Package 'warbleR'

November 30, 2015

Type Package

Title Streamline Bioacoustic Analysis

Version 1.1.0

Date 2015-11-30

Author Marcelo Araya-Salas, Grace Smith Vidaurre, Hua Zhong

Maintainer Marcelo Araya-Salas <marceloa27@gmail.com>

Description Offers three overarching categories of functions to obtain bird vocalization recordings from the web, manage sound files, and facilitate (bio)acoustic analysis in R.

License GPL (>= 2)

Imports rjson, pbapply, RCurl, fftw, utils, stats, grDevices, graphics

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

LazyData TRUE

NeedsCompilation no

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

Repository CRAN

Date/Publication 2015-11-30 17:18:51

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 = FALSE)
```

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 a 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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used.

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 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 (http://marceloarayasalas.weebly.com/). Implements a modified version of the timer function from seewave.

Examples

```
## Not run:
# First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

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 = 10, env = "abs", 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)

#check this folder!!
getwd()

#remove example directory
unlink(getwd(),recursive = TRUE)

## End(Not run)
```

---

checkwavs *Check .wav files*

Description

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

Usage

```
checkwavs()
```

Details

This function has no associated arguments, although users must set the working directory where they wish to check .wav files beforehand.

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 (http://marceloarayasalas.weebly.com/)

Examples

```
## Not run:
# First create empty folder
setwd(tempdir())

# save wav file examples
```

```
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")

checkwavs()


## 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)
```

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.

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

Graphs of the singing events in the input data frame are saved in the working directory.

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/)

Examples

```
## Not run:

# First set temporary folder
setwd(tempdir())

# load simulate singing events  (see data documentation)
data(coor.sing)

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


#' # make coor.graphs in graphic device format
coor.graph(X = coor.sing, ovlp = T, only.coor = F, img = FALSE)

## End(Not run)
```

`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. Three events are simulated: overlapping signals (ovlp), alternating signals (altern) and non-synchronized signals (no.sync). .*

Description

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. Three events are simulated: overlapping signals (ovlp), alternating signals (altern) and non-synchronized signals (no.sync). .

Usage

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

Format

coor.sing Selections of coordinated singing

coor.test *Randomization test for singing coordination*

Description

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

Usage

```
coor.test(X, iterations = 1000, less.than.chance = TRUE, parallel = 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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

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 (http://marceloarayasalas.weebly.com/)

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(coor.sing)

# testing if coordination happens less than expected by chance
coor.test(coor.sing, iterations = 1000, less.than.chance = T)

# testing if coordination happens more than expected by chance
coor.test(coor.sing, iterations = 1000, less.than.chance = F)

## End(Not run)
```

---

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

Description

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

Usage

```
dfts(X, wl = 512, flim = c(0, 22), length.out = 20, 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, gr = FALSE, sc = FALSE, bp = c(0, 22), cex = 1,
  threshold = 15, col = "dodgerblue", pch = 16,  mar = 0.05,
  lpos = "topright", it = "jpeg", img = TRUE, parallel = 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.
flim	A numeric vector of length 2 for the frequency limit of the spectrogram (in kHz), as in spectro. Default is c(0, 22).
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.
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.
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). Default is c(0, 22).
cex	Numeric vector of length 1, specifies relative size of points plotted for frequency measurements and legend font/points, respectively. See spectro.
threshold	amplitude threshold (%) for dominant frequency detection. Default is 15.
col	Vector of length 1 specifying colors of points plotted to mark dominant frequency measurements. Default is "dodgerblue".
pch	Numeric vector of length 1 specifying plotting characters for the frequency measurements. Default is 16.
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".

img	Logical argument. If FALSE, image files are not produced. Default is TRUE.
parallel	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

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.

Value

A data frame with the dominant frequency values measured across the signals. 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 (http://marceloarayasalas.weebly.com/)

See Also

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

Other spectrogram.creators: snrspecs; specreator; trackfreqs

Examples

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

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

# run function
dfts(manualoc.df, length.out = 30, flim = c(1, 12), picsize = 2, res = 100, bp = c(2, 9))


## 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)
```

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).
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.

Value

A single data frame with the information from 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.file 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 (http://marceloarayasalas.weebly.com/)

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 = F, sep= "\t")
```

```
write.table(selection.files[[2]],file = "1023-Arremonops rufivirgatus.selections.txt",
row.names = F, sep= "\t")

#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)
```

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.

Value

A single data frame with the information from 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.file 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 (http://marceloarayasalas.weebly.com/)

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 = F,
 col.names = F, sep= "\t")

write.table(selection.files[[4]],file = "Phae.long4.wav.txt",row.names = F,
col.names = F, 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 = FALSE, wl = 512, gr = FALSE, pal = reverse.gray.colors.2,
cex = 1, it = "jpeg", flist = NULL)
```

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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.
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.

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 a equivalent data frame), the function delimits and labels the selections. This function aims to facilitate 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 (http://marceloarayasalas.weebly.com/) and Hua Zhong

Examples

```
## Not run:
# First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

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

lspec(sxrow = 2, rows = 8, pal = reverse.heat.colors)
lspec(sxrow = 2, rows = 8, X = manualoc.df, pal = reverse.heat.colors) #including selections

#remove example directory
unlink(getwd(),recursive = TRUE)

## 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)
```

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.

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.

Value

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

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/) and Hua Zhong

Examples

```
## Not run:
#First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

# 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()
# need to use the buttoms to manipulate function
# check working directory for .csv file after stopping function

#remove example directory
unlink(getwd(),recursive = TRUE)

## End(Not run)
```

---

manualoc.df *Data frame of manualoc() selections.*

Description

A data frame containing information for calls selected using manualoc.

Usage

```
data(manualoc.df)
```

Format

A data frame with 11 rows and 6 variables:

sound.files recording names

selec selection numbers within recording

start start times of selected call or element

end end times of selected call or element

sel.comment selection comments

rec.comment recording comments

Source

Marcelo Araya Salas, warbleR

`mp32wav` *Convert .mp3 files to .wav*

Description

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

Usage

```
mp32wav(samp.rate = 44.1)
```

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.

Details

convert all .mp3 files in working directory to .wav format. Function used internally to read .mp3 files (`readMP3`) sometimes crashes This should be fixed in the next version of tuneR.

Value

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

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/) 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` *Acoustic recording #1 of* Phaethornis longirostris *(Long-billed Hermit).*

Description

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

Description

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

Description

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

Description

Acoustic 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, parallel = FALSE)
```

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/).
download	Logical argument. Downloads recording file names and associated metadata if FALSE. If TRUE, recordings are also downloaded to working directory as .mp3 files. Default is FALSE.
X	Data frame with the same columns as the output of the function, or at least the following columns: Genus, Specific_epithet and Recording_ID. 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.
parallel	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS. Only used for downloading files.

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 can also download the associated sound files.

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.

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/) and Hua Zhong

Examples

```
## Not run:
# First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))
df1 <- querxc("Phaethornis anthophilus", download = FALSE)
View(df1)

#downloading files
querxc("Phaethornis anthophilus", download = TRUE)
#check this folder!!
getwd()

# remove example directory
unlink(getwd(),recursive = TRUE)

## End(Not run)
```

`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'

`sig2noise` *Measure signal-to-noise ratio*

Description

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

Usage

```
sig2noise(X, mar, parallel = 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).
mar	numeric vector of length 1. Specifies the margins adjacent to the start and end points of selection over which to measure noise.
parallel	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

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 (http://marceloarayasalas.weebly.com/) 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","manualoc.df"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files

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

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

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

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

Description

Simulated coordinated singing events.

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", 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, trel =
  FALSE, propwidth = FALSE, xl=1, osci = FALSE, gr = FALSE, sc = FALSE, mar =
  0.2, snrmar = 0.1, it = "jpeg", parallel = 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).
`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. See Details.
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.
trel	Logical argument to add a time axis scale relative to the wave. Default is FALSE.
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.
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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

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 (http://marceloarayasalas.weebly.com/) 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: dfts; specreator; trackfreqs

Examples

```
## Not run:
# First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

data(list = c("Phae.long1", "Phae.long2"))
data(manualoc.df)
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(manualoc.df, 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")

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

snrspecs(manualoc.df[grepl(c("Phae.long1"), manualoc.df$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)

#check this folder!!
getwd()

#remove example directory
unlink(getwd(),recursive = TRUE)

## End(Not run)
```

specan *Measure acoustic parameters in batches of sound files*

Description

specan measures 22 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 = 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.
bp	numeric vector of length 2 giving the lower and upper limits of the frequency bandpass filter (in kHz). Default is c(0, 22).
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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

Details

The ouptut of manualoc or autodetec can be used directly without any additional modification. The function measures 22 acoustic parameters on each selection in the data frame. Most parameters are produced internally by specprop, fpeaks, fund, and dfreq from the package seewave.

Value

Data frame with the following acoustic parameters:

- duration: length of signal
- meanfreq: mean frequency (in kHz)
- sd: standard deviation of frequency
- median: median frequency (in kHz)
- Q25: first quantile (in kHz)
- Q75: third quantile (in kHz)
- IQR: interquantile range (in kHz)

- skew: skewness (see note in specprop description)
- kurt: kurtosis (see note in specprop description)
- sp.ent: spectral entropy
- sfm: spectral flatness
- mode: mode frequency
- centroid: frequency centroid (see specprop)
- peakf: peak frequency (frequency with highest energy)
- meanfun: average of fundamental frequency measured across acoustic signal
- minfun: minimum fundamental frequency measured across acoustic signal
- maxfun: maximum fundamental frequency measured across acoustic signal
- meandom: average of dominant frequency measured across acoustic signal
- mindom: minimum of dominant frequency measured across acoustic signal
- maxdom: maximum of dominant frequency measured across acoustic signal
- dfrange: range of dominant frequency measured across acoustic signal
- modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/), Grace Smith Vidaurre and Hua Zhong

Examples

```
## Not run:
#First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

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")
data(manualoc.df)

a <- specan(X = manualoc.df, bp = c(0, 22))

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

# remove example directory
unlink(getwd(), recursive = TRUE)

## 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, trel = FALSE,
  propwidth = FALSE, xl = 1, osci = FALSE, gr = FALSE,  sc = FALSE, line = TRUE,
  mar = 0.05, it = "jpeg", parallel = 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.

trel	Logical argument to add a time axis scale relative to the wave. Default is FALSE.
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. 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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

Details

This 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.

Value

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

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/) 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: dfts; snrspecs; trackfreqs

Examples

```
## Not run:
# First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))
```

```
data(list = c("Phae.long1", "Phae.long2","manualoc.df"))
writeWave(Phae.long1, "Phae.long1.wav") #save sound files
writeWave(Phae.long2, "Phae.long2.wav")

# make spectrograms

specreator(manualoc.df, 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)
#check this folder!!
getwd()

#remove example directory
unlink(getwd(),recursive = TRUE)

## End(Not run)
```

---

trackfreqs *Spectrograms with frequency measurements*

Description

trackfreqs creates spectrograms to visualize dominant and fundametal frequency measurements 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(0.8, 1),
  threshold = 15, col = c("chartreuse3", "dodgerblue"), pch = c(17, 16),  mar = 0.05,
  lpos = "topright", it = "jpeg", parallel = 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.
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). Default is c(0, 22).
cex	Numeric vector of length 1, 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.
col	Vector of length 2 specifying colors of points plotted to mark fundamental and dominant frequency measurements. Default is c("chartreuse3", "dodgerblue").
pch	Numeric vector of length 2 specifying plotting characters for the frequency measurements. Default is c(17, 16).
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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

Details

This function provides visualization of frequency measurements made by specan. 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.

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 (http://marceloarayasalas.weebly.com/)

See Also

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

Other spectrogram.creators: dfts; snrspecs; specreator

Examples

```
## Not run:
#First create empty folder
dir.create(file.path(getwd(),"temp"))
setwd(file.path(getwd(),"temp"))

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

# make  spectrograms

trackfreqs(manualoc.df, 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, bp = c(0, 14), cex = c(1.5, 2),
col = c("blue", "red"),  mar = 0.09, lpos = "bottomright", it = "jpeg")

# make only Phae.long1 spectrograms

trackfreqs(manualoc.df[manualoc.df$sound.files == "Phae.long1.wav", ], 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,
bp = c(3, 14), cex = c(1.5, 2), col = c("blue", "red"),  mar = 0.09,
lpos = "bottomright", it = "tiff")

# remove example directory
unlink(getwd(),recursive = TRUE)
```

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

---

warbleR *warbleR: A package to streamline bioacoustic analysis*

Description

warbleR is a package designed to streamline analysis of (bio)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 in a batch process. 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.

Details

The warbleR package offers three overarching categories of functions:

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

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

checkwavs: Check whether .wav files can be read by subsequent functions

Streamlining analysis of acoustic signal structure in R

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

lspec: Produce spectrograms of whole recordings split into multiple rows

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

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 as a time series

coor.graph: Creat graphs of coordinated singing

coor.test: Assess statistical significance of singing coordination

Author(s)

Marcelo Araya-Salas, Grace Smith Vidaurre, Hua Zhong

Maintainer: Marcelo Araya-Salas (marceloa27@gmail.com)

---

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)
```

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.

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 (http://marceloarayasalas.weebly.com/) and Grace Smith Vidaurre

Examples

```
## Not run:
X <- querxc("Phaethornis anthophilus", download = FALSE)
View(X)
xcmaps(X)
xcmaps(X, img = FALSE, it = "jpeg")

## 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 = FALSE)
```

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. If not provided (NULL) the dfts function is used internally to define the higher and lower dominant frequency in the signals to be analyzed. 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	Either logical or numeric. Controls wehther parallel computing is applied. If TRUE 2 cores are employed. If numeric, it specifies the number of cores to be used. Not available for windows OS.

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. This function is a modified version of the corMatch and makeTemplate from the awesome R package 'monitoR'.

Value

A list that includes 1) a data frame with the correlation statistic for each "sliding" step, 2) a matrix with the maximum (peak) correlation for each pairwise comparison, and 3) the frequency range.

Author(s)

Marcelo Araya-Salas (http://marceloarayasalas.weebly.com/)

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 temporal working directory
setwd(tempdir())
```

```
#load data
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4","manualoc.df"))
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 = manualoc.df, 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` 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 (http://marceloarayasalas.weebly.com/)

See Also

xcorr

Examples

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

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

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

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

## End(Not run)
```

Index