

Package ‘rLakeHabitat’

July 30, 2026

Type Package

Title Interpolate Bathymetry and Quantify Physical Aquatic Habitat

Version 1.0.3

Date 2026-07-29

URL <https://gitlab.com/tristanblechinger/rlakehabitat>

BugReports <https://gitlab.com/tristanblechinger/rlakehabitat/-/issues>

Depends R (>= 4.3.0)

Imports dplyr, terra, gstat, sf, ggplot2, gganimate, tidyterra,
rLakeAnalyzer, isoband

Maintainer Tristan Blechinger <tblechin@uwyo.edu>

Description Offers bathymetric interpolation using Inverse Distance Weighted and Ordinary Kriging via the 'gstat' and 'terra' packages. Other functions focus on quantifying physical aquatic habitats (e.g., littoral, epilimnion, metalimnion, hypolimnion) from interpolated digital elevation models (DEMs). Functions were designed to calculate these metrics across water levels for use in reservoirs but can be applied to any DEM and will provide values for fixed conditions. Parameters like Secchi disk depth or estimated photic zone, thermocline depth, and water level fluctuation depth are included in most functions.

License GPL (>= 3)

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spbal, ggspatial,
httr, maptiles, mapview, tidyr

Config/testthat/edition 3

NeedsCompilation no

Author Tristan Blechinger [aut, cre] (ORCID:
<<https://orcid.org/0009-0005-6730-8856>>),
Sean Bertalot [aut] (ORCID: <<https://orcid.org/0000-0001-8099-0144>>)

Repository CRAN

Date/Publication 2026-07-30 11:10:08 UTC

Contents

animBathy	2
bathyMap	3
calcHyps	5
calcLittoral	6
calcPhotic	7
calcSDI	8
calcVolume	9
contourPoints	10
crossValidate	11
estThermo	13
genStack	14
interpBathy	15
littoralVol	18
optimizeParams	19
partitionSites	21
rarify	23
samplingDensity	24
saveContours	30
Index	32

animBathy	<i>Generate Animated Plot</i>
-----------	-------------------------------

Description

Generate an animated plot of littoral area at different water level increments from a raster digital elevation model (DEM).

Usage

```
animBathy(  
  DEM,  
  units = "m",  
  littoral = TRUE,  
  secchi = NULL,  
  photic = NULL,  
  stop = NULL,  
  by = 1  
)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the <code>rast()</code> function in 'terra'
units	character describing depth units of DEM. Can be meters ("m") or feet ("ft"). Default = "ft"
littoral	logical indicating if littoral zone should be plotted (T) or entire waterbody (F), default = TRUE
secchi	number giving the average Secchi depth of the waterbody, photic zone estimated as 2.5m * secchi
photic	number giving the average photic depth of the waterbody, overwrites Secchi
stop	optional numeric value specifying depth at which to stop animation, default = NULL (all depths)
by	numeric value specifying depth increments between plots. Higher values will result in lower resolution. Default = 1

Details

We recommend using the `calcPhotic` function to estimate photic depth instead of using a Secchi disk measurement

Value

an animated ggplot object

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))

#run function
animBathy(DEM, units = 'm', littoral = TRUE, secchi = 1, by = 5)
```

bathyMap

Plot Bathymetry Map

Description

Generate a bathymetry map from a provided DEM raster with optional contours and depth labels.

Usage

```
bathyMap(
  DEM,
  contours = TRUE,
  start = NULL,
  end = NULL,
  by = 5,
  breaks = NULL,
  units = "m",
  labels = TRUE,
  textSize = 1.5,
  plotTitle = NULL
)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the <code>rast()</code> function in 'terra'
contours	logical indicating whether contours should included (TRUE) or not (FALSE), default = TRUE
start	numeric value describing what value contours should start at, default = 0
end	numeric value describing what value contours should end at, default = max depth
by	numeric value describing contour intervals, default = 5
breaks	optional numeric vector describing specific contours to include if contours = T, default = NULL
units	character describing units of depth measurement, default = "ft"
labels	logical indicating whether labels should be included (TRUE) or not (FALSE), default = TRUE
textSize	number describing text size of contour labels if included, default = 1.5
plotTitle	optional character string adding title to output plot

Value

ggplot object

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
bathyMap(DEM, contours = TRUE, units = 'm', labels = TRUE)
```

calcHyps	<i>Calculate Hypsography</i>
----------	------------------------------

Description

Calculates area at each depth for a given waterbody.

Usage

```
calcHyps(DEM, DEMunits = "m", depthUnits = "m", by = 1, output = "values")
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the <code>rast()</code> function in 'terra'
DEMunits	character describing units of raster coordinate system. Can be meters, kilometers, or hectares ("m", "km", "ha"), default = "m"
depthUnits	character describing units of depth measurement. Can be either feet or meters ("ft", "m"), default = "ft"
by	numeric increment per unit by which volumes are calculated. Higher values will result in lower resolution. Default = 1
output	character describing desired output, can either be a data frame of values ("values") or a hypsography plot ("plot"). Default = "values"

Value

data frame of areas at each depth unit ("values") or a hypsography plot ("plot")

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
calcHyps(DEM, DEMunits = 'm', depthUnits = 'm', by = 1, output = 'values')
```

calcLittoral

*Calculate Littoral Area***Description**

Calculates littoral surface area (2D) of a given waterbody across water levels based on an average photic depth value.

Usage

```
calcLittoral(
  DEM,
  photic = NULL,
  secchi = NULL,
  DEMunits = "m",
  depthUnits = "m",
  by = 1,
  stop = NULL
)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the rast() function in 'terra'
photic	number giving the average photic depth, overwrites Secchi depth
secchi	number giving the average secchi depth, photic zone estimated as 2.5m * secchi
DEMunits	character describing units of raster coordinate system. Can be meters, kilometers, or hectares ("m", "km", "ha"), default = "m"
depthUnits	character describing units of depth measurement (secchi and DEM). Can be either feet or meters ("ft", "m"), default = "ft"
by	numeric increment per unit depth by which areas are calculated. Higher values will result in lower resolution. Default = 1
stop	optional numeric value specifying depth at which to stop calculations, default = NULL

Details

We recommend using the calcPhotic function to estimate photic depth instead of using a Secchi disk measurement

Value

data frame of areas in specified units for each depth, as well as the littoral percentage of total surface area

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
calcLittoral(DEM, secchi = 1, depthUnits = "m", DEMunits = "m")
```

calcPhotic

Calculate Euphotic Depth

Description

Calculates euphotic depth from a Secchi disk measurement (meters) based on relationships between light attenuation (K_d), optical properties of water (F), and a measured Secchi depth (Z_{sd}). Based on equations specified by Kirk (1994) and Koenings and Edmundson (1991).

Usage

```
calcPhotic(Z, F = 1.99)
```

Arguments

Z	numeric value for measured Secchi disk depth in meters
F	numerical constant that accounts for optical properties of water, default = 1.99

Details

Kirk, J. T. O. (1994). Light and Photosynthesis in Aquatic Ecosystems. Cambridge University Press. Koenings, J. P., & Edmundson, J. A. (1991). Secchi disk and photometer estimates of light regimes in Alaskan lakes: Effects of yellow color and turbidity. Limnology and Oceanography, 36(1), 91–105. <https://doi.org/10.4319/lo.1991.36.1.0091>

We recommend finding a value from the literature that most accurately reflects the water quality of the study system, but values for generalized conditions are included from Koenings and Edmundson (1991). $F = 2.76$ for stained lakes $F = 1.99$ for clear lakes $F = 1.05$ for turbid lakes

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
calcPhotic(Z = 4, F = 1.99)
```

calcSDI	<i>Calculate Shoreline Development Index</i>
---------	--

Description

Calculates Shoreline Development Index value across water levels for a given waterbody.

Usage

```
calcSDI(DEM, units = "m", by = 1, stop = NULL)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the <code>rast()</code> function in 'terra'
units	character describing units of raster coordinate system. Can be meters, kilometers, or hectares ("m", "km", "ha"), default = "m"
by	numeric increment per unit depth by which areas are calculated. Higher values will result in lower resolution. Default = 1
stop	optional numeric value specifying depth at which to stop calculations, default = NULL

Value

data frame of perimeter lengths and SDI values for given depths

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
calcSDI(DEM, units = 'm')
```

calcVolume

*Calculate Pelagic Habitat Volumes***Description**

Calculates epilimnion, metalimnion, and hypolimnion volumes based on defined thermocline depths across water levels.

Usage

```
calcVolume(
  DEM,
  thermo_depth = NULL,
  thermo_high,
  thermo_low,
  DEMunits = "m",
  depthUnits = "m",
  by = 1,
  stop = NULL
)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the rast() function in 'terra'
thermo_depth	number giving the estimated middle of thermocline, results in calculation of only epilimnion and hypolimnion volumes. Default = NULL, cannot use in conjunction with thermo_low and thermo_high
thermo_high	number giving the upper bound of thermocline depth, results in calculation of epilimnion, metalimnion, and hypolimnion values
thermo_low	number giving the lower bound of thermocline depth, results in calculation of epilimnion, metalimnion, and hypolimnion values
DEMunits	character describing units of raster coordinate system. Can be meters, kilometers, or hectares ("m", "km", "ha"), default = "m"
depthUnits	character describing units of depth measurement. Can be either feet or meters ("ft", "m"), default = "ft"
by	numeric increment per unit by which volumes are calculated. Higher values will result in lower resolution. Default = 1
stop	optional numeric value specifying depth at which to stop habitat volume calculations, default = NULL

Value

a data frame of volumes in cubic meters calculated for each habitat (epilimnion, metalimnion, hypolimnion)

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
calcVolume(DEM, thermo_depth = 3, DEMunits = 'm', depthUnits = 'm')
```

contourPoints

Contour Lines to Points

Description

Get point coordinates and depth values along predetermined contours at a specified density.

Usage

```
contourPoints(object, depths = NULL, geometry = "geometry", density = 10)
```

Arguments

object	polygon or multipolygon shapefile (.shp) with depths included as an attribute column. Can be an sf or spatVector object.
depths	character string describing column name of depth attribute
geometry	character string describing column name of geometries. Default = "geometry"
density	numeric value describing distance between points in meters, default = 10m

Value

dataframe of coordinates and associated depths

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
# load test data
data <- sf::read_sf(system.file("extdata", "example_contour.shp", package = 'rLakeHabitat'))
#run function
contourPoints(data, depths = "Z", geometry = "geometry", density = 50)
```

crossValidate

*Cross Validate Interpolated Bathymetry***Description**

Obtain residual mean square error (RMSE) from K-fold cross validation of bathymetry interpolation.

Usage

```
crossValidate(
  outline,
  df,
  x,
  y,
  z,
  zeros = FALSE,
  separation = NULL,
  k = 5,
  res = 5,
  seed = NULL,
  method = "IDW",
  nmax = 20,
  idp = 2,
  model = "Sph",
  psill = NULL,
  range = NULL,
  nugget = 0,
  kappa = NULL,
  trend_order = 1,
  zero_threshold = 0.05
)
```

Arguments

outline	shapefile outline of a waterbody. Accepts a SpatVector, an sf object, or anything terra::vect() can read.
df	dataframe of coordinates and depths for a given waterbody. Coordinates are assumed to be in the same CRS as 'outline'.
x	character giving name of longitude column
y	character giving name of latitude column
z	character giving name of depth column
zeros	logical describing if bounding zeros are needed (FALSE) or provided (TRUE), default = FALSE
separation	number describing distance between points, in meters

k	numeric value describing the number of folds to test, default = 5
res	number describing desired cell resolution in meters, default = 5
seed	optional numeric value used to seed the random number generator, so that fold assignment (and therefore the resulting RMSE) is reproducible across runs. Default = NULL (not seeded).
method	character describing method of interpolation, options include Inverse Distance Weighted ("IDW"), Ordinary Kriging ("OK"), or Universal Kriging ("UK"). Default = "IDW"
nmax	numeric value describing number of neighbors used in interpolation, default = 20
idp	numeric value describing inverse distance power value for IDW interpolation
model	character describing type of model used in Ordinary Kriging, options include 'Sph', 'Exp', 'Gau', 'Sta', default = 'Sph'
psill	numeric value describing the partial sill value for OK interpolation, default = NULL
range	numeric describing distance beyond which there is no spatial correlation in Ordinary Kriging models, default = NULL
nugget	numeric describing variance at zero distance in Ordinary/Universal Kriging models, default = 0
kappa	numeric value describing model smoothness, default = NULL
trend_order	numeric value (1 or 2) giving the polynomial trend order for Universal Kriging ("UK" only), default = 1
zero_threshold	numeric proportion (0-1) of surface area that must interpolate to exactly 0 before the automatic zero re-interpolation pass runs - passed through to interpBathy(). Default = 0.05.

Details

Folds are assigned in two stages: points are first split into 5 depth strata (quintiles of observed depth), then within each stratum, points are spatially clustered into 'k' groups (via k-means on their coordinates) and each spatial cluster becomes one fold. This keeps depth ranges reasonably balanced across folds while avoiding the optimistic bias that comes from randomly scattering spatially autocorrelated points across folds. 'res' is required and is always in meters, regardless of the CRS 'outline' was originally supplied in.

For the model argument there are four different methods included here that are supported by `gstat::vgm` ("Sph", "Exp", "Gau", "Mat"). "Sph" = The default `gstat::vgm` method. Spherical model characterized by a curve that rises steeply to defined range then flattens, indicates no spatial correlation between points beyond that range. "Exp" = Exponential model characterized by spatial correlation decaying rapidly with distance, results in a rougher surface. "Gau" = Gaussian model similar to spatial model but with slower decay over distance, results in a smoother surface. "Mat" = Matern model that uses kappa to define the variogram relationship. High kappa values approach a Gaussian model (smooth surface), and low kappa values approach the Exponential model (kappa = 0.5 is equivalent to Exponential). Three parameters (psill, range, kappa) are incorporated from a fitted variogram (default = NULL). If specified in function input, chosen values will overwrite variogram values.

Value

a named numeric value giving the mean RMSE across k folds

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load example outline
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
#load example xyz data
data <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))
#run function
crossValidate(outline, data, "x", "y", "z", zeros = FALSE, separation = 10, k = 5,
res = 50, method = "IDW", nmax = 4, idp = 1.5, seed = 123)
```

estThermo

Estimate Average Thermocline Depth

Description

Estimate average thermocline depth across multiple sites and dates.

Usage

```
estThermo(data, site, date, depth, temp, combine = "all", type = "midpoint")
```

Arguments

data	data frame of water column temperature profiles
site	character giving the name of the site column
date	character giving the name of the date column
depth	character giving the name of the depth column
temp	character giving the name of the temp column
combine	character indicating whether to average across sites ("sites"), dates ("dates"), sites and dates ("all"), or not at all ("none"), default = "all". When combine = "none", each site/date combination is returned individually with no averaging.
type	character indicating whether to estimate a single thermocline midpoint depth ("midpoint") or the upper and lower depths bounding the metalimnion ("top_bottom"), default = "midpoint". "midpoint" wraps <code>rLakeAnalyzer::thermo.depth()</code> ; "top_bottom" wraps <code>rLakeAnalyzer::meta.depths()</code> .

Value

a data frame. When type = "midpoint": thermocline midpoint depths, standard deviations, and n (combine = "sites", "dates", or "all"), or one row per site/date combination giving that profile's midpoint depth (combine = "none"). When type = "top_bottom": upper and lower metalimnion boundary depths, standard deviations, and n (combine = "sites", "dates", or "all"), or one row per site/date combination giving that profile's upper and lower boundary depths (combine = "none").

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
# load test profile data
data <- read.csv(system.file("extdata", "example_profile_data.csv", package = 'rLakeHabitat'))
data$date <- base::as.Date(data$date)
#run function
estThermo(data = data, site = "site", date = "date",
depth = "depth", temp = "temp", combine = "all")
#or, to get each site/date profile's thermocline depth individually:
estThermo(data = data, site = "site", date = "date",
depth = "depth", temp = "temp", combine = "none")
#or, to estimate upper and lower metalimnion boundaries instead of a single midpoint:
estThermo(data = data, site = "site", date = "date",
depth = "depth", temp = "temp", combine = "none", type = "top_bottom")
```

genStack

Create Raster Stack

Description

Create a raster stack from a single raster, option to save as file.

Usage

```
genStack(
  DEM,
  by = 1,
  stop = NULL,
  save = TRUE,
  file_name = NULL,
  file_type = "COG"
)
```

Arguments

DEM	raster object
by	numeric increment per unit depth by which layers are split. Default = 1
stop	optional numeric value specifying depth at which to stop stacking rasters, default = NULL
save	logical, save raster stack (TRUE) or not (FALSE), default = TRUE
file_name	character string used to name saved raster stack
file_type	character string defining file type to save, default = "COG"

Value

a raster stack of specified depth increments for a given waterbody. Raster stack is either stored as an object (save = FALSE) or written to a file in the directory (save = TRUE).

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
genStack(DEM, by = 1, save = FALSE)
```

 interpBathy

Interpolate bathymetry

Description

Generate a bathymetric digital elevation model (DEM) for a given waterbody using Inverse Distance Weighting (IDW), Ordinary Kriging (OK), or Universal Kriging (UK) interpolation. For high densities of point data, we recommend rarifying prior to interpolation to improve accuracy and reduce computation time (see rarify function).

Usage

```
interpBathy(
  outline,
  df,
  x,
  y,
  z,
  zeros = FALSE,
  separation = NULL,
  res = 10,
```

```

method = "IDW",
nmax = 20,
idp = 2,
model = "Sph",
psill = NULL,
range = NULL,
nugget = NULL,
kappa = NULL,
trend_order = 1,
zero_threshold = 0.05
)

```

Arguments

outline	shapefile outline of a waterbody. Accepts a SpatVector, an sf object, or anything terra::vect() can read (e.g., a file path).
df	dataframe of coordinates and depths for a given waterbody. Coordinates are assumed to be in the same CRS as 'outline'.
x	character giving name of longitude column
y	character giving name of latitude column
z	character giving name of depth column
zeros	logical describing if bounding zeros are needed (FALSE) or provided (TRUE), default = FALSE
separation	number describing distance between points, in meters
res	number describing desired cell resolution in meters, default = 10
method	character describing method of interpolation: Inverse Distance Weighted ("IDW"), Ordinary Kriging ("OK"), or Universal Kriging ("UK"). Default = "IDW"
nmax	numeric value describing number of neighbors used in interpolation, default = 20
idp	numeric value describing inverse distance power value for IDW interpolation
model	character describing type of model used in Ordinary/Universal Kriging, options include 'Sph', 'Exp', 'Gau', 'Mat', default = 'Sph'
psill	numeric value describing the partial sill value for OK/UK interpolation, default = NULL
range	numeric describing distance beyond which there is no spatial correlation in Ordinary/Universal Kriging models, default = NULL
nugget	numeric describing variance at zero distance in Ordinary/Universal Kriging models, default = NULL
kappa	numeric value describing model smoothness, default = NULL
trend_order	numeric value (1 or 2) giving the order of the polynomial trend surface fit for Universal Kriging. 1 = linear trend ($z \sim x + y$), 2 = quadratic trend. Default = 1.
zero_threshold	numeric proportion (0-1) of the waterbody's surface area that must interpolate to exactly 0 before the automatic zero re-interpolation pass runs, default = 0.05 (5%). A handful of scattered zero cells won't trigger it; a large contiguous block collapsing to zero (typically an interpolation artifact, often from the shoreline zero ring dominating a narrow bay or inlet) will.

Details

The function automatically detects whether 'outline' (and therefore 'df', which is assumed to share its CRS) is in a geographic (decimal degree) or projected (meters) coordinate system. If geographic, the outline and point data are internally reprojected to their best-fit UTM zone so that all distance-based calculations (resolution, nmax neighbor selection, IDW power, kriging variogram parameters, and boundary point separation) operate on meters rather than degrees. The final DEM is reprojected back to the original CRS of 'outline' before being returned. The CRS used for interpolation, and progress through the major steps, are printed/reported as the function runs. 'res' is required and is always in meters, regardless of the CRS 'outline' was originally supplied in.

For the model argument there are four different methods included here that are supported by `gstat::vgm` ("Sph", "Exp", "Gau", "Mat"). "Sph" = The default `gstat::vgm` method. Spherical model characterized by a curve that rises steeply to defined range then flattens, indicates no spatial correlation between points beyond that range. "Exp" = Exponential model characterized by spatial correlation decaying rapidly with distance, results in a rougher surface. "Gau" = Gaussian model similar to spatial model but with slower decay over distance, results in a smoother surface. "Mat" = Matern model that uses kappa to define the variogram relationship. High kappa values approach a Gaussian model (smooth surface), and low kappa values approach the Exponential model (kappa = 0.5 is equivalent to Exponential). Three parameters (psill, range, kappa) are incorporated from a fitted variogram (default = NULL). If specified in function input, chosen values will overwrite variogram values - and any parameter that is auto-fit is fit with knowledge of the others you did supply (including nugget), rather than fitting as if the rest were still at their `gstat` defaults. Universal Kriging ("UK") differs from Ordinary Kriging in that it fits a polynomial trend surface across x/y (see 'trend_order') and models spatial correlation in the residuals from that trend, rather than assuming a constant mean across the whole waterbody. This can help for reservoirs with a strong directional depth gradient (e.g. a river-fed arm sloping steadily toward a dam), where OK's constant-mean assumption doesn't hold well.

Value

the interpolated DEM. For "IDW", a single-layer `SpatRaster`. For "OK" and "UK", a two-layer `SpatRaster`: layer 'depth' (the interpolated values) and layer 'error' (the associated standard error of each estimate).

Author(s)

Tristan Blechinger & Sean Bertalot, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load example outline
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
#load example xyz data
data <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))
#run function
interpBathy(outline, data, "x", "y", "z", zeros = FALSE, separation = 10,
res = 5, method = "IDW", nmax = 4, idp = 2)
```

littoralVol

*Calculate Littoral Volume***Description**

Calculate littoral and pelagic volume across water levels from a DEM based on estimated photic depth.

Usage

```
littoralVol(
  DEM,
  photic,
  secchi = NULL,
  DEMunits = "m",
  depthUnits = "m",
  by = 1
)
```

Arguments

DEM	SpatRaster object of a given waterbody, rasters can be transformed to SpatRaster via the rast() function in 'terra'
photic	number giving the average photic depth, overwrites Secchi depth
secchi	number giving the average secchi depth, photic zone estimated as 2.5m * secchi
DEMunits	character describing units of raster coordinate system. Can be meters, kilometers, or hectares ("m", "km", "ha"), default = "m"
depthUnits	character describing units of depth measurement. Can be either feet or meters ("ft", "m"), default = "ft"
by	numeric increment per unit by which volumes are calculated. Higher values will result in lower resolution. Default = 1

Details

We recommend using the calcPhotic function to estimate photic depth instead of using a Secchi disk measurement

Value

data frame of littoral and pelagic volume estimates

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
#load raster
DEM <- terra::rast(system.file("extdata", "example_raster.tif", package = 'rLakeHabitat'))
#run function
littoralVol(DEM, photic = 2, DEMunits = "m", depthUnits = "m", by = 1)
```

optimizeParams	<i>Optimize Point Density and DEM Resolution From Existing Depth Data</i>
----------------	---

Description

Given a dense depth dataset, determine what point spacing the data can be rarified to, and what DEM cell resolution to interpolate at, without meaningfully losing accuracy. Unlike `samplingDensity()`, this works directly with measured depth data: at every combination of candidate rarefaction spacing and candidate DEM resolution, the data are thinned with `rarify()` and scored with `crossValidate()`, building an RMSE surface across the full spacing x resolution grid.

Usage

```
optimizeParams(
  outline,
  df,
  x,
  y,
  z,
  spacings,
  res_values,
  k = 5,
  zeros = FALSE,
  separation = NULL,
  method = "IDW",
  nmax = 20,
  idp = 2,
  model = "Sph",
  psill = NULL,
  range = NULL,
  nugget = NULL,
  kappa = NULL,
  trend_order = 1,
  zero_threshold = 0.05,
  tolerance = 0.1,
  plot = TRUE,
  seed = NULL
)
```

Arguments

outline	shapefile outline of a waterbody. Accepts a SpatVector, an sf object, or anything terra::vect() can read.
df	dataframe of coordinates and depths for a given waterbody. Coordinates are assumed to be in the same CRS as 'outline'.
x	character giving name of longitude column
y	character giving name of latitude column
z	character giving name of depth column
spacings	numeric vector of candidate raryfy() target spacings to test, in meters.
res_values	numeric vector of candidate DEM cell resolutions to test, in meters.
k	numeric value describing the number of cross-validation folds to use at each grid point, default = 5
zeros	logical describing if bounding zeros are needed (FALSE) or provided (TRUE), default = FALSE
separation	number describing distance between boundary points, in meters (required if zeros = FALSE)
method	character describing method of interpolation, "IDW", "OK", or "UK. Default = "IDW"
nmax	numeric value describing number of neighbors used in interpolation, default = 20
idp	numeric value describing inverse distance power value for IDW interpolation
model	character describing type of model used in Ordinary Kriging, options include 'Sph', 'Exp', 'Gau', 'Mat', default = 'Sph'
psill	numeric value describing the partial sill value for OK interpolation, default = NULL
range	numeric describing distance beyond which there is no spatial correlation in Ordinary Kriging models, default = NULL
nugget	numeric describing variance at zero distance in Ordinary Kriging models, default = 0
kappa	numeric value describing model smoothness, default = NULL
trend_order	numeric value (1 or 2) giving the order of the polynomial trend surface fit for Universal Kriging. 1 = linear trend ($z \sim x + y$), 2 = quadratic trend. Default = 1.
zero_threshold	numeric proportion (0-1) of the waterbody's surface area that must interpolate to exactly 0 before the automatic zero re-interpolation pass runs, default = 0.05 (5%). A handful of scattered zero cells won't trigger it; a large contiguous block collapsing to zero (typically an interpolation artifact, often from the shoreline zero ring dominating a narrow bay or inlet) will.
tolerance	numeric value (proportion) used to pick the recommended (spacing, resolution) pair: among all grid points whose mean RMSE is within 'tolerance' of the single best (lowest-RMSE) grid point, the coarsest spacing is chosen (ties broken by coarsest resolution). Default = 0.1 (10%).

plot	logical: should a diagnostic heatmap of RMSE across the spacing x resolution grid be drawn? Default = TRUE.
seed	optional numeric value used to seed the random number generator (fold assignment in crossValidate() is stochastic), for reproducible results across runs. Default = NULL (not seeded).

Details

Runtime scales with $\text{length}(\text{spacings}) \times \text{length}(\text{res\_values}) \times k$, since each grid point runs a full crossValidate() call (which itself fits k separate DEMs). Start with a small grid (e.g. 3x3) to confirm the analysis runs end-to-end before scaling up to a finer search.

Value

a list with:

results a data frame of spacing, res, mean RMSE, and the number of points remaining after rarefaction at each spacing

recommended_spacing the recommended rarefaction spacing, in meters

recommended_res the recommended DEM cell resolution, in meters

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
data <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))
optimizeParams(outline, data, "x", "y", "z", spacings = c(10, 25, 50),
res_values = c(5, 10, 20), seed = 123)
```

partitionSites

Random Stratified Site Selection

Description

Randomly assign locations across a waterbody using Halton Iterative Partitioning (HIP; Robertson et al. 2018, via the spbal package), stratified by depth. Depth bins are defined from an interpolated DEM (e.g. the output of interpBathy()), and a customizable number of spatially determined locations is drawn within each bin - so, for example, you can request more locations in a shallow littoral bin and fewer in a deep pelagic bin.

Usage

```
partitionSites(
  dem,
  depth_bins,
  n_per_bin,
  water_level_drop = 0,
  min_spacing = NULL,
  iterations = 7,
  plot = TRUE,
  seed = NULL
)
```

Arguments

dem	a SpatRaster of interpolated bathymetry (e.g. from <code>interpBathy()</code>), used both to define depth bins and as the pool of candidate locations. Assumed to represent the waterbody at full pool/elevation unless 'water_level_drop' is used.
depth_bins	numeric vector of depth bin edges, e.g. <code>c(0, 5, 15, 30, Inf)</code> for three bins: 0-5 m, 5-15 m, and 15-30+ m. Must have at least 2 values (i.e. at least 1 bin). Bins are half-open, [lower, upper).
n_per_bin	numeric vector giving how many locations to draw from each bin, in the same order as 'depth_bins' implies (length must be <code>length(depth_bins) - 1</code>). A single value is also accepted and recycled across all bins (e.g. <code>n_per_bin = 5</code> draws 5 locations from every bin).
water_level_drop	optional single non-negative numeric value giving how far the water level has dropped, in the same depth units as 'dem', relative to the full-pool elevation 'dem' is assumed to represent. If greater than 0, the DEM is rebuilt before sampling: every cell's depth is reduced by this amount, and any cell whose adjusted depth is at or below zero (i.e., now exposed/dry) is excluded from sampling. Default = 0 (no water level adjustment; 'dem' is used as-is).
min_spacing	optional numeric value giving the minimum allowed distance, in meters, between selected locations within a bin (passed to <code>spbal::HIP()</code> 's <code>minRadius</code> argument). Default = <code>NULL</code> (no minimum spacing enforced).
iterations	numeric value giving the number of Halton partition levels used by <code>spbal::HIP()</code> (see <code>spbal::HIP()</code> for details), default = 7. If HIP fails for a given bin (e.g. too few candidate cells), the function automatically retries with fewer partition levels before giving up on that bin with a warning.
plot	logical: should a map of the resulting locations, colored by depth bin and shown over the (adjusted) DEM, be drawn? Default = <code>TRUE</code> .
seed	optional numeric value used to seed the random number generator, for reproducible site selection across runs. Default = <code>NULL</code> (not seeded).

Details

This function requires the 'spbal' and 'sf' packages to be installed (not hard dependencies of this package, since they are only needed for this function). The function automatically detects whether

'dem' is in a geographic (decimal degree) or projected (meters) coordinate system; if geographic, sampling and 'min_spacing' are carried out in the waterbody's best-fit UTM zone (determined from 'dem's own extent) so that distances are measured in meters, and returned locations are projected back to the original CRS of 'dem'. If a depth bin contains fewer candidate DEM cells than the number of locations requested for it, all available cells are used and a warning is issued. No outline shapefile is required - a boundary for the plot (if requested) is derived automatically from the non-NA footprint of 'dem' (or the water-level-adjusted DEM, if 'water_level_drop' > 0).

Value

a list with:

locations a data frame of selected locations, with columns 'x', 'y' (in the original CRS of 'dem'), 'depth_bin' (the bin label), and 'bin_index' (the bin's position in 'depth_bins'). Depth bins/labels reflect the water-level-adjusted DEM if 'water_level_drop' > 0.

map a recorded base R plot (see `grDevices::recordPlot()`) of the (adjusted) DEM and selected locations by depth bin, or NULL if plot = FALSE.

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
data <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))
dem <- interpBathy(outline, data, "x", "y", "z", zeros = FALSE,
  separation = 10, res = 10, nmax = 8, method = "IDW")
partitionSites(dem, depth_bins = c(0, 5, 15, 30, Inf),
  n_per_bin = c(8, 6, 4, 2), seed = 123)

# with a 2 m water level drop from full pool
partitionSites(dem, depth_bins = c(0, 5, 15, 30, Inf),
  n_per_bin = c(8, 6, 4, 2), water_level_drop = 2, seed = 123)
```

rarify

Rarify Depth Data

Description

Reduce density of mapped depth data to improve accuracy and computation time.

Usage

```
rarify(outline, df, x, y, z, res = 10)
```

Arguments

outline	shapefile outline of a waterbody. Accepts a SpatVector, an sf object, or anything terra::vect() can read.
df	dataframe of coordinates and depths for a given waterbody. Coordinates are assumed to be in the same CRS as 'outline'.
x	character giving name of longitude column
y	character giving name of latitude column
z	character giving name of depth column
res	number describing the target spacing between rarified points, in meters, default = 10

Details

The function automatically detects whether 'outline' (and therefore 'df') is in a geographic (decimal degree) or projected (meters) coordinate system, the same way interpBathy() does. If geographic, points are rarified in the waterbody's best-fit UTM zone so that 'res' is honored as a true physical spacing in meters, then the rarified points are returned in the original CRS of 'outline'.

Value

dataframe of rarified xyz coordinates (columns named 'x', 'y', 'z' regardless of the input column names), in the same CRS as the original input

Author(s)

Sean Bertalot & Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
depths <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))
rarify(outline = outline, df = depths, x = "x", y = "y", z = "z", res = 100)
```

samplingDensity

Pre-Survey Power Analysis for Bathymetric Sampling Design

Description

Before any depth data are collected, estimate how far apart survey transects can be spaced and still produce an accurate bathymetric DEM. Because no real depth data exist yet, this works as a simulation study: a plausible "true" bathymetry is generated for the waterbody, a survey at each candidate transect spacing is simulated by sampling that truth, a DEM is reconstructed from those simulated samples (via interpBathy), and the reconstruction is compared back against the generated DEM. This is repeated many times per spacing to average out randomness, and across all candidate spacings, to build a curve of expected DEM accuracy vs. transect spacing - along with a recommended spacing and a map of what that survey design looks like on the actual waterbody outline.

Before any depth data are collected, estimate how far apart survey transects can be spaced and still produce an accurate bathymetric DEM. Because no real depth data exist yet, this works as a simulation study: a plausible "true" bathymetry is generated for the waterbody, a survey at each candidate transect spacing is simulated by sampling that truth, a DEM is reconstructed from those simulated samples (via `interpBathy`), and the reconstruction is compared back against the generated DEM. This is repeated many times per spacing to average out randomness, and across all candidate spacings, to build a curve of expected DEM accuracy vs. transect spacing - along with a recommended spacing and a map of what that survey design looks like on the actual waterbody outline.

Before any depth data are collected, estimate how far apart survey transects can be spaced and still produce an accurate bathymetric DEM. Because no real depth data exist yet, this works as a simulation study: a plausible synthetic "true" bathymetry is generated for the waterbody (a bowl-shaped depth profile plus spatially-correlated random roughness), a survey at each candidate transect spacing is simulated by sampling that synthetic truth, a DEM is reconstructed from those simulated samples (via `interpBathy`), and the reconstruction is compared back against the known synthetic truth. This is repeated many times per spacing to average out randomness, and across all candidate spacings, to build a curve of expected DEM accuracy vs. transect spacing - along with a recommended spacing and a map of what that survey design looks like on the actual lake outline.

Usage

```
samplingDensity(
  outline,
  max_depth,
  spacings = NULL,
  n_sim = 10,
  shape = 1,
  truth_model = "Exp",
  truth_range = NULL,
  truth_sill = NULL,
  truth_nugget = 0,
  along_track_interval = NULL,
  orientation = NULL,
  res = NULL,
  method = "IDW",
  nmax = 20,
  idp = 2,
  model = "Sph",
  psill = NULL,
  range = NULL,
  nugget = 0,
  kappa = NULL,
  trend_order = 1,
  zero_threshold = 0.05,
  tolerance = 0.1,
  n_truth_examples = 3,
  plot = TRUE,
  seed = NULL
)
```

```
samplingDensity(  
  outline,  
  max_depth,  
  spacings = NULL,  
  n_sim = 10,  
  shape = 1,  
  truth_model = "Exp",  
  truth_range = NULL,  
  truth_sill = NULL,  
  truth_nugget = 0,  
  along_track_interval = NULL,  
  orientation = NULL,  
  res = NULL,  
  method = "IDW",  
  nmax = 20,  
  idp = 2,  
  model = "Sph",  
  psill = NULL,  
  range = NULL,  
  nugget = 0,  
  kappa = NULL,  
  trend_order = 1,  
  zero_threshold = 0.05,  
  tolerance = 0.1,  
  n_truth_examples = 3,  
  plot = TRUE,  
  seed = NULL  
)
```

```
samplingDensity(  
  outline,  
  max_depth,  
  spacings = NULL,  
  n_sim = 10,  
  shape = 1,  
  truth_model = "Exp",  
  truth_range = NULL,  
  truth_sill = NULL,  
  truth_nugget = 0,  
  along_track_interval = NULL,  
  orientation = NULL,  
  res = NULL,  
  method = "IDW",  
  nmax = 20,  
  idp = 2,  
  model = "Sph",  
  psill = NULL,
```

```

    range = NULL,
    nugget = 0,
    kappa = NULL,
    trend_order = 1,
    zero_threshold = 0.05,
    tolerance = 0.1,
    n_truth_examples = 3,
    plot = TRUE,
    seed = NULL
  )

```

Arguments

outline	shapefile outline of a waterbody. Accepts a SpatVector, an sf object, or anything terra::vect() can read.
max_depth	expected maximum depth of the waterbody, in meters. This drives the synthetic bathymetry and has a large effect on the results - use the best estimate available (a known max depth, an old chart, or a knowledgeable guess).
spacings	numeric vector of candidate transect spacings to test, in meters. Default = NULL, in which case a sequence scaled to the waterbody's size is generated automatically (roughly 2%-40% of the lake's characteristic length).
n_sim	number of independent synthetic "true" bathymetries to simulate and test each spacing against, default = 10. Higher values give a more stable accuracy estimate at the cost of runtime.
shape	numeric value controlling the synthetic depth profile's shape: $\text{depth} = \text{max\_depth} * (\text{relative distance to shore})^{\text{shape}}$. shape = 1 (default) gives a linear, cone-like profile; shape < 1 gives a broader, flatter deep basin; shape > 1 gives a narrower, steeper-sided basin.
truth_model	character variogram model used to generate the synthetic bathymetry's random roughness (see gstat::vgm options), default = "Exp".
truth_range	numeric range parameter (in meters) for the synthetic roughness's spatial autocorrelation, default = NULL, in which case it is set to one quarter of the waterbody's characteristic length (sqrt(area)).
truth_sill	numeric sill (variance, in squared meters) for the synthetic roughness, default = NULL, in which case it is set to $(\text{max\_depth}/5)^2$.
truth_nugget	numeric nugget for the synthetic roughness variogram, default = 0.
along_track_interval	numeric spacing (in meters) at which simulated samples are drawn along each transect line, representing how frequently a sonar unit records depth while underway. Default = NULL, in which case it is set to 1/10th of the smallest tested spacing.
orientation	numeric angle, in degrees (0 = along the x-axis/east, increasing counterclockwise), giving the orientation of transect lines. Default = NULL, in which case the orientation is chosen automatically to align with the waterbody's long axis.

res	numeric DEM cell resolution (in meters) used both for generating the synthetic truth surface and for reconstructing the DEM from simulated samples. Default = NULL, in which case it is set to 1/5th of the smallest tested spacing. This should generally be left fine relative to 'spacings'.
method	character describing method of interpolation used to reconstruct DEMs, "IDW", "OK", or "UK". Default = "IDW" (recommended for this use, since it is run many times and OK's variogram fitting adds considerable runtime).
nmax	numeric value describing number of neighbors used in interpolation, default = 20
idp	numeric value describing inverse distance power value for IDW interpolation
model	character describing type of model used in Ordinary Kriging, options include 'Sph', 'Exp', 'Gau', 'Sta', default = 'Sph'
psill	numeric value describing the partial sill value for OK interpolation, default = NULL
range	numeric describing distance beyond which there is no spatial correlation in Ordinary Kriging models, default = NULL
nugget	numeric describing variance at zero distance in Ordinary Kriging models, default = 0
kappa	numeric value describing model smoothness, default = NULL
trend_order	numeric value (1 or 2) giving the polynomial trend order for Universal Kriging ("UK" only), default = 1
zero_threshold	numeric proportion (0-1) of surface area that must interpolate to exactly 0 before the automatic zero re-interpolation pass runs - passed through to interpBathy(). Default = 0.05.
tolerance	numeric value (proportion) used to pick the recommended spacing: the coarsest tested spacing whose mean RMSE is still within 'tolerance' of the best (finest-spacing) RMSE. Default = 0.1 (10%%).
n_truth_examples	numeric value giving how many of the n_sim synthetic "true" bathymetries to save and return/plot, so you can visually check whether the assumed depth profile and roughness look like your lake before trusting the rest of the results. Default = 3 (capped at n_sim).
plot	logical: should diagnostic plots (accuracy-vs-spacing curve, and outline + recommended transects map) be drawn? Default = TRUE.
seed	optional numeric value used to seed the random number generator, for reproducible results across runs. Default = NULL (not seeded).

Details

This is a power analysis under assumed, not observed, bathymetry - it can only be as realistic as the 'max_depth', 'shape', 'truth_range', and 'truth_sill' inputs. Once real depth data have been collected, use crossValidate(), interpBathy(), and optimizeParams() on the actual data to check whether observed accuracy matches what was predicted here, and adjust future survey effort accordingly. Runtime scales with $n\_sim \times \text{length}(\text{spacings}) \times (\text{cost of one synthetic-surface simulation} + \text{one})$

interpBathy call). For a first look, consider a smaller `n_sim` (e.g. 5) and a coarser `'res'` before committing to a longer run.

This is a power analysis under assumed, not observed, bathymetry - it can only be as realistic as the `'max_depth'`, `'shape'`, `'truth_range'`, and `'truth_sill'` inputs. Once real depth data have been collected, use `crossValidate()` and `interpBathy()` on the actual data to check whether observed accuracy matches what was predicted here, and adjust future survey effort accordingly.

This is a power analysis under assumed, not observed, bathymetry - it can only be as realistic as the `'max_depth'`, `'shape'`, `'truth_range'`, and `'truth_sill'` inputs. Once real depth data have been collected, use `crossValidate()` and `interpBathy()` on the actual data to check whether observed accuracy matches what was predicted here, and adjust future survey effort accordingly. Runtime scales with `n_sim` x `length(spacings)` x (cost of one synthetic-surface simulation + one `interpBathy` call). For a first look, consider a smaller `n_sim` (e.g. 5) and a coarser `'res'` before committing to a long run.

Value

a list with:

results a data frame of spacing, mean RMSE, and SD of RMSE across simulations

recommended_spacing the recommended transect spacing, in meters

transects a `SpatVector` of the recommended transect lines, in the original CRS of `'outline'`

total_transect_length total length of the recommended transects, in meters

truth_examples a multi-layer `SpatRaster` (one layer per saved example) of synthetic "true" bathymetries used during the simulation, in the original CRS of `'outline'`, for visually sanity-checking the assumed depth profile

a list with:

results a data frame of spacing, mean RMSE, and SD of RMSE across simulations

recommended_spacing the recommended transect spacing, in meters

transects a `SpatVector` of the recommended transect lines, in the original CRS of `'outline'`

total_transect_length total length of the recommended transects, in meters

truth_examples a multi-layer `SpatRaster` (one layer per saved example) of synthetic "true" bathymetries used during the simulation, in the original CRS of `'outline'`, for visually sanity-checking the assumed depth profile

a list with:

results a data frame of spacing, mean RMSE, and SD of RMSE across simulations

recommended_spacing the recommended transect spacing, in meters

transects a `SpatVector` of the recommended transect lines, in the original CRS of `'outline'`

total_transect_length total length of the recommended transects, in meters

truth_examples a multi-layer `SpatRaster` (one layer per saved example) of synthetic "true" bathymetries used during the simulation, in the original CRS of `'outline'`, for visually sanity-checking the assumed depth profile

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
samplingDensity(outline, max_depth = 40, n_sim = 5, seed = 123)

outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
samplingDensity(outline, max_depth = 40, n_sim = 5, seed = 123)

outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))
samplingDensity(outline, max_depth = 40, n_sim = 5, seed = 123)
```

saveContours

Generate and Export Bathymetric Contours

Description

Generate depth contours from an interpolated DEM and save them to a file format usable outside R - a GPS unit or chartplotter (.gpx), Google Earth or similar (.kml), or standard GIS formats (.shp, .gpkg).

Usage

```
saveContours(
  dem,
  by = NULL,
  levels = NULL,
  file_type = "gpx",
  file_name = "contours"
)
```

Arguments

dem	a SpatRaster of interpolated bathymetry (e.g. from interpBathy()). If 'dem' has multiple layers (e.g. the depth/error output of OK or UK), the layer named 'depth' is used if present, otherwise the first layer.
by	numeric value giving a regular contour interval, e.g. by = 5 contours every 5 depth units starting at 5 (the 0 contour is skipped, since it's just the shoreline itself). Exactly one of 'by' or 'levels' must be specified.
levels	numeric vector of specific depth values to contour, e.g. c(1, 5, 10, 20). Exactly one of 'by' or 'levels' must be specified.
file_type	character giving the output format: "gpx", "kml", "shp", or "gpkg". Default = "gpx".
file_name	character giving the output file name, with or without a path. Default = "contours".

Details

GPX and KML both require geographic (WGS84 lon/lat) coordinates - if 'dem' is in a projected CRS, the contours are automatically reprojected to EPSG:4326 before being written (.shp/.gpkg outputs keep 'dem's original CRS).

Value

an sf object of the generated contour lines (with a 'depth' column), invisibly saved to 'file_name' as a side effect. Returned regardless of 'file_type' so you can plot or inspect it without re-reading the saved file.

Author(s)

Tristan Blechinger, Department of Zoology & Physiology, University of Wyoming

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {  
  outline <- terra::vect(system.file("extdata", "example_outline.shp", package = 'rLakeHabitat'))  
  data <- read.csv(system.file("extdata", "example_depths.csv", package = 'rLakeHabitat'))  
  dem <- interpBathy(outline, data, "x", "y", "z", zeros = FALSE, separation = 10,  
    res = 10, method = "IDW", nmax = 4)  
  saveContours(dem, by = 2, file_type = "gpx", file_name = tempfile("contours"))  
}
```

Index

`animBathy`, [2](#)

`bathyMap`, [3](#)

`calcHyps`, [5](#)

`calcLittoral`, [6](#)

`calcPhotic`, [7](#)

`calcSDI`, [8](#)

`calcVolume`, [9](#)

`contourPoints`, [10](#)

`crossValidate`, [11](#)

`estThermo`, [13](#)

`genStack`, [14](#)

`interpBathy`, [15](#)

`littoralVol`, [18](#)

`optimizeParams`, [19](#)

`partitionSites`, [21](#)

`rarify`, [23](#)

`samplingDensity`, [24](#)

`saveContours`, [30](#)