

Package: crputils (via r-universe)

October 9, 2026

Title Miscellaneous R Utilities Useful to CRP

Version 0.3.0053

Date 2026-09-09

Maintainer Selene Fregosi <selene.fregosi@noaa.gov>

Description A collection of miscellaneous utilities that are useful for various research activities conducted by the Cetacean Research Program (CRP) at NOAA NMFS Pacific Islands Fisheries Science Center. This includes utilities for working with latitude and longitude data, gpx file creation, and more to come.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports utils, lubridate, dplyr, stringr, swfscDAS, googledrive, PAMpal, PamBinaries

Remotes SWFSC/swfscDAS

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libabsl-dev cmake libfftw3-dev libgdal-dev gdal-bin libgeos-dev libglpk-dev make libicu-dev libpng-dev libsndfile1-dev libuv1-dev libxml2-dev libnetcdf-dev libssl-dev libproj-dev python3 libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

Repository <https://pifsc-protected-species-division.r-universe.dev>

Date/Publication 2026-09-09 18:20:03 UTC

RemoteUrl <https://github.com/PIFSC-Protected-Species-Division/crputils>

RemoteRef HEAD

RemoteSha 772be6671dfb1a03ccb734bed59c6a8b5d69ff14

Contents

CSVToGPX_track	2
DASTrackToGPX	3
decdeg2degmin	4
decdeg2degminsec	4
degmin2decdeg	5
degmin2degminsec	6
degminsec2decdeg	7
degminsec2degmin	8
downloadGoogleDriveFolder	9
er_clickSummary	9
er_concatenatedSlope	10
er_filterClicks	11
er_plotClickDurationHist	12
er_plotClickSNRHist	13
er_plotConcatSpec	13
er_plotSpectra	15
er_plotWhistleBeginFreqHist	16
er_plotWhistleContours	16
er_plotWhistleMedFreqHist	17
er_whistleSummary	17
extractDASTrack	18
flac2wav	18
installSavedPackages	19
lldistkm	20
loadMultiBinaries	21
saveInstalledPackages	22
wav2flac	23
Index	25

CSVToGPX_track	<i>CSVToGPX_track</i>
----------------	-----------------------

Description

convert a .csv of vessel track segments to a .gpx file.

The input data.frame must have 1 column 'uid' containing a unique identifier for that track/segment, 2 columns for start latitude and longitude, and 2 columns for stop latitude and longitude. Additional columns for cruise, date, vessel name, etc will be ignored

This is a modified version of trackToGPX.R that is included in the 'cruise-maps-live' repository that tries to generalize the function to convert any input data.frame, not just the output from a .das file.

Usage

```
CSVToGPX_track(inCSV, outGPX)
```

Arguments

inCSV	filename of csv containing track data to be processed. Must include the following columns: uid, startLat, startLon, stopLat, stopLon where uid is a unique track identifier, startLat/startLon is the track start location and stopLat/stopLon is the track end location. All locations should be in decimal degrees. Can optionally include a startDateTime and stopDateTime column. Assumes UTC timezone for GPX output.
outGPX	fullpath filename of gpx to be written example: outGPX <- './crputils/exampleData/exampleVessel'

Value

none, will write a file

Author(s)

Selene Fregosi

DASTrackToGPX	<i>DASTrackToGPX</i>
---------------	----------------------

Description

create a .gpx file based on the effort track data recorded in .DAS files. The track data needs to have been extracted from the .DAS file using `extractDASTrack()` which creates the 'et' dataframe

Track segments within days are plotted separately but are connected. Tracks across days are not connected.

This was generalized from the cruise-maps-live repository's `trackToGPX()`

Usage

```
DASTrackToGPX(et, outGPX)
```

Arguments

et	data.frame of effort as tracks, can be 'et' cumulative over a survey or 'et' for just a single DAS
outGPX	fullpath filename to save example: outGPX <- paste0('newEffortTracks_', dasName, '_', Sys.Date(), '.gpx')

Value

none, will write a file

Author(s)

Selene Fregosi

decdeg2degmin	<i>decdeg2degmin</i>
---------------	----------------------

Description

Utility to convert latitude and longitude coordinates from decimal degrees to degrees decimal minutes

Usage

```
decdeg2degmin(decdeg)
```

Arguments

decdeg N-by-1 vector of coordinates in decimal degrees e.g., decdeg <- 30.48667 OR
decdeg <- c(30.48667, -155.61496)

Value

N-by-2 data.frame of coordinates with columns 'deg' for degrees and 'min' for decimal minutes

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
decdeg <- 30.48667
degmin <- decdeg2degmin(decdeg)

# multiple input coordinates
decdeg <- c(30.48667, -155.61496)
degmin <- decdeg2degmin(decdeg)
```

decdeg2degminsec	<i>decdeg2degminsec</i>
------------------	-------------------------

Description

Utility to convert latitude and longitude coordinates from degrees minutes to degrees minutes second

Usage

```
decdeg2degminsec(decdeg)
```

Arguments

decdeg N-by-1 vector of coordinates in decimal degrees e.g., decdeg <- 30.48667 OR
decdeg <- c(30.48667, -155.61496)

Value

N-by-3 data.frame of coordinates with columns 'deg' for degrees, 'min' for minutes, and 'sec' for decimal seconds

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
decdeg <- 30.48667
degminsec <- decdeg2degminsec(decdeg)

# multiple input coordinates
decdeg <- c(30.48667, -155.61496)
degminsec <- decdeg2degminsec(decdeg)
```

degmin2decdeg	<i>Degrees decimal minutes to decimal degrees</i>
---------------	---

Description

Utility to convert latitude and longitude coordinates from degrees minutes to decimal degrees

Usage

```
degmin2decdeg(degmin)
```

Arguments

degmin N-by-2 matrix of coordinates with each column representing degrees and minutes. e.g., matrix(c(30, 29.2020), nrow = 1, ncol = 2)
e.g., matrix(c(30, 29.2020, -155, 36.8973), nrow = 2, ncol = 2, byrow = TRUE)

Value

N-by-1 vector of coordinates in decimal degrees

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
degmin <- matrix(c(30, 29.2020), nrow = 1, ncol = 2)
decdeg <- degmin2decdeg(degmin)

# multiple input coordinates
degmin <- matrix(c(30, 29.2020,
                  -155, 36.8973),
                nrow = 2, ncol = 2, byrow = TRUE)
decdeg <- degmin2decdeg(degmin)
```

degmin2degminsec	<i>degmin2degminsec</i>
------------------	-------------------------

Description

Utility to convert latitude and longitude coordinates from degrees minutes to degrees minutes seconds

Usage

```
degmin2degminsec(degmin)
```

Arguments

degmin	N-by-2 matrix of coordinates with each column representing degrees and minutes e.g., <code>matrix(c(30, 29.2020), nrow = 1, ncol = 2)</code> e.g., <code>matrix(c(30, 29.2020, -155, 36.8973), nrow = 2, ncol = 2, byrow = TRUE)</code>
--------	--

Value

N-by-3 data.frame of coordinates with columns 'deg' for degrees and 'min' for minutes and 'sec' for decimal seconds

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
degmin <- matrix(c(30, 29.2020), nrow = 1, ncol = 2)
degminsec <- degmin2degminsec(degmin)

# multiple input coordinates
degmin <- matrix(c(30, 29.2020,
                  -155, 36.8973),
```

```
      nrow = 2, ncol = 2, byrow = TRUE)
degminsec <- degmin2degminsec(degmin)
```

degminsec2decdeg	<i>degminsec2decdeg</i>
------------------	-------------------------

Description

Utility to convert latitude and longitude coordinates from degrees minutes seconds to decimal degrees

Usage

```
degminsec2decdeg(degminsec)
```

Arguments

degminsec N-by-3 matrix of coordinates with each column representing degrees, minutes, and seconds e.g., `matrix(c(30, 29, 12), nrow = 1, ncol = 3)`
e.g., `matrix(c(30, 29, 12, -155, 36, 53.838), nrow = 2, ncol = 3, byrow = TRUE)`

Value

N-by-1 vector of coordinates in decimal degrees

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
degminsec = matrix(c(30, 29, 12), nrow = 1, ncol = 3)
decdeg = degminsec2decdeg(degminsec)

# multiple input coordinates
degminsec = matrix(c(30, 29, 12,
                     -155, 36, 53.838),
                  nrow = 2, ncol = 3, byrow = TRUE)
decdeg = degminsec2decdeg(degminsec)
```

degminsec2degmin	<i>degminsec2degmin</i>
------------------	-------------------------

Description

Utility to convert latitude and longitude coordinates from degrees minutes seconds to degrees decimal minutes

Usage

```
degminsec2degmin(degminsec)
```

Arguments

degminsec N-by-3 matrix of coordinates with each column representing degrees, minutes, and seconds e.g., `matrix(c(30, 29, 12), nrow = 1, ncol = 3)`
 e.g., `matrix(c(30, 29, 12, -155, 36, 53.838), nrow = 2, ncol = 3, byrow = TRUE)`

Value

N-by-2 data.frame of coordinates with columns 'deg' for degrees and 'min' for minutes

Author(s)

Selene Fregosi

Examples

```
# single input coordinates
degminsec <- matrix(c(30, 29, 12), nrow = 1, ncol = 3)
degmin <- degminsec2degmin(degminsec)

# multiple input coordinates
degminsec <- matrix(c(30, 29, 12,
                     -155, 36, 53.838),
                   nrow = 2, ncol = 3, byrow = TRUE)
degmin <- degminsec2degmin(degminsec)
```

downloadGoogleDriveFolder
<i>downloadGoogleDriveFolder</i>

Description

recursively download all files and subdirectories in a google drive folder

Usage

downloadGoogleDriveFolder(folder_id, local_path)

Arguments

folder_id	Google Drive folder ID, parsed as the 'id' column finding the drive info using googledrive::drive_get or using using googledrive::as_id and the folder URL
local_path	full path to the folder to save the downloaded files

Author(s)

Selene Fregosi

er_clickSummary	<i>er_clickSummary</i>
-----------------	------------------------

Description

Summarize clicks for a single AcousticStudy event, to be used within the event_summary_report Rmd. Summarizes total number of clicks, runs `PAMpal::calculateAverageSpectra()`, identifies 'good clicks' as those with a pseudo-snr above input snrThreshold, calculates some event-level summary statistics (peak and center frequency, 3 and 10 dB bandwidths, click duration) and outputs the raw click data and a formatted summary table

last updated: 22 August 2026

Usage

er_clickSummary(acSt, eventUID, snrThreshold = 15)

Arguments

acSt	AcousticStudy produced with <code>PAMpal::processPgDetections()</code>
eventUID	ID of a single acoustic event
snrThreshold	threshold for "good" clicks with snr equal to or above this threshold (in dB). Default is 15

Value

'cl' list with 7 elements: total number of clicks, PAMpal click data for all clicks, PAMpal spec data for all clicks (output from `PAMpal::calculateAverageSpectra()`), the number of clicks above `snrThreshold` (default 15 dB if not specified), PAMpal click data for these 'good' clicks only, and 'mt', a formatted table of median summary statistics

See Also

`er_plotClickDurationHist()` `er_plotClickSNRHist()` `er_plotSpectra()`

Examples

```
## Not run:
cl <- er_clickSummary(acStFilt, eventUID)

## End(Not run)
```

`er_concatenatedSlope` *er_concatenatedSlope*

Description

Model the slope of peak frequencies of clicks sorted by peak peak frequency as displayed in a concatenated spectrogram using linear regression. This function models the slope in two ways (1) for all clicks and (2) for just clicks within the 25-75th percentiles of the mean.

Usage

```
er_concatenatedSlope(cl, quantLimit = NULL)
```

Arguments

<code>cl</code>	list returned by <code>clickSummary()</code> that has a elements <code>spec</code> and <code>snr</code>
<code>quantLimit</code>	optional argument to limit the clicks to a particular quantile range e.g., only clicks in the 25-75th quantiles, 5-95th, etc. Default is <code>NULL</code> (all clicks), input in format <code>c(.25, .75)</code>

Value

'cs' the linear regression model output

Examples

```
# all clicks
## Not run: cs <- concatenatedSlope(cl)
# 25-75th quantile only
## Not run: cs <- concatenatedSlope(cl, c(0.25, 0.75))
```

er_filterClicks	<i>er_filterClicks</i>
-----------------	------------------------

Description

filter all click detections to remove clicks that are unlikely to be odontocetes and are likely noise. Filter settings were tuned by Jennifer McCullough, see McCullough et al. 2021

Filter settings: Remove clicks with 10dB bandwidth < 5 kHz Remove clicks with peak frequency below 5 kHz and above 80 kHz Remove clicks with durations of less than 2 ms or greater than 1000 ms For each individual click detector (classifier), remove clicks with peak frequencies outside the specified limits (remove erroneous miss-classifications)

last updated: 31 July 2026

Usage

```
er_filterClicks(acSt)
```

Arguments

acSt	AcousticStudy object generated with processPgDetections()
------	---

Value

an AcousticStudy object containing only filtered detections

Author(s)

Selene Fregosi <selene.fregosi@noaa.gov>

Examples

```
## Not run:  
detsFiltered <- er_filterClicks(dets)  
  
## End(Not run)
```

`er_plotClickDurationHist`
er_plotClickDurationHist

Description

Plot histogram of click durations of high SNR clicks only

last updated: 21 August 2026

Usage

```
er_plotClickDurationHist(c1)
```

Arguments

`c1` list returned by `er_clickSummary()` containing elements `goodClicks` (data frame with column `duration`) and `nGoodClicks` (count)

Value

creates a plot

Author(s)

Selene Fregosi <selene.fregosi@noaa.gov>

See Also

[er_clickSummary\(\)](#) [er_plotClickSNRHist\(\)](#) [er_plotSpectra\(\)](#) [er_plotConcatSpec\(\)](#)

Examples

```
## Not run:
c1 <- er_clickSummary(acSt)
er_plotClickDurationHist(c1)

## End(Not run)
```

er_plotClickSNRHist *er_plotClickSNRHist*

Description

Plot histogram of click SNR for all clicks with a vertical dashed line at the specified cutoff

last updated: 22 August 2026

Usage

```
er_plotClickSNRHist(spec, snrThreshold = 15)
```

Arguments

spec	output of PAMpal::calculateAverageSpectra() that includes variable snr-Vals
snrThreshold	SNR in dB to plot as the threshold for 'good' clicks. Default is 15 dB # @seealso er_clickSummary() er_plotClickDurationHist() er_plotSpectra() er_plotConcatSpec()

Examples

```
## Not run:
er_plotClickSNRHist(spec, snrThreshold = 15)

## End(Not run)
```

er_plotConcatSpec *er_plotConcatSpec*

Description

Plot a concatenated spectrogram of all clicks with snr above snrThreshold, sorted by peak frequency. This plotting code has been extracted from [PAMpal::calculateAverageSpectra](#) to be run on outputs of that same function.

last updated: 21 August 2026

Usage

```
er_plotConcatSpec(
  spec,
  flim = NULL,
  snrThreshold = 15,
  sort = TRUE,
  filterfrom_khz = 0,
  filterto_khz = NULL,
  q = 0.01,
  brightness = 0,
  contrast = 0,
  cmap = hcl.colors(30, "YlOrRd", rev = TRUE),
  showBreaks = FALSE
)
```

Arguments

spec	output of <code>PAMpal::calculateAverageSpectra()</code> and includes variables freq, avgSpec, avgNoise, allSpec, and snrVals
flim	optional argument for frequency limits. Default is full frequency range of recordings. Define with format <code>c(0, 100)</code> , specify in kHz
snrThreshold	optional argument for SNR threshold (in dB) for filtering which clicks to include in plot. Default is 15
sort	logical flag to sort concatenated spectrogram by peak frequency. Default is TRUE (this is different than <code>PAMpal::calculateAverageSpectra()</code> 's default.)
filterfrom_khz	frequency in kHz of highpass filter to apply, or the lower bound of a bandpass filter if <code>filterto_khz</code> is not NULL. Default is 0.
filterto_khz	if a bandpass filter is desired, set this as the upper bound. If only a highpass filter is desired, leave as the default NULL value. Currently only highpass and bandpass filters are supported, so if <code>filterfrom_khz</code> is left as zero then this parameter will have no effect.
q	lower and upper quantiles to remove for scaling concatenated spectrogram. Or if a single value, then quantiles <code>q</code> and <code>q</code> will be used. Ex. if <code>q=0.01</code> (default), then the bottom 1% and top 1% of values are truncated before plotting the image. This is done purely for cosmetic reasons, no output data is affected
brightness	value from -255 to 255, positive values increase brightness, negative values decrease brightness of concatenated spectrogram image
contrast	value from -255 to 255, positive values increase contrast, negative values decrease contrast of concatenated spectrogram image
cmap	colors to use for concatenated click spectrogram, either a palette function or vector of colors
showBreaks	logical flag to show lines separating events when plotting multiple events. This DOES not apply for event summary reports (as they plot) event-by-event but it is included here just in case. Default is FALSE

See Also

[er_clickSummary\(\)](#) [er_plotClickDurationHist\(\)](#) [er_plotClickSNRHist\(\)](#) [er_plotSpectra\(\)](#)

Examples

```
## Not run:
cl <- er_clickSummary(acStFilt, eventUID)
# use all defaults
er_plotConcatSpec(cl$spec)
# modify plot appearance
er_plotConcatSpec(cl$spec, q = 0.05, brightness = 25, contrast = -50)

## End(Not run)
```

er_plotSpectra	<i>er_plotSpectra</i>
----------------	-----------------------

Description

Plot spectra of all clicks - mean, median, noise, and any specified reference spectra

last updated: 22 August 2026

Usage

```
er_plotSpectra(spec, refSpecs = NULL, flim = NULL, snrThreshold = 15)
```

Arguments

spec	output of PAMpal::calculateAverageSpectra() and includes variables freq, avgSpec, avgNoise, allSpec, and snrVals
refSpecs	optional argument with list of reference spectra to include as overlays in the plot
flim	optional argument for frequency limits. Default is full frequency range of recordings. Define with format <code>c(0, 100)</code> , specify in kHz
snrThreshold	optional argument for SNR threshold (in dB) for filtering clicks used in calculation of median spectrum. Default is 15

See Also

[er_clickSummary\(\)](#) [er_plotClickDurationHist\(\)](#) [er_plotClickSNRHist\(\)](#) [er_plotConcatSpec\(\)](#)

Examples

```
## Not run:
er_plotSpectra(spec, refSpecs = refSpecs, flim = c(0,100), snrThreshold = 15)

## End(Not run)
```

```
er_plotWhistleBeginFreqHist
      er_plotWhistleBeginFreqHist
```

Description

Plot histogram begin frequency of whistles

Usage

```
er_plotWhistleBeginFreqHist(wl)
```

Arguments

wl list returned by whistleSummary() that has a elements wh

Examples

```
## Not run:
er_plotWhistleBeginFreqHist(wl)

## End(Not run)
```

```
er_plotWhistleContours
      er_plotWhistleContours
```

Description

Plot all whistle contours

Usage

```
er_plotWhistleContours(dets, eventUID, xMax = 1.5, yMax = 20)
```

Arguments

dets AcousticStudy containing whistle detections
 eventUID unique ID for the event being processed (to reference within the AcousticStudy)
 xMax optional argument for upper limit of x axis, in seconds. Default is 1.5 seconds
 yMax optional argument for upper limit of y axis, in kHz. Default is 20 kHz

Examples

```
## Not run:
er_plotWhistleContours(dets, eventUID, xMax = 1.5, yMax = 20)

## End(Not run)
```

```
er_plotWhistleMedFreqHist
      er_plotWhistleMedFreqHist
```

Description

Plot histogram median frequency of whistles

Usage

```
er_plotWhistleMedFreqHist(wl)
```

Arguments

wl list returned by whistleSummary() that has a elements wh

Examples

```
## Not run:
er_plotWhistleMedFreqHist(wl)

## End(Not run)
```

```
er_whistleSummary      er_whistleSummary
```

Description

Summarize whistles for a single AcousticStudy event, to be used within the event_summary_report Rmd. Summarizes total number of whistles and calculates some summary statistics (begin and end frequency, mean frequency, duration, and frequency range) output into a nicely formatted table.

Usage

```
er_whistleSummary(asF, eventUID)
```

Arguments

asF AcousticStudy that has been filtered to remove noise
eventUID ID of a single acoustic event

Value

'wl' list with 3 elements: total number of whistles, PAMpal whistle data for each whistle, and 'mt', the formatted table of median summary statistics

Examples

```
## Not run:
w1 <- er_whistleSummary(detsFilt, eventUID)

## End(Not run)
```

extractDASTrack	<i>extractDASTrack</i>
-----------------	------------------------

Description

Pull effort tracks from a relatively raw daily .das file into a simpler dataframe. Utilizes the package `swfscDAS` and then cleans up those outputs a bit. Generalized from `cruise-maps-live`'s `extractTrack()`

Usage

```
extractDASTrack(df_proc)
```

Arguments

`df_proc` processed das file (with `swfscDAS::das_process`)

Value

a dataframe of effort tracks with date and lat/lon

Author(s)

Selene Fregosi

flac2wav	<i>flac2wav</i>
----------	-----------------

Description

Utility to decode FLAC files to WAV format using the command line `flac.exe` encoder.

This function converts a directory of .flac files to .wav and saves the new files to the specified output directory. It requires the free FLAC codec/executable which can be downloaded from: <https://xiph.org/flac/download.html>.

Usage

```
flac2wav(path_flac, inDir, outDir, numCh = 1)
```

Arguments

path_flac	Character. Path to flac executable e.g., "C:\flac-1.5.0-win\Win64\flac.exe"
inDir	Character. Path to the folder containing FLAC files
outDir	Character. Path to the folder where WAV files will be saved. <i>This path MUST end in a final slash. A check will ensure that it does.</i>
numCh	numeric. Number of channels. Default is 1

Value

None. Writes WAV files to disk.

Author(s)

Selene Fregosi

See Also

[tk_choose.dir](#) Note: This function uses `tcltk::tk_choose.dir()` which may require a working installation of Tcl/Tk. On Linux and macOS, you may need to install Tcl/Tk separately.

[wav2flac](#)

Examples

```
# single channel data
## Not run:
flac2wav('C:\\users\\user.name\\programs\\flac-1.5.0-win\\Win64\\flac.exe',
'F:\\flacFiles', 'F:\\wavFiles\\');

## End(Not run)
# multichannel data
## Not run:
flac2wav('C:\\users\\user.name\\programs\\flac-1.5.0-win\\Win64\\flac.exe',
'F:\\flacFiles', 'F:\\wavFiles\\', 4);

## End(Not run)
```

installSavedPackages *installSavedPackages*

Description

Re-install a set of saved packages (created with `saveInstalledPackages()`) after updating R

Usage

```
installSavedPackages(fileName)
```

Arguments

fileName filename of saved packages. This can be the full file path with filename or if no path is specified it will look for the file in your current working directory. You can use `getwd()` to check current working directory)

Author(s)

Yvonne Barkley and Selene Fregosi

lldistkm

Calculate distance in km between to lat/lons

Description

Utility to calculate the distance (in kilometers) between a pair of latitude/longitude coordinates. It returns two distances: d1km which is calculated using the Haversine formula which accounts for the shape of the earth and d2km based on Pythagoras' theorem and is not accurate over long distances. Over short distances d1km and d2km will be very similar.

Usage

```
lldistkm(latlon1, latlon2)
```

Arguments

latlon1 1-by-2 numeric vector of lat/lon coordinates (in decimal degrees) with latitude first, longitude second e.g., `c(21, -157)`

latlon2 1-by-2 numeric vector of lat/lon coordinates (in decimal degrees) with latitude first, longitude second e.g., `c(21, -157)`

Value

list with d1km (distance calculated by haversine formula) and d2km (distance calculated based on Pythagorean theorem)

Author(s)

Selene Fregosi

Examples

```
#short distance
#create input coordinates
l11 <- c(21, -157)
l12 <- c(22, -158)
#calculate distance
dkm <- lldistkm(l11, l12)
dkm
# $d1km
# [1] 151.8795
#
# $d2km
# [1] 151.8809

#long distance
#calculate distance
dkm <- lldistkm(c(21, -157), c(13, 144))
dkm
# $d1km
# [1] 6304.003
#
# $d2km
# [1] 32019.57
#d2km not accurate at longer distances
```

loadMultiBinaries	<i>loadMultiBinaries</i>
-------------------	--------------------------

Description

Load in multiple Pamguard binary files and compile the data lists into a single list. For now it works with WMD data...

Usage

```
loadMultiBinaries(wmBinFiles)
```

Arguments

wmBinFiles character array of whistle moan binary files to be loaded

Value

binOut a list of binary WMD data

Examples

```
## Not run:
# map the needed binary files
binFiles <- dets@events[[eventUID]]@files$binaries
wmFileIdx <- grep(pattern = '^.*WhistlesMoans_Whistle_and_Moan_Detector_Contours.*\\.pgdf$',
                  binFiles)
wmFiles <- dets@events[[eventUID]]@files$binaries[wmFileIdx]
# now load the binaries
loadMultiBinaries(wmFiles)

## End(Not run)
```

```
saveInstalledPackages  saveInstalledPackages
```

Description

Save a vector of currently installed R packages as a RDS named fileName. Run this before updating R, then update R, then run installSavedPackages to re-install all packages

Usage

```
saveInstalledPackages(fileName)
```

Arguments

fileName	filename to save to (can be full file path with name or if just a name is specified then will save in the current working directory use getwd() to check current working directory).
----------	--

Value

installedPkgs a dataframe of effort tracks with date and lat/lon

Author(s)

Yvonne Barkley

wav2flac

wav2flac

Description

Utility to convert WAV files to FLAC format using the command line flac.exe encoder.

This function converts a directory of .wav files to .flac and saves the new files to the specified output directory. It requires the free FLAC codec/executable which can be downloaded from: <https://xiph.org/flac/download.html>.

Usage

```
wav2flac(path_flac, inDir, outDir, numCh = 1)
```

Arguments

path_flac	Character. Path to flac executable e.g., "C:\flac-1.5.0-win\Win64\flac.exe"
inDir	Character. Path to the folder containing WAV files
outDir	Character. Path to the folder where FLAC files will be saved. <i>This path MUST end in a final slash. A check will ensure that it does.</i>
numCh	numeric. Number of channels. Default is 1

Value

None. Writes FLAC files to disk.

Author(s)

Selene Fregosi

See Also

[tk_choose.dir](#) Note: This function uses `tk_choose.dir()` which may require a working installation of Tcl/Tk. On Linux and macOS, you may need to install Tcl/Tk separately.

[flac2wav](#)

Examples

```
# single channel data
## Not run:
wav2flac('C:\\users\\user.name\\programs\\flac-1.5.0-win\\Win64\\flac.exe',
'F:\\wavFiles', 'F:\\flacFiles\\')

## End(Not run)
# multichannel data
## Not run:
wav2flac('C:\\users\\user.name\\programs\\flac-1.5.0-win\\Win64\\flac.exe',
```

```
'F:\\wavFiles', 'F:\\flacFiles\\', 4)
```

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

Index

CSVToGPX_track, [2](#)

DASTrackToGPX, [3](#)
decdeg2degmin, [4](#)
decdeg2degminsec, [4](#)
degmin2decdeg, [5](#)
degmin2degminsec, [6](#)
degminsec2decdeg, [7](#)
degminsec2degmin, [8](#)
downloadGoogleDriveFolder, [9](#)

er_clickSummary, [9](#)
er_clickSummary(), [12](#), [13](#), [15](#)
er_concatenatedSlope, [10](#)
er_filterClicks, [11](#)
er_plotClickDurationHist, [12](#)
er_plotClickDurationHist(), [10](#), [13](#), [15](#)
er_plotClickSNRHist, [13](#)
er_plotClickSNRHist(), [10](#), [12](#), [15](#)
er_plotConcatSpec, [13](#)
er_plotConcatSpec(), [12](#), [13](#), [15](#)
er_plotSpectra, [15](#)
er_plotSpectra(), [10](#), [12](#), [13](#), [15](#)
er_plotWhistleBeginFreqHist, [16](#)
er_plotWhistleContours, [16](#)
er_plotWhistleMedFreqHist, [17](#)
er_whistleSummary, [17](#)
extractDASTrack, [18](#)

flac2wav, [18](#), [23](#)

installSavedPackages, [19](#)

lldistkm, [20](#)
loadMultiBinaries, [21](#)

PAMpal::calculateAverageSpectra, [13](#)
PAMpal::calculateAverageSpectra(), [9](#),
[10](#), [13–15](#)
PAMpal::processPgDetections(), [9](#)

saveInstalledPackages, [22](#)

tk_choose.dir, [19](#), [23](#)

wav2flac, [19](#), [23](#)