow = 3, ovlp = 95,
  cols = viridis::magma(4, alpha = 0.3),
  palette = viridis::magma
)

# missing some data, 2 rows
plot_degradation(
  X = Y[-3, ], nrow = 2, ovlp = 95,
  cols = viridis::mako(4, alpha = 0.4), palette = viridis::mako, wl = 200
)

# changing marging and high overlap
plot_degradation(X = Y, margins = c(5, 1), nrow = 6, ovlp = 95)

# more rows than needed (will adjust it automatically)
plot_degradation(X = Y, nrow = 10, ovlp = 90)
```

`set_reference_sounds` *Set reference for test sounds*

Description

`set_reference_sounds` set rows to be used as reference for each test sound.

Usage

```
set_reference_sounds(
  X,
  method = getOption("method", 1),
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances (and transects if more than 1) and 8) "distance": distance (numeric) at which each test sound was re-recorded. A 'transect' column labeling those sounds recorded in the same transect is required if 'method = 2'. 'X' can only have 1 copy for any given sound id in a distance or a transect-distance combination (if column 'transect' is supplied). In addition, 'selec' column values in 'X' cannot be duplicated within a sound file ('sound.files' column) as this combination is used to refer to specific rows in the output 'reference' column.
method	Integer vector of length 1 to indicate the 'experimental design' for measuring degradation. Two methods are available: • 1: compare sounds (by 'sound.id') with their counterpart that was recorded at the closest distance to source (e.g. compare sounds recorded at 5m, 10m and 15m with its counterpart recorded at 1m). This is the default method. The function will try to use references from the same transect. However, if there is another test sound from the same 'sound.id' at a shorter distance in other transects, it will be used as reference instead. This behavior aims to account for the fact that in this type of experiments reference sounds are typically recorded at 1 m and at single transect. • 2: compare all sounds with their counterpart recorded at the distance immediately before within a transect (e.g. a sound recorded at 10m compared with the same sound recorded at 5m, then sound recorded at 15m compared with same sound recorded at 10m and so on). 'transect' column in 'X' is required. Can be set globally for the current R session via the "method" option (see options).
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

Details

This function adds a 'reference' column defining which sounds will be used by other functions as reference. Two methods are available (see 'methods' argument description). For method 1 the

function will attempt to use re-recorded sounds from the shortest distance in the same transect as reference. However, if there is another re-recorded sound from the same 'sound.id' at a shorter distance in other transects, it will be used as reference instead. This behavior aims to account for the fact that in this type of experiments reference sounds are typically recorded at 1 m and at single transect. Note that if users want to define their own reference sound this can be set manually. If so, NAs must be used to indicate rows to be ignored. References must be indicated as a the combination of the 'sound.files' and 'selec' column. For instance, '10m.wav-1' indicates that the row in which the 'selec' column is '1' and the sound file is '10m.wav' should be used as reference. The function also checks that the information 'X' is in the right format so it wont produce errors in downstream analysis (see 'X' argument description for details on format). The function will ignore rows in which the column 'sound.id' is equals to 'ambient', 'start_marker' or 'end_marker'.

Value

An object similar to 'X' with one additional column, 'reference', with the ID of the sounds to be used as reference by degradation-quantifying functions in downstream analyses. The ID is created as `paste(X$sound.files, X$selec, sep = "-")`.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., & Smith-Vidaurre, G. (2017). warbleR: An R package to streamline analysis of animal acoustic signals. Methods in Ecology and Evolution, 8(2), 184-191.

See Also

check_sound_files, check_sels

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

# save wav file examples
X <- test_sounds_est[test_sounds_est$sound.files != "master.wav", ]

# method 1
Y <- set_reference_sounds(X = X)

# method 2
Y <- set_reference_sounds(X = X, method = 2)
}
```

signal_to_noise_ratio *Measure attenuation as signal-to-noise ratio*

Description

signal_to_noise_ratio measures attenuation as signal-to-noise ratio of sounds referenced in an extended selection table.

Usage

```
signal_to_noise_ratio(
  X,
  mar = NULL,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  eq.dur = FALSE,
  noise.ref = c("adjacent", "custom"),
  snr.formula = 1,
  bp = "freq.range",
  hop.size = getOption("hop.size", 1),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 0),
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations (typically the output of align_test_files). Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for band-pass, 6) "top.freq": high frequency for bandpass and 7) "sound.id": ID of sounds used to identify counterparts across distances (only needed for "custom" noise reference, see "noise.ref" argument). If the "sound.id" column is supplied then SNR is only computed for those rows with a sound.id different from "ambient", "start_marker" or "end_marker".
mar	numeric vector of length 1. Specifies the margins adjacent to the start point of the annotation over which to measure ambient noise.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).

`eq.dur` Logical. Controls whether the ambient noise segment that is measured has the same duration to that of the sound (if `TRUE`. Default is `FALSE`). If `TRUE` then 'mar' and 'noise.ref' arguments are ignored.

`noise.ref` Character vector of length 1 to determined which noise segment must be used for measuring ambient noise. Two options are available:

- `adjacent`: measure ambient noise right before test sounds (using argument 'mar' to define duration of ambient noise segments).
- `custom`: measure ambient noise segments referenced in the selection table (labeled as 'ambient' in the 'sound.id' column). Those segments will be used to apply the same ambient noise reference to all sounds in a sound file. Therefore, at least one 'ambient' selection for each sound file must be provided. If several 'ambient' selections by sound file are supplied, then the root mean square of the amplitude envelope will be averaged across those selections.

`snr.formula` Integer vector of length 1. Selects the formula to be used to calculate the signal-to-noise ratio (S = signal , N = background noise):

- `1`: ratio of S amplitude envelope root mean square to N amplitude envelope root mean square (`20 * log10(rms(env(S))/rms(env(N)))`) as described by Darden (2008).
- `2`: ratio of the difference between S amplitude envelope root mean square and N amplitude envelope root mean square to N amplitude envelope root mean square (`20 * log10((rms(env(S)) - rms(env(N)))/rms(env(N)))`, as described by Dabelsteen et al (1993).

`bp` Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Alternatively, when set to 'freq.range' (default), the function will use the 'bottom.freq' and 'top.freq' for each sound as the bandpass range.

`hop.size` A numeric vector of length 1 specifying the time window duration (in ms). Default is 1 ms, which is equivalent to ~45 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see `options`).

`wl` A numeric vector of length 1 specifying the window length of the spectrogram, default is NULL. Ignored if `bp = NULL`. If supplied, 'hop.size' is ignored. Note that lower values will increase time resolution, which is more important for amplitude ratios calculations.

`ovlp` Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in `spectro`. Default is 0. Only used for bandpass filtering. Can be set globally for the current R session via the "ovlp" option (see `options`).

`path` Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see `options`).

Details

Signal-to-noise ratio (SNR) measures sound amplitude level in relation to ambient noise. Noise is measured on the background noise immediately before the test sound. A general margin in which ambient noise will be measured must be specified. Alternatively, a selection of ambient noise can be used as reference (see 'noise.ref' argument). When margins overlap with another sound nearby, SNR will be inaccurate, so margin length should be carefully considered. Any SNR less than or equal to one suggests background noise is equal to or overpowering the sound. The function will measure signal-to-noise ratio within the supplied frequency range (e.g. bandpass) of the reference signal ('bottom.freq' and 'top.freq' columns in 'X') by default (that is, when `bp = 'freq.range'`. SNR can be ~0 when both tail and signal have very low amplitude.

Value

Object 'X' with an additional column, 'signal.to.noise.ratio', with the signal-to-noise ratio values (in dB).

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Holland J, Dabelsteen T, Pedersen SB, Paris AL (2001) Potential ranging cues contained within the energetic pauses of transmitted wren song. Bioacoustics 12(1):3-20. Darden, SK, Pedersen SB, Larsen ON, & Dabelsteen T. (2008). Sound transmission at ground level in a short-grass prairie habitat and its implications for long-range communication in the swift fox *Vulpes velox*. The Journal of the Acoustical Society of America, 124(2), 758-766.

See Also

excess_attenuation

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), spcc(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

  # using measure ambient noise reference selections
  signal_to_noise_ratio(X = test_sounds_est, mar = 0.05, noise.ref = "custom")

  # using margin for ambient noise of 0.05 and adjacent measure ambient noise reference
  signal_to_noise_ratio(X = test_sounds_est, mar = 0.05, noise.ref = "adjacent")
```

```
}
```

---

spcc *Measure spectrographic cross-correlation as a measure of sound distortion*

Description

spcc measures spectrographic cross-correlation as a measure of sound distortion in sounds referenced in an extended selection table.

Usage

```
spcc(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  cor.method = c("pearson", "spearman", "kendall"),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 90),
  wn = "hanning",
  path = getOption("sound.files.path", ".")
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
cor.method	Character string indicating the correlation coefficient to be applied ("pearson", "spearman", or "kendall", see cor).

hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 90. High values of ovlp slow down the function but produce more accurate results. Can be set globally for the current R session via the "ovlp" option (see options).
wn	A character vector of length 1 specifying the window name as in ftwindow.
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).

Details

Spectrographic cross-correlation measures frequency distortion of sounds as a similarity metric. Values close to 1 means very similar spectrograms (i.e. little sound distortion has occurred). Cross-correlation is measured of sounds in which a reference playback has been re-recorded at increasing distances. The 'sound.id' column must be used to indicate the function to only compare sounds belonging to the same category (e.g. song-types). The function compares each sound to the corresponding reference sound within the supplied frequency range (e.g. bandpass) of the reference sound ('bottom.freq' and 'top.freq' columns in 'X'). Two methods for computing cross-correlation are provided (see 'method' argument). The function is a wrapper on warbleR's cross_correlation function.

Value

Object 'X' with an additional column, 'cross.correlation', containing the computed spectrogram cross-correlation coefficients.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Clark, C.W., Marler, P. & Beeman K. (1987). Quantitative analysis of animal vocal phonology: an application to Swamp Sparrow song. Ethology. 76:101-115.

See Also

blur_ratio, manual_realign, cross_correlation

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spectrum_blur_ratio(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

  # add reference to X
  X <- set_reference_sounds(X = test_sounds_est)

  # get spcc
  spcc(X = X)
}
```

---

spectrum_blur_ratio *Measure blur ratio in the frequency domain*

Description

spectrum_blur_ratio measures blur ratio of frequency spectra from sounds referenced in an extended selection table.

Usage

```
spectrum_blur_ratio(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  spec.smooth = getOption("spec.smooth", 5),
  spectra = FALSE,
  res = 150,
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
  path = getOption("sound.files.path", "."),
  n.bins = 100
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
spec.smooth	Numeric vector of length 1 determining the length of the sliding window used for a sum smooth for power spectrum calculation (in kHz). Default is 5.
spectra	Logical to control if power spectra are returned (as attributes). Default is FALSE.
res	Numeric argument of length 1. Controls image resolution. Default is 150 (faster) although 300 - 400 is recommended for publication/presentation quality.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Applied to both spectra and spectrograms on image files.
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 70. Applied to both spectra and spectrograms on image files. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
n.bins	Numeric vector of length 1 specifying the number of frequency bins to use for representing power spectra. Default is 100. If null the raw power spectrum is used (note that this can result in high RAM memory usage for large data sets). Power spectrum values are interpolated using approx.

Details

Spectral blur ratio measures the degradation of sound as a function of the change in sound power in the frequency domain, analogous to the blur ratio proposed by Dabelsteen et al (1993) for the time

domain (and implemented in blur_ratio). Low values indicate low degradation of sounds. The function measures the blur ratio of spectra from sounds in which a reference playback has been re-recorded at different distances. Spectral blur ratio is measured as the mismatch between power spectra (expressed as probability density functions) of the reference sound and the re-recorded sound. The function compares each sound type to the corresponding reference sound. The 'sound.id' column must be used to tell the function to only compare sounds belonging to the same category (e.g. song-types). Two methods for setting the experimental design are provided. All wave objects in the extended selection table must have the same sampling rate so the length of spectra is comparable. The function uses spec internally to compute power spectra. NA is returned if at least one the power spectra cannot be computed.

Value

Object 'X' with an additional column, 'spectrum.blur.ratio', containing the computed spectrum blur ratio values. If spectra = TRUE the output would include power spectra for all sounds as attributes ('attributes(X)$spectra').

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Dabelsteen, T., Larsen, O. N., & Pedersen, S. B. (1993). Habitat-induced degradation of sound signals: Quantifying the effects of communication sounds and bird location on blur ratio, excess attenuation, and signal-to-noise ratio in blackbird song. The Journal of the Acoustical Society of America, 93(4), 2206. Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481

See Also

blur_ratio

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_correlation(), tail_to_signal_ratio()

Examples

```
{
  # load example data
  data("test_sounds_est")

# add reference to X
X <- set_reference_sounds(X = test_sounds_est)

  # get spetrum blur ratio
  spectrum_blur_ratio(X = X)
```

```
  # using method 2
   X <- set_reference_sounds(X = test_sounds_est, method = 2)
  spectrum_blur_ratio(X = X)

  # get power spectra
  sbr <- spectrum_blur_ratio(X = X, spectra = TRUE)


  # plot spectra
  spctr <- attributes(sbr)$spectra

  # make distance a factor for plotting
  spctr$distance <- as.factor(spctr$distance)

  # plot
  rlang::check_installed("ggplot2")
  library(ggplot2)

  ggplot(spctr[spctr$freq > 0.3, ], aes(y = amp, x = freq,
  col = distance)) +
  geom_line() +
  facet_wrap(~sound.id) +
  scale_color_viridis_d(alpha = 0.7) +
  labs(x = "Frequency (kHz)", y = "Amplitude (PMF)") +
  coord_flip() +
  theme_classic()

}
```

`spectrum_correlation` *Measure frequency spectrum correlation*

Description

`spectrum_correlation` measures frequency spectrum correlation of sounds referenced in an extended selection table.

Usage

```
spectrum_correlation(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  cor.method = c("pearson", "spearman", "kendall"),
  spec.smooth = getOption("spec.smooth", 5),
  hop.size = getOption("hop.size", 11.6),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 70),
```

```
  path = getOption("sound.files.path", "."),
  n.bins = 100
)
```

Arguments

X	The output of set_reference_sounds which is an object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function selection_table from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances and 8) "reference": identity of sounds to be used as reference for each test sound (row). See set_reference_sounds for more details on the structure of 'X'.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
cor.method	Character string indicating the correlation coefficient to be applied ("pearson", "spearman", or "kendall", see cor).
spec.smooth	Numeric vector of length 1 determining the length of the sliding window used for a sum smooth for power spectrum calculation (in kHz). Default is 5.
hop.size	A numeric vector of length 1 specifying the time window duration (in ms). Default is 11.6 ms, which is equivalent to 512 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see options).
wl	a vector with a single even integer number specifying the window length of the spectrogram, default is NULL. If supplied, 'hop.size' is ignored. Odd integers will be rounded up to the nearest even number. Can be set globally for the current R session via the "wl" option (see options).
ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in spectro. Default is 70. Can be set globally for the current R session via the "ovlp" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
n.bins	Numeric vector of length 1 specifying the number of frequency bins to use for representing power spectra. Default is 100. If null the raw power spectrum is used (note that this can result in high RAM memory usage for large data sets). Power spectrum values are interpolated using approx.

Details

spectral correlation measures the similarity of two sounds in the frequency domain. The function measures the spectral correlation coefficients of sounds in which a reference playback has been re-recorded at increasing distances. Values range from 1 (identical frequency spectrum, i.e. no degradation) to 0. The 'sound.id' column must be used to indicate the function to only compare sounds belonging to the same category (e.g. song-types). The function will then compare each sound to the corresponding reference sound. Two methods for computing spectral correlation are provided (see 'method' argument). The function uses `meanspec` internally to compute power spectra. Use `spectrum_blur_ratio` to extract raw spectra values. NA is returned if at least one the power spectra cannot be computed.

Value

Object 'X' with an additional column, 'spectrum.correlation', containing the computed frequency spectrum correlation coefficients.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Apol, C.A., Sturdy, C.B. & Proppe, D.S. (2017). Seasonal variability in habitat structure may have shaped acoustic signals and repertoires in the black-capped and boreal chickadees. Evol Ecol. 32:57-74.

See Also

`envelope_correlation`, `spectrum_blur_ratio`

Other quantify degradation: `blur_ratio()`, `detection_distance()`, `envelope_correlation()`, `plot_blur_ratio()`, `plot_degradation()`, `set_reference_sounds()`, `signal_to_noise_ratio()`, `spcc()`, `spectrum_blur_ratio()`, `tail_to_signal_ratio()`

Examples

```
{
  # load example data
  data("test_sounds_est")

  # method 1
  # add reference column
  Y <- set_reference_sounds(X = test_sounds_est)

  # run spectrum correlation
  spectrum_correlation(X = Y)

  # method 2
  Y <- set_reference_sounds(X = test_sounds_est, method = 2)
```

```
  # spectrum_correlation(X = Y)
}
```

---

spot_ambient_noise *Find a segment of ambient noise to be used as reference*

Description

spot_ambient_noise finds a segment of ambient noise to be used as reference by other functions.

Usage

```
spot_ambient_noise(
  X,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  path = getOption("sound.files.path", "."),
  length = NULL,
  ovlp = 0,
  fun = function(x) which.min(abs(x - mean(x)))
)
```

Arguments

X	Object of class 'data.frame', or 'selection_table' (a class are created by the function selection_table from the warbleR package) with the test sound files' annotations ('extended_selection_table' are not supported). Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass, 6) "top.freq": high frequency for bandpass, 7) "sound.id": ID of sounds used to identify counterparts across distances/transects. 'selec' column values in 'X' cannot be duplicated within a sound file ('sound.files' column) as this combination is used to refer to specific rows.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see options).
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
path	Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see options).
length	Numeric. Length (in s) of the segments to be used as ambient noise. Must be supplied. Default is NULL.

ovlp	Numeric vector of length 1 specifying the percentage of overlap between two consecutive segments. Default is 0. Can be set globally for the current R session via the "ovlp" option (see options).
fun	Function to be applied to select the segment to be used as ambient noise. It must be a function that takes a numeric vector (peak sound pressure level values for each candidate segment) and a single value with the index of the value to keep. Default is function(x) which.min(abs(x - mean(x))).

Details

This function finds a segment of ambient noise to be used as reference by other functions. The function first finds candidate segments that do not overlap with annotated sounds in 'X'. Then, it calculates the peak sound pressure level (SPL) of each candidate segment and applies the function supplied by the argument 'fun' to select a single segment. By default 'fun' searches for the segment with the closest value to the mean peak SPL across all candidate segments. Ambient noise annotations are added as a new row in 'X'. Ambient noise annotations are used by the functions signal_to_noise_ratio and noise_profile to determine background noise levels. Note that this function does not work with annotations in 'extended_selection_table' format.

Value

An object similar to 'X' with one additional row for each sound file, containing the selected 'ambient' reference.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

#' Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Araya-Salas, M., & Smith-Vidaurre, G. (2017). warbleR: An R package to streamline analysis of animal acoustic signals. Methods in Ecology and Evolution, 8(2), 184-191.

See Also

signal_to_noise_ratio, noise_profile

Other prepare acoustic data: master_sound_file(), synth_sounds()

Examples

```
{
# set temporary directory
td <- tempdir()
# load example data
data("test_sounds_est")
########## save acoustic data (This doesn't have to be done
# with your own data as you will have them as sound files already.)
```

```
 # save example files in working director
for (i in unique(test_sounds_est$sound.files)[1:2]) {
 writeWave(object = attr(test_sounds_est, "wave.objects")[[i]],
           file.path(tempdir(), i))
}
test_sounds_df <- as.data.frame(test_sounds_est)
test_sounds_df <- test_sounds_df[test_sounds_df$sound.id != "ambient", ]
test_sounds_df <-
 test_sounds_df[test_sounds_df$sound.files %in%
  unique(test_sounds_est$sound.files)[1:2], ]
####
# closest to mean (default)
spot_ambient_noise(X = test_sounds_df, path = td, length = 0.12, ovlp = 20)

# min peak
spot_ambient_noise(X = test_sounds_df, path = td, length = 0.12, ovlp = 20, fun = which.min)
}
```

---

synth_sounds *Create synthetic sounds*

Description

synth_sounds create synthetic sounds

Usage

```
synth_sounds(
  replicates = 1,
  frequencies,
  durations,
  nharmonics = 1,
  fm = FALSE,
  am = FALSE,
  am.amps = rep(c(1:4, 3:2), length.out = 11),
  mar = 0.05,
  seed = NULL,
  sig2 = 0.3,
  shuffle = FALSE,
  hrm.freqs = c(1/2, 1/3, 2/3, 1/4, 3/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10),
  sampling.rate = 44.1,
  pb = TRUE,
  freq.range = 2
)
```

Arguments

replicates	Numeric vector of length 1 indicating the number of replicates for each treatment combination. Default is 1. Useful for measuring variation in transmission parameters.

frequencies	Numeric vector with the different frequencies (in seconds) to synthesize. A Brownian bridge motion stochastic process (diff.fun == "BB") is used to simulate frequency modulation (see simulate_songs).
durations	Numeric vector with the different durations (in seconds) to synthesize.
nharmonics	Numeric vector of length 1 specifying the number of harmonics to simulate. 1 indicates that only the fundamental frequency harmonic will be simulated.
fm	Logical to control if both frequency modulated sounds and pure tones (i.e. non-modulated sounds) are synthesize. If FALSE (default) only pure tones are synthesized.
am	Logical to control if both amplitude modulated sounds and non-modulated sounds are synthesize. If FALSE (default) only non-modulated sounds are synthesized.
am.amps	Numeric vector with the relative amplitude for each time step to simulate amplitude modulation (only applied to the fundamental frequency). The default value (rep(c(1:4, 3:2), length.out = 11)) has 2 amplitude peaks (although only applied if 'am = TRUE')
mar	Numeric vector with the duration of margins of silence around sounds in seconds. Default is 0.05.
seed	Numeric vector of length 1. This allows users to get the same results in different runs (using se.seed internally). Default is NULL.
sig2	Numeric vector of length 1 defining the sigma value of the brownian motion model (used for simulating frequency modulation). Default is 0.3.
shuffle	Logical to control if the position of sounds is randomized. Having all sounds from the same treatment in a sequence can be problematic if an environmental noise masks them. Hence 'shuffle' is useful to avoid having sounds from the same treatment next to each other. Default is FALSE.
hrm.freqs	Numeric vector with the frequencies of the harmonics relative to the fundamental frequency. The default values are c(1/2, 1/3, 2/3, 1/4, 3/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10).
sampling.rate	Numeric vector of length 1. Sets the sampling frequency of the wave object (in kHz). Default is 44.1.
pb	Logical argument to control if progress bar is shown. Default is TRUE. Can be set globally for the current R session via the "pb" option (see options).
freq.range	Numeric vector of length 1 with the frequency range around the simulated frequency in which signals will modulate. Default is 2 which means that sounds will range +/- 1 kHz around the target frequency.

Details

This function creates synthetic sounds that can be used for playback experiments to understand the link between signal structure and its transmission properties. The function can add variation in signal structure in 5 features:

- frequency: continuous, argument 'frequencies'.
- duration: continuous, argument 'durations'.

- harmonic structure: binary (harmonics vs no-harmonics), arguments 'nharmonics' and 'hrm.freqs'.
- frequency modulation: variation in fundamental frequency across time. Binary (modulated vs non-modulated), arguments 'fm' and 'sig2'.
- amplitude modulation: variation in amplitude across time. Binary (modulated vs non-modulated), arguments 'am' and 'am.amps'.

Sound for all possible combinations of the selected structure dimensions will be synthesized. The output is an extended selection table, which can be input into master_sound_file to create the .wav file. The functions uses simulate_songs internally for synthesizing individual sounds. A Brownian bridge motion stochastic process (diff.fun == "BB") is used to simulate frequency modulation. The output table contains columns for each of the varying features and a 'treatment' column (useful to tell sound from the same combination of features when using replicates).

Value

An extended selection table, which can be input into master_sound_file to create the .wav file. The table contains columns for each of the varying features a 'treatment' column (useful to tell the acoustic features of each sound) and a 'replicate' column indicating the replicates for each 'treatment'.

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., & Smith-Vidaurre, G. (2017). warbleR: An R package to streamline analysis of animal acoustic signals. Methods in Ecology and Evolution, 8(2), 184-191.

See Also

simulate_songs from the package warbleR.

Other prepare acoustic data: master_sound_file(), spot_ambient_noise()

Examples

```
## Not run:

synthetic_est <- synth_sounds(
  mar = 0.01,
  frequencies = c(1, 2, 3, 5),
  durations = 0.1,
  fm = TRUE,
  am = TRUE,
  nharmonics = 4,
  shuffle = TRUE,
  replicates = 3
)
```

```
## End(Not run)
```

`tail_to_signal_ratio` *Measure reverberations as tail-to-signal ratio*

Description

`tail_to_signal_ratio` measures reverberations as tail-to-signal ratio of sounds referenced in an extended selection table.

Usage

```
tail_to_signal_ratio(
  X,
  mar,
  cores = getOption("mc.cores", 1),
  pb = getOption("pb", TRUE),
  tsr.formula = 1,
  bp = "freq.range",
  hop.size = getOption("hop.size", 1),
  wl = getOption("wl", NULL),
  ovlp = getOption("ovlp", 0),
  path = getOption("sound.files.path", ".")
)
```

Arguments

`X`	Object of class 'data.frame', 'selection_table' or 'extended_selection_table' (the last 2 classes are created by the function `selection_table` from the warbleR package) with the test sound files' annotations . Must contain the following columns: 1) "sound.files": name of the .wav files, 2) "selec": unique selection identifier (within a sound file), 3) "start": start time and 4) "end": end time of selections, 5) "bottom.freq": low frequency for bandpass and 6) "top.freq": high frequency for bandpass.
`mar`	numeric vector of length 1. Specifies the margins adjacent to end of the sound over which to measure tail power.
`cores`	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing). Can be set globally for the current R session via the "mc.cores" option (see `options`).
`pb`	Logical argument to control if progress bar is shown. Default is `TRUE`. Can be set globally for the current R session via the "pb" option (see `options`).
`tsr.formula`	Integer vector of length 1. Determine the formula to be used to calculate the tail-to-signal ratio (S = signal, T = tail, N = background noise):

- 1: log10 of the ratio of T amplitude envelope quadratic mean to S amplitude envelope quadratic mean (`log10(rms(env(T))/rms(env(S)))`) as described by Dabelsteen et al. (1993).
- 2: ratio of T amplitude envelope quadratic mean to N amplitude envelope quadratic mean (`log10(rms(env(T))/rms(env(N)))`). N is measure in the margin right before the sound. So tsr.formula 2 actually measures tail-to-noise ratio.

`bp` Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Alternatively, when set to 'freq.range' (default), the function will use the 'bottom.freq' and 'top.freq' for each sound as the bandpass range.

`hop.size` A numeric vector of length 1 specifying the time window duration (in ms). Default is 1 ms, which is equivalent to ~45 wl for a 44.1 kHz sampling rate. Ignored if 'wl' is supplied. Can be set globally for the current R session via the "hop.size" option (see `options`). Note that this might be internally adjusted if the number of samples in the tail is lower than the hop.size.

`wl` A numeric vector of length 1 specifying the window length of the spectrogram, default is NULL. Ignored if `bp = NULL`. If supplied, 'hop.size' is ignored. Note that lower values will increase time resolution, which is more important for amplitude calculations.

`ovlp` Numeric vector of length 1 specifying the percentage of overlap between two consecutive windows, as in `spectro`. Default is 0. Only used for bandpass filtering. Can be set globally for the current R session via the "ovlp" option (see `options`).

`path` Character string containing the directory path where the sound files are found. Only needed when 'X' is not an extended selection table. If not supplied the current working directory is used. Can be set globally for the current R session via the "sound.files.path" option (see `options`).

Details

Tail-to-signal ratio (TSR) measures the ratio of power in the tail of reverberations to that in the test sound. A general margin in which reverberation tail will be measured must be specified. The function will measure TSR within the supplied frequency range (e.g. bandpass) of the reference sound ('bottom.freq' and 'top.freq' columns in 'X'). Two methods for computing reverberations are provided (see 'tsr.formula' argument). Note that 'tsr.formula' 2 is not equivalent to the original description of TSR in Dabelsteen et al. (1993) and is better referred to as tail-to-noise ratio. Tail-to-signal ratio values are typically negative as signals tend to have higher power than that in the reverberating tail. TSR can be ~0 when both tail and signal have very low amplitude.

Value

Object 'X' with an additional column, 'tail.to.signal.ratio', with the tail-to-signal ratio values (in dB).

Author(s)

Marcelo Araya-Salas (<marcelo.araya@ucr.ac.cr>)

References

Araya-Salas, M., Grabarczyk, E. E., Quiroz-Oliva, M., Garcia-Rodriguez, A., & Rico-Guevara, A. (2025). Quantifying degradation in animal acoustic signals with the R package baRulho. Methods in Ecology and Evolution, 00, 1-12. https://doi.org/10.1111/2041-210X.14481 Darden, SK, Pedersen SB, Larsen ON, & Dabelsteen T. (2008). Sound transmission at ground level in a short-grass prairie habitat and its implications for long-range communication in the swift fox *Vulpes velox*. The Journal of the Acoustical Society of America, 124(2), 758-766. Mathevon, N., Dabelsteen, T., & Blumenrath, S. H. (2005). Are high perches in the blackcap Sylvia atricapilla song or listening posts? A sound transmission study. The Journal of the Acoustical Society of America, 117(1), 442-449.

See Also

excess_attenuation

Other quantify degradation: blur_ratio(), detection_distance(), envelope_correlation(), plot_blur_ratio(), plot_degradation(), set_reference_sounds(), signal_to_noise_ratio(), spcc(), spectrum_blur_ratio(), spectrum_correlation()

Examples

```
{
  # load example data

  data("test_sounds_est")

  # set global options
  options(pb = FALSE)

  # using margin for noise of 0.01
  tsr <- tail_to_signal_ratio(X = test_sounds_est, mar = 0.01)

  # use tsr.formula 2 which is equivalent to tail-to-noise ratio
  tsr <- tail_to_signal_ratio(X = test_sounds_est, mar = 0.01, tsr.formula = 2)
}
```

---

test_sounds_est *Extended selection table with re-recorded playbacks*

Description

Extended selection table (est) of 7 re-recorded synthetic sounds. The synthetic sounds are 2 s long, frequency modulated, amplitude modulated, and were broadcast and re-recorded at 3 distances (1, 10 and 30m, column 'distance'). The data was created by the function selection_table from the warbleR package. The master sound file data used to create this test sounds is found in the example object master_est.

Usage

```
data(test_sounds_est)
```

Format

Extended selection table object in the warbleR format, which contains annotations and acoustic data.

Source

Marcelo Araya-Salas

See Also

Other data sets: master_est

Index

∗ **data sets**
  master_est, 28
  test_sounds_est, 65
∗ **datasets**
  master_est, 28
  test_sounds_est, 65
∗ **miscellaneous**
  add_noise, 2
  attenuation, 7
  noise_profile, 31
∗ **prepare acoustic data**
  master_sound_file, 29
  spot_ambient_noise, 58
  synth_sounds, 60
∗ **quantify degradation**
  blur_ratio, 10
  detection_distance, 13
  envelope_correlation, 16
  plot_blur_ratio, 37
  plot_degradation, 40
  set_reference_sounds, 44
  signal_to_noise_ratio, 47
  spcc, 50
  spectrum_blur_ratio, 52
  spectrum_correlation, 55
  tail_to_signal_ratio, 63
∗ **test sound alignment**
  align_test_files, 5
  auto_realign, 8
  find_markers, 22
  manual_realign, 24
  plot_aligned_sounds, 34

add_noise, 2
add_noise(), *8*, *33*
align_test_files, *3*, 5, *23*, *25–27*, *36*, *47*
align_test_files(), *10*, *23*, *27*, *36*
approx, *12*, *53*, *56*
attenuation, 7, *15, 16*
attenuation(), *4*, *33*
auto_realign, 8, *23*, *27*, *36*
auto_realign(), *6*, *23*, *27*, *36*

blur_ratio, *10*, 10, *18*, *38–40*, *43*, *52*, *54*
blur_ratio(), *16*, *18*, *40*, *43*, *46*, *49*, *52*, *54*, *57*, *65*

check_sels, *46*
check_sound_files, *46*
cor, *17*, *50*, *56*
cross_correlation, *9, 10*, *51, 52*

detection_distance, 13
detection_distance(), *13*, *18*, *40*, *43*, *46*, *49*, *52*, *54*, *57*, *65*

env, *11*, *15*, *17*, *38*, *41*
envelope, *43*
envelope_correlation, *13*, 16, *21*, *40*, *57*
envelope_correlation(), *13*, *16*, *40*, *43*, *46*, *49*, *52*, *54*, *57*, *65*
excess_attenuation, 19, *33*, *49*, *65*
exp_raven, *30*

find_markers, *5, 6*, *9*, 22, *25*, *27*, *36*
find_markers(), *6*, *10*, *27*, *36*
ftwindow, *9*, *51*

gray.2, *26*, *35*

manual_realign, *6*, *23*, 24, *36*, *52*
manual_realign(), *6*, *10*, *23*, *36*
master_est, 28, *65, 66*
master_sound_file, *22, 23*, 29, *62*
master_sound_file(), *59*, *62*
meanspec, *33*, *57*

noise, *3*
noise_profile, 31, *59*
noise_profile(), *4*, *8*

options, *3*, *5*, *9*, *11, 12*, *14, 15*, *17*, *19, 20*, *22, 23*, *25*, *30*, *32, 33*, *35*, *38, 39*, *41, 42*, *45*, *47, 48*, *50, 51*, *53*, *56*, *58, 59*, *61*, *63, 64*

plot_aligned_sounds, *6*, 34, *43*
plot_aligned_sounds(), *6*, *10*, *23*, *27*
plot_blur_ratio, 37
plot_blur_ratio(), *13*, *16*, *18*, *43*, *46*, *49*, *52*, *54*, *57*, *65*
plot_degradation, 40, *43*
plot_degradation(), *13*, *16*, *18*, *40*, *46*, *49*, *52*, *54*, *57*, *65*

se.seed, *61*
selection_table, *3–5*, *8, 9*, *11*, *14*, *17*, *19*, *22*, *25*, *28, 29*, *32*, *35*, *37*, *41*, *45*, *47*, *50*, *53*, *56*, *58*, *63*, *65*
set_reference_sounds, *11*, *14*, *17*, *19*, *37, 38*, *41*, 44, *50*, *53*, *56*
set_reference_sounds(), *13*, *16*, *18*, *40*, *43*, *49*, *52*, *54*, *57*, *65*
signal_to_noise_ratio, *4*, 47, *59*
signal_to_noise_ratio(), *13*, *16*, *18*, *40*, *43*, *46*, *52*, *54*, *57*, *65*
simulate_songs, *61, 62*
spcc, *21*, 50
spcc(), *13*, *16*, *18*, *40*, *43*, *46*, *49*, *54*, *57*, *65*
spec, *43*, *54*
spectro, *9*, *12*, *17*, *20*, *25, 26*, *35, 36*, *38, 39*, *41–43*, *48*, *51*, *53*, *56*, *64*
spectrum_blur_ratio, *13*, *18*, *38–40*, 52, *57*
spectrum_blur_ratio(), *13*, *16*, *18*, *40*, *43*, *46*, *49*, *52*, *57*, *65*
spectrum_correlation, 55
spectrum_correlation(), *13*, *16*, *18*, *40*, *43*, *46*, *49*, *52*, *54*, *65*
spot_ambient_noise, 58
spot_ambient_noise(), *30*, *62*
synth_sounds, 60
synth_sounds(), *30*, *59*

tail_to_signal_ratio, 63
tail_to_signal_ratio(), *13*, *16*, *18*, *40*, *43*, *46*, *49*, *52*, *54*, *57*
template_correlator, *23*
template_detector, *23*
test_sounds_est, *28*, 65

viridis, *26*, *35*, *38*, *42*