Package 'warbleR'

April 27, 2017

Type Package

Title Streamline Bioacoustic Analysis

Version 1.1.8

Date 2017-04-26

Author Marcelo Araya-Salas & Grace Smith Vidaurre

Maintainer Marcelo Araya-Salas <araya-salas@cornell.edu>

Description A tool to streamline the analysis of animal acoustic signal structure. The package offers functions for downloading avian vocalizations from the open-access online repository Xeno-Canto, displaying the geographic extent of the recordings, manipulating sound files, detecting acoustic signals or importing detected signals from other software, assessing performance of methods that measure acoustic similarity, conducting cross-correlations, dynamic time warping, measuring acoustic parameters and analysing interactive vocal signals, among others. Most functions working iteratively allow parallelization to improve computational efficiency.

License GPL (>= 2)

Imports bitops, doParallel, dtw, fftw, foreach, graphics, grDevices, iterators, jpeg, monitoR, parallel, pbapply, proxy, RCurl, rjson, stats, signal, utils

Depends R (>= 3.2.1), maps, tuneR, seewave (>= 2.0.1)

LazyData TRUE

URL https://github.com/maRce10/warbleR

BugReports https://github.com/maRce10/warbleR/issues

NeedsCompilation no

Suggests knitr, ggplot2, pbmcapply

VignetteBuilder knitr

RoxygenNote 5.0.1

Repository CRAN

Date/Publication 2017-04-27 17:32:58 UTC

R topics documented:

autodetec *Automatically detect vocalizations in sound files*

Description

autodetec automatically detects the start and end of vocalizations in sound files based on amplitude, duration, and frequency range attributes.

Usage

```
autodetec(X = NULL, threshold = 15, envt = "abs", ssmooth = NULL, msmooth = NULL,
  power = 1, bp = NULL, osci = FALSE, wl = 512, xl = 1, picsize = 1, res = 100,
  flim = c(0,22), ls = FALSE, sxrow = 10, rows = 10, mindur = NULL, maxdur =
 NULL, redo = FALSE, img = TRUE, it = "jpeg", set = FALSE, flist = NULL, smadj = NULL,
  parallel = 1, path = NULL, pb = TRUE)
```

Arguments

X	Data frame with results from manualoc function or any data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end).
threshold	A numeric vector of length 1 specifying the amplitude threshold for detecting signals (in %).
envt	Character vector of length 1 specifying the type of envelope to be used: "abs" for absolute amplitude envelope or "hil" for Hilbert amplitude envelope. Default is "abs".
ssmooth	A numeric vector of length 1 to smooth the amplitude envelope with a sum smooth function. Default is NULL.
msmooth	A numeric vector of length 2 to smooth the amplitude envelope with a mean sliding window. The first component is the window length and the second is the overlap between successive windows (in %). Faster than ssmooth but time detection is much less accurate. Will be deprecated in future versions. Default is NULL.
power	A numeric vector of length 1 indicating a power factor applied to the amplitude envelope. Increasing power will reduce low amplitude modulations and increase high amplide modulations, in order to reduce background noise. Default is 1 (no change).
bp	Numeric vector of length 2 giving the lower and upper limits of a frequency bandpass filter (in kHz). Default is c(0, 22).
osci	Logical argument to add an oscillogram underneath spectrogram, as in spectro. Default is FALSE. Not applied if ls is TRUE.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
xl	Numeric vector of length 1, a constant by which to scale spectrogram width. Default is 1.

picsize	Numeric argument of length 1. Controls the relative size of the spectrogram. Default is 1.
res	Numeric argument of length 1 controling resolution of images. Default is 100 (faster) although 300 - 400 is recommended for publication/ presentation quality.
flim	A numeric vector of length 2 for the frequency limit in kHz of the spectrogram, as in spectro. Default is c(0, 22).
ls	Logical argument. If TRUE, long spectrograms as in lspec are produced.
sxrow	A numeric vector of length 1. Specifies seconds of spectrogram per row when creating long spectrograms. Default is 10. Applied when ls = TRUE and/or when X is not provided.
rows	A numeric vector of length 1. Specifies number of rows per image file when creating long spectrograms. Default is 10. Applied when ls = TRUE and/or when X is not provided.
mindur	Numeric vector of length 1 giving the shortest duration (in seconds) of the signals to be detected. It removes signals below that threshold.
maxdur	Numeric vector of length 1 giving the longest duration (in seconds) of the signals to be detected. It removes signals above that threshold.
redo	Logical argument. If TRUE all selections will be analyzed again when code is rerun. If FALSE only the selections that do not have an 'autodetec' generated image file in the working directory will be analyzed. Default is FALSE.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
set	A logical argument indicating wheter the settings of the autodetection process should be included in the image file name. If TRUE, threshold (th), envelope (envt), bandpass (bp), power (pw), smooth (smo, either mmsooth[1] or ssmooth), maxdur (mxdu), and mindur (midu) are included.
flist	character vector or factor indicating the subset of files that will be analyzed. Ignored if X is provided.
smadj	adjustment for amplitude smoothing. Character vector of length one indicating whether start end values should be adjusted. "start", "end" or "both" are the inputs admitted by this argument. Amplitude smoothing through ssmooth generates a predictable deviation from the actual start and end positions of the signals, determined by the threshold and ssmooth values. This deviation is more obvious (and problematic) when the increase and decrease in amplitude at the start and end of the signal (respectively) is not gradual. Ignored if ssmooth is NULL.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

Details

This function determines the start and end of signals in the segments of the sound files listed in the input data frame. Alternatively, if no data frame is provided, the function detects signals across each entire sound file and creates long spectrograms highlighting the start and of the detected signals for all sound files in the working directory. The input data frame should have the following columns: c("sound.files","selec","start","end"). The ouptut of `manualoc` can be used as the input data frame. This function uses a modified version of the `timer` function from seewave package to detect signals.

Value

Image files with spectrograms showing the start and end of the detected signals. It also returns a data frame containing the start and end of each signal by sound file and selection number.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>). Implements a modified version of the timer function from seewave.

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

ad <- autodetec(threshold = 5, env = "hil", ssmooth = 300, power=1,
bp=c(2,9), xl = 2, picsize = 2, res = 200, flim= c(1,11), osci = TRUE,
wl = 300, ls = FALSE, sxrow = 2, rows = 4, mindur = 0.1, maxdur = 1, set = TRUE)

#run it with different settings
ad <- autodetec(threshold = 90, env = "abs", ssmooth = 300, power = 1, redo = TRUE,
bp=c(2,9), xl = 2, picsize = 2, res = 200, flim= c(1,11), osci = TRUE,
wl = 300, ls = FALSE,  sxrow = 2, rows = 4, mindur=0.1, maxdur=1, set = TRUE)

#check this folder!!
getwd()

## End(Not run)
```

catalog *Create catalog of vocal signals*

Description

catalog produces spectrograms of selections (signals) split into multiple rows and columns.

Usage

```
catalog(X, flim = c(0, 22), nrow = 4, ncol = 3, same.time.scale = TRUE,
collev = seq(-40, 0, 1), ovlp = 50, parallel = 1, mar = 0.05, wl = 512, gr = FALSE,
pal = reverse.gray.colors.2, it = "jpeg", path = NULL, pb = TRUE, fast.spec = FALSE,
res = 100, orientation = "v", labels = c("sound.files", "selec"), height = NULL,
width = NULL, tags = NULL, tag.pal = list(temp.colors, heat.colors), legend = 3,
cex = 1, leg.wd = 1, img.suffix = NULL, tag.widths = c(1, 1), hatching = 0,
breaks = c(5, 5))
```

Arguments

X	Data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). Default is NULL.
flim	A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrogram, as in spectro. Default is c(0,22).
nrow	A numeric vector of length 1. Specifies number of rows. Default is 4.
ncol	A numeric vector of length 1. Specifies number of columns. Default is 3.
same.time.scale	Logical. Controls if all spectrograms are in the same time scale (i.e. have the same duration).
collev	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.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 50. High values of ovlp slow down the function but produce more accurate selection limits (when X is provided).
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
mar	Numeric vector of length 1. Specifies the margins adjacent to the start and end points of selections, dealineating spectrogram limits. Default is 0.05.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
gr	Logical argument to add grid to spectrogram. Default is FALSE.

pal — Color palette function for spectrogram. Default is reverse.gray.colors.2. See spectro for more palettes. Palettes as gray.2 may work better when fast.spec = T.

it — A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".

path — Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.

pb — Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast.spec' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.

res — Numeric argument of length 1. Controls image resolution. Default is 100 (faster) although 300 is recommended for publication/presentation quality. Note that high resolution produce significantly bigger image files. This could be problematic when creating pdf files using catalog.

orientation — String. Indicates whether a letter page size image is produced in vertical ('v' option) or horizontal orientation ('h' option). Note that width and height can also be specified.

labels — String vector. Provides the column names that will be used as labels above the corresponding spectrograms.

height — Numeric. Single value (in inches) indicating the height of the output image files. Default is 11 for vertical orientation.

width — Numeric. Single value (in inches) indicating the width of the output image files. Default is 8.5 for vertical orientation.

tags — String vector. Provides the column names that will be used for the color tagging legend above. Tags can also be numeric. Continuous variables would be break down in 10 color classes. spectrograms.

tag.pal — List of color palette function for tags. Should be of length 1 or 2. Default is list(temp.colors, heat.colors).

legend — A numeric vector of length 1 controling a legend for color tags is added. Ignored if no tags are provided. Four values are allowed:

- 0: No label
- 1: Label for the first color tag
- 2: Label for the second color tag
- 3: Labels both color tags

Default is 3.

cex — A numeric vector of length 1 giving the amount by which text (including labels and axis) should be magnified. Default is 1.

`leg.wd` Numeric. Controls the width of the legend column. Default is 1.

`img.suffix` A character vector of length 1 with a suffix (label) to add at the end of the names of image files. Default is `NULL` (no suffix). Can be useful to label catalogs from different individuals, species or sites.

`tag.widths` A numeric vector of length 2 to control de relative width of the color tags (when 2 tags are provided).

`hatching` A numeric vector of length 1 controling cross-hatching is used for color tags. Several cross-hatching patterns are used to make tags with similar colors more distinguishable. Four values are allowed:

- `0`: No cross-hatching
- `1`: Cross-hatching the first color tag
- `2`: Cross-hatching the second color tag
- `3`: Cross-hatching both color tags

`breaks` Numeric vector of length 1 or 2 controling the number of intervals in which a numeric tag will be divided. The numbers control the first and second tags respectively. Ignored if tags are not numeric. Default is c(5, 5).

Details

This functions aims to simplify the visual exploration of multiple vocalizations. The function plots a matrix of spectrograms from a selection table. Spectrograms can be labeled or color tagged to facilitate exploring variation related to a parameter of interest (e.g. location, song type). A legend will be added to help match colors with tag levels (if legend is > 0). Different color palettes can be used for each tag. Numeric tags are split in intervals (the number of intervals can be controlled with break argument). The width and height can also be adjusted to fit more column and/or rows. This files can be put together in a single pdf file with `catalog2pdf`. We recommend using low resolution (~60-100) and smaller dimensions (width & height < 10) if aiming to generate pdfs (otherwise pdfs could be pretty big).

Value

Image files with spectrograms of whole sound files in the working directory. Multiple pages can be returned, depending on the length of each sound file.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

https://marce10.github.io/2017-03-17-Creating_song_catalogs/ `catalog2pdf`

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())
# save sound file examples
```

```
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
 writeWave(Phae.long3,"Phae.long3.wav")
 writeWave(Phae.long4,"Phae.long4.wav")


catalog(X = selec.table, flim = c(1, 10), nrow = 4, ncol = 2, same.time.scale = T,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200, gr = FALSE,
 orientation = "v",  labels = c("sound.files", "selec"), legend = 0)

 #different time scales and tag palette
catalog(X = selec.table, flim = c(1, 10), nrow = 4, ncol = 2, same.time.scale = F,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200,
 orientation = "v",  labels = c("sound.files", "selec"), legend = 0,
 tag.pal = list(terrain.colors))

 #adding tags and changing spectro palette
catalog(X = selec.table, flim = c(1, 10), nrow = 4, ncol = 2, same.time.scale = F,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200, pal = reverse.heat.colors,
 orientation = "v",  labels = c("sound.files", "selec"), legend = 1,
 tag.pal = list(terrain.colors), tags = "sound.files")

 #create a bigger selection table
 X <- rbind(selec.table, selec.table, selec.table, selec.table)
 X <- rbind(X, X)

 #create some simulated labels
 X$songtype <- sample(letters[13:15], nrow(X), replace = T)
 X$indiv <- sample(letters[1:12], nrow(X), replace = T)

# 12 columns in 5 rows, 2 tags
catalog(X = X, flim = c(1, 10), nrow = 5, ncol = 12, same.time.scale = F,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200,
 orientation = "v",  labels = c("sound.files", "selec"), legend = 3,
 collev = seq(-65, 0, 5), tag.pal = list(terrain.colors), tags = c("songtype", "indiv"))


# with legend
catalog(X = X, flim = c(1, 10), nrow = 5, ncol = 12, same.time.scale = F,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200, gr = FALSE,
 orientation = "v",  labels = c("sound.files", "selec"), legend = 3,
 width = 20, collev = seq(-65, 0, 5), tag.pal = list(terrain.colors),
  tags = c("songtype", "indiv"))

  # horizontal orientation
catalog(X = X, flim = c(1, 10), nrow = 5, ncol = 12, same.time.scale = F,
 ovlp = 90, parallel = 1, mar = 0.01, wl = 200, gr = FALSE,
 orientation = "h",  labels = c("sound.files", "selec"), legend = 3,
 width = 20, collev = seq(-65, 0, 5), tag.pal = list(terrain.colors),
  tags = c("songtype", "indiv"))
check this floder
getwd()
```

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

---

catalog2pdf catalog2pdf *combines* catalog *images into pdfs*

Description

catalog2pdf combines catalog images into pdfs

Usage

```
catalog2pdf(keep.jpeg = TRUE, overwrite = FALSE, parallel = 1, path = NULL,
pb = TRUE, by.img.suffix = FALSE, ...)
```

Arguments

keep.jpeg	Logical argument. Indicates whether jpeg files should be kept (default) or remove. (including sound file and page number) should be magnified. Default is 1.
overwrite	Logical argument. If TRUE all jpeg pdf will be produced again when code is rerun. If FALSE only the ones missing will be produced. Default is FALSE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
by.img.suffix	Logical. If TRUE catalogs with the same image suffix will be put together in a single pdf (so one pdf per image suffix in the catalog images). Default is FALSE (i.e. no suffix).
...	Additional arguments to be passed to the internal pdf creating function pdf for customizing output.

Details

The function combines catalog images in .jpeg format from the catalog function into pdfs. Note that using lower resolution and smaller dimension (width and height) when creating catalogs will substantially decrease the size of pdf files (which could be pretty big).

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

catalog2pdf, https://marce10.github.io/2017-03-17-Creating_song_catalogs/

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

# save sound file examples
data(list = c("Phae.long1", "Phae.long2"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")

catalog(X = selec.table, nrow = 2, ncol = 4)

#now create single pdf removing jpeg
catalog2pdf(keep.jpeg = FALSE)

check this floder
getwd()

## End(Not run)
```

---

checksels *Check selection data frames*

Description

`checksels` checks whether selections can be read by subsequent functions.

Usage

```
checksels(X, parallel =  1, path = NULL, check.header = FALSE)
```

Arguments

`X`	data frame with the following columns: 1) "sound.files": name of the .wav files, 2) "sel": number of the selections, 3) "start": start time of selections, 4) "end": end time of selections. The ouptut of `manualoc` or `autodetec` can be used as the input data frame.
`parallel`	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
`path`	Character string containing the directory path where the sound files are located. If `NULL` (default) then the current working directory is used.
`check.header`	Logical. Controls whether sound file headers correspond to the actual file properties (i.e. if is corrupted). This could significantly affect the performance of the function (much slower) particularly with long sound files.

Details

This function checks 1) if the selections listed in the data frame correspond to .wav files in the working directory, 2) if the sound files can be read and if so, 3) if the start and end time of the selections are found within the duration of the sound files. Note that the sound files should be in the working directory (or the directory provided in 'path'). This is useful for avoiding errors in dowstream functions (e.g. specan, xcorr, catalog, dfDTW). Note that corrupt files can be fixed using fixwavs) ('sox' must be installed to be able to run this function).

Value

A data frame including the columns in the input data frame (X) and 2 additional columns: "check.res" (check selections), and "min.n.samples" (the smallest number of samples). Note the number of samples available in a selection limits the minimum window length (wl argument in other functions) that can be used in batch analyses.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

checkwavs

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

# save wav file examples
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")

checksels(X = selec.table)

## End(Not run)
```

---

checkwavs *Check .wav files*

Description

checkwavs checks whether .wav files can be read by subsequent functions.

Usage

```
checkwavs(X = NULL, path = NULL)
```

Arguments

X	Optional. Data frame with the following columns: 1) "sound.files": name of the .wav files, 2) "sel": number of the selections, 3) "start": start time of selections, 4) "end": end time of selections. The ouptut of manualoc or autodetec can be used as the input data frame. If provided the function also returns the smallest number of samples from the listed selections, which limits the minimum window length (wl argument in other functions) that can be used in batch analyses. This could be useful for avoiding errors in dowstream functions (e.g. specan).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.

Details

This function checks if .wav files in the working directory can be read. Users must set the working directory where they wish to check .wav files beforehand. If X is provided it also returns the smallest number of samples from the selections listed in X (if all files can be read). Note that corrupt files can be fixed using fixwavs) ('sox' must be installed to be able to run this function). The function is intended for a "quick and dirty" check of the .wav files in a selections data frame. For a more thourough analysis see checksels.

Value

If all .wav files are ok, returns message "All files can be read!". Otherwise returns the names of the corrupted .wav files.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

checksels seltailor

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

# save wav file examples
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

# without selection data frame
checkwavs()

# without selection data frame
```

```
checkwavs(X = selec.table)

## End(Not run)
```

---

compare.methods *Assessing the performance of acoustic distance measurements*

Description

compare.methods creates graphs to visually assess performance of acoustic distance measurements

Usage

```
compare.methods(X = NULL, flim = c(0, 22), bp = c(0, 22), mar = 0.1, wl = 512, ovlp = 90,
res = 150, n = 10, length.out = 30, methods = c("XCORR", "dfDTW", "ffDTW", "SP"),
it = "jpeg", parallel = 1, path = NULL, sp = NULL, pb = TRUE, gr = TRUE,
clip.edges = TRUE, threshold = 15, na.rm = FALSE, scale = FALSE,
 pal = reverse.gray.colors.2, img = TRUE, ...)
```

Arguments

X	Data frame with results from manualoc function, autodetec function, or any data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). Default NULL.
flim	A numeric vector of length 2 for the frequency limit in kHz of the spectrogram, as in spectro. Default is c(0, 22).
bp	numeric vector of length 2 giving the lower and upper limits of the frequency bandpass filter (in kHz) used in the acoustic distance methods. Default is c(0, 22). Note that for XCORR this argument sets the frange argument from the xcorr function.
mar	Numeric vector of length 1. Specifies plot margins around selection in seconds. Default is 0.1.
wl	A numeric vector of length 1 specifying the window length of the spectrogram and cross-correlation, default is 512.
ovlp	Numeric vector of length 1 specifying the percent overlap between two consecutive windows, as in spectro. Default is 90.
res	Numeric argument of length 1. Controls image resolution. Default is 150.
n	Numeric argument of length 1. Defines the number of plots to be produce. Default is 10.
length.out	A character vector of length 1 giving the number of measurements of fundamental or dominant frequency desired (the length of the time series). Default is 30.

methods	A character vector of length 2 giving the names of the acoustic distance methods that would be compared. The methods available are: cross-correlation (XCORR, from xcorr), dynamic time warping on dominant frequency time series (dfDTW, from dtw applied on dfts output), dynamic time warping on dominant frequency time series (ffDTW, from dtw applied on ffts output), spectral parameters (SP, from specan).
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
sp	Data frame with acoustic parameters as the one generated by specan. Must contain 'sound.files' and "selec' columns and the same selections as in 'X'.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
gr	Logical argument to control the presence of a grid on the spectrograms (default is TRUE).
clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed when using dfDTW and ffDTW methods. If TRUE this edges will be excluded and signal contour will be calculated on the remainging values. Default is TRUE.
threshold	amplitude threshold (%) for dominant and/or fundamental frequency detection when using dfDTW, ffDTW and SP methods. Default is 15.
na.rm	Logical. If TRUE all NAs produced when pairwise cross-correlations failed are removed from the results. This means that all selections with at least 1 cross-correlation that failed are excluded in both methods under comparison. Only apply if XCORR is one of the methods being compared.
scale	Logical. If TRUE dominant and/or fundamental frequency values are z-transformed using the scale function, which "ignores" differences in absolute frequencies between the signals in order to focus the comparison in the frequency contour, regardless of the pitch of signals. Default is TRUE.
pal	A color palette function to be used to assign colors in the spectrograms, as in spectro. Default is reverse.gray.colors.2.
img	A logical argument specifying whether an image files would be produced. Default is TRUE.
...	Additional arguments to be passed to a modified version of spectro for customizing graphical output. This includes fast.spec, an argument that speeds up the plotting of spectrograms (see description in specreator).

Details

This function produces graphs with spectrograms from 4 signals in the provided data frame that allow visual inspection of the performance of acoustic distance methods at comparing those signals.

The signals are randomly picked up from the provided data frame (X argument).The spectrograms are all plotted with the same frequency and time scales. The function compares 2 methods at a time. The methods available are: cross-correlation (XCORR, from `xcorr`), dynamic time warping on dominant frequency time series (dfDTW, from `dtw` applied on `dfts` output), dynamic time warping on dominant frequency time series (ffDTW, from `dtw` applied on `ffts` output), spectral parameters (SP, from `specan`). The graph also contains 2 scatterplots (1 for each method) of the acoustic space of all signals in the input data frame 'X'. The compared selections are randomly picked up from the pool of selections in the input data frame. The argument 'n' defines the number of comparisons (i.e. graphs) to be produced. The acoustic pairwise distance between signals is shown next to the arrows linking them. The font color of a distance value correspond to the font color of the method that generated it, as shown in the scatterplots. Distances are standardized, being 0 the distance of a signal to itself and 1 the farthest pairwise distance in the pool of signals. Principal Component Analysis (`princomp`) is applied to calculate distances when using spectral parameters (SP). In that case the first 2 PC's are used. Classical Multidimensional Scalling (also known as Principal Coordinates Analysis, (`cmdscale`)) is used for all other methods. Note that SP can only be used with at least 22 selections (number of rows in input data frame) as PCA only works with more units than variables. The graphs are return as image files in the working directory. The file name contains the methods being compared and the rownumber of the selections. This function uses internally a modified version of the `spectro` function from seewave package to create spectrograms.

Value

Image files with 4 spectrograms of the selection being compared and scatterplots of the acoustic space of all signals in the input data frame 'X'.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>). It uses internally a modified version of the `spectro` function from seewave package to create spectrograms.

See Also

https://marce10.github.io/2017-02-17-Choosing_the_right_method_for_measuring_acoustic_signal_structure/

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

compare.methods(X = selec.table, flim = c(0, 10), bp = c(0, 10), mar = 0.1, wl = 300,
ovlp = 90, res = 200, n = 10, length.out = 30,
methods = c("XCORR", "dfDTW"), parallel = 1, it = "jpeg")
```

```
#remove progress bar
compare.methods(X = selec.table, flim = c(0, 10), bp = c(0, 10), mar = 0.1, wl = 300,
ovlp = 90, res = 200, n = 10, length.out = 30,
methods = c("XCORR", "dfDTW"), parallel = 1, it = "jpeg", pb = FALSE)

#check this folder!
getwd()


#compare SP and XCORR
#first we need to create a larger data set as the PCA that summarizes the spectral parameters
#needs more units (rows) that variables (columns)
#so I just create a new selection table repeating 3 times selec.table
st2 <- rbind(selec.table, selec.table, selec.table)

#note that the selection labels should be also changed
st2$selec <- 1:nrow(st2)
#now we can compare SP method against XCORR
compare.methods(X = st2, flim = c(0, 10), bp = c(0, 10), mar = 0.1, wl = 300,
ovlp = 90, res = 200, n = 10, length.out = 30,
methods = c("XCORR", "SP"), parallel = 1, it = "jpeg")

#compare SP method against dfDTW
compare.methods(X = st2, flim = c(0, 10), bp = c(0, 10), mar = 0.1, wl = 300,
ovlp = 90, res = 200, n = 10, length.out = 30,
methods = c("dfDTW", "SP"), parallel = 1, it = "jpeg")

#alternatively we can provide our own SP matrix
sp <- specan(selec.table, bp = c(0, 10))

#and selec just a few variables to avoid the problem of # observations vs # parameters in PCA
sp <- sp[, 1:7]

compare.methods(X = selec.table, flim = c(0, 10), sp = sp, bp = c(0, 10), mar = 0.1, wl = 300,
ovlp = 90, res = 200, n = 10, length.out = 30,
methods = c("XCORR", "SP"), parallel = 1, it = "jpeg")

#note that "SP" should also be included as a method in 'methods'
#again, all images are saved in the working directory

## End(Not run)
```

---

coor.graph *Coordinated singing graphs*

Description

coor.graph creates graphs of coordinated singing and highlights the signals that overlap in time. The signals are represented by polygons of different colors.

Usage

```
coor.graph(X, only.coor = FALSE, ovlp = TRUE, xl = 1, res= 80, it = "jpeg", img = TRUE,
tlim = NULL)
```

Arguments

X	Data frame containing columns for singing event (sing.event), individual (indiv), and start and end time of signal (start and end).
only.coor	Logical. If TRUE only the segment in which both individuals are singing is included (solo singing is removed). Default is FALSE.
ovlp	Logical. If TRUE the vocalizations that overlap in time are highlighted. Default is TRUE.
xl	Numeric vector of length 1, a constant by which to scale spectrogram width. Default is 1.
res	Numeric argument of length 1. Controls image resolution. Default is 80.
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
img	Logical argument. If FALSE, image files are not produced. Default is TRUE. Note that images are return
tlim	Numeric vector of length 2 indicating the start and end time of the coordinated singing events to be displayed in the graphs.

Details

This function provides visualization for coordination of acoustic signals. Signals are shown as polygon across a time axis. It also shows which signals overlap, the amount of overlap, and highlights the individual responsible for the overlap using a color code. The width of the polygons depicting the time of overlap.

Value

The function returns a list of graphs, one for each singing event in the input data frame. The graphs can be plotted by simply calling the list. If 'img' is TRUE then the graphs are also saved in the working directory as files.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

Examples

```
## Not run:

# First set temporary folder
setwd(tempdir())

# load simulate singing events  (see data documentation)
```

```
, data(sim.coor.sing)

# make coor.graphs in tiff format
coor.graph(X = sim.coor.sing, ovlp = TRUE, only.coor = FALSE, xl =2, res =80,
it = "tiff", img = TRUE)


#' # make coor.graphs in graphic device format
cgs <- coor.graph(X = sim.coor.sing, ovlp = TRUE, only.coor = FALSE, img = FALSE)

cgs

## End(Not run)
```

`coor.test` *Randomization test for singing coordination*

Description

Monte Carlo randomization test to assess the statistical significance of singing coordination

Usage

```
coor.test(X, iterations = 1000, less.than.chance = TRUE, parallel = 1, pb = TRUE,
rm.imcomp = FALSE, cutoff = 2, rm.solo = FALSE)
```

Arguments

`X`	Data frame containing columns for singing event (sing.event), individual (indiv), and start and end time of signal (start and end).
`iterations`	number of iterations for shuffling and calculation of the expected number of overlaps. Default is 1000.
`less.than.chance`	Logical. If `TRUE` the test evaluates whether overlaps occur less often than expected by chance. If `FALSE` the opposite pattern is evaluted (whether overlaps occur more often than expected by chance). Default is `TRUE`.
`parallel`	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
`pb`	Logical argument to control progress bar. Default is `TRUE`. Note that progress bar is only used when parallel = 1.
`rm.imcomp`	Logical. If `TRUE` removes the events that don't have 2 interacting individuals. Default is `FALSE`.
`cutoff`	Numeric. Determines the minimum number of signals per individual in a singing event. Events not meeting this criterium are removed if rm.imcomp is `TRUE`. If rm.icomp is `FALSE` cutoff is ignored. Default is 2. Note that randomization tests are not reliable with very small sample sizes. Ideally 10 or more signals per individual should be available in each singing event.

`rm.solo` Logical. Controls if signals that are not intercalated at the start or end of the sequence are removed (if `TRUE`). For instances the sequence of signals A-A-A-B-A-B-A-B-B-B (in which A and B represent different individuals, as in the 'indiv' column) would be subset to A-B-A-B-A-B. Default is `FALSE`.

Details

This function calculates the probability of finding and equal or lower number (or higher if les.than.chance is `TRUE`) of song overlaps in a coordinated singing event. The function shuffles the sequences of signals and silence-between-signals for both individuals to produce a null distribution of expected number of overlaps by chance. The observed number of overlaps is compared to this expected values. The p-values are calculated as the proportion of random expected values that were lower (or higher) than the observed value. The function runs one test for each singing event in the input data frame. The function is equivalent to the "KeepGaps" methods described in Masco et al. 2015.

Value

A data frame with the observed number of overlaps (obs.overlaps), mean number of overlaps expected by chance, and p value.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

References

Masco, C., Allesina, S., Mennill, D. J., and Pruett-Jones, S. (2015). The Song Overlap Null model Generator (SONG): a new tool for distinguishing between random and non-random song overlap. Bioacoustics. 1-12.

Examples

```
## Not run:
#load  simulated singing data (see data documentation)
, data(sim.coor.sing)

# testing if coordination happens less than expected by chance
coor.test(sim.coor.sing, iterations = 100, less.than.chance = TRUE)

# testing if coordination happens more than expected by chance
coor.test(sim.coor.sing, iterations = 100, less.than.chance = FALSE)

## End(Not run)
```

`Cryp.soui` *Acoustic recording of* Crypturellus soui *(Little Tinamou).*

Description

Acoustic recording of *Crypturellus soui* (Little Tinamou).

Usage

```
data(Cryp.soui)
```

Format

One .wav file:

Cryp.soui *Crypturellus soui* recording

Source

http://www.xeno-canto.org/154190

`dfDTW` *Acoustic dissimilarity using dynamic time warping on dominant frequency contours*

Description

`dfDTW` calculates acoustic dissimilarity of dominant frequency contours using dynamic time warping. Internally it applies the `dtwDist` function from the `dtw` package.

Usage

```
dfDTW(X, wl = 512, length.out = 20, wn = "hanning", ovlp = 70, bp = c(0, 22),
  threshold = 5, img = TRUE, parallel = 1, path = NULL, img.suffix = "dfDTW", pb = TRUE,
  clip.edges = TRUE, window.type = "none", open.end = FALSE, scale = FALSE, ...)
```

Arguments

`X`	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of `manualoc` or `autodetec` can be used as the input data frame.
`wl`	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
`length.out`	A character vector of length 1 giving the number of measurements of dominant frequency desired (the length of the time series).

wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz). Default is c(0, 22).
threshold	amplitude threshold (%) for dominant frequency detection. Default is 5.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not availble in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
img.suffix	A character vector of length 1 with a suffix (label) to add at the end of the names of image files. Default is NULL.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed. If TRUE (default) this edges will be excluded and signal contour will be calculated on the remainging values. Note that DTW cannot be applied if missing values (e.i. when amplitude is not detected).
window.type	dtw windowing control parameter. Character: "none", "itakura", or a function (see dtw).
open.end	dtw control parameter. Performs open-ended alignments (see dtw).
scale	Logical. If TRUE dominant frequency values are z-transformed using the scale function, which "ignores" differences in absolute frequencies between the signals in order to focus the comparison in the frequency contour, regardless of the pitch of signals. Default is TRUE.
...	Additional arguments to be passed to trackfreqs for customizing graphical output.

Details

This function extracts the dominant frequency values as a time series and then calculates the pairwise acoustic dissimilarity using dynamic time warping. The function uses the approx function to interpolate values between dominant frequency measures. If 'img' is TRUE the function also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the dominant frequencies.

Value

A matrix with the pairwise dissimilarity values. If img is FALSE it also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the dominant frequencies.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

specreator for creating spectrograms from selections, snrspecs for creating spectrograms to optimize noise margins used in sig2noise and dfts, ffts, ffDTW for frequency contour overlaid spectrograms. https://marce10.github.io/2016-09-12-Similarity_of_acoustic_signals_with_dynamic_time_warping_(DTW)/

Other spectrogram.creators: dfts, ffDTW, ffts, snrspecs, sp.en.ts, specreator, trackfreqs

Examples

```
## Not run:
# set the temp directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long2, "Phae.long2.wav") #save sound files
writeWave(Phae.long1, "Phae.long1.wav")

# run function
dfDTW(selec.table, length.out = 30, flim = c(1, 12), bp = c(2, 9), wl = 300)


## End(Not run)
```

---

dfts *Extract the dominant frequency values as a time series*

Description

dfts extracts the dominant frequency values as a time series. of signals selected by manualoc or autodetec.

Usage

```
dfts(X, wl = 512, length.out = 20, wn = "hanning", ovlp = 70, bp = c(0, 22),
  threshold = 15, img = TRUE, parallel = 1, path = NULL, img.suffix = "dfts", pb = TRUE,
  clip.edges = FALSE, leglab = "dfts", ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.

wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
length.out	A character vector of length 1 giving the number of measurements of dominant frequency desired (the length of the time series).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz). Default is c(0, 22).
threshold	amplitude threshold (%) for dominant frequency detection. Default is 15.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located.
img.suffix	A character vector of length 1 with a sufix (label) to add at the end of the names of image files.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed. If TRUE this edges will be excluded and signal contour will be calculated on the remainging values. Default is FALSE.
leglab	A character vector of length 1 or 2 containing the label(s) of the frequency contour legend in the output image.
...	Additional arguments to be passed to trackfreqs for customizing graphical output.

Details

This function extracts the dominant frequency values as a time series. The function uses the approx function to interpolate values between dominant frequency measures. If there are no frequencies above the amplitude theshold at the begining or end of the signals then NAs will be generated. On the other hand, if there are no frequencies above the amplitude theshold in between signal segments in which amplitude was detected then the values of this adjacent segments will be interpolated to fill out the missing values (e.g. no NAs in between detected amplitude segments).

Value

A data frame with the dominant frequency values measured across the signals. If img is TRUE it also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the dominant frequencies (see trackfreqs description for more details).

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

sig2noise, trackfreqs, ffts, ffDTW, dfDTW

Other spectrogram.creators: dfDTW, ffDTW, ffts, snrspecs, sp.en.ts, specreator, trackfreqs

Examples

```
## Not run:
# set the temp directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long2, "Phae.long2.wav") #save sound files
writeWave(Phae.long1, "Phae.long1.wav")

# run function
dfts(X = selec.table, length.out = 30, flim = c(1, 12), bp = c(2, 9), wl = 300)


## End(Not run)
```

---

ffDTW *Acoustic dissimilarity using dynamic time warping on fundamental frequency contours*

Description

ffDTW calculates acoustic dissimilarity of fundamental frequency contours using dynamic time warping. Internally it applies the dtwDist function from the dtw package.

Usage

```
ffDTW(X, wl = 512, length.out = 20, wn = "hanning", ovlp = 70,
bp = c(0, 22), threshold = 5, img = TRUE, parallel = 1, path = NULL,
img.suffix = "ffDTW", pb = TRUE, clip.edges = TRUE, window.type = "none",
open.end = FALSE, scale = FALSE, ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.

length.out	A character vector of length 1 giving the number of measurements of fundamental frequency desired (the length of the time series).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz). Default is c(0, 22).
threshold	amplitude threshold (%) for fundamental frequency detection. Default is 5.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
img.suffix	A character vector of length 1 with a sufix (label) to add at the end of the names of image files. Default is NULL.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed. If TRUE (default) this edges will be excluded and signal contour will be calculated on the remainging values. Note that DTW cannot be applied if missing values (e.i. when amplitude is not detected).
window.type	dtw windowing control parameter. Character: "none", "itakura", or a function (see dtw).
open.end	dtw control parameter. Performs open-ended alignments (see dtw).
scale	Logical. If TRUE dominant frequency values are z-transformed using the scale function, which "ignores" differences in absolute frequencies between the signals in order to focus the comparison in the frequency contour, regardless of the pitch of signals. Default is TRUE.
...	Additional arguments to be passed to trackfreqs for customizing graphical output.

Details

This function extracts the fundamental frequency values as a time series and then calculates the pairwise acoustic dissimilarity of the selections using dynamic time warping. The function uses the approx function to interpolate values between fundamental frequency measures. If 'img' is TRUE the function also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the fundamental frequencies. Note that if no amplitude is detected at the begining or end of the signals then NAs will be generated. On the other hand, if amplitude is not detected in between signal segments in which amplitude was detected then the values of this adjacent segments will be interpolated to fill out the missing values (e.g. no NAs in between detected amplitude segments).

Value

A matrix with the pairwise dissimilarity values. If img is FALSE it also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the fundamental frequencies.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

specreator for creating spectrograms from selections, snrspecs for creating spectrograms to optimize noise margins used in sig2noise

dfDTW dfts, ffts, dfDTW

Other spectrogram.creators: dfDTW, dfts, ffts, snrspecs, sp.en.ts, specreator, trackfreqs

Examples

```
## Not run:
# set the temp directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long2, "Phae.long2.wav") #save sound files
writeWave(Phae.long1, "Phae.long1.wav")

# run function
ffDTW(selec.table, length.out = 30, flim = c(1, 12), img = T, bp = c(1, 9), wl = 300)


## End(Not run)
```

---

ffts *Extract the fundamental frequency values as a time series*

Description

ffts extracts the fundamental frequency values as a time series of signals selected by manualoc or autodetec.

Usage

```
ffts(X, wl = 512, length.out = 20, wn = "hanning", ovlp = 70, bp = c(0, 22),
  threshold = 15, img = TRUE, parallel = 1, path = NULL, img.suffix = "ffts", pb = TRUE,
  clip.edges = FALSE, leglab = "ffts", ff.method = "seewave", ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
length.out	A character vector of length 1 giving the number of measurements of fundamental frequency desired (the length of the time series).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz). Default is c(0, 22).
threshold	amplitude threshold (%) for fundamental frequency detection. Default is 15.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
img.suffix	A character vector of length 1 with a suffix (label) to add at the end of the names of image files.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed. If TRUE this edges will be excluded and signal contour will be calculated on the remainging values. Default is FALSE. #' @param leglab A character vector of length 1 or 2 containing the label(s) of the frequency contour legend in the output image.
leglab	A character vector of length 1 or 2 containing the label(s) of the frequency contour legend in the output image.
ff.method	Character. Selects the method used to calculate the fundamental frequency. Either 'tuneR' (using FF) or 'seewave' (using fund). Default is 'seewave'. 'tuneR' performs faster (and seems to be more accurate) than 'seewave'.
...	Additional arguments to be passed to trackfreqs. for customizing graphical output.

Details

This function extracts the fundamental frequency values as a time series. The function uses the approx function to interpolate values between fundamental frequency #' measures. If there are no

frequencies above the amplitude theshold at the begining or end of the signals then NAs will be generated. On the other hand, if there are no frequencies above the amplitude theshold in between signal segments in which amplitude was detected then the values of this adjacent segments will be interpolated to fill out the missing values (e.g. no NAs in between detected amplitude segments).

Value

A data frame with the fundamental frequency values measured across the signals. If img is TRUE it also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the fundamental frequencies (see trackfreqs description for more details).

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

sig2noise, trackfreqs, dfts, ffDTW, dfDTW

Other spectrogram.creators: dfDTW, dfts, ffDTW, snrspecs, sp.en.ts, specreator, trackfreqs

Examples

```
## Not run:
# set the temp directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files
writeWave(Phae.long2, "Phae.long2.wav") #save sound files

# run function
ffts(selec.table, length.out = 50, flim = c(1, 12), bp = c(2, 9), wl = 300)

Note that fundamental frequency is not accurate for noisy signals, works better with pure tones


## End(Not run)
```

---

filtersels *Subset selection data frames based on manually filtered image files*

Description

filtersels subsets selection data frames based on image files that have been manually filtered.

Usage

```
filtersels(X, path = NULL, lspec = FALSE, img.suffix = NULL, it = "jpeg",
 incl.wav = TRUE, missing = FALSE, index = FALSE)
```

Arguments

X	data frame with the following columns: 1) "sound.files": name of the .wav files, 2) "sel": number of the selections. The ouptut of manualoc or autodetec can be used as the input data frame.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
lspec	A logical argument indicating if the image files to be use for filtering were produced by the function lspec. All the image files that correspond to a sound file must be deleted in order to be filtered out.
img.suffix	A character vector of length 1 with the suffix (label) at the end of the names of the image files. Default is NULL (i.e. no suffix as in the images produced by specreator). Ignored if lspec = TRUE.
it	A character vector of length 1 giving the image type ("tiff", "jpeg" or "pdf") Default is "jpeg". Note that pdf files can only be generated by lspec2pdf.
incl.wav	Logical. To indicate if sound files extensions (".wav") are included (TRUE, default) or not in the image file names.
missing	Logical. Controls whether the output data frame (or row index if is index = TRUE) contains the selections with images in the working directory (Default, missing = FALSE) or the ones with no image.
index	Logical. If TRUE and missing = FALSE the row index for the selections with images in the working directory is returned. If missing = TRUE) then the row index of the ones with no image is returned instead. Default is FALSE.

Details

This function subsets selections (or sound files if lspec is TRUE) listed in a data frame based on the image files from spectrogram-creating functions (e.g. specreator) in the working directory. Only the selections/sound files with and image in the working directory will remain. This is useful for excluding selections from undesired signals. Note that the image files should be in the working directory (or the directory provided in 'path').

Value

If all .wav files are ok, returns message "All files are ok!". Otherwise returns "These file(s) cannot be read" message with names of the corrupted .wav files.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

# save wav file examples
```

```
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")

specreator(selec.table, flim = c(0, 11), inner.mar = c(4,4.5,2,1), outer.mar = c(4,2,2,1),
picsize = 2, res = 300, cexlab = 2, mar = 0.05, wl = 300)

#go to the working directory and delete some images

#filter selection data frame

#this data frame does not have the selections corresponding to the images that were deleted
fmloc

#now using lspec images
lspec(sxrow = 2, rows = 8, pal = reverse.heat.colors, wl = 300, ovlp = 10)

#go to the working directory and delete lspec images (the ones with several rows of spectrograms)

#filter selection data frame


## End(Not run)
```

`fixwavs` *Fix .wav files to allow importing them into R*

Description

`fixwavs` fixes sound files in .wav format so they can be imported into R.

Usage

```
fixwavs(checksels = NULL, files = NULL, samp.rate = NULL, bit.rate = NULL,
 path = NULL, ...)
```

Arguments

`checksels`	Data frame with results from `checksels`.
`files`	Character vector with the names of the wav files to fix. Default is `NULL`. Default is `NULL`.
`samp.rate`	Numeric vector of length 1 with the sampling rate (in khz) for output files. Default is `NULL`.
`bit.rate`	Numeric vector of length 1 with the dynamic interval (i.e. bit rate) for output files. Default is `NULL`.
`path`	Character string containing the directory path where the sound files are located. If `NULL` (default) then the current working directory is used.
`...`	Additional arguments to be passed to `sox`.

Details

This function aims to simplify the process of converting sound files that cannot be imported into R to a format that can actually be imported. Problematic files can be determined using checksels. The checksels output can be directly input using the argument 'checksels'. Alternatively a vector of file names to be "fixed" can be provided (argument 'files'). Internally the function calls 'sox' through the sox function. 'sox' must be installed to be able to run this function.

Value

A folder inside the working directory (or path provided) all 'converted sound files', containing sound files in a format that can be imported in R.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) #last modification on march-15-2017 (MAS)

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

fixwavs(files = selec.table$sound.files)

#check this folder
getwd()

## End(Not run)
```

---

imp.raven *Import Raven selections*

Description

imp.raven imports Raven selection data from many files simultaneously. Files must be in .txt format.

Usage

```
imp.raven(path = NULL, sound.file.col = NULL, all.data = FALSE, recursive = FALSE,
 name.from.file = FALSE, ext.case = NULL, freq.cols = TRUE)
```

Arguments

path	A character string indicating the path of the directory in which to look for the text files. If not provided (default) the function searches into the current working directory. Default is NULL).
sound.file.col	A character string with the name of the column listing the sound files in the selection text files. Default is NULL). If provided, the output data frame will contained all columns needed for subsequent analysis in warbleR. Duplicated rows, as when "waveform" and "spectrogram" information are included for the same selection, will be removed. All selection files must contain "Selection", "Begin.Time" and "End.Time" columns.
all.data	Logical. If TRUE) all columns in text files are returned, keeping the name columns as in the raven files (not in "warbleR" format). Default is FALSE). Columns absent in some selection files will be filled with NA's.
recursive	Logical. If TRUE) the listing recurse into sub-directories.
name.from.file	Logical. If TRUE) the sound file names are extracted from the selection text file name. It asssumes that selections files contained the suffix "Table.1.selections.txt" or "selections.txt". Note that by default it will assume that the extension file name is ".wav". This can be control using the argumet 'ext.wav'. Default is FALSE). Ignored if sound.file.col' is provided and/or all.data is TRUE).
ext.case	Character string of length 1 to specify whether sound file extensions are in upper or lower case. This should match the extension of the of the .wav files from which the selection were made. It must be either 'upper' or 'lower'. Only needed when 'name.from.file' is TRUE). Ignored if 'sound.file.col' is provided and/or all.data is TRUE).
freq.cols	Logical. If TRUE) 'Low Freq' and 'High Freq' columns are also imported. Ignored if all.data is TRUE.

Details

The function import raven selection data from many files simultaneously. Files must be in .txt format. Note that selection files including data from mulitple recordings cannot be imported. Make sure that NO OTHER TEXT FILES are found in the working directory, only raven generated selections files.

Value

A single data frame with information of the selection files. If all.data argument is set to FALSE) the data frame contains the following columns: selec, start, end, and selec.file. If sound.file.col is provided the data frame will also contain a 'sound.files' column. In addition, all rows with duplicated data are removed. This is useful when both spectrogram and waveform views are included in the Raven selection files. If all.data is set to TRUE then all columns in selection files are returned.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

imp.syrinx

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

data(selection.files)

write.table(selection.files[[1]],file = "100889-Garrulax monileger.selections.txt",
row.names = FALSE, sep= '\t')

write.table(selection.files[[2]],file = "1023-Arremonops rufivirgatus.selections.txt",
row.names = FALSE, sep= '\t')

## MAKE SURE THERE ARE NO OTHER .txt FILES IN THE WORKING DIRECTORY
 #providing the name of the column with the sound file names
rav.dat<-imp.raven(sound.file.col = "End.File", all.data = FALSE)

View(rav.dat)

#getting all the data
rav.dat<-imp.raven(all.data = TRUE)
View(rav.dat)

## End(Not run)
```

---

imp.syrinx *Import Syrinx selections*

Description

imp.syrinx imports Syrinx selection data from many files simultaneously. All files must be have the same columns.

Usage

```
imp.syrinx(path = NULL, all.data = FALSE, recursive = FALSE, exclude = FALSE)
```

Arguments

path	A character string indicating the path of the directory in which to look for the text files. If not provided (default) the function searches into the current working directory. Default is NULL).
all.data	Logical. If TRUE) all columns in text files are returned. Default is FALSE). Note that all files should contain exactly the same columns in the same order.

recursive	Logical. If TRUE) the listing recurse into sub-directories.
exclude	Logical. Controls whether files that cannot be read are ignored (TRUE). Default is FALSE.

Value

A single data frame with information of the selection files. If all.data argument is set to FALSE) the data frame contains the following columns: selec, start, end, and selec.file. If sound.file.col is provided the data frame will also contain a 'sound.files' column. In addition, all rows with duplicated data are removed. This is useful when both spectrogram and waveform views are included in the Syrinx selection files. If all.data is set to TRUE then all columns in selection files are returned.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

imp.raven

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

#load data
data(selection.files)

write.table(selection.files[[3]],file = "harpyeagle.wav.txt",row.names = FALSE,
 col.names = FALSE, sep= "\t")

write.table(selection.files[[4]],file = "Phae.long4.wav.txt",row.names = FALSE,
col.names = FALSE, sep= "\t")

syr.dat<-imp.syrinx(all.data = FALSE)

View(syr.dat)

#getting all the data
syr.dat<-imp.syrinx(all.data = TRUE)

View(syr.dat)

## End(Not run)
```

lspec *Create long spectrograms of whole sound files*

Description

lspec produces image files with spectrograms of whole sound files split into multiple rows.

Usage

```
lspec(X = NULL, flim = c(0,22), sxrow = 5, rows = 10, collev = seq(-40, 0, 1),
ovlp = 50, parallel = 1, wl = 512, gr = FALSE, pal = reverse.gray.colors.2,
cex = 1, it = "jpeg", flist = NULL, redo = TRUE, path = NULL, pb = TRUE,
fast.spec = FALSE)
```

Arguments

X	Data frame with results from manualoc or any data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). If given, two red dotted lines are plotted at the start and end of a selection and the selections are labeled with the selection number (and selection comment, if available). Default is NULL.
flim	A numeric vector of length 2 indicating the highest and lowest frequency limits (kHz) of the spectrogram, as in spectro. Default is c(0,22).
sxrow	A numeric vector of length 1. Specifies seconds of spectrogram per row. Default is 5.
rows	A numeric vector of length 1. Specifies number of rows per image file. Default is 10.
collev	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).
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 50. High values of ovlp slow down the function but produce more accurate selection limits (when X is provided).
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
gr	Logical argument to add grid to spectrogram. Default is FALSE.
pal	Color palette function for spectrogram. Default is reverse.gray.colors.2. See spectro for more palettes.
cex	A numeric vector of length 1 giving the amount by which text (including sound file and page number) should be magnified. Default is 1.
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".

flist	character vector or factor indicating the subset of files that will be analyzed. Ignored if X is provided.
redo	Logical argument. If TRUE all selections will be analyzed again when code is rerun. If FALSE only the selections that do not have a image file in the working directory will be analyzed. Default is FALSE.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.

Details

The function creates spectrograms for complete sound files, printing the name of the sound files and the "page" number (p1-p2...) at the upper right corner of the image files. If results from manualoc are supplied (or an equivalent data frame), the function delimits and labels the selections. This function aims to facilitate visual inspection of multiple files as well as visual classification of vocalization units and the analysis of animal vocal sequences.

Value

image files with spectrograms of whole sound files in the working directory. Multiple pages can be returned, depending on the length of each sound file.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())
# save sound file examples
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")

lspec(sxrow = 2, rows = 8, pal = reverse.heat.colors, wl = 300)

# including selections
lspec(sxrow = 2, rows = 8, X = selec.table, pal = reverse.heat.colors, redo = TRUE, wl = 300)
```

```
check this floder
getwd()

## End(Not run)
```

---

lspec2pdf lspec2pdf *combines* lspec *images in .jpeg format to a single pdf file.*

Description

lspec2pdf combines lspec images in .jpeg format to a single pdf file.

Usage

```
lspec2pdf(keep.jpeg = TRUE, overwrite = FALSE, parallel = 1, path = NULL, pb = TRUE)
```

Arguments

keep.jpeg	Logical argument. Indicates whether jpeg files should be kept (default) or remove. (including sound file and page number) should be magnified. Default is 1.
overwrite	Logical argument. If TRUE all jpeg pdf will be produced again when code is rerun. If FALSE only the ones missing will be produced. Default is FALSE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

Details

The function combines spectrograms for complete sound files from the lspec function into a single pdf (for each sound file).

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

lspec, catalog2pdf, https://marce10.github.io/2017-01-07-Create_pdf_files_with_spectrograms_of_full_recordings/

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

# save sound file examples
data(list = c("Phae.long1", "Phae.long2"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")

lspec(sxrow = 2, rows = 8, pal = reverse.heat.colors, wl = 300, it = "jpeg")

#now create single pdf removing jpeg
lspec2pdf(keep.jpeg = FALSE)

check this floder
getwd()

## End(Not run)
```

---

manualoc *Interactive view of spectrograms*

Description

manualoc produces an interactive spectrographic view in which the start and end times of acoustic signals can be measured.

Usage

```
manualoc(wl = 512, flim = c(0,12), seltime = 1, tdisp = NULL, reccomm =
  FALSE, wn = "hanning", title = TRUE, selcomm = FALSE, osci = FALSE, player =
  NULL, pal = reverse.gray.colors.2, path = NULL, flist = NULL, fast.spec = FALSE)
```

Arguments

wl	A numeric vector of length 1 specifying the spectrogram window length. Default is 512.
flim	A numeric vector of length 2 specifying the frequency limit (in kHz) of the spectrogram, as in the function spectro. Default is c(0,12).
seltime	A numeric vector of length 1 indicating the time interval in seconds at which the spectrograms are produced with higher resolution (ovlp = 70) and oscilograms (if osci = TRUE). Default is 1 second.
tdisp	A numeric vector of length 1 specifying the length in seconds of the total sound file to be displayed. Default is NULL which displays the full sound file.
reccomm	Logical argument. If TRUE pops up a comment window at the end of each sound file. The comment needs to be quoted. Default is FALSE.

wn	A character vector of length 1 specifying the window function (by default "hanning"). See function ftwindow for more options.
title	Logical argument. If TRUE the name of the sound file will be printed as the main title of the spectrogram window. Default is TRUE
selcomm	Logical argument. If TRUE pops up a comment window after each selection. The comment is printed as a label on the selected unit. The comment must be quoted. Default is FALSE
osci	Logical argument. If TRUE adds a oscillogram whenever the spectrograms are produced with higher resolution (see seltime). Default is FALSE.
player	Path to or name of a program capable of playing a wave file by invocation from the command line. If under Windows and no player is given, windows player will be chosen as the default. "vlc" works in Linux if vlc player is installed. The external program must be closed before resuming analysis. Default is NULL.
pal	A color palette function to be used to assign colors in the plot, as in spectro. Default is reverse.gray.colors.2. See Details.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
flist	character vector or factor indicating the subset of files that will be analyzed. Ignored if X is provided.
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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.

Details

Users can zoom-in a specific sound file segment by clicking at the start and end (left side and right side) of the segment. To select the start and end of a vocalization unit the users need to click at the end and then at the start (right side and left side) of the unit. In addition, 6 "buttons" are provided at the upper right side of the spectrogram that allow to display a full view of the spectrogram ("Full view"), go back to the previous view ("Previous view"), stop the analysis ("Stop"), go to the next sound file ("Next rec"), play the current view using external software ("Play", see "player" argument), or delete the last manual selection in the current sound file ("Del-sel"). When a unit has been selected, the function plots a red circle with the selection number in the middle point of the selection in the spectrogram. It also plots vertical dotted lines at the start and end of the selection. The circle and lines "disappear" when the selection is deleted ("Del-sel" button). Only the last selection can be deleted.

The function produces a .csv file (manualoc_output.csv) with information about the .wav file name, selection number, start and end time, selection comment (selcomm), and sound file comment (reccomm). The file is saved in the working directory and is updated every time the user moves into the next sound file (Next rec "button") or stop the process (Stop "button"). When resuming the process (after "stop" and re-running the function in the same working directory), the function will

keep the previous selections and will only pick up .wav files that are not present in the .csv file (not previously analyzed). When users go to the next sound file (Next rec "button") without making any selection the file is still included in the .csv file, with NA's in the "end", "time" and "selec" field.

Windows length (wl) controls the temporal and frequency precision of the spectrogram. A high "wl" value increases the frequency resolution but reduces the temporal resolution, and vice versa. Any color palette that comes with the seewave package can be used: temp.colors, reverse.gray.colors.1, reverse.gray.colors.2, reverse.heat.colors, reverse.terrain.colors, reverse.topo.colors, reverse.cm.colors, heat.colors, terrain.colors, topo.colors, cm.colors. The function is slow when working on files of length > 5min. In most cases other sound analysis softwares for manually selecting acoustic signals (e.g. Raven, Syrinx) should be prefered.

Value

.csv file saved in the working directory with start and end time of selections.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

seltailor

Examples

```
## Not run:
#Set temporary working directory
setwd(tempdir())

# save wav file examples
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

manualoc(wl = 300)
# need to use the buttoms to manipulate function
# check working directory for .csv file after stopping function
#check here:
getwd()

## End(Not run)
```

mp32wav *Convert .mp3 files to .wav*

Description

mp32wav converts several .mp3 files in working directory to .wav format

Usage

```
mp32wav(samp.rate = 44.1, parallel = 1, from.path = NULL, to.path = NULL,
normalize = NULL, pb = TRUE)
```

Arguments

samp.rate	Sampling rate at which the .wav files should be written. The maximum permitted is 44.1 kHz (default). Units should be kHz.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not availble in Windows OS.
from.path	Character string containing the directory path where the .mp3 files are located. If NULL (default) then the current working directory is used.
to.path	Character string containing the directory path where the .wav files will be saved. If NULL (default) then the current working directory is used.
normalize	Character string containing the units to be used for amplitude normalization. Check (normalize) for details. If NULL (default) no normalization is carried out.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

Details

convert all .mp3 files in working directory to .wav format. Function used internally to read .mp3 files (readMP3) sometimes crashes.

Value

.wav files saved in the working directory with same name as original mp3 files.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

#Then download mp3 files from xeno-canto
querxc(qword = "Phaethornis aethopygus", download = TRUE)

# Convert all files to .wav format
mp32wav()

#check this folder!!
getwd()

## End(Not run)
```

Phae.long1 *Audio recording #1 of* Phaethornis longirostris

Description

Audio recording #1 of *Phaethornis longirostris* (Long-billed Hermit).

Usage

```
data(Phae.long1)
```

Format

One .wav file:

Phae.long1 *Phaethornis longirostris* #1 recording

Source

http://www.xeno-canto.org/contributor/EMCWQLLKEW

Phae.long2 *Audio recording #2 of* Phaethornis longirostris

Description

Audio recording #2 of *Phaethornis longirostris* (Long-billed Hermit).

Usage

```
data(Phae.long2)
```

Format

One .wav file:

Phae.long2 *Phaethornis longirostris* #2 recording

Source

http://www.xeno-canto.org/contributor/EMCWQLLKEW

Phae.long3 *Audio recording #3 of* Phaethornis longirostris

Description

Audio recording #3 of *Phaethornis longirostris* (Long-billed Hermit).

Usage

```
data(Phae.long3)
```

Format

One .wav file:

Phae.long3 *Phaethornis longirostris* #3 recording

Source

http://www.xeno-canto.org/contributor/EMCWQLLKEW

Phae.long4 *Audio recording #1 of* Phaethornis longirostris

Description

Audio recording #4 of *Phaethornis longirostris* (Long-billed Hermit).

Usage

```
data(Phae.long4)
```

Format

One .wav file:

Phae.long4 *Phaethornis longirostris* #4 recording

Source

http://www.xeno-canto.org/contributor/EMCWQLLKEW

querxc *Access Xeno-Canto recordings and metadata*

Description

querxc downloads recordings and metadata from Xeno-Canto (http://www.xeno-canto.org/).

Usage

```
querxc(qword, download = FALSE, X = NULL, file.name = c("Genus", "Specific_epithet"),
parallel = 1, path = NULL, pb = TRUE)
```

Arguments

qword	Character vector of length one indicating the genus, or genus and species, to query Xeno-Canto database. For example, *Phaethornis* or *Phaethornis longirostris*. (http://www.xeno-canto.org/). More complex queries can be done by using search terms that follow the xeno-canto advance query syntax.This syntax uses tags to search within a particular aspect of the recordings (e.g. country, location, sound type). Tags are of the form tag:searchterm'. For instance, 'type:song' will search for all recordings in which the sound type description contains the word 'song'. Several tags can be included in the same query. The query "phaethornis cnt:belize' will only return results for birds in the genus *Phaethornis* that were recorded in Belize. See http://www.xeno-canto.org/help/search for a full description and see examples below for queries using terms with more than one word.
download	Logical argument. If FALSE only the recording file names and associated metadata are downloaded. If TRUE, recordings are also downloaded to the working directory as .mp3 files. Default is FALSE. Note that if the recording is already in the working directory (as when the downloading process has been interrupted) it will be skipped. Hence, resuming downloading processes will not start from scratch.
X	Data frame with a 'Recording_ID' column and any other column listed in the file.name argument. Only the recordings listed in the data frame will be download (download argument is automatically set to TRUE). This can be used to select the recordings to be downloaded based on their attributes.
file.name	Character vector indicating the tags (or column names) to be included in the sound file names (if download = TRUE). Several tags can be included. If NULL only the Xeno-Canto recording identification number ("Recording_ID") is used. Default is c("Genus", "Specific_epithet"). Note that recording id is always used (whether or not is listed by users) to avoid duplicated names.
parallel	Numeric. Controls whether parallel computing is applied when downloading mp3 files. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Currently only applied when downloading files. Might not work improve performance on Windows OS.

path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

Details

This function queries for avian vocalization recordings in the open-access online repository Xeno-Canto (http://www.xeno-canto.org/). It can return recordings metadata or download the associated sound files. Complex queries can be done by using search terms that follow the xeno-canto advance query syntax (check "qword" argument description). Files are double-checked after downloading and "empty" files are re-downloaded. File downloading process can be interrupted and resume later as long as the working directory is the same. Maps of recording coordinates can be produced using xcmaps.

Value

If X is not provided the function returns a data frame with the following recording information: recording ID, Genus, Specific epithet, Subspecies, English name, Recordist, Country, Locality, Latitude, Longitude, Vocalization type, Audio file, License, URL, Quality, Time, Date. Sound files in .mp3 format are downloaded into the working directory if download = TRUE or if X is provided; a column indicating the names of the downloaded files is included in the output data frame.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

xcmaps, https://marce10.github.io/2016-12-22-Download_a_single_recording_for_each_species_in_a_site_from_Xeno-Canto/

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

# search without downloading
df1 <- querxc(qword = 'Phaethornis anthophilus', download = FALSE)
View(df1)

# downloading files
querxc(qword = 'Phaethornis anthophilus', download = TRUE)

# check this folder
getwd()

## search using xeno-canto advance query ###
orth.pap <- querxc(qword = 'gen:orthonyx cnt:papua loc:tari', download = FALSE)
```

```
# download file using the output data frame as input
querxc(X = orth.pap)

# use quotes for queries with more than 1 word (e.g. Costa Rica),note that the
# single quotes are used for the whole 'qword' and double quotes for the 2-word term inside
#Phaeochroa genus in Costa Rica
phae.cr <- querxc(qword = 'gen:phaeochroa cnt:"costa rica"', download = FALSE)

# several terms can be searched for in the same field
# search for all female songs in sound type
femsong <- querxc(qword = 'type:song type:female', download = FALSE)

## End(Not run)
```

`selec.table` *Data frame of selections (i.e. selection table).*

Description

A data frame containing the start, end, low and hig frequency of *Phaethornis longirostris* (Long-billed Hermit) songs from the example sound files included in this package.

Usage

```
data(selec.table)
```

Format

A data frame with 11 rows and 6 variables:

sound.files recording names

selec selection numbers within recording

start start times of selected signal

end end times of selected signal

low.f lower limit of frequency range

high.f upper limit of frequency range

sel.comment selection comments

rec.comment recording comments

Source

Marcelo Araya Salas, warbleR

selection.files *Selections files from Raven and Syrinx.*

Description

Selections files from Raven and Syrinx.

Usage

```
data(selection.files)
```

Format

selection.files Selections from the commercial software 'Raven' and 'Syrinx'

seltailor *Interactive view of spectrograms to tailor start and end of selections*

Description

`seltailor` produces an interactive spectrographic view (similar to `manualoc`) in which the start and end times of acoustic signals listed in a data frame can be adjusted.

Usage

```
seltailor(X = NULL, wl = 512, flim = c(0,22), wn = "hanning", mar = 0.5,
 osci = TRUE, pal = reverse.gray.colors.2, ovlp = 70, auto.next = FALSE, pause = 1,
  comments = TRUE, path = NULL, frange = FALSE, fast.spec = FALSE, ext.window = TRUE,
   width = 15, height = 5, ...)
```

Arguments

`X`	data frame with the following columns: 1) "sound.files": name of the .wav files, 2) "selec": number of the selections, 3) "start": start time of selections, 4) "end": end time of selections. The ouptut of `manualoc` or `autodetec` can be used as the input data frame. Other data frames can be used as input, but must have at least the 4 columns mentioned above. Required. Notice that, if an output file ("seltailor_output.csv") is found in the working directory it will be given priority over an input data frame.
`wl`	A numeric vector of length 1 specifying the spectrogram window length. Default is 512.
`flim`	A numeric vector of length 2 specifying the frequency limit (in kHz) of the spectrogram, as in the function `spectro`. Default is c(0,22).
`wn`	A character vector of length 1 specifying the window function (by default "hanning"). See function `ftwindow` for more options.

mar	Numeric vector of length 1. Specifies the margins adjacent to the start and end points of the selections to define spectrogram limits. Default is 0.5.
osci	Logical argument. If TRUE adds a oscillogram whenever the spectrograms are produced with higher resolution (see seltime). Default is TRUE. The external program must be closed before resuming analysis. Default is NULL.
pal	A color palette function to be used to assign colors in the plot, as in spectro. Default is reverse.gray.colors.2. See Details.
ovlp	Numeric vector of length 1 specifying the percent overlap between two consecutive windows, as in spectro. Default is 70.
auto.next	Logical argument to control whether the functions moves automatically to the next selection. The time interval before moving to the next selection is controled by the 'pause' argument.
pause	Numeric vector of length 1. Controls the duration of the waiting period before moving to the next selection (in seconds). Default is 1.
comments	Logical argument specifying if 'sel.comment' (when in data frame) should be included in the title of the spectrograms. Default is TRUE.
path	Character string containing the directory path where the sound files are located.
frange	Logical argument specifying whether limits on frequency range should be recorded. If NULL (default) then only the time limits are recorded.
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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.
ext.window	Logical. If TRUE then and external graphic window is used. Default dimensions can be set using the 'width' and 'height' arguments. Default is TRUE.
width	Numeric of length 1 controling the width of the external graphic window. Ignored if ext.window = FALSE. Default is 15.
height	Numeric of length 1 controling the height of the external graphic window. Ignored if ext.window = FALSE. Default is 5.
...	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 produces an interactive spectrographic view (similar to manualoc) in which users can select a new start and end of a vocalization unit (e.g. elements) by clicking at the end and at the start of the signal (in any order). In addition, 2 "buttons" are provided at the upper right side of the spectrogram that allow to stop the analysis ("Stop") or go to the next sound file ("next sel"). When a unit has been selected, the function plots red dotted lines in the start and end of the selection in the spectrogram. The lines "disappear" when a new selections is made. Only the last selection is

kept for each selection that is adjusted. The function produces a .csv file (seltailor_output.csv) with the same information than the input data frame, except for the new time coordinates, plus a new column (X$tailored) indicating if the selection has been tailored. The file is saved in the working directory and is updated every time the user moves into the next sound file (next sel "button") or stop the process (Stop "button"). If no selection (by clicking on the 'next' buttom) the original time coordinates are kept. When resuming the process (after "stop" and re-running the function in the same working directory), the function will continue working on the selections that have not been analyzed.

Windows length (wl) controls the temporal and frequency precision of the spectrogram. A high "wl" value increases the frequency resolution but reduces the temporal resolution, and vice versa. Any color palette that comes with the seewave package can be used: temp.colors, reverse.gray.colors.1, reverse.gray.colors.2, reverse.heat.colors, reverse.terrain.colors, reverse.topo.colors, reverse.cm.colors, heat.colors, terrain.colors, topo.colors, cm.colors. Note that, unlike `manualoc`, you cannot zoom in the spectrogram `seltailor`. The zoom can be adjusted by setting the `mar` argument.

Value

.csv file saved in the working directory with start and end time of selections.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

`manualoc`

Examples

```
## Not run:
#Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

seltailor(X =  selec.table, flim = c(1,12), wl = 300, auto.next = TRUE)

# Read output .csv file
seltailor.df <- read.csv("seltailor_output.csv")
seltailor.df

# check this directory for .csv file after stopping function
getwd()

## End(Not run)
```

sig2noise *Measure signal-to-noise ratio*

Description

sig2noise measures signal-to-noise ratio across multiple files.

Usage

```
sig2noise(X, mar, parallel = 1, path = NULL, pb = TRUE, type = 1, eq.dur = FALSE,
in.dB = TRUE, before = FALSE, lim.dB = TRUE)
```

Arguments

X	Data frame with results from manualoc or any data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end).
mar	numeric vector of length 1. Specifies the margins adjacent to the start and end points of selection over which to measure noise.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
type	Numeric. Determine the formula to be used to calculate the signal-to-noise ratio (S = signal , N = background noise): • 1: ratio of S mean amplitude envelope to N mean amplitude envelope (mean(env(S))/mean(env(N))) • 2: ratio of S amplitude envelope quadratic mean to N amplitude envelope quadratic mean (rms(env(S))/rms(env(N))) • 3: ratio of the difference between S amplitude envelope quadratic mean and N amplitude envelope quadratic mean to N amplitude envelope quadratic mean ((rms(env(S)) - rms(env(N)))/rms(env(N)))
eq.dur	Logical. Controls whether the noise segment that is measured has the same duration than the signal (if TRUE, default FALSE). If TRUE then mar argument is ignored.
in.dB	Logical. Controls whether the signal-to-noise ratio is returned in decibels (20*log10(SNR)). Default is TRUE.
before	Logical. If TRUE noise is only measured right before the signal (instead of before and after). Default is FALSE.
lim.dB	Logical. If TRUE the lowest signal-to-noise would be limited to -40 dB (if in.dB = TRUE). This would remove NA's that can be produced when noise segments have a higher amplitude than the signal itself. Default is TRUE.

Details

Signal-to-noise ratio (SNR) is a measure of the level of a desired signal compared to background noise. The function divides the mean amplitude of the signal by the mean amplitude of the background noise adjacent to the signal. A general margin to apply before and after the acoustic signal must be specified. Setting margins for individual signals that have been previously clipped from larger files may take some optimization, as for calls within a larger file that are irregularly separated. When margins overlap with another acoustic signal nearby, the signal-to-noise ratio (SNR) will be inaccurate. Any SNR less than or equal to one suggests background noise is equal to or overpowering the acoustic signal. snrspecs can be used to troubleshoot different noise margins.

Value

Data frame similar to autodetec output, but also includes a new variable with the signal-to-noise values.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

Source

https://en.wikipedia.org/wiki/Signal-to-noise_ratio

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

data(list = c("Phae.long1","selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files

# specifying the correct margin is important
# use snrspecs to troubleshoot margins for sound files
sig2noise(selec.table[grep("Phae.long1", selec.table$sound.files), ], mar = 0.2)

# this smaller margin doesn't overlap neighboring signals
sig2noise(selec.table[grep("Phae.long1", selec.table$sound.files), ], mar = 0.1)

## End(Not run)
```

---

sim.coor.sing *Simulated coordinated singing events.*

Description

sim.coor.sing Selections of simulated interactive singing events.The simulated events use the mean and standard deviation of real lekking *Phaethornis longirostris* (Long-billed Hermit hummingbird) songs and intervals between songs (e.i gaps). Three events are simulated: overlapping signals (ovlp), alternating signals (altern) and non-synchronized signals (uncoor).

Usage

```
data(sim.coor.sing)
```

Format

sim.coor.sing Simulated coordinated singing events that overlap and do not overlap most of the time, for use with coor.test

---

snrspecs *Spectrograms with background noise margins*

Description

snrspecs creates spectrograms to visualize margins over which background noise will be measured by sig2noise.

Usage

```
snrspecs(X, wl = 512, flim = c(0, 22), wn = "hanning", ovlp = 70,
inner.mar = c(5, 4, 4, 2), outer.mar = c(0, 0, 0, 0), picsize = 1,
res = 100, cexlab = 1, title = TRUE,
  propwidth= FALSE, xl=1, osci = FALSE, gr = FALSE, sc = FALSE, mar = 0.2,
   snrmar = 0.1, it = "jpeg", parallel = 1, path = NULL, pb = TRUE)
```

Arguments

X	Data frame with results from manualoc or any data frame with columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end).
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
flim	A numeric vector of length 2 for the frequency limit in kHz of the spectrogram, as in spectro. Default is c(0, 22).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.

inner.mar	Numeric vector with 4 elements, default is c(5,4,4,2). Specifies number of lines in inner plot margins where axis labels fall, with form c(bottom, left, top, right). See par.
outer.mar	Numeric vector with 4 elements, default is c(0,0,0,0). Specifies number of lines in outer plot margins beyond axis labels, with form c(bottom, left, top, right). See par.
picsize	Numeric argument of length 1, controls relative size of spectrogram. Default is 1.
res	Numeric argument of length 1 that controls image resolution. Default is 100 (faster) although 300 - 400 is recommended for publication/ presentation quality.
cexlab	Numeric vector of length 1 specifying relative size of axis labels. See spectro.
title	Logical argument to add a title to individual spectrograms. Default is TRUE.
propwidth	Logical argument to scale the width of spectrogram proportionally to duration of the selected call. Default is FALSE.
xl	Numeric vector of length 1, a constant by which to scale spectrogram width if propwidth = TRUE. Default is 1.
osci	Logical argument to add an oscillogram underneath spectrogram, as in spectro. Default is FALSE.
gr	Logical argument to add grid to spectrogram. Default is FALSE.
sc	Logical argument to add amplitude scale to spectrogram, default is FALSE.
mar	Numeric vector of length 1. Specifies the margins adjacent to the start and end points of the selections to define spectrogram limits. Default is 0.2. If snrmar is larger than mar, then mar is set to be equal to snrmar.
snrmar	Numeric vector of length 1. Specifies the margins adjacent to the start and end points of the selections where noise will be measured. Default is 0.1.
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

Details

This function can be used to test different margins to facilitate accurate SNR measurements when using sig2noise down the line. Setting margins for individual calls that have been previously clipped from larger files may take some optimization, as for calls within a larger file that are irregularly separated. Setting inner.mar to c(4,4.5,2,1) and outer.mar to c(4,2,2,1) works well when picsize = 2 or 3. Title font size, inner.mar and outer.mar (from mar and oma in par) don't work well when osci or sc = TRUE, this may take some optimization by the user.

Value

Spectrograms per selection marked with margins where background noise will be measured.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

Source

https://en.wikipedia.org/wiki/Signal-to-noise_ratio

See Also

trackfreqs for creating spectrograms to visualize frequency measurements by specan, specreator for creating spectrograms after using manualoc

Other spectrogram.creators: dfDTW, dfts, ffDTW, ffts, sp.en.ts, specreator, trackfreqs

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound.files
writeWave(Phae.long2, "Phae.long2.wav")

# make Phae.long1 and Phae.long2 spectrograms
# snrmar needs to be smaller before moving on to sig2noise()

snrspecs(selec.table, flim = c(0, 14), inner.mar = c(4,4.5,2,1), outer.mar = c(4,2,2,1),
picsize = 2, res = 300, cexlab = 2, mar = 0.2, snrmar = 0.1, it = "jpeg", wl = 300)

# make only Phae.long1 spectrograms
# snrmar now doesn't overlap neighboring signals

snrspecs(selec.table[grepl(c("Phae.long1"), selec.table$sound.files), ], flim = c(3, 14),
inner.mar = c(4,4.5,2,1), outer.mar = c(4,2,2,1), picsize = 2, res = 300, cexlab = 2,
mar = 0.2, snrmar = 0.01, wl = 300)

#check this folder!!
getwd()

## End(Not run)
```

sp.en.ts *Extract the spectral entropy across signals as a time series*

Description

sp.en.ts spectral entropy across signals as a time series. of signals selected by manualoc or autodetec.

Usage

```
sp.en.ts(X, wl = 512, length.out = 20, wn = "hanning", ovlp = 70, bp = NULL,
  threshold = 15, img = TRUE, parallel = 1, path = NULL, img.suffix = "sp.en.ts",
   pb = TRUE, clip.edges = FALSE, leglab = "sp.en.ts", sp.en.range = c(2, 10), ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512. Note that this is particularly important for measuring spectral entropy. Low values (~100) generate a very detail contour of the variation in spectral entropy that is probably not useful for assesing signal similarity.
length.out	A character vector of length 1 giving the number of measurements of spectral entropy desired (the length of the time series).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz). Default is NULL.
threshold	amplitude threshold (%) for dominant frequency detection. Default is 15.
img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located.
img.suffix	A character vector of length 1 with a sufix (label) to add at the end of the names of image files.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.

clip.edges	Logical argument to control whether edges (start or end of signal) in which amplitude values above the threshold were not detected will be removed. If TRUE this edges will be excluded and signal contour will be calculated on the remainging values. Default is FALSE.
leglab	A character vector of length 1 or 2 containing the label(s) of the frequency contour legend in the output image.
sp.en.range	Numeric vector of length 2. Range of frequency in which to display the entropy values on the spectrogram (when img = TRUE). Default is c(2, 10). Negative values can be used in order to stretch more the range.
...	Additional arguments to be passed to trackfreqs for customizing graphical output.

Details

This function spectral entropy across signals as a time series. The function uses the approx function to interpolate values between spectral entropy measures (calculated with csh). If there are no frequencies above the amplitude theshold at the begining or end of the signals then NAs will be generated. On the other hand, if there are no frequenciesabove the amplitude theshold in between signal segments in which amplitude was detected then the values of this adjacent segments will be interpolated to fill out the missing values (e.g. no NAs in between detected amplitude segments). Missing values at the start of end can be removed with "clip.edges".

Value

A data frame with the dominant frequency values measured across the signals. If img is TRUE it also produces image files with the spectrograms of the signals listed in the input data frame showing the location of the dominant frequencies (see trackfreqs description for more details).

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

specreator for creating spectrograms from selections, snrspecs for creating spectrograms to optimize noise margins used in sig2noise

Other spectrogram.creators: dfDTW, dfts, ffDTW, ffts, snrspecs, specreator, trackfreqs

Examples

```
## Not run:
# set the temp directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2",  "Phae.long3",  "Phae.long4","selec.table"))
writeWave(Phae.long2, "Phae.long2.wav") #save sound files
writeWave(Phae.long1, "Phae.long1.wav")
writeWave(Phae.long3, "Phae.long3.wav") #save sound files
```

```
writeWave(Phae.long4, "Phae.long4.wav")

# without clip edges
sp.en.ts(X = selec.table, threshold = 10, bp = NULL, clip.edges = F, length.out = 10,
 type = "b", sp.en.range = c(-25, 10))

# with clip edges and length.out 10
sp.en.ts(X = selec.table, threshold = 10, bp = c(2, 12), clip.edges = T, length.out = 10)


## End(Not run)
```

---

specan *Measure acoustic parameters in batches of sound files*

Description

specan measures acoustic parameters on acoustic signals for which the start and end times are provided.

Usage

```
specan(X, bp = c(0,22), wl = 512, threshold = 15, parallel = 1, fast = TRUE, path = NULL,
pb = TRUE, ovlp = 50, ff.method = "seewave", wn = "hanning")
```

Arguments

X	Data frame with the following columns: 1) "sound.files": name of the .wav files, 2) "sel": number of the selections, 3) "start": start time of selections, 4) "end": end time of selections. The ouptut of manualoc or autodetec can be used as the input data frame.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz) or "frange" to indicate that values in low.f and high.f columns will be used as bandpass limits. Default is c(0, 22). Lower limit of bandpass is not applied to fundamental frequencies.
wl	A numeric vector of length 1 specifying the spectrogram window length. Default is 512.
threshold	amplitude threshold (%) for fundamental frequency and dominant frequency detection. Default is 15.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). For windows OS the parallelsugar package should be installed.
fast	Logical. If TRUE (default) then the peakf acoustic parameter (see below) is not computed, which substantially increases performance (~9 times faster).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.

pb	Logical argument to control progress bar and messages. Default is TRUE. Note that progress bar is only used when parallel = 1.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, used for fundamental frequency (using fund or FF) and dominant frequency (using dfreq). Default is 50.
ff.method	Character. Selects the method used to calculate the fundamental frequency. Either 'tuneR' (using FF) or 'seewave' (using fund). Default is 'seewave'. Use trackfreqs to decide which method works the best. 'tuneR' performs faster (and seems to be more accurate) than 'seewave'.
wn	Character vector of length 1 specifying window name. Default is hanning'. See function ftwindow for more options.

Details

The ouptut of manualoc or autodetec can be used directly without any additional modification. The function measures 29 acoustic parameters (if fast = TRUE) on each selection in the data frame. Most parameters are produced internally by specprop, fpeaks, fund, and dfreq from the package seewave and FF from the package tuneR. NAs are produced for fundamental and dominant frequency measures when there are no amplitude values above the threshold.

Value

Data frame with 'sound.files' and 'selec' as in the input data frame, plus the following acoustic parameters:

- duration: length of signal (in s)
- meanfreq: mean frequency. Weighted average of frequency by amplitude (in kHz)
- sd: standard deviation of frequency weighted by amplitude
- freq.median: median frequency. The frequency at which the signal is divided in two frequency intervals of equal energy (in kHz)
- freq.Q25: first quartile frequency. The frequency at which the signal is divided in two frequency intervals of 25% and 75% energy respectively (in kHz)
- freq.Q75: third quartile frequency. The frequency at which the signal is divided in two frequency intervals of 75% and 25% energy respectively (in kHz)
- freq.IQR: interquartile frequency range. Frequency range between 'freq.Q25' and 'freq.Q75' (in kHz)
- time.median: median time. The time at which the signal is divided in two time intervals of equal energy (in s)
- time.Q25: first quartile time. The time at which the signal is divided in two time intervals of 25% and 75% energy respectively (in s). See acoustat
- time.Q75: third quartile time. The time at which the signal is divided in two time intervals of 75% and 25% energy respectively (in s). See acoustat
- time.IQR: interquartile time range. Time range between 'time.Q25' and 'time.Q75' (in s). See acoustat
- skew: skewness. Asymmetry of the spectrum (see note in specprop description)

- kurt: kurtosis. Peakedness of the spectrum (see note in specprop description)
- sp.ent: spectral entropy. Energy distribution of the frequency spectrum. Pure tone ~ 0; noisy ~ 1. See sh
- time.ent: time entropy. Energy distribution on the time envelope. Pure tone ~ 0; noisy ~ 1. See th
- entropy: spectral entropy. Product of time and spectral entropy sp.ent * time.ent. See H
- sfm: spectral flatness. Similar to sp.ent (Pure tone ~ 0; noisy ~ 1). See sfm
- peakf: peak frequency. Frequency with highest energy
- meanfun: average of fundamental frequency measured across the acoustic signal
- minfun: minimum fundamental frequency measured across the acoustic signal
- maxfun: maximum fundamental frequency measured across the acoustic signal
- meandom: average of dominant frequency measured across the acoustic signal
- mindom: minimum of dominant frequency measured across the acoustic signal
- maxdom: maximum of dominant frequency measured across the acoustic signal
- dfrange: range of dominant frequency measured across the acoustic signal
- modindx: modulation index. Calculated as the cumulative absolute difference between adjacent measurements of dominant frequencies divided by the dominant frequency range. 1 means the signals is not modulated.
- startdom: dominant frequency measurement at the start of the signal
- enddom: dominant frequency measurement at the end of the signal
- dfslope: slope of the change in dominant through time ((enddom-startdom)/duration). Units are kHz/s.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

Examples

```
## Not run:
# First set temporary folder
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")

a <- specan(X = selec.table, bp = c(0, 22))

# using a diferent threshold
a <- specan(X = selec.table, bp = c(0, 22), threshold = 20)
# View(a)


## End(Not run)
```

specreator *Spectrograms of selected signals*

Description

specreator creates spectrograms of signals selected by manualoc or autodetec.

Usage

```
specreator(X, wl = 512, flim = c(0, 22), wn = "hanning", pal
  = reverse.gray.colors.2, ovlp = 70, inner.mar = c(5, 4, 4, 2), outer.mar =
  c(0, 0, 0, 0), picsize = 1, res = 100, cexlab = 1, title = TRUE,
  propwidth = FALSE, xl = 1, osci = FALSE, gr = FALSE,  sc = FALSE, line = TRUE,
  mar = 0.05, it = "jpeg", parallel = 1, path = NULL, pb = TRUE, fast.spec = FALSE, ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signals (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
flim	A numeric vector of length 2 for the frequency limit (in kHz) of the spectrogram, as in spectro. Default is c(0, 22).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
pal	A color palette function to be used to assign colors in the plot, as in spectro. Default is reverse.gray.colors.2.
ovlp	Numeric vector of length 1 specifying the percent overlap between two consecutive windows, as in spectro. Default is 70.
inner.mar	Numeric vector with 4 elements, default is c(5,4,4,2). Specifies number of lines in inner plot margins where axis labels fall, with form c(bottom, left, top, right). See par.
outer.mar	Numeric vector with 4 elements, default is c(0,0,0,0). Specifies number of lines in outer plot margins beyond axis labels, with form c(bottom, left, top, right). See par.
picsize	Numeric argument of length 1. Controls relative size of spectrogram. Default is 1. Ignored when propwidth is TRUE.
res	Numeric argument of length 1. Controls image resolution. Default is 100 (faster) although 300 - 400 is recommended for publication/ presentation quality.
cexlab	Numeric vector of length 1 specifying the relative size of axis labels. See spectro.
title	Logical argument to add a title to individual spectrograms. Default is TRUE.

propwidth	Logical argument to scale the width of spectrogram proportionally to duration of the selection. Default is FALSE.
xl	Numeric vector of length 1. A constant by which to scale spectrogram width if propwidth = TRUE. Default is 1.
osci	Logical argument to add an oscillogram underneath spectrogram, as in spectro. Default is FALSE.
gr	Logical argument to add grid to spectrogram. Default is FALSE.
sc	Logical argument to add amplitude scale to spectrogram, default is FALSE.
line	Logical argument to add red lines at start and end times of selection (or box if low.f and high.f columns are provided). Default is TRUE.
mar	Numeric vector of length 1. Specifies the margins adjacent to the start and end points of selections, dealineating spectrogram limits. Default is 0.05.
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.
...	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 provides access to bath process of (a modified version of) the spectro function from the 'seewave' package. The function creates spectrograms for visualization of vocalizations. Setting inner.mar to c(4,4.5,2,1) and outer.mar to c(4,2,2,1) works well when picsize = 2 or 3. Title font size, inner.mar and outer.mar (from mar and oma) don't work well when osci or sc = TRUE, this may take some optimization by the user. Setting 'fast' argument to TRUE significantly increases speed, although some options become unavailable, as collevels, and sc (amplitude scale). This option is indicated for signals with high background noise levels.

Value

Image files containing spectrograms of the signals listed in the input data frame.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

See Also

trackfreqs for creating spectrograms to visualize frequency measurements by specan, snrspecs for creating spectrograms to optimize noise margins used in sig2noise

Other spectrogram.creators: dfDTW, dfts, ffDTW, ffts, snrspecs, sp.en.ts, trackfreqs

Examples

```
## Not run:
# First set empty folder
setwd(tempdir())
data(list = c("Phae.long1", "Phae.long2","selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files
writeWave(Phae.long2, "Phae.long2.wav")

# make spectrograms

specreator(selec.table, flim = c(0, 11), res = 300, mar = 0.05, wl = 300)

 #' #check this folder!!
getwd()

## End(Not run)
```

---

trackfreqs *Spectrograms with frequency measurements*

Description

trackfreqs creates spectrograms to visualize dominant and fundametal frequency measurements (contours) of signals selected by manualoc or autodetec.

Usage

```
trackfreqs(X, wl = 512, flim = c(0, 22), wn = "hanning", pal =
  reverse.gray.colors.2, ovlp = 70, inner.mar = c(5, 4, 4, 2), outer.mar =
  c(0, 0, 0, 0), picsize = 1, res = 100, cexlab = 1, title = TRUE, propwidth = FALSE,
  xl = 1, osci = FALSE, gr = FALSE, sc = FALSE, bp = c(0, 22), cex = c(0.6, 1),
  threshold = 15, contour = "both", col = c("skyblue", "red2"),
   pch = c(21, 24),  mar = 0.05, lpos = "topright", it = "jpeg", parallel = 1,
   path = NULL, img.suffix = NULL, custom.contour = NULL, pb = TRUE, type = "p",
   leglab = c("Ffreq", "Dfreq"), col.alpha = 0.6, line = TRUE,
   fast.spec = FALSE, ff.method = "seewave", ...)
```

Arguments

X	Data frame with results containing columns for sound file name (sound.files), selection number (selec), and start and end time of signal (start and end). The ouptut of manualoc or autodetec can be used as the input data frame.
wl	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
flim	A numeric vector of length 2 for the frequency limit of the spectrogram (in kHz), as in spectro. Default is c(0, 22).
wn	Character vector of length 1 specifying window name. Default is "hanning". See function ftwindow for more options.
pal	A color palette function to be used to assign colors in the plot, as in spectro. Default is reverse.gray.colors.2.
ovlp	Numeric vector of length 1 specifying % of overlap between two consecutive windows, as in spectro. Default is 70.
inner.mar	Numeric vector with 4 elements, default is c(5,4,4,2). Specifies number of lines in inner plot margins where axis labels fall, with form c(bottom, left, top, right). See par.
outer.mar	Numeric vector with 4 elements, default is c(0,0,0,0). Specifies number of lines in outer plot margins beyond axis labels, with form c(bottom, left, top, right). See par.
picsize	Numeric argument of length 1. Controls relative size of spectrogram. Default is 1.
res	Numeric argument of length 1. Controls image resolution. Default is 100 (faster) although 300 - 400 is recommended for publication/ presentation quality.
cexlab	Numeric vector of length 1 specifying the relative size of axis labels. See spectro.
title	Logical argument to add a title to individual spectrograms. Default is TRUE.
propwidth	Logical argument to scale the width of spectrogram proportionally to duration of the selected call. Default is FALSE.
xl	Numeric vector of length 1. A constant by which to scale spectrogram width. Default is 1.
osci	Logical argument to add an oscillogram underneath spectrogram, as in spectro. Default is FALSE.
gr	Logical argument to add grid to spectrogram. Default is FALSE.
sc	Logical argument to add amplitude scale to spectrogram, default is FALSE.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz) or "frange" to indicate that values in low.f and high.f columns will be used as bandpass limits. Default is c(0, 22).
cex	Numeric vector of length 2, specifies relative size of points plotted for frequency measurements and legend font/points, respectively. See spectro.
threshold	amplitude threshold (%) for fundamental frequency and dominant frequency detection. Default is 15.

contour	Character vector, one of "df", "ff" or "both", specifying whether the dominant or fundamental frequencies or both should be plotted. Default is "both".
col	Vector of length 1 or 2 specifying colors of points plotted to mark fundamental and dominant frequency measurements respetively (if both are plotted). Default is c("skyblue", "red2").
pch	Numeric vector of length 1 or 2 specifying plotting characters for the frequency measurements. Default is c(21, 24).
mar	Numeric vector of length 1. Specifies the margins adjacent to the selections to set spectrogram limits. Default is 0.05.
lpos	Character vector of length 1 or numeric vector of length 2, specifiying position of legend. If the former, any keyword accepted by xy.coords can be used (see below). If the latter, the first value will be the x coordinate and the second value the y coordinate for the legend's position. Default is "topright".
it	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing).
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
img.suffix	A character vector of length 1 with a suffix (label) to add at the end of the names of image files. Default is NULL.
custom.contour	A data frame with frequency contours for exactly the same sound files and selection as in X. The frequency values are assumed to be equally spaced in between the start and end of the signal. The first 2 colums of the data frame should contain the 'sound.files' and 'selec' columns and should be identical to the corresponding columns in X (same order).
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
type	A character vector of length 1 indicating the type of frequency contour plot to be drawn. Possible types are "p" for points, "l" for lines and "b" for both.
leglab	A character vector of length 1 or 2 containing the label(s) of the frequency contour legend in the output image.
col.alpha	A numeric vector of length 1 within [0,1] indicating how transparent the lines/points should be.
line	Logical argument to add red lines (or box if low.f and high.f columns are provided) at start and end times of selection. 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, and sc (amplitude scale). This option is indicated for signals with high background noise levels. Palette colors gray.1, gray.2, gray.3, topo.1 and rainbow.1 (which should be imported from the package monitoR) seem to work better with 'fast' spectograms. Palette colors gray.1, gray.2, gray.3 offer decreasing darkness levels. THIS IS STILL BEING TESTED.

ff.method	Character. Selects the method used to calculate the fundamental frequency. Either 'tuneR' (using FF) or 'seewave' (using fund). Default is 'seewave'. 'tuneR' performs faster (and seems to be more accurate) than 'seewave'.
...	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 provides visualization of frequency measurements as the ones made by specan. Frequency measures can be made by the function or input by the user (see 'custom.contour' argument) Arguments that are accepted by xy.coords and can be used for 'lpos' are: "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". Setting inner.mar to c(4,4.5,2,1) and outer.mar to c(4,2,2,1) works well when picsize = 2 or 3. Title font size, inner.mar and outer.mar (from mar and oma) don't work well when osci or sc = TRUE, this may take some optimization by the user. Note that if no amplitude was detected for a particular time bin, then the image will show a dark dot at the bottom of the time bin.

Value

Spectrograms of the signals listed in the input data frame showing the location of the dominant and fundamental frequencies.

Author(s)

Grace Smith Vidaurre and Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

specreator for creating spectrograms from selections, snrspecs for creating spectrograms to optimize noise margins used in sig2noise

Other spectrogram.creators: dfDTW, dfts, ffDTW, ffts, snrspecs, sp.en.ts, specreator

Examples

```
## Not run:
#Set temporary working directory
setwd(tempdir())

#load data
data("Cryp.soui")
writeWave(Cryp.soui, "Cryp.soui.wav") #save sound files

#autodetec location of signals
ad <- autodetec(threshold = 6, bp = c(1, 3), mindur = 1.2,
maxdur = 3, img = FALSE, ssmooth = 600, wl = 300, flist = "Cryp.soui.wav")

#track dominant frequency graphs
trackfreqs(X = ad[!is.na(ad$start),], flim = c(0, 5), ovlp = 90, it = "tiff",
bp = c(1, 3), contour = "df", wl = 300)
```

```
#using users frequency data (custom.contour argument)
#first get contours using dfts
df <- dfts(X = ad[!is.na(ad$start),], flim = c(0, 5), ovlp = 90, img = FALSE,
bp = c(1, 3),  wl = 300)

# now input the dfts output into trackfreqs
trackfreqs(X = ad[!is.na(ad$start),], custom.contour = df ,flim = c(0, 5), ovlp = 90, it = "tiff")

# Check this folder
getwd()

#track both frequencies
trackfreqs(X = ad[!is.na(ad$start),], flim = c(0, 5), ovlp = 90, it = "tiff",
bp = c(1, 3), contour = "both", wl = 300)


## End(Not run)
```

---

warbleR *warbleR: A package to streamline bioacoustic analysis*

Description

warbleR is a package designed to streamline analysis of animal acoustic signals in R. This package allows users to collect open-access avian vocalizations data or input their own data into a workflow that facilitates spectrographic visualization and measurement of acoustic parameters. warbleR makes fundamental sound analysis tools from the R package seewave, as well as new tools not yet offered in the R environment, readily available for batch process analysis. The functions facilitate searching and downloading avian vocalizations from Xeno-Canto http://www.xeno-canto.org/, creating maps of Xeno-Canto recordings, converting .mp3 files to .wav files, checking .wav files, automatically detecting acoustic signals, selecting them manually, printing spectrograms of whole recordings or individual signals, measuring signal to noise ratio, cross-correlation and performing acoustic measurements.

The warbleR package offers three overarching categories of functions:

- Obtaining avian vocalization data
- Sound file management
- Streamlined (bio)acoustic analysis in R

Details

License: GPL (>= 2)

Obtaining avian vocalization data

querxc: Download recordings and metadata from Xeno-Canto

xcmaps: Create maps to visualize the geographic spread of Xeno-Canto recordings

imp.syrinx: Importing Syrinx selections

imp.raven: Importing Raven selections

Managing sound files

mp32wav: Convert several .mp3 files in working directory to .wav format

checksels: Check whether selections can be read by subsequent functions

checkwavs: Check whether .wav files can be read by subsequent functions and the minimum windows length ("wl" argument) that can be used

fixwavs: Fix .wav files to allow importing them into R

wavdur: Determine the duration of sound files

Exploring/analyzing signal structure

autodetec: Automatically detect start and end of acoustic signals

manualoc: Interactive spectrographic view to measure start and end of acoustic signals

autodetec: Automatic detection of acoustic signals based on ampltiude

seltailor: Interactive view of spectrograms to tailor start and end of selections

lspec: Produce spectrograms of whole recordings split into multiple rows

lspec2pdf: Combine lspec images to single pdf files

specreator: Create spectrograms of manualoc selections

snrspecs: Create spectrograms to visualize margins over which noise will be measured by sig2noise

sig2noise: Measure signal-to-noise ratio across multiple files

trackfreqs: Create spectrograms to visualize frequency measurements

filtersels: Filter selection data frames based on filtered image files

specan: Measure acoustic parameters on selected acoustic signals

xcorr: Pairwise cross-correlation of multiple signals

xcorr.graph: Pairwise cross-correlation of multiple signals

dfts: Extract the dominant frequency values across the signal as a time series

ffts: Extract the fundamental frequency values across the signal as a time series

sp.en.ts: Extract the spectral entropy values across the signal as a time series

dfDTW: Calculate acoustic dissimilarity using dynamic time warping on dominant frequency contours

ffDTW: Calculate acoustic dissimilarity using dynamic time warping on fundamental frequency contours

compare.methods: Produce graphs to visually assess performance of acoustic distance measurements

catalog: Produce a vocalization catalog with spectrograms in and array with several rows and columns

catalog2pdf: Combine catalog images to single pdf files

coor.graph: Creat graphs of coordinated singing

coor.test: Assess statistical significance of singing coordination

Author(s)

Marcelo Araya-Salas & Grace Smith Vidaurre

Maintainer: Marcelo Araya-Salas (<araya-salas@cornell.edu>)

---

wavdur *Measure the duration of sound files*

Description

wavdur measures the duration of sound files in '.wav' format

Usage

```
wavdur(files = NULL, path = NULL)
```

Arguments

files	Character vector with the names of the sound files to be measured. The sound files should be in the working directory or in the directory provided in 'path'.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.

Details

This function returns the duration (in seconds) of sound files.

Value

A data frame with the duration (in seconds) of the sound files.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

Examples

```
## Not run:
# Set temporary working directory
setwd(tempdir())

data(list = c("Phae.long1", "Phae.long2", "Phae.long3"))
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")

wavdur()

## End(Not run)
```

`xcmaps` *Maps of Xeno-Canto recordings by species*

Description

`xcmaps` creates maps to visualize the geographic spread of Xeno-Canto recordings.

Usage

```
xcmaps(X, img = TRUE, it = "jpeg", res = 100, labels = F)
```

Arguments

`X`	Data frame output from `querxc`.
`img`	A logical argument specifying whether an image file of each species map should be returned, default is `TRUE`.
`it`	A character vector of length 1 giving the image type to be used. Currently only "tiff" and "jpeg" are admitted. Default is "jpeg".
`res`	Numeric argument of length 1. Controls image resolution. Default is 100 (faster) although 300 - 400 is recommended for publication/ presentation quality.
`labels`	A logical argument defining whether dots depicting recording locations are labeled. If `TRUE` then the Recording_ID is used as label.

Details

This function creates maps for visualizing the geographic spread of recordings from the open-access online repository Xeno-Canto (`http://www.xeno-canto.org/`). The function takes the output of `querxc` as input. Maps can be displayed in the graphic devide or saved as images in the working directory.

Value

A map of Xeno-Canto recordings per species (image file), or a faceted plot of species map(s) in the active graphic device.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>) and Grace Smith Vidaurre

Examples

```
## Not run:
# search in xeno-canto
X <- querxc("Phaethornis anthophilus", download = FALSE)

#create image in R graphic device
xcmaps(X, img = FALSE)

#or save it as a file in the working directory
xcmaps(X)


## End(Not run)
```

---

xcorr *Spectrogram cross-correlation*

Description

`xcorr` estimates the similarity of two spectrograms by means of cross-correlation

Usage

```
xcorr(X, wl =512, frange= NULL, ovlp=90, dens=0.9, bp= NULL, wn='hanning',
cor.method = "pearson", parallel = 1, path = NULL, pb = TRUE, na.rm = FALSE,
 dfrange = FALSE, cor.mat = TRUE)
```

Arguments

`X`	Data frame containing columns for sound files (sound.files), selection number (selec), and start and end time of signal (start and end).
`wl`	A numeric vector of length 1 specifying the window length of the spectrogram, default is 512.
`frange`	A numeric vector of length 2 setting the upper and lower frequency limits (in kHz) in which to compare the signals. Must be provided. The `dfts` function can be used to determine this parameter if `dfrange = TRUE`. This method is more adequate for pure tone signals. Default is `NULL`.

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.
dens	Numeric vector of length 1 specifying the approximate density of points in which to sample amplitude. See makeTemplate. Deafult is 0.9.
bp	A numeric vector of length 2 for the lower and upper limits of a frequency bandpass filter (in kHz) in which to detect dominant frequency. Only applied when frange is NULL. Default is NULL.
wn	A character vector of length 1 specifying the window name as in ftwindow.
cor.method	A character vector of length 1 specifying the correlation method as in cor.
parallel	Numeric. Controls whether parallel computing is applied. It specifies the number of cores to be used. Default is 1 (i.e. no parallel computing). Not available in Windows OS.
path	Character string containing the directory path where the sound files are located. If NULL (default) then the current working directory is used.
pb	Logical argument to control progress bar. Default is TRUE. Note that progress bar is only used when parallel = 1.
na.rm	Logical. If TRUE all NAs produced when pairwise cross-correlations failed are removed from the results. This means that all selections with at least 1 cross-correlation that failed are excluded.
dfrange	Logical. If TRUE the dfts function can is used to determine the frequency range in which to compare signals.
cor.mat	Logical. If TRUE only the correlation matrix is returned. Default is TRUE.

Details

This function calculates the pairwise similarity of multiple signals by means of spectrogram cross-correlation. This method "slides" one spectrogram over the other calculating a correlation of the amplitude values at each step. The function runs pairwise cross-correlations on several signals and returns a list including the correlation statistic for each "sliding" step as well as the maximum (peak) correlation for each pairwise comparison. To accomplish this the margins of the signals are expanded by half the duration of the signal both before and after the provided time coordinates. The correlation matrix could have NA's if some of the pairwise correlation did not work (common when sound files have been modified by band-pass filters). This function is a modified version of the corMatch and makeTemplate from the awesome R package 'monitoR'.

Value

If corr.mat is TRUE the function returns a matrix with the maximum (peak) correlation for each pairwise comparison. Otherwise it will return a list that includes 1) a data frame with the correlation statistic for each "sliding" step, 2) a matrix with the maximum correlation for each pairwise comparison, and 3) the frequency range.

Author(s)

Marcelo Araya-Salas <araya-salas@cornell.edu>)

Source

H. Khanna, S.L.L. Gaunt & D.A. McCallum (1997). Digital spectrographic cross-correlation: tests of sensitivity. Bioacoustics 7(3): 209-234

See Also

xcorr.graph

Examples

```
## Not run:
#First set temporary working directory
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4","selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files
writeWave(Phae.long2, "Phae.long2.wav")
writeWave(Phae.long3, "Phae.long3.wav")
writeWave(Phae.long4, "Phae.long4.wav")

xcor <- xcorr(X = selec.table, wl = 300, frange = c(2, 9), ovlp = 90,
dens = 1, wn = 'hanning', cor.method = "pearson")


## End(Not run)
```

---

xcorr.graph *Pairwise plots of spectrogram cross-correlation scores*

Description

xcorr.graph generates pairwise plots showing the spectrogram cross-correlation scores against the time sliding.

Usage

```
xcorr.graph(X, cex.cor = 1, cex.lab = 1, cex.axis.lab = 1, rel.cex = FALSE, labs = NULL)
```

Arguments

X	Output from xcorr function.
cex.cor	A numeric vector of length 1 giving the amount by which correlation scores (in the upper triangle of the multipannel plot) should be magnified. Default is 1.
cex.lab	A numeric vector of length 1 giving the amount by which signal selection labels (in diagonal of the multipannel plot) should be magnified. Default is 1.
cex.axis.lab	A numeric vector of length 1 giving the amount by which the axis labels should be magnified. Default is 1.

rel.cex	Logical. Controls whether the size of the correlation scores (in the upper triangle of the multipannel plot) should be relative to the correlation score.
labs	Alternative selection labels. If not provided the combined name of sound files and selection numbers are used as labels. Default is FALSE.

Details

This function generates pairwise plots of the spectrogram cross-correlation scores by sliding step. The function takes the output of xcorr (when cor.mat is FALSE in xcorr) as input. The colors of the lines in the lower triangle of the plot matrix represent the strenght of the similarity between the two signals. The x axis shows the time difference between the two signals for each sliding step (0 means perfectly centered signals). Note that large number of signals may not display well in the default graphic device. In such cases saving the plot as and image file is adviced.

Author(s)

Marcelo Araya-Salas (<araya-salas@cornell.edu>)

See Also

xcorr

Examples

```
## Not run:
#load data
#First set temporary working directory]
setwd(tempdir())

#load data
data(list = c("Phae.long1", "Phae.long2", "selec.table"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files
writeWave(Phae.long2, "Phae.long2.wav")

 #run cross correlation first
 xcor<-xcorr(X = selec.table[1:5,], wl =300, frange= c(2, 9), ovlp=90, dens=0.8,
 wn='hanning', cor.method = "pearson", cor.mat = FALSE)

 #plot pairwise scores
  xcorr.graph(X = xcor, cex.cor = 2, cex.lab = 1, rel.cex = FALSE)

## End(Not run)
```

Index