Package ‘echogram’

April 13, 2025

Version 0.1.3

Title Echogram Visualisation and Analysis

Date 2025-04-13

Encoding UTF-8

Description Easily import multi-frequency acoustic data stored in 'HAC' files (see <doi:10.17895/ices.pub.5482> for more information on the format), and produce echogram visualisations with predefined or customized color palettes. It is also possible to merge consecutive echograms; mask or delete unwanted echogram areas; model and subtract background noise; and more important, develop, test and interpret different combinations of frequencies in order to perform acoustic filtering of the echogram's data.

Depends R (>= 3.0.0)

License GPL-3

URL `https://github.com/hvillalo/echogram`

BugReports `https://github.com/hvillalo/echogram/issues`

Imports geosphere, readHAC, pals

Author Héctor Villalobos [aut, cre] (<`https://orcid.org/0000-0002-6424-4050`>)

Maintainer Héctor Villalobos <`hvillalo@ipn.mx`>

NeedsCompilation no

Repository CRAN

Date/Publication 2025-04-13 21:50:02 UTC

Contents

add.echogram *Add two echograms*

Description

This function allows addition or subtraction of Sv data matrices of corresponding echograms from two frequencies in the linear or logarithmic domain.

Usage

```
add.echogram(echogram1, echogram2, operator = c("plus", "minus"),
    domain = c("linear", "dB"))
```

Arguments

`echogram1`	an object of `class` “echogram” as returned by `read.echogram`.
`echogram2`	an object of `class` “echogram” from a different acoustic frequency than `echogram1` above.
`operator`	a string indicating addition (“plus”) or subtraction (“minus”). May be abbreviatted.
`domain`	a string indicating the domain where the operation will be performed (see details).

Details

Corresponding echograms refers to data acquired at the same time with different acoustic frequencies. In order to add echograms, the Sv data matrices must have the same dimensions. If they don’t, `match.echogram` can be used for this purpose. It is also important to mask undesired echoes beforehand, as those belonging to the bottom and below. When `domain = "dB"` (the default), the Sv matrices are added as they are. When `domain = "linear"`, the Sv values are transformed with 10^(Sv/10) before addition, and the result (X) is then back transformed to dB (10*log10(X)).

Value

An object of `class` “echogram” where the Sv component is the result of the performed operation.

Author(s)

Héctor Villalobos

See Also

match.echogram, mask.echogram

Examples

```
# import 38 and 120 kHz data from an HAC file
hacfile <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile, channel = 1)
echo2.120 <- read.echogram(hacfile, channel = 2)

## Not run:
# attempting to add the two frequencies with unequal number of pings gives an error
add.echogram(echo2.038, echo2.120, "plus", "dB")

## End(Not run)

# running match.echogram() solves this
tmp <- match.echogram(echo2.038, echo2.120)
str(tmp) # both frequencies are in a list, need to split
echo2.038 <- tmp$echogram1
echo2.120 <- tmp$echogram2

# we don't want to add bottom echoes, mask bottom and surface from both frequencies
echo2.038m <- mask.echogram(echo2.038, surf.off = 2, bott.off = 0.2)
echo2.120m <- mask.echogram(echo2.120, surf.off = 2, bott.off = 0.2)

# adding Sv values and plot result
echo.sum <- add.echogram(echo2.038m, echo2.120m, "plus", "dB")
Min <- min(as.vector(echo.sum$Sv), na.rm=TRUE) # useful to set Sv threshold
echogram(echo.sum, Svthr=floor(Min), scheme = "EK500")

# subtract 38 from 120 kHz
echo.minus <- add.echogram(echo2.120m, echo2.038m, "minus", "dB")
Min <- min(as.vector(echo.minus$Sv), na.rm=TRUE)
echogram(echo.minus, Svthr=floor(Min), scheme = "EK500")
```

bottom.hac *Read detected bottom range from an HAC file*

Description

This function imports, for a given acoustic channel, the detected bottom range stored in the ping tuple of an HAC file.

Usage

```
bottom.hac(hac, channel = NULL, plot = FALSE, maxDepth = NULL)
```

Arguments

hac	name of an HAC file.
channel	acoustic channel number.
plot	logical. if TRUE a plot is produced.
maxDepth	maximum depth (in m) represented in the plot.

Details

The acoustic channel is an integer, normally between 1 and n, where n is the number of frequencies used during data acquisition. When channel = 1, data from the lowest acoustic frequency is imported, while channel = n refers to the highest frequency present in the HAC file. By default, the function finds out the smallest channel number, because in some HAC files channel = 0. When a graphical representation is desired (plot = TRUE), the maximum displayed in the echogram depth can be provided as a negative integer with argument maxDepth.

Value

A data frame with two variables where every row represents one emitted ping:

pingTime	time of emitted ping.
detBottom	detected depth range in m.

Author(s)

Héctor Villalobos

Examples

```
hacfile <- system.file("hac", "D20150510-T202221.hac", package="echogram")
bottom.hac( hacfile )
bottom.hac( hacfile, plot = TRUE )
```

---

echo.noise *Sample echogram data (120 kHz)*

Description

Echogram for background noise estimation.

Usage

```
data("echo.noise")
```

Details

For this echogram, recording range (500 m) has been set above the sea bottom, and there is no evident presence of biological scatteres in order to facilitate the background noise estimation.

Examples

```
data("echo.noise")
echogram(echo.noise)
```

echogram *Echogram visualisation*

Description

This function allows to produce echogram visualisations from imported hac data. The user can define the visualisation Sv threshold and select between two built-in color schemes or define a custom scheme.

Usage

```
echogram(echogram, Svthr = -70, Svmax = 0, col.sep = NULL, col.nb = NULL,
         scheme = NULL, depth.grid = NULL, x.grid = NULL,
         x.ref = c("pings", "nmi", "seconds"), seabed = FALSE,
         depth.max = NULL, ping.ini = NULL, ping.max = NULL, colbar=TRUE,
         main = NULL, tformat = "%H:%M", ...)
```

Arguments

`echogram`	an object of `class` “echogram” as returned by `read.echogram`.
`Svthr`	Sv visualisation threshold, in decibels (dB).
`Svmax`	maximum Sv visualisation value, in dB.
`col.sep`	separation between colors. Defaults to 1 dB.
`col.nb`	number of colors.
`scheme`	color scheme for echogram, “parula” (the default), “EK500”, or “echov”. It can also be a vector of valid color names, or a function generating color names.
`depth.grid`	spacing between depth labels (in m).
`x.grid`	spacing between labels in the horizontal dimension according to x.ref.
`x.ref`	horizontal reference in echogram: “pings” (the default), “nmi” or “seconds”.
`seabed`	logical. When `TRUE` and data on detected bottom is present, a line is added to the echogram.
`depth.max`	maximum depth to visualise.
`ping.ini`	initial ping to visualise.
`ping.max`	last ping to visualise.

`colbar` logical. If `TRUE` a color bar is added to the echogram.

`main` the acoustic frequency, by default.

`tformat` time format for annotating when horizontal dimension when x.ref = 'seconds'.

`...` other options to `image`.

Details

Besides the two built-in color schemes, the user can define its own by specifying a vector of valid color names (see examples). This function uses `imageScale` function from `sinkr` package by Marc Taylor.

Author(s)

Héctor Villalobos

See Also

`palette.echogram`.

Examples

```
# import hac file
hacfile <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile)

# echogram by default
echogram(echo2.038)

# using alternative color schemes
echogram(echo2.038, Svthr = -70, col.sep = 1.5, scheme = "EK500")
echogram(echo2.038, Svthr = -70, col.sep = 3, scheme = c("white", "blue", "grey", "black"))
```

`join.echogram` *Merge echograms*

Description

This function allows to join two echograms.

Usage

```
join.echogram(echogram1, echogram2)
```

Arguments

`echogram1` an object of `class` "echogram" as returned by `read.echogram`.

`echogram2` an object of `class` "echogram", preferentially contiguous in space and time with `echogram1` above.

Details

This function is designed to join echograms of the same acoustic frequency, giving an error if frequencies differ. Desirably, echograms should be contiguous in space and time, but as this is not verified, it is possible to join non-contiguous echograms.

Value

An object of class "echogram" resulting from the merging operation.

Author(s)

Héctor Villalobos

Examples

```
# import 38 kHz data from two consecutive HAC files
hacfile1 <- system.file("hac", "D20150510-T202221.hac", package = "echogram")
echo1.038 <- read.echogram(hacfile1, channel = 1)

hacfile2 <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile2, channel = 1)

# join into one echogram
echo.038 <- join.echogram(echo1.038, echo2.038)
str(echo.038)
echogram(echo.038)
```

mask.echogram *Mask an echogram*

Description

This function creates, and optionally applies, a mask by "blanking" portions of the Sv data matrix of an echogram.

Usage

```
mask.echogram(echogram, surf.off = NULL, bott.off = NULL, mask = TRUE)
```

Arguments

echogram	an object of class "echogram" as returned by read.echogram.
surf.off	surface offset in m defining the upper layer (refered to the surface) to be masked.
bott.off	bottom offset in m defining the bottom layer (refered to the bottom) to be masked.
mask	logical. If FALSE, the function returns a masking matrix. If TRUE (the default), the function returns a masked echogram (see details).

Details

The masking process consists in producing a matrix of the same dimensions as the original Sv data matrix with NA's in the masked portion and 1's otherwise. The product of both matrices gives the masked echogram.

Value

When mask = FALSE, a masking matrix is returned. When mask = TRUE (the default), an object of class "echogram" with the mask applied.

Author(s)

Héctor Villalobos

Examples

```
# import 38 kHz data from HAC file
hacfile <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile, channel = 1)

# make a copy of the original echogram
tmp <- echo2.038

# Create a mask, which is a matrix with 1's and NA's
mask <- mask.echogram(tmp, surf.off = 1, bott.off = -1, mask = FALSE)
image(t(mask[nrow(mask):1, ]))

# Apply mask to echogram
tmp$Sv <- tmp$Sv * mask
echogram(tmp)

# By default, the function returns the masked echogram
echo2.038mask <- mask.echogram(echo2.038, surf.off = 2, bott.off = 0.2)
echogram(echo2.038mask)
```

match.echogram *Match ping times from two echograms*

Description

This function verifies ping times between corresponding echograms from two frequencies, eliminating non-matching and duplicated pings.

Usage

```
match.echogram(echogram1, echogram2)
```

Arguments

`echogram1` an object of `class` “echogram” as returned by `read.echogram`.

`echogram2` an object of `class` “echogram” from a different acoustic frequency than `echogram1` above.

Details

Corresponding echograms refers to data acquired at the same time with different acoustic frequencies. Unmatching pings, i.e. those present in only one frequency, and duplicated pings, are identified by it’s associated time and subsequently eliminated.

Value

A list with the two matched echograms.

Author(s)

Héctor Villalobos and Violeta E. González-Maynez

See Also

`add.echogram`

Examples

```
# import 38 and 120 kHz data from an HAC file
hacfile <- system.file("hac", "D20150510-T202221.hac", package = "echogram")
echo1.038 <- read.echogram(hacfile, channel = 1)
echo1.120 <- read.echogram(hacfile, channel = 2)

# Sv matrices have different number of pings
dim(echo1.038$Sv); dim(echo1.120$Sv)

# Apply match ping times
tmp <- match.echogram(echo1.038, echo1.120)

# split the list in the two echograms
echo1.038 <- tmp$echogram1
echo1.120 <- tmp$echogram2

# number of pings and ping times are now the same for both frequencies
dim(echo1.038$Sv); dim(echo1.120$Sv)
```

mergeSvmat *Merge inequal Sv data matrices*

Description

Internal function called by `read.echogram` and `merge.echogram`.

Usage

```
mergeSvmat(m1, m2)
```

Arguments

`m1`	First Sv data matrix.
`m2`	Second Sv data matrix.

Author(s)

Héctor Villalobos

navigation.hac *Compute bearing, navigated distance and speed*

Description

This function computes navigation course (bearing), navigated distance, time diference and navigation speed between GPS fixes in position data imported from an HAC file.

Usage

```
navigation.hac(pos)
```

Arguments

`pos`	geographic position data from an HAC file, as imported with `position.hac`.

Details

The bearing and navigated distance are computed with functions `bearingRhumb` and `distVincentyEllipsoid` from package `geosphere`. This function is intended to be called inside `read.echogram`, rather than being used directly.

Value

A data frame with seven variables:

`time.cpu`	date and time from the computer CPU during data acquisition.
`lon`	longitudes.
`lat`	latitudes.
`bearing`	navigation course between two consecutive GPS fixes.
`navdist`	navigated distance between two consecutive GPS fixes.
`time.dif`	time difference between two consecutive GPS fixes.
`navspeed`	navigation speed between two consecutive GPS fixes.

Author(s)

Héctor Villalobos

See Also

`position.hac`, `bearingRhumb`, `distVincentyEllipsoid`.

Examples

```
hacfile <- system.file("hac", "D20150510-T202221.hac", package="echogram")
pos <- position.hac( hacfile )
pos
pos2 <- navigation.hac(pos)
pos2
```

`noise.echogram` *Modelling ambient noise in echograms*

Description

This function allows to estimate a model of the background noise in an echogram by fitting the equation proposed by De Robertis and Higginbottom (2007).

Usage

```
noise.echogram(echogram, ping = NULL, dB1m = NULL, alpha = NULL, plot = TRUE, out = FALSE)
```

Arguments

`echogram`	an object of `class` "echogram".
`ping`	ping number for which the Sv values are to be modeled.
`dB1m`	noise level at 1m from the face of the transducer.
`alpha`	absortion coefficient of sound in sea water for echogram's frequency.
`plot`	logical. If `TRUE` (the default) a plot of the data and adjusted model is produced.
`out`	logical. If `TRUE` an echogram with the modeled noise is returned.

Details

The estimation of an ambient noise model for a particular acoustic frequency, eventually allows the "cleaning" of echograms by subtracting this noise.

Value

When `plot = TRUE` and `out = FALSE` (the default), only a plot is produced. With `out = TRUE`, the function returns an object of `class` "echogram" with the noise modelled.

Author(s)

Héctor Villalobos

References

De Robertis, A. and I. Higginbottom. 2007. A post-processing technique to estimate the signal-to-noise ratio and remove echosounder background noise 64:1282–1291.

Examples

```
# load echogram for noise estimation at 120 kHz (deep waters, no scatterers)
data("echo.noise")
attr(echo.noise$Sv, "frequency")
echogram(echo.noise, xref = "ping")

# a first look to the Sv values at ping 2
noise.echogram(echo.noise, ping = 2)

# To better adjust the model, we need to provide the absortion coefficient for 120 kHz and adjust
# the dB1m parameter. For this example, using data from a nearby CTD profile, alpha was calculated
# as being 0.03550554, while -131 dB is choosen for dB1m
noise <- noise.echogram(echo.noise, ping = 2, dB1m = -131, alpha = 0.03550554, out = TRUE)
echogram(noise)
```

`palette.echogram` *Design color palettes for echograms*

Description

This function allows to design and visualise color palettes to be used in echograms.

Usage

```
palette.echogram(Svthr, Svmax, col.sep = NULL, col.nb = NULL, scheme = NULL, visu = FALSE)
```

Arguments

`Svthr` lower visualisation limit in decibels (dB).

`Svmax` upper visualisation limit in dB.

`col.sep` separation between colors in dB.

`col.nb` number of colors.

`scheme` color scheme for echogram, “parula” (the default), “EK500”, or “echov”. It can also be a vector of valid color names, or a function generating color names.

`visu` logical. If `TRUE`, a visual representation of the palette is created.

Details

This function is mainly intended to be called by `plot.echogram`, however it is possible to use it directly in order to have a first impression of a custom color palette.

Value

A list with two elements

`palette` a vector of colors

`breaks` a vector of color breaks

Author(s)

Héctor Villalobos

See Also

`echogram`

Examples

```
palette.echogram()
palette.echogram(Svthr=-75, col.sep=1.5, scheme="EK500", visu=TRUE)
palette.echogram(Svthr=-81, col.sep=3, scheme=c("white", "blue", "black"), visu=TRUE)
```

`position.hac` *Read geographic position data from an HAC file*

Description

This function imports time and geographic positions recorded by a GPS in an HAC file during data acquisition.

Usage

```
position.hac(hac)
```

Arguments

hac	name of an HAC file

Details

The function looks for the Position tuple (20) in the HAC file, and if found, imports the time, latitude and longitude of GPS fixes stored in the digital echogram, as well as the CPU time of the acquisition PC.

Value

A data frame with four variables:

time.gps	date and time from the GPS during data acquisition.
time.cpu	date and time from the computer CPU during data acquisition.
lon	longitudes.
lat	latitudes.

Note

If during acoustic data acqusition the PC clock is set to UTC time, as recomended, time.gps and time.cpu will be approximately equal, because a fraction of a second is added to time.cpu to obtain a precision of 0.0001 s.

Author(s)

Héctor Villalobos

References

ICES, 2005. Description of the ICES HAC Standard Data Exchange Format, Version 1.60. Technical Report 278, ICES Cooperative Research Report.

See Also

navigation.hac

Examples

```
hacfile <- system.file("hac", "D20150510-T202221.hac", package="echogram")
pos <- position.hac( hacfile )
pos
```

read.echogram *Read echogram data from an HAC file*

Description

This function imports from different tuples in the HAC file, the necessary information to visualise and analyse an echogram in R.

Usage

```
read.echogram(hac, channel = NULL)
```

Arguments

hac	name of an HAC file.
channel	acoustic channel number.

Details

This function calls internally other echogram's functions (postion.hac, navigation.hac and bottom.hac) to import data from an HAC file. The acoustic channel is an integer, normally between 1 and n, where n is the number of frequencies used during data acquisition. When channel = 1, data from the lowest acoustic frequency is imported, while channel = n refers to the highest frequency present in the HAC file. By default, the function finds out the smallest channel number, because in some HAC files channel = 0. A text string with the frequency value (in kilohertz) is stored as an attribute of the Sv matrix (see examples below).

Value

An object of class "echogram" (a list) with components:

depth	a vector of mean sample depth (in m) of length p.
Sv	a p by k matrix of sampled values, currently the mean volume backscattering strength (Sv, in dB).
pings	a k by four data frame with ping time, detected bottom depth, vessel speed and cummulated traveled distance.

Note

Currently, read.echogram has been successfully tested importing HAC data from the following ping tuples: 10000 (U-32); 10030 (U-16) and 10040 (C-16).

Author(s)

Héctor Villalobos

References

ICES, 2005. Description of the ICES HAC Standard Data Exchange Format, Version 1.60. Technical Report 278, ICES Cooperative Research Report.

Examples

```
hacfile <- system.file("hac", "D20150510-T202221.hac", package = "echogram")
echo1 <- read.echogram(hacfile, channel = 1)
class(echo1)
str(echo1)
attr(echo1$Sv, "frequency")
echogram(echo1)
```

`sample.echogram` *Select and sample data values from an echogram*

Description

This function allows to select individual pixels from an echogram and returns the Sv value, ping time and depth of the sampled pixel.

Usage

```
sample.echogram(echogram, plot = TRUE, coords = NULL, col = "black")
```

Arguments

`echogram`	an object of `class` "echogram" as returned by `read.echogram`.
`plot`	logical. If `TRUE` (the default), the echogram to be sampled is plotted.
`coords`	(x, y) coordinates (in plot units) of the desired samples. They could result from previous sampling of another frequency.
`col`	color for the sampled points pixels.

Details

The selection of pixels to sample can be done by clicking on the echogram or by passing the coordinates of the desired pixels to the function. The coordinates should be in plot units, and therefore, these typically come from a previous selection by clicking on another frequency's echogram (see examples).

`sample.echogram` makes use of `locator` function, and therefore it only works in devices supported by the latter, such as `X11`, `windows` and `quartz`.

Value

A data frame with seven variables:

`id`	pixel id.
`x`	x coordinate in plot units.
`y`	y coordinate in plot units.
`d`	distance in plot units from the selected location to a valid pixel.
`pingTime`	time of sampled ping.
`depth`	depth of the sample.
`Sv`	Sv value.

Author(s)

Héctor Villalobos

Examples

```
# import 38 and 120 kHz data from an HAC file
hacfile <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile, channel = 1)
echo2.120 <- read.echogram(hacfile, channel = 2)

# plot 38 kHz echogram
echogram(echo2.038)

## Not run:
# select points coordinates with the mouse
# click to select several locations and escape when done
pts038 <- sample.echogram(echo2.038)
pts038

# plot 120 kHz echogram
echogram(echo2.120)

# use the points previously selected for 38 kHz
pts120 <- sample.echogram(echo2.120, coords = pts038[ , 2:3])
pts120

## End(Not run)
```

`trim.echogram` *Trim an echogram vertically or horizontally*

Description

This function allows to trim an echogram by depth or ping number by actually trimming the underlying data matrices and vectors.

Usage

```
trim.echogram(echogram, depth.max = NULL, ping.ini = 1, ping.end = NULL)
```

Arguments

echogram	an object of class “echogram” as returned by read.echogram.
depth.max	maximum depth to keep in the echogram.
ping.ini	start ping to keep.
ping.end	end ping to keep.

Details

This funcion has been conceived to discard undesired data below a given depth (e.g. the sea bottom), therefore, the initial depth is always the surface, so the vertical trimming is limited to select the maximum depth.

Value

An object of class “echogram”.

Author(s)

Héctor Villalobos

Examples

```
# import 38 kHz data from an HAC file
hacfile <- system.file("hac", "D20150510-T202500.hac", package = "echogram")
echo2.038 <- read.echogram(hacfile, channel = 1)

# echogram by default
echogram(echo2.038)

# trim the echogram
echo.tmp <- trim.echogram(echo2.038, depth.max = 70, ping.end = 250)
echogram(echo.tmp)
```

Index