Package 'Rraven'

May 23, 2019

Type Package

Title Connecting R and 'Raven' Sound Analysis Software

Version 1.0.6

Date 2019-05-22

Author Marcelo Araya-Salas [aut, cre]

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

Description A tool to exchange data between R and 'Raven' sound analysis software <http://www.birds.cornell.edu/brp/raven/RavenOverview.html> (Cornell Lab of Ornithology). Functions work on data formats compatible with the R package 'warbleR'.

License GPL (>= 2)

Imports pbapply, warbleR, utils, stats, seewave, tuneR, NatureSounds

Depends R (>= 3.2.1)

LazyData TRUE

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

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

NeedsCompilation no

Suggests knitr, vegan, dplyr, kableExtra

VignetteBuilder knitr

RoxygenNote 6.1.1

Repository CRAN

Date/Publication 2017-11-17 08:12:11

R topics documented:

`exp_empty_sels` *Export a 'Raven' selection for all sound files in a folder*

Description

`exp_empty_sels` exports a 'Raven' selection data in .txt format that includes empty selections for all sound files in a folder.

Usage

```
exp_empty_sels(path = NULL, file.name = NULL, pb = TRUE)
```

Arguments

`path`	A character string indicating the path of the directory in which to look for sound files. If not provided (default) the function will use the current working directory.
`file.name`	Name of the output .txt file. If `NULL` then the folder name is used instead.
`pb`	Logical argument to control progress bar. Default is `TRUE`.

Details

The function saves a selection file in '.txt' format (that can be directly opened in Raven) that will display all sound files in the provided directory (argument 'path'). Useful to simplify the making of selections from several sound files that need to be displayed simultaneously (e.g. several recordings from the same individual). The selection file is saved in the provided directory ('path').

Value

The function saves a selection table in '.txt' format that can be directly opened in Raven. No objects are returned in the R environment.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

exp_raven

Examples

```
# Load data
library(warbleR)
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))

# Export a single selection table including multiple files
writeWave(Phae.long1, "Phae.long1.wav", extensible = FALSE) #save sound files
writeWave(Phae.long2, "Phae.long2.wav", extensible = FALSE)
writeWave(Phae.long3, "Phae.long3.wav", extensible = FALSE)
writeWave(Phae.long4, "Phae.long4.wav", extensible = FALSE)

# export with no file name
exp_empty_sels()

# export with file name
exp_empty_sels(file.name = "Phaethornis.longirostris")
```

---

exp_est *Export wave objects of extended selection tables as sound files*

Description

exp_est exports wave objects of an extended selection table as sound files

Usage

```
exp_est(X, file.name = NULL, path = NULL, single.file = FALSE,
selection.table = TRUE, pb = TRUE, normalize = TRUE, parallel = 1)
```

Arguments

X	object of class 'extended_selection_table' (objects produced by selection_table). More details about these objects can be found on this link.
file.name	character string indicating the name of the sound file (if single.file = TRUE) and/or the selection table (if selection.table = TRUE). Default is NULL.
path	A character string indicating the path of the directory where sound files and/or selection table will be saved. If not provided the function uses the current working directory. Default is NULL.
single.file	Logical argument to control if all wave objects are pooled together in a single sound file (if TRUE) or each one as an individual sound file (default). If exporting a single sound file the files are pasted in the same sequences as in the extended selection table.

`selection.table`
: Logical argument to determine if a Raven sound selection table ('.txt' file) is also exported. Default is `TRUE`. If `FALSE` then selection table is return as an object in the R environment. If exporting multiple sound files (if `single.file = FALSE`) the function stil exports a single selection table (in this case a multiple sound selection table).

`pb`
: Logical argument to control progress bar when exporting multiple sound files. Default is `TRUE`.

`normalize`
: Logical argument to control if wave objects are individually normalized before exporting (or before being pasted together if `single.file = TRUE`). Normalization rescales amplitude values to a 16 bit dynamic range. 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).

Details

The function takes wave objects contained as attributes in extended selection tables and saves them as sound files in '.wav' format. A single or several sound files can be produced (see 'single.file' argument). In addition, a Raven sound selection table can be saved along with the sound files. The exported selection table can be open in Raven for exploring/manipulating selections in 'X'.

Value

Sound file(s) are saved in the provided path or current working directory. If `selection.table = TRUE` a Raven sound selection table with the data in 'X' will also be saved.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

exp_raven;

Examples

```
{
# First set temporary folder
# setwd(tempdir())

# load example data
data(list = "Phae.long.est", package = "NatureSounds")

# subset to 10 selections
X <- Phae.long.est[1:10, ]

# Export data to a single sound file
exp_est(X, file.name = "test", single.file = TRUE)

# Export data to a single sound file and normalizing, no pb
```

```
exp_est(X, file.name = "test2", single.file = TRUE, normalize = TRUE, pb = FALSE)

# several files
exp_est(X, single.file = FALSE, file.name = "test3")
}
```

---

exp_raven *Export 'Raven' selections*

Description

exp_raven exports selection tables as 'Raven' selection data in .txt format.

Usage

```
exp_raven(X, path = NULL, file.name = NULL, khz.to.hz = TRUE,
sound.file.path = NULL, single.file = TRUE, parallel = 1, pb = TRUE)
```

Arguments

X	Object of class data frame or selection_table containing columns for sound file (sound.files), selection (selec), start and end time of signals ('start' and 'end') and low and high frequency ('bottom.freq' and 'top.freq', optional). See example data 'selec_table' in the warbleR) package.
path	A character string indicating the path of the directory in which to save the selection files. If not provided (default) the function saves the file into the current working directory.
file.name	Name of the output .txt file. If NULL then the sound file names are used instead. If multiple selection files are generated (see 'single.file') then the sound files names are added to the provided 'file.name'.
khz.to.hz	Logical. Controls if frequency variables should be converted from kHz (the unit used by other bioacoustic analysis R packages like warbleR) to Hz (the unit used by Raven). Default is TRUE.
sound.file.path	A character string indicating the path of the directory containing the sound file(s). Providing this information allows to open both sound file and selection table simultaneously. This can be done by using the 'File > Open selection table' option in 'Raven' (or drag/drop the selection file into Raven). Default is NULL. This argument is required when exporting selections from multiple sound files.
single.file	Logical. Controls whether a single selection file (TRUE; default) or a selection file for each sound file (FALSE, hence, only applicable when several sound files are included in 'X') are generated. Note that 'sound.file.path' must be provided when exporting several sound files into a single selection file as the duration of the sound files is required.

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.

Details

The function exports selection tables (as the ones used in the R package warbleR) into the 'Raven' selection file format ('.txt'). This can be useful to obtain additional Raven measurements on existing selections by adding new measurements to the selection table once in Raven. Note that selection labels must be numeric and unduplicated when exporting them to Raven. If that is not the case the function will relabeled the selections and the previous selection labels will be retained in a new column ('old.selec').

Value

The function saves a selection table in '.txt' format that can be directly opened in Raven. If several sound files are available users can either export them as a single selection file or as multiple selection files (one for each sound file). No objects are returned in the R environment.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_raven; imp_syrinx

Examples

```
# Load data
library(warbleR)
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))

# Select data for a single sound file
st1 <- selec.table[selec.table$sound.files == "Phae.long1.wav", ]

# Export data of a single sound file
exp_raven(st1, file.name = "Phaethornis 1")

# Export a single selection table including multiple files
writeWave(Phae.long1, "Phae.long1.wav", extensible = FALSE) #save sound files
writeWave(Phae.long2, "Phae.long2.wav", extensible = FALSE)
writeWave(Phae.long3, "Phae.long3.wav", extensible = FALSE)
writeWave(Phae.long4, "Phae.long4.wav", extensible = FALSE)

exp_raven(X = selec.table, file.name = "Phaethornis multiple sound files",
 single.file = TRUE, sound.file.path = getwd())
```

extract_ts *Extract time series parameters from data imported from 'Raven'*

Description

extract_ts extracts time series parameters from data imported from 'Raven' bioacoustic software.

Usage

```
extract_ts(X, ts.column, equal.length = FALSE, as.time.series = FALSE,
length.out = 30, parallel = 1, pb = TRUE)
```

Arguments

X	Data frame imported from Raven. It should include at least columns for: sound file names, selection labels, a parameters encoded as a time series (e.g. several numbers separated by semicolon)
ts.column	Name of the column with the time series data to be extracted. Default is NULL.
equal.length	Logical. Controls whether time series are kept as in the original data (most of the time with unequal lengths) or numbers are interpolated to equalize series length (using the approx function). All series will be interpolated to match the length of the longest series in the data. Default is FALSE.
as.time.series	Logical. Controls if data is converted to the time series format (using the as.ts function). Default is FALSE.
length.out	A numeric vector of length 1 giving the number of measurements to be interpolated (the length of the time series). default is 30. Ignored if equal.length 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).
pb	Logical argument to control progress bar. Default is TRUE.

Details

The function extracts parameters enconded as time series in 'Raven' selection files. The resulting data frame can be directly input into functions for time series analysis of acoustic signals as dfDTW.

Value

A data frame with columns for sound file name (sound.files), selection label (selec) and the time series for each selection.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_raven; exp_raven

Examples

```
# Load data
data(selection_files)

#save 'Raven' selection tables in the temporary directory
writeLines(selection_files[[5]], con = names(selection_files)[5])


# import data to R
rvn.dat <- imp_raven(all.data = TRUE)

# Peak freq dif length
extract_ts(X = rvn.dat, ts.column = "Peak.Freq.Contour..Hz.")

# Peak freq equal length
extract_ts(X = rvn.dat, ts.column = "Peak.Freq.Contour..Hz.", equal.length = T)

# Peak freq equal length 10 measurements
extract_ts(X = rvn.dat, ts.column = "Peak.Freq.Contour..Hz.",
equal.length = TRUE, length.out = 10)
```

---

fix_path *Modify sound file path in Raven's selection tables*

Description

fix_path modifies the path column in selection tables and sound selection tables

Usage

```
fix_path(path = NULL, dest.path = NULL, recursive = FALSE, parallel = 1, pb = TRUE,
new.begin.path, sound.file.col)
```

Arguments

path	A character string indicating the path of the directory in which to look for the 'Raven' selection (text) files. If not provided (default) the function searches into the current working directory.
dest.path	A character string indicating the path of the directory in which sound selection tables will be saved. If not provided (default) files will be save in the current directory.

recursive	Logical. If TRUE the listing recurse into sub-directories.
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.
new.begin.path	A character string indicating the path of the directory where sound files would be located. This argument is required.
sound.file.col	A character string with the name of the column containing the sound file names in the selection text files. Required.

Details

The function modifiies the path field in Raven's selection tables or sound selection tables. This is useful when sound files have been moved to a different location (or computer). Note the abililty to open selections and sound files simultaneously works as long as the "begin.path" column is refering to the directory containing the sound files.

Value

Selection table file(s) saved in 'dest.path' or in the working directory (by default, which overwrites existing files).

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

to_sound_selection; imp_raven

Examples

```
{

# load warbleR for sound file examples
library(warbleR)

#load data
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selection_files"))

# set temporary directory
# setwd(tempdir())

# save sound files
writeWave(Phae.long1,"Phae.long1.wav")
writeWave(Phae.long2,"Phae.long2.wav")
writeWave(Phae.long3,"Phae.long3.wav")
writeWave(Phae.long4,"Phae.long4.wav")
# save 'Raven' selection tables in the temporary directory
out <- lapply(1:2, function(x)
writeLines(selection_files[[x]], con = names(selection_files)[x]))
```

```
# try drag and drop selection files into Raven (shouldn't work)

# now fix files
fix_path(path = getwd(),
sound.file.col = "Begin File", new.begin.path = "YOUR NEW LOCATION HERE")

# try drag and drop into Raven again (should work now)
}
```

`imp_corr_mat` *Import 'Raven' batch correlator output*

Description

`imp_corr_mat` imports the output of 'Raven' batch correlator.

Usage

```
imp_corr_mat(file, path = NULL)
```

Arguments

`file`	A character string with the name of the output '.txt' file generated by Raven.
`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.

Details

The function imports the output of a batch correlation routine in Raven. Both the correlation and lag matrices contained in the output ' .txt' file are read and both waveform and spectrogram (cross-correlation) correlations can be imported.

Value

A list with 2 matrices. The first one contains the correlation coefficients and the second one the time lags of the peak correlations.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

`imp_raven`; `exp_raven`

Examples

```
## Not run:
# Load data
library(warbleR)
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4", "selec.table"))

writeWave(Phae.long1, "Phae.long1.wav", extensible = FALSE) #save sound files
writeWave(Phae.long2, "Phae.long2.wav", extensible = FALSE)
writeWave(Phae.long3, "Phae.long3.wav", extensible = FALSE)
writeWave(Phae.long4, "Phae.long4.wav", extensible = FALSE)

#create new folder to put cuts
dir.create("cuts")

# cut files
cut_sels(X = selec.table, mar = 0.05, path = tempdir(), dest.path = file.path(tempdir(), "cuts"))

#Now run 'Raven' batch correlator un the cuts and save the output in the same folder

# Import output (change the name of the file if you used a different one)
xcorr.rav <- imp_corr_mat(file = "BatchCorrOutput.txt",
path = file.path(tempdir(), "cuts"))

# check results

## correlation matrix
xcorr.rav[[1]]

## time lag matrix
xcorr.rav[[2]]

## End(Not run)
```

`imp_raven` *Import 'Raven' selections*

Description

`imp_raven` imports several 'Raven' selection files simultaneously. Files must be in '.txt' format.

Usage

```
imp_raven(path = NULL, warbler.format = FALSE,  all.data = FALSE, files = NULL,
only.spectro.view = TRUE, recursive = FALSE, name.from.file = FALSE,
ext.case = NULL, freq.cols = TRUE, waveform = FALSE, parallel = 1, pb = TRUE,
unread = FALSE, rm.dup = FALSE,  sound.file.col = NULL)
```

Arguments

path	A character string indicating the path of the directory in which to look for the 'Raven' selection (text) files. If not provided (default) the function searches into the current working directory.
warbler.format	Logical. If TRUE columns are renamed using the standard names for a selection table as in the package 'warbleR', frequency limit columns (high and low frequency) in 'Hz' are converted to 'kHz' (as in warbleR selection tables) and only the spectrogram view measurements are kept. Default is FALSE.
all.data	Logical. If TRUE all columns in the selection files are returned, keeping the name columns as in the 'Raven' files. Default is FALSE. Columns absent in some selection files will be filled with NA's. This argument WILL BE DEPRECATED as it is being replaced by 'warbler.format'.
files	Character vector indicating the name of selection files (in .txt format) to be imported. Optional. Default is NULL.
only.spectro.view	Logical. If TRUE (default) only the mesaurements in the Raven spectrogram view ('View' column) are returned. Ignored if warbler.format == TRUE (only spectrogram view measurements are kept).
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.
waveform	Logical to control if waveform view data should be included (this data is typically duplicated in spectrogram view data). Default is FALSE (not to include it). This argument WILL BE DEPRECATED as it is being replaced by 'only.spectro.view'.
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.
unread	Logical. If TRUE a list (instead of a data frame). The first element of the list contains the selections whole data. The second and third elements are character vectors with the names of sound files that could not be read or that contain multiple sound files but no 'File Offset' column and could not be imported. Default is FALSE.
rm.dup	Logical. If TRUE duplicated rows and columns are removed. Usefull when selection files have been duplicated. Default is FALSE.

`sound.file.col` A character string with the name of the column containing the sound files in the selection text files. Default is `NULL`. This argument WILL BE DEPRECATED as the function now searches for columns containing the sound file names.

Details

The function import 'Raven' selection data from many files simultaneously. All selection files in the working directory or 'path' supplied will be imported (unless 'files' argument is also supplied). It has been created using Raven Pro 1.5 so selection tables created with other versions might not be read properly. Files must be in '.txt' format. Selection files including data from mulitple recordings can also be imported.

Value

A single data frame with information of the selection files. If `unread = TRUE` the function returns a list of length 3 with the selection data frame and a vector with the names of files that could not be read (see 'unread' argument). If 'warbler.format' argument is set to `TRUE` the data frame contains the following columns: sound.files, selec, channel,start, end, top.freq, bottom.freq and selec.file. If all.data is set to `TRUE` then all columns in the 'Raven' selection files are returned. If individual selection files contain information about multiple sound files the function will import the file and correct the time parameters (start and end) only if 1) the 'File Offset (s)' is found in the selection table.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_syrinx

Examples

```
#load data
data(selection_files)

# set temporary directory
# setwd(tempdir())

#save 'Raven' selection tables in the temporary directory
out <- lapply(1:2, function(x)
writeLines(selection_files[[x]], con = names(selection_files)[x]))


#providing the name of the column with the sound file names
rvn.dat <- imp_raven(sound.file.col = "Begin.File", all.data = FALSE)

# View(rvn.dat)
```

imp_syrinx *Import 'Syrinx' selections*

Description

imp_syrinx imports 'Syrinx' selection data from many files simultaneously. All files must have the same columns.

Usage

```
imp_syrinx(path = NULL, all.data = FALSE, recursive = FALSE,
exclude = FALSE, hz.to.khz = TRUE, parallel = 1, pb = 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.
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.
hz.to.khz	Logical. Controls if frequency variables should be converted from Hz (the unit used by Syrinx) to kHz (the unit used by warbleR and other bioacoustic analyssis packages in R). Default if TRUE. Ignored if all.data 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.

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. If all.data is set to TRUE then all columns in selection files are returned.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_raven

Examples

```
## Not run:
#load data
data(selection_files)

#save 'Raven' selection tables in the temporary directory
writeLines(selection_files[[7]], con = names(selection_files)[7])

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)
```

`match_wav_case` *Fix the extension case of sound files*

Description

`match_wav_case` fixes the extension case of sound files in a selection table.

Usage

```
match_wav_case(X, path = NULL, output = "data.frame", verbose = TRUE)
```

Arguments

`X`	Data frame containing columns for sound file (sound.files) and selection (selec). See example data 'selec.table' in the `warbleR`) package.
`path`	A character string indicating the path of the directory in which to look for sound files. If not provided (default) the function searches into the current working directory.
`output`	Character string. Controls whether a complete data frame ('data.frame') or only the sound file names ("names") are returned. Default is 'data.frame'.
`verbose`	Logical to control if messages are printed (`TRUE`, default).

Details

The function returns the data from the input data frame with extension file names in the 'sound.files' column matching those of the sound files (in case there was any mismatch). The function needs the path to the sound files to compare extension names.

Value

The same data as in the input data frame but with the case of the extension file names in the 'sound.files' column matching those of the sound files themselves.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

relabel_colms

Examples

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

writeWave(Phae.long1, "Phae.long1.wav", extensible = FALSE) #save sound files
writeWave(Phae.long2, "Phae.long2.wav", extensible = FALSE)
writeWave(Phae.long3, "Phae.long3.wav", extensible = FALSE)
writeWave(Phae.long4, "Phae.long4.wav", extensible = FALSE)

# change one extension
selec.table$sound.files <- as.character(selec.table$sound.files)
selec.table$sound.files[1] <- gsub("\\.wav$", ".WAV", selec.table$sound.files[1])

 # fixed extension an return data frame
 match_wav_case(X = selec.table)

 # fixed extension an return sound file names
 match_wav_case(X = selec.table, output = "names")
```

---

raven_batch_detec *Run 'Raven' batch detector*

Description

raven_batch_detec Runs 'Raven' batch detector on multiple sound files sequentially

Usage

```
raven_batch_detec(raven.path = NULL, sound.files, path = NULL,
detector.type, detector.preset = "Default",
view.preset = "Default", relabel_colms = TRUE, pb = TRUE)
```

Arguments

raven.path	A character string indicating the path of the directory in which to look for the 'Raven' executable file (where 'Raven' was installed).
sound.files	character vector indicating the files that will be analyzed. In OSX (mac) only one file at the time can be run (use loops instead!). If NULL (default) then 'Raven' will be run without opening any file.
path	A character string indicating the path of the directory in which to look for the sound files. If not provided (default) the function searches into the current working directory. Default is NULL.
detector.type	Character string specifying the type of detector to be called. There are 3 options available in 'Raven': 'Amplitude Detector', 'Band Limited Energy Detector' and 'Band Limited Entropy Detector'. Must be provided.
detector.preset	Character string specifying the name of the customized detector to be called. If NULL (default) then the 'Default' detector for the specific detector type is used (see 'detector.type' argument). Custom detectors must be found in one of the default 'Raven' detector directories (usually within "'raven.path'/Presets/Detector").
view.preset	Character string specifying the name of the window preset to be used. Not require for 'Amplitude Detector' (see 'detector.type' argument). If NULL (default) then the 'Default' window preset is used.
relabel_colms	Logical. If TRUE (default) colums are labeled to match the selection table format from the acoustic analysis package warbleR
pb	Logical argument to control progress bar. Default is TRUE.

Details

The function runs 'Raven' sound analysis software (Cornell Lab of Ornithology), detector on multiple sound files sequentially. 'Raven' Pro must be installed. Note that batch detection in 'Raven' can also take sound files in 'mp3', 'flac' and 'aif' format.

Value

A data frame with the selections produced during the detection. See imp_raven for more details on how selections are imported.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_raven; imp_syrinx; run_raven

Examples

```
## Not run:

# here replace with the path where 'Raven' is install in your computer
raven.path <- "PATH_TO_RAVEN_DIRECTORY_HERE"

# Run detector on raven example sound files

# single sound file using 'Amplitude Detector'
detec.res <- raven_batch_detec(raven.path = raven.path,
sound.files = "BlackCappedVireo.aif", path = file.path(raven.path, "Examples"),
detector.type = "Amplitude Detector")

# on raven examples  2 files
detec.res <- raven_batch_detec(raven.path = raven.path,
sound.files = c("BlackCappedVireo.aif", "CanyonWren.wav"),
path = file.path(raven.path, "Examples"), detector.type = "Amplitude Detector")

# using 'Band Limited Energy Detector'
detec.res <- raven_batch_detec(raven.path = raven.path,
sound.files = c("BlackCappedVireo.aif", "CanyonWren.wav"),
path = file.path(raven.path, "Examples"), detector = "Band Limited Energy Detector")

## End(Not run)
```

`relabel_colms` *Relabel columns to match the selection table format*

Description

`relabel_colms` relabels columns to match the selection table format (as in the R package `warbleR`)

Usage

```
relabel_colms(X, extra.cols.name = NULL, extra.cols.new.name = NULL,
khz.to.hz = FALSE, hz.to.khz = FALSE, waveform = FALSE)
```

Arguments

`X`	Data frame imported from Raven.
`extra.cols.name`	Character vector with the names of additional columns to be relabeled. Default is `NULL`. 'extra.cols.new.name' must also be provided.
`extra.cols.new.name`	Character vector with the new names for the additional columns to be relabeled. Default is `NULL`. 'extra.cols.name' must also be provided.

khz.to.hz	Logical. Controls if frequency variables ('top.freq' and 'bottom.freq') should be converted from kHz (the unit used by other bioacoustic analysis R packages like warbleR) to Hz (the unit used by Raven). Default is FALSE.
hz.to.khz	Logical. Controls if frequency variables ('top.freq' and 'bottom.freq') should be converted from Hz (the unit used by other bioacoustic analysis R packages like Raven) to kHz (the unit used by warbleR). Default is FALSE. Ignored if 'kHz.to.hz' is TRUE.
waveform	Logical to control if 'waveform' related data should be included (this data is typically duplicated in 'spectrogram' data). Default is FALSE (not to include it).

Details

This function relabels columns to match the selection table format to match then ones used by other bioacoustic analysis R packages like warbleR.

Value

The function returns the input data frame with new column names for time and frequency 'coordinates' and sound files and selections.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_raven; exp_raven

Examples

```
# Load data
data(selection_files)

#save 'Raven' selection tables in the temporary directory
writeLines(selection_files[[5]], con = names(selection_files)[5])


#'# import data to R
rvn.dat <- imp_raven(all.data = TRUE)

names(rvn.dat)

# Select data for a single sound file
rvn.dat2 <-  relabel_colms(rvn.dat)

names(rvn.dat2)

# plus 1 additional column
rvn.dat2 <-  relabel_colms(rvn.dat, extra.cols.name = "selec.file", "Raven selection file")
```

```
names(rvn.dat2)

# plus 2 additional column
rvn.dat2 <- relabel_colms(rvn.dat, extra.cols.name = c("selec.file", "View"),
c("Raven selection file", "Raven view"))

names(rvn.dat2)
```

Rraven *Rraven: Exchange data and open sound files in 'Raven' from R*

Description

Rraven is a package designed to facilitate the exchange of data between R and 'Raven' sound analysis software (Cornell Lab of Ornithology)

Details

License: GPL (>= 2)

@section Functions:

exp_raven: Export R selection tables into 'Raven' selection file format

extract_ts: Extract time series parameters from data imported from Raven

imp_corr_mat: Import 'Raven' batch correlator output

imp_raven: Importing 'Raven' selections

imp_syrinx: Importing 'Syrinx' selections

match_wav_case: Fix the extension case of sound files

raven_batch_detec: Run 'Raven' batch detector

relabel_colms: Relabel columns to match the selection table format

run_raven: Open sound files in Raven

to_sound_selection: Convert Raven's selection files into sound selection files

Author(s)

Marcelo Araya-Salas

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

run_raven *Open sound files in 'Raven' sound analysis software*

Description

run_raven opens several sound files in 'Raven' sound analysis software

Usage

```
run_raven(raven.path = NULL, sound.files = NULL, path = NULL, at.the.time = 10,
import = FALSE, redo = FALSE, view.preset = NULL, pb = TRUE, ...)
```

Arguments

raven.path	A character string indicating the path of the directory in which to look for the 'Raven' executable file (where 'Raven' was installed).
sound.files	character vector indicating the files that will be analyzed. If NULL (default) then 'Raven' will be run without opening any file.
path	A character string indicating the path of the directory in which to look for the sound files. If not provided (default) the function searches into the current working directory. Default is NULL.
at.the.time	Numeric vector of length 1 controling how many files will be open in 'Raven' at the same time. Note that opening too many files at once could make 'Raven' run out of memory. You need to close 'Raven' every time the batch of files is analyzed, so the next batch is opened. Default is 10. Not available in OSX (mac).
import	Logical. Controls if the selection tables generated should be returned as a data frame into the R environment. This only works if the selections are saved in the "Selections" folder in the 'Raven' directory. This argument calls the imp_raven internally. Additional arguments can be passed to imp_raven to control the way the data is imported.
redo	Logical. Controls whether only the subset of files with no 'Raven' selections (.txt file) in the 'Raven' 'selections' folder are analyzed (if FALSE). Useful when resuming the analysis. Default is FALSE.
view.preset	Character string defining the 'Raven' view preset to be used. It should match exactly the name of the present in the 'Raven' folder 'Presets/Sound Window'. If not provided the default view preset is used.
pb	Logical argument to control progress bar. Default is TRUE.
...	Additional arguments to be passed to imp_raven for customizing how selections are imported (ignored if import = FALSE).

Details

The function runs 'Raven' sound analysis software (Cornell Lab of Ornithology), opening many files simultaneously. 'Raven' will still run if no sound files are provided (i.e. `sound.files = NULL`). At the end of the analysis the data can be automatically imported back into R using the 'import' argument. 'Raven' Pro must be installed. Note that 'Raven' can also take sound files in 'mp3', 'flac' and 'aif' format.

Value

If `import = TRUE` a data frame with the selections produced during the analysis will be return as an data frame. See `imp_raven` for more details on how selections are imported.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

`imp_raven`; `imp_syrinx`

Examples

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

# save sound files
library(warbleR)
data(list = c("Phae.long1", "Phae.long2", "Phae.long3", "Phae.long4"))
writeWave(Phae.long1, "Phae.long1.wav", extensible = FALSE)
writeWave(Phae.long2, "Phae.long2.wav", extensible = FALSE)

# here replace with the path where 'Raven' is install in your computer
raven.path <- "PATH_TO_RAVEN_DIRECTORY_HERE"

# run function
run_raven(raven.path = raven.path, sound.files = c("Phae.long1.wav", "Phae.long2.wav"),
 at.the.time = 2, import = T, name.from.file = T, ext.case = "upper",
 all.data = TRUE, path = tempdir())

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

writeWave(Phae.long3, "Phae.long3.wav", extensible = FALSE)
writeWave(Phae.long4, "Phae.long4.wav", extensible = FALSE)

# run function on all the wav files in the working directory 3 at the time
run_raven(raven.path = raven.path, sound.files = list.files(pattern = "\\.wav$",
ignore.case = TRUE, path = tempdir()), at.the.time = 3, import = FALSE,
path = tempdir())
```

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

`selection_files` *A list of 'Raven' selection tables.*

Description

List of 'Raven' and 'Syrinx' selection tables containing *Phaethornis longirostris* (Long-billed Hermit) songs from the example sound files included in the `warbleR` package (times 1 to 5), a 'Raven' correlation matrix (item 6) and one selection from cane-brake wrens (item 7; no sound file available).

Usage

```
data(selection_files)
```

Format

List of 7 selection tables. The first 5 items are 'Raven' selection tables of Long billed hermit songs. The 5th table contains a multiple sound file selection table. The 6th item is a 'Raven' correlation matrix and the last one a table in 'Syrinx' format.

Source

Marcelo Araya-Salas

`to_sound_selection` *Convert Raven's selection files into sound selection files*

Description

`to_sound_selection` converts Raven's selection files into sound selection files

Usage

```
to_sound_selection(path = NULL, dest.path = NULL, recursive = FALSE,
 parallel = 1, pb = TRUE, sound.file.path, sound.file.col)
```

Arguments

path	A character string indicating the path of the directory in which to look for the 'Raven' selection (text) files. If not provided (default) the function searches into the current working directory.
dest.path	A character string indicating the path of the directory in which sound selection tables will be saved. If not provided (default) files will be save in the current directory.
recursive	Logical. If TRUE the listing recurse into sub-directories.
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.
sound.file.path	A character string indicating the path of the directory containing the sound file(s). This argument is required.
sound.file.col	A character string with the name of the column containing the sound file names in the selection text files. Required.

Details

The function converts Raven's selection tables to sound selection tables. Sound selection table is a more convenient format as it can be open directly in Raven (or drag-and-drop) and will automatically open the associated sound file. Multiple files can be simultaneously converted. Files must be in '.txt' format. Selection files including data from mulitple recordings can be converted only if all the correspondent sound files are found in the same directory. Note that no data is imported into the R environment.

Value

Sound selection table file(s) saved in 'dest.path' or in the working directory.

Author(s)

Marcelo Araya-Salas (<marceloa27@gmail.com>)

See Also

imp_syrinx; imp_raven

Examples

```
{

#load data
data(selection_files)

# set temporary directory
# setwd(tempdir())
```

```
# save 'Raven' selection tables in the temporary directory
out <- lapply(1:2, function(x)
writeLines(selection_files[[x]], con = names(selection_files)[x]))

# try drag and drop selection files into Raven (shouldn't work)

# now convert files
to_sound_selection(sound.file.path = getwd(),
sound.file.col = "Begin Path")

# try drag and drop into Raven again (should work now)
}
```

Index