Package 'tlocoh'

November 12, 2016

Type Package

Title Constructs homeranges and explores time use patterns from location data

Version 1.40.02

Date 2016-11-13

Author Andy Lyons

Maintainer Andy Lyons <tlocoh@gmail.com>

BugReports Andy Lyons <tlocoh@gmail.com>

Depends R (>= 3.0.0),
  sp

Imports FNN,
  pbapply,
  rgeos

Suggests rgdal,
  move,
  png,
  raster,
  gpclib,
  XML,
  dismo

Description Constructs homeranges and explores time use patterns from location
  data.

License GPL-2

URL <http://tlocoh.r-forge.r-project.org>

RoxygenNote 5.0.1

R topics documented:

---

aoi *Select an area of interest on a plot using the mouse*

Description

Allows the user to select an area of interest (box) by clicking on the active plot window

Usage

```
aoi(draw.poly = TRUE, draw.poly.col = "black", status = TRUE)
```

Arguments

draw.poly	Whether to draw the box on the plot. T/F.
draw.poly.col	The color to use if drawing the box on the plot. Color value. Ignored if draw.poly=FALSE.
status	Show messages to the user

Value

A two-column xy data frame containing the upper left point and the lower right point in the first and second rows respectively

Note

This function requires a plot window to be open

---

aoi2box   *Converts a dataframe with upper-left / lower-right coordinates of a*
                                           box to a closed polygon

---

Description

Converts a dataframe with upper-left / lower-right coordinates of a box to a closed polygon

Usage

```
aoi2box(aoi)
```

Arguments

aoi   A two-column two-row data frame or matrix with x in the first column and y in
                     the second columm. The upper left point and the lower right point in the first
                     and second rows respectively

Value

Two column dataframe containing the xy coordinates of a closed rectangle that can be plotted using
[points](#) or [polygons](#)

Note

You can use the aoi function to select two coordinates on the active plot window with the mouse.
This function facilitates plotting that rectangle.

See Also

[aoi](#)

Examples

```
## Not run:
myaoi <- aoi()
polygons(myaoi)

## End(Not run)
```

---

auto.a   *Parameters to select a reasonable value of 'a'*

Description

Returns a list of parameters that other functions accept for the 'a' parameter

Usage

```
auto.a(meth = "nn", ptp = 0.98, nnn = 2, tct = 1.05)
```

Arguments

meth   The name of the method for finding a (default is "nn" method for 'nearest neigh-
        bor' method.

ptp   The proportion of total points [0..1]

nnn   The number of nearest neighbors

tct   The temporal continuity threshhold, see details.

Details

If multiple values are passed for any of the parameters, the resulting data frame will have multiple
rows and other parameters will be recycled as needed.

This function does not actually find nearest neighgors. Rather it returns a list of parameters that can
be used as input in other functions such as `lxy.nn.add`.

Value

A data frame with columns meth, ptp, nnn, and tct which are the settings that other functions will
use to find a value for 'a' should be automatically selected. Columns include:

- *meth* The name of the method for finding a
- *ptp* The proportion of parent points that should have at least a.nn neighbors (default 0.98)
- *nnn* The minimum number of nearest neighbors that each point will get (default 2, so each
  point can be part of a triangle)
- *tct* A time threshhold (expressed as a percentage of the median sampling frequency), such that
  if a point is temporally disconnected from its 2 nearest temporal neighbors by more than than
  tct times the median sampling frequency, the point will be be excluded from the computation
  of a. This is to avoid bias from internal time gaps (default 1.05)

See Also

`lxy.nn.add`

Examples

```
## Not run:
lxy <- lxy.nn.add(lxy, ptsh=0.5, a=auto.a())

## End(Not run)
```

---

chop2plot   *Chops a long character object into separate lines for adding to a plot*

Description

Breaks a long character object into separate lines at specific characters

Usage

```
chop2plot(str, width, char.break = " ", char.break.keep = FALSE, cex = 1)
```

Arguments

`str`	The string to format
`width`	The maximum width of any single line in inches
`char.break`	The character where the string should be broken
`char.break.keep`	
	Whether to keep the char.break character when the string gets broken. T/F
`cex`	The character expansion factor that will be used to display the text

Value

A two-item character vector containing [1] the formatted character string with \n inserted and [2] number of lines

Note

This function is designed to break up a potentially long character object that is going to be added to a plot either as a title or text. Line breaks will be inserted so that the string will wrap appropriately at `width` inches when plotted with `cex` character expansion factor.

The character object will be broken at the `char.break` characters such that each line will be `<=` `width` when plotted. Breaks are replaced with carriage returns. If `char.break.keep` is `TRUE`, the break character will be kept, otherwise it will be removed. If no occurrence of the /codechar.break is available, the line will be chopped at the last character that fits.

---

clean_slivers   *Clean slivers*

Description

Remove 'slivers' from a SpatialPolygons* object

Usage

```
clean_slivers(sp, status = TRUE)
```

Arguments

sp	A SpatialPolygons or SpatialPolygonsDataFrame object
status	Show messages. T/F

Details

As defined here, a 'sliver' is polygon or hole that has < 3 unique nodes (perhaps due to a data type conversion or rounding error). This seems to be a particularly common type of geometry error that crops in when unioning hulls into isopleths.

Value

Returns a list with two elements:

sp - the (cleaned) SpatialPolygons* object results - a dataframe of the polygon and Polygon indices of removed slivers

---

cube	*Cubes a number*

Description

Cubes a number

Usage

```
cube(x)
```

Arguments

x	a number

Value

x^3

---

cw	*Console wrap. Formats a string for printing to the console window*

Description

Takes a potentially long character object and adds carriage returns and blank spaces for better appearance when printing to the terminal window

Usage

```
cw(x, final.cr = TRUE, indent = 0, exdent = 0)
```

Arguments

x	A character object
final.cr	Whether to add a carriage return at the end. T/F.
indent	First line indent (number of blank spaces)
exdent	Subsequent line(s) indent (number of blank spaces)

Value

character object with embedded \n characters

Note

This function can prevent long messages from extending beyond the width of the R-console. Works with any function that prints to the console (e.g., cat, print, stop)

---

findonpath   *findonpath*

Description

Searches for a file on the system environment path

Usage

```
findonpath(fn, status = TRUE, quote = TRUE)
```

Arguments

fn	The filename to search for (without a path)
status	Show messages
quote	Enclose returned filename in quotes

Value

The full path and name of the found file, or NULL if not found

Note

This will return the first found occurence of file fn, searching 1) the current working directory, 2) the user's R 'home' directory, then 3) the directories on the operating system environment 'path'

---

get.vals   *Get x and/or y values from a plot using the mouse*

Description

Get x and/or y values from a plot using the mouse

Usage

```
get.vals(axis = c("x", "y")[1], round.to = 1, print = TRUE)
```

Arguments

axis	Which axes to return values for, character vector
round.to	An integer number that values will be rounded to. For example if round.to=100, values will be rounded to the nearest 100.
print	Whether to print the values in the console, T/F

Details

This function allows you to get the x and/or y values by clicking on the active plot window. When you're done clicking, right-click and select 'Stop' from the pop-up menu.

Value

A vector or matrix of the x and/or y values where the mouse was clicked.

---

hist.locoh.lhs   *Plot histograms of hull metrics*

Description

Plot histograms of hull metrics

Usage

```
## S3 method for class 'locoh.lhs'
hist(x, lhs, id = NULL, k = NULL, r = NULL,
  a = NULL, s = NULL, hs.names = NULL, metric = "area",
  include.missing.hulls = TRUE, hmap = NULL, hmap.in.title = TRUE,
  hs.name.in.title = TRUE, title = NULL, title.show = TRUE,
  figs.per.page = NULL, mar = c(3, 3, if (title.show) 2.8 else 0.7, 0.5),
  mgp = c(1.8, 0.5, 0), same.axes.for.all = FALSE, ufat = TRUE,
  breaks = "Sturges", col = "gray80", lo.margins.set, lo.save = TRUE,
  png.dir = NULL, png.dir.make = TRUE, png.width = 800,
  png.height = png.width, png.overwrite = TRUE, png.pointsize = 12 +
  (png.width - 480)/80, title.two.id = FALSE,
  indicate.missing.hulls.in.axis.lbl = TRUE, panel.num = NULL,
  panel.num.inside.plot = !title.show, ...)
```

Arguments

x	A [LoCoH-hullset](#) object
lhs	Deprecated, use x instead
id	The names of the individual(s) to include in the plot.
k	The k value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
r	The r value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
a	The a value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
s	The s value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
hs.names	The name(s) of saved hullsets to include in the plot
metric	The name(s) of hull metric(s); one histogram will be created for each hull metric.
include.missing.hulls	
	Whether to include missing hulls on the histogram (using the default value assigned to missing hulls by the hull metrics). T/F
hmap	A named list of hull metric auxillary parameters, the name of each list element is the name of the variable.
hmap.in.title	Include the hmap value(s) in the plot subtitle. T/F
hs.name.in.title	
	Include the name of the hullset in the plot subtitle. T/F
title	A title for the plot (over-writes the construction of a title)
title.show	Whether to add a title on the plot. T/F
figs.per.page	Number of plots per page
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
same.axes.for.all	
	Whether to use the same range on the axes for all histograms generated (helps to visually see differences)
ufat	User-friendly axis title. T/F
breaks	Number of breaks or a clustring function. See [hist](#).
col	Color value for the bars
lo.margins.set	Deprecated, use lo.save instead
lo.save	Whether to set the plot device options (margins, rows and columns). T/F.
png.dir	The directory for a PNG file (filename will be constructed automatically).
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image. Ignored if png.fn is passed.
png.height	The height of the PNG image. Ignored if png.fn is passed.
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
png.pointsize	The pointsize (in pixels) for the PNG image (increase to make labels appear larger). Determines the height or width of a character in pixels.

title.two.id	Construct a title with the names of two ids (for so.* and to.* assocation metrics). T/F
indicate.missing.hulls.in.axis.lbl	
	Specify whether missing hulls are excluded in the axes label, T/F
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. Ignored if panel.num is NULL. T/F
...	Other parameters, including any auxillary parameters to specify particular hull metrics

Value

A named list of values of the hull metrics. There is one list element for each metric.

Note

To see the names of hull metrics, type `hm.expr()`. Only hull metrics which have been computed can be plotted.

---

hist.locoh.lxy	*Plot histograms of properties of a LoCoH-xy object*

Description

Displays histogram(s) of point-to-point step length, velocity, and sampling frequency for a LoCoH-xy object

Usage

```
## S3 method for class 'locoh.lxy'
hist(x, lxy, id = NULL, dt = TRUE, d = TRUE,
  delta.t = TRUE, v = TRUE, figs.per.page = NULL, col = "gray80",
  lo.save = TRUE, dt.bins.base = c("secs", "mins", "hours", "days")[4],
  dt.bins.width = 3600 * 24 * 7, delta.t.num.sd = NULL, d.tct = 1.2,
  time.unit = "auto", overlay.median = TRUE, breaks = 20, ...)
```

Arguments

x	A [LoCoH-xy](#) object
lxy	Deprecated, use x instead
id	The id value(s) to be plotted
dt	Include a histogram of the number of locations over time) (T/F)
d	Include a histogram of distance travelled per adjacent points (i.e., step length) (T/F)
delta.t	Include a histogram of the time between points (i.e., sampling frequency) (T/F)
v	Include a histogram of point velocity (distance over delta.t) (T/F)
figs.per.page	Number of plots per page

col	The color of the bars
lo.save	Whether to set the plot device options (margins, rows and columns). T/F.
dt.bins.base	Where to start the first bin ((for histogram of number of points over time): "secs", "mins", "hours", "days"
dt.bins.width	The width of the time bins (for histogram of number of points over time). In seconds.
delta.t.num.sd	Number of standard deviations for delta.t to be included in the histogram. To omit outliers from appearing in the histogram (which can make the central data more difficult to discern, set delta.t.num.sd to ~2. Ignored if delta.t=F.
d.tct	temporal connectivity thsesshold for the distance between two points to be included in the histogram of step length. In other words, detlta.t must be <= median delta.t * d.tct). Ignored if d=F.
time.unit	The unit of time on the x-axis (character). Ignored if delta.t=F.
overlay.median	Plot the median value on the histogram (T/F)
breaks	Argument passed to the hist function, see hist
...	Other parameters to be passed to hist

Value

A list of frequencies with one element for each of the histograms plotted.

---

hm.expr	*Hull metrics expressions*

Description

Returns a list of expression objects that when evaluated in functions will return various hull metrics

Usage

```
hm.expr(names.only = TRUE, desc = names.only, print = names.only)
```

Arguments

names.only	Return only the names of hull metrics (not the expressions themselves). T/F.
desc	Include descriptions of the hull metrics in the object returned. T/F.
print	Display the list of hull metrics on the console. T/F.

Details

Hull metrics are used by many functions. This function returns a list of the expressions to 'pull out' hull metric values from the data structure of a LoCoH-hullset object and other properties of hull metrics. These objects are used by other T-LoCoH functions for plotting, exporting, creating isopleths, and other things you can do with hull metrics.

Most users will not need to use this function other than to see a list of hull metric names (which you need to know to specify a hull metric in a plotting function for example) and the hull metric descriptions.

This function does not compute any hull metrics, only provide a comprehensive list of all possible metrics. Some hull metrics are computed by default when you create a hullset (see [lxy.lhs](lxy.lhs)). Others metrics must be computed with a special function (e.g., lhs.ellipses.add. To see which hull metrics exist for a specific LoCoH-hullset, use summary.locoh.lhs.

Value

A named list with the following elements:

1. ufat. User-friendly axis title

2. desc. Description (used to build plot captions)

3. def. Definition. If NULL, then the definition is the same as $desc

4. ufipt. User-friendly isopleth plot title. Used when plotting isopleths constructed by sorting hulls on the metric.

5. nhv. No hull value (the value to use for parent points that have no hulls, used to highlight the presence of points without hulls in histograms

6. iso.dec. Sort hulls in descending order when used to create isopleths). T/F

7. expr Expression object that when evaulated returns the hull metric values

8. zero2na Whether 0 values should be colored as NA (i.e., hidden) when classifying hull parent points, T/F

9. discrete Whether the metric assumes discrete (integer) values, used in classifying hull parent points T/F

10. auto.jiggle The maximum value of a uniform distribution used to apply a random offset to the metric when plotted on scatterplots to better see point density

11. spao.x include the metric as a default for the x-axis when the 'auto' option is used in lhs.plot.scatter.auto

12. spao.y include the metric as a default for the y-axis when the 'auto' option is used in lhs.plot.scatter.auto

13. req.metrics The underlying metric(s) that need to be saved in the hullset for successful computation. Character vector. This will usually simply be the name of the metric itself, but may be different in the case of derived metrics such as perimeter-area ratio.

14. req.ap The name(s) of required auxillary variables that are needed to extract the metric. Character vector.

15. req.ap.def A named list of default values of any required auxillary variables

16. req.ap.subtitle A named list of expression objects (one for each hmap) that returns a string of hmap values that will be used in the subtitle of plots

17. req.ap.desc An named list of expression objects (one for each hmap) that returns a string of hmap values that will be used in the description of plots

18. fun The name of the function that creates the hull metric

19. iso.hm2 Second hull metric used to break ties when creating isopleths (used mostly when the metric returns discrete values)

20. iso.hm2.dec Sort second hull metric descending

See Also

lhs.ellipses.add, lhs.visit.add

hsp2expr   *Returns a R expression that will recreate a saved scatterplot of hull*
                            metrics

Description

Returns a R expression that will recreate a saved scatterplot of hull metrics

Usage

```
hsp2expr(hsp, reg_nodes_prec = 3, var_name = "hsp")
```

Arguments

hsp	A list of objects of class locoh.hsp
reg_nodes_prec	The number of decimal places that will be preserved when printing the coordinates of the nodes of manually defined polygon regions in scatterplot space
var_name	The name(s) of the variable that will be initialized in the output code

Details

Hull metric scatterplot settings can be saved as an object for reuse as a map legend or subsetting data (see *T-LoCoH Tips* on website). Saved scatterplot settings can include the boundaries of manually defined (i.e., with the mouse) regions of interest, that can be used as a map legend or for data subsetting. The purpose of this function is to generate R code so that polygons that were manually defined with the mouse in a hull metric scatterplot can be recreated (e.g., in a markdown document) for the purposes of reproducible research. hsp2expr will return a valid chunk of R code that recreates the saved scatterplot settings. This code can be copy-pasted into a RMarkdown document or other report format so the work may be recreated.

Value

A chunk of R code that will reproduce the hsp object

hulls   *Extract hulls from a LoCoH-hullset*

Description

Extract hulls from a LoCoH-hullset

Usage

```
hulls(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals
k	The k value of hullsets
r	The r value of hullsets
a	The a value of hullsets
s	The s value of hullsets
hs.names	The name(s) of saved hullsets

Details

This function returns a list of SpatialPolygonDataFrame objects that contain hulls. There will be one list element for each hullset in lhs. The data table will contain the hull metrics.

Value

list of SpatialPolygonDataFrame objects

See Also

[isopleths](#), [lhs.exp.shp](#), [lhs.exp.csv](#), Vignette on T-LoCoH data classes

---

isopleths	*Extract isopleths from a LoCoH-hullset*

Description

Extract isopleths from a LoCoH-hullset

Usage

```
isopleths(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, iso.idx = NULL, iso.sort.metric = NULL)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals
k	The k value of hullsets
r	The r value of hullsets
a	The a value of hullsets
s	The s value of hullsets
hs.names	The name(s) of saved hullsets
iso.idx	The index(s) of the isopleths to extract. Use [summary.locoh.lhs](#) to see the indices of the isopleths.
iso.sort.metric	
	The name(s) of sort metric(s) for the isopleths that will be extracted. Character. See [hm.expr](#).

Details

This function returns a list of SpatialPolygonDataFrame objects that contain isopleths. There will be one list element for each isopleth in lhs. Note that a LoCoH-hullset object can contain multiple hullsets, and a single hullset can have multiple isopleths. Accordingly, the names of elements in the returned list are a combination of a hullset name and an isopleth name.

Value

list of SpatialPolygonDataFrame objects

See Also

[hulls](#), [lhs.exp.shp](#), [lhs.exp.csv](#), Vignette on T-LoCoH data classes

---

lhs.anv.add   *Adds an ancillary value to a hullset*

Description

Add ancillary values to the points of a hullset, which can then be used as hull metrics

Usage

```
lhs.anv.add(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, anv, anv.desc = NULL, overwrite = FALSE)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals to export
k	The k value of hullsets to add an ancillary value to
r	The r value of hullsets to add an ancillary value to
a	The a value of hullsets to add an ancillary value to
s	The s value of hullsets to add an ancillary value to
hs.names	The name(s) of saved hullsets to export
anv	A vector, named list, or data frame with the same number of values as points
anv.desc	A character vector of descriptions
overwrite	Whether to overwrite existing ancillary values with the same names (T/F)

Details

This will add one or more ancillary values (columns) to the point locations in a [LoCoH-hullset](#) object. The name of the columns will be the name of the object passed (in the case where anv is a vector), the list element names (in the case of a named list), or column names (in the case of a data frame).

In other functions, you can sort (or filter) hulls based on ancillary values of the parent points using the hull metric 'anv' and passing another argument (called anv) set to the name of the ancillary value.

Value

A [LoCoH-hullset](#) object

See Also

[lhs.iso.add](#)

---

lhs.anv.del   *Delete ancillary variables from a LoCoH-hullset*

Description

Delete ancillary variables from a LoCoH-hullset

Usage

```
lhs.anv.del(lhs, anv = NULL, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, status = TRUE)
```

Arguments

lhs	A LoCoH-hullset object
anv	The name(s) of ancillary variables to remove
id	The name(s) of individuals to process
k	The k value of hullsets to process
r	The r value of hullsets to process
a	The a value of hullsets to process
s	The s value of hullsets to process
hs.names	The name(s) of hullsets to process
status	Show status messages (T/F)

Value

A LoCoH-hullset object

Note

To see which ancillary variables a hullset contains, use the summary() function.

Copying ancillary variables is optional when creating a hullset with [lxy.lhs](#).

See Also

[summary.locoh.lhs](#)

lhs.dr.add	*Define directional routes*

Description

Defines 'directional routes' by identifying parent points that are 1) temporally contiguous and 2) have hulls in the top n

Usage

```
lhs.dr.add(lhs, metric = c("ecc", "par")[2], thresh.val = 0.95,
  thresh.type = c("q", "v")[1], smooth = 1, status = TRUE,
  show.elong.hist = FALSE)
```

Arguments

lhs	A LoCoH-hullset object
metric	The name of the hull metric used as a proxy for directionality
thresh.val	The threshhold above which a hull is considered part of a directional route
thresh.type	The type of threshhold used. If 'q' for quantile, then thresh.val is taken to be a percentile of the full range of the directionality metric values (i.e., 0 < thresh.val < 1). If 'v', thresh.val is taken to be an actual value of the threshhold metric.
smooth	The amount of temporal smoothing applied, expressed as the number of points on either side (temporally) of the parent point whose average of the directionality metric is used for determining if the hull constructed around the parent point should be considered a part of a directional route. For no smoothing, set smooth = 0.
status	Display status messages
show.elong.hist	Whether to display histogram(s) of the distribution of the directionality metric before and after smoothing. T/F

Value

A LoCoH-hullset object

Note

Directional routes are line segments defined by connecting temporally contiguous hull parent-points that are in the top N percent of hull elongation. Hull elongation is proxied by one of two hull metrics: perimeter:area ratio ('par') or the eccentricity of the bounding ellipse ('ecc'). Hull perimeter:area ratios are automatically computed when a hullset is created; bounding ellipses must be computed separately using lhs.ellipses.add. Once computed, you can plot directional routes by passing dr=TRUE to plot.locoh.lhs.

---

lhs.ellipses.add *Compute hull bounding ellipses*

Description

Creates bounding ellipses around each hull and computes the eccentricity as a metric of ellongation

Usage

```
lhs.ellipses.add(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, status = TRUE, beep = FALSE,
  save.ellipses = TRUE, existing.ellipses = c("overwrite", "abort")[1])
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals to analyze
k	The k value of hullsets to analyze
r	The r value of hullsets to analyze
a	The a value of hullsets to analyze
s	The s value of hullsets to analyze
hs.names	The name(s) of saved hullsets to analyze
status	Show status messages (T/F)
beep	Beep when done (T/F)
save.ellipses	Save ellipses in the lhs object? (T/F). If False, only the eccentricity values of the ellipses are saved as a hull metric.
existing.ellipses	
	Whether to overwrite or abort if existing ellipses are saved

Value

A LoCoH-hullset object

Note

Saving the ellipses is optional. If ellipses are not saved, the eccentricity values will still be computed and saved as a hull metric. The only real reason to save ellipses is if you wish to plot them.

Note creating ellipses is computationally intensive and can take a long time.

lhs.exp.csv	*Export a LoCoH-hullset to csv*

Description

Export hull metrics to a csv file / data frame

Usage

```
lhs.exp.csv(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, csv.save = TRUE, dir = ".", csv.fn.pre = NULL,
  csv.fn.mid = c("none", "short", "long")[2], hm = "all", anv = NULL,
  hsp = NULL, status = TRUE)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals to export
k	The k value of hullsets to export
r	The r value of hullsets to export
a	The a value of hullsets to export
s	The s value of hullsets to export
hs.names	The name(s) of saved hullsets to analyze
csv.save	Whether to create csv file(s). If FALSE, a list of data frames will still be returned. T/F
dir	The directory where the csv file(s) will be saved
csv.fn.pre	A prefix for the csv file name(s)
csv.fn.mid	Style of the filename mid-section: none, short, or long
hm	The name of hull metric(s) that will be exported. Character vector or comma separated string.
anv	The name(s) of ancillary variables(s) of the parent point of each hull to be exported. Default is NULL, can also be set to 'all'. Character vector or comma separated string.
hsp	A list containing one hull scatterplot object with regions saved (i.e., object returned by `lhs.plot.scatter`), or the index of a hull scatter plot saved in the hullset (see `lhs.hsp.add`). If passed, a column for the region number will be included in the attribute table. Applicable only to hulls and hull parent points.
status	Show messages. T/F

Value

List (one list element for each hullset) of data frames of hull metrics and possibly ancillary variables

Note

This function exports hull metrics only. If you want to export the coordinates of the hulls themselves for further analysis in a GIS, see /link{lhs.exp.shp}.

If `file` is left blank, no csv file will be written to disk but the function will still return a list of data frames (one for each hullset)

If you have created a scatterplot legend with manually defined regions (see tutorial for details), the region number of each hull parent point can be included in the csv file if you pass the hullscatter plot as parameter `hsp`.

See Also

lhs.exp.shp

---

lhs.exp.mov   *Create a Quicktime animation from a LoCoH-hullset object*

Description

Create a Quicktime animation from a LoCoH-hullset object

Usage

```
lhs.exp.mov(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, all.ids.at.once = TRUE,
  all.ids.col.unique = all.ids.at.once, all.ids.col = NULL,
  all.ids.legend = c("bottomright", "bottom", "bottomleft", "left", "topleft",
  "top", "topright", "right", "center")[5], all.ids.legend.cex = 0.8,
  dt.start = NULL, dt.end = NULL, frame.method = c("auto", "time",
  "location")[1], frame.rtd = "auto", rtd.center = TRUE,
  clean.multihull.frames = 4, xlim = NULL, ylim = NULL, dt.label = TRUE,
  dt.label.col = "black", dt.label.col.bg = NA, dt.label.x = NULL,
  dt.label.y = NULL, title = NULL, title.show = TRUE, axes.show = TRUE,
  axes.ticks = axes.show, axes.titles = FALSE, mar.map = c(0.7 + (if
  (axes.ticks) 0.9 else 0) + (if (axes.titles) 1.3 else 0), 0.5 + (if
  (axes.ticks) 0.9 else 0) + (if (axes.titles) 1.3 else 0), if (title.show) 2.1
  else 0.5, 0.5), mgp.map = c(0.4 + if (axes.ticks) 1.2 else 0, 0.4, 0),
  col.hull.active = "red", col.hull.alpha = 255, tz.local = NULL,
  tz.local.check = TRUE, col.by.hour.of.day = FALSE,
  col.hod = colorRampPalette(colors()[c(24, 30, 553, 121, 26, 121, 553, 30,
  24)])(24), width = if (screen.test) 7 else 608, height = NULL,
  max.frames = NULL, png.pointsize = 16 + (width - 480)/80,
  screen.test = FALSE, tmp.dir = NULL, tmp.files.delete = TRUE,
  prompt.continue = TRUE, fn.mov = NULL, fn.mov.dir = getwd(),
  fn.mov.exists = c("auto.increment", "overwrite", "stop", "ask")[1],
  duration = NULL, fps = NULL, skip = NULL, ffmpeg = "ffmpeg.exe",
  create.mov = TRUE, info.only = TRUE, shp.csv = NULL, layers = NULL,
  tiff.fn = NULL, tiff.pct = FALSE, tiff.bands = c(4, 3, 2),
  tiff.buff = 500, tiff.fill.plot = TRUE, bg2png = !is.null(layers),
  crop.layers.to.extent = TRUE, date.bar = 0.85, date.bar.bins = 12,
  col.db = "darkblue", cex.axis.db = 0.7, beep = FALSE,
  report.time = TRUE, status = TRUE)
```

Arguments

lhs	A [LoCoH-hullset](#) object
id	The id value(s) to be on the plot
k	The k value of hullsets to export
r	The r value of hullsets to export
a	The a value of hullsets to export
s	The s value of hullsets to export
hs.names	The name(s) of saved hullsets to export
all.ids.at.once	
	Display all the individual ids simultaneously. If False, an animation will be created for each id. T/F.
all.ids.col.unique	
	Whether to use unique colors for each individual when plotting multiple individuals simultaneously, T/F
all.ids.col	A named list of color values; the element names must match the name(s) of the ids
all.ids.legend	Where to place a legend showing the color of each id (for an animation showing the movement of multiple individuals simultaneously): 'bottomright', 'bottom', 'bottomleft', 'left', 'topleft', 'top', 'topright', 'right', or 'center'. May also be NULL, in which case the legend will not be displayed. Ignored if ids.legend.bln=FALSE or all.ids.col.unique = FALSE.
all.ids.legend.cex	
	The character expansion factor for the id legend. See parameter all.ids.legend above.
dt.start	The starting date-time. An object of class POSIXt or one that can be coerced to POSIXt. If NULL, the earliest date-time in the series will be used.
dt.end	The ending date-time. An object of class POSIXt or one that can be coerced to POSIXt. If NULL, the last date-time in the series will be used.
frame.method	How each frame should be defined temporally, "time" - each frame represents a fixed amount of time, "location" each frame is a point in the series
frame.rtd	The real-time duration of each frame (in seconds). If "auto" (default), the lowest median sampling frequency will be used
rtd.center	Whether to center the start and end time of the first frame around the time stamp of the first location, to help ensure that each frame has only one hull in it. T/F.
clean.multihull.frames	
	A numeric value that determines whether and how much a hull can be time shifted to avoid having two active hulls in the same frame. Pass 0 to disable cleaning multi-hull frames. See details.
xlim	A two-element numeric vector for the range of the x-axis (in map units)
ylim	A two-element numeric vector for the range of the y-axis (in map units)
dt.label	Add a label for the date of the frame
dt.label.col	A color value/name for the date label (ignored if dt.label=FALSE)
dt.label.col.bg	
	The background color for the date label (NA for transparent background, ignored if dt.label=FALSE)
dt.label.x	The x-coordinate for the date label (ignored if dt.label=FALSE)

`dt.label.y`	The y-coordinate for the date label (ignored if `dt.label=FALSE`)
`title`	A title for the map
`title.show`	Whether to show the title. T/F.
`axes.show`	Whether to show the axes (ticks, labels and titles). Can be over-written by `axes.ticks` and `axes.titles`. T/F.
`axes.ticks`	Whether to show the tick marks and labels on the axes. T/F.
`axes.titles`	Whether to show axes titles. T/F.
`mar.map`	Margin settings for the map, see `par`
`mgp.map`	Locations of the axes elements for the map, see `par`
`col.hull.active`	The color of the active hull
`col.hull.alpha`	A number 0..255 for the alpha value for the hull color (semi-transparency); 0=completely transparent, 255=opaque
`tz.local`	The name of the time zone of the study area
`tz.local.check`	Check whether tz.local is a valid timezone name (not implemented) T/F.
`col.by.hour.of.day`	Whether to color the active point by the hour of day (i.e., dark colors at night, orange for day time locations). T/F
`col.hod`	A vector of 24 color values used to symbolize the color of the active point by the hour of day. Ignored if `col.by.hour.of.day = FALSE`.
`width`	The width of each frame in pixels (if `screen.test=FALSE`) or inches (if `screen.test=TRUE`).
`height`	The height of each frame in pixels (if `screen.test=FALSE`) or inches (if `screen.test=TRUE`).
`max.frames`	The maximum number of frames to produce.
`png.pointsize`	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
`screen.test`	Create up to three sample frame(s) on the screen (instead of PNG files)
`tmp.dir`	A directory where temporary PNG files for each frame will be created, character.
`tmp.files.delete`	Delete the temporary PNG files when done, T/F
`prompt.continue`	Whether to present a summary of the encoding settings and get user confirmation before continuing, T/F
`fn.mov`	The path and filename of output Quicktime *.mov file. If NULL a filename will be automatically generated
`fn.mov.dir`	The directory where the animation will be saved (ignored if a value for `fn.mov` is passed)
`fn.mov.exists`	What to do if the animation file already exists: "auto.increment", "overwrite", "stop", or "ask".
`duration`	The desired duration of the animation (in seconds)
`fps`	A numeric value for frames per second
`skip`	Output every nth frame. To include every frame set skip=1. Integer.
`ffmpeg`	The name of the ffmpeg file. See notes.
`create.mov`	Whether to actually create the mov file. Set to FALSE preview a few frames without actually encoding them.

`info.only`	Only return info
`shp.csv`	The path and filename of a csv file that contains information about shapefiles, including layer names, file, and symbology.
`layers`	The name(s) of layers in shp.csv to display in the background. Will be displayed using the symbology in shp.csv. Character vector or comma delimited string.
`tiff.fn`	The path and name of a GeoTIFF file (e.g., satellite image) that will be displayed in the background. See notes.
`tiff.pct`	Whether or to convert the GeoTIFF to an indexed 256 color RGB image, which may speed up drawing. T/F.
`tiff.bands`	A vector of threee integers corresponding to the bands of the GeoTIFF image that will be mapped to the red, green and blue color guns respectively.
`tiff.buff`	A numeric buffer distance that the range of the plot will be expanded so the points are not right on the edge of the GeoTIFF.
`tiff.fill.plot`	Whether to fill the entire plot area with the GeoTIFF. T/F.
`bg2png`	Save the plot background elements as a static raster image (to improve speed), ignored if `screen.test=TRUE`
`crop.layers.to.extent`	
	Whether to crop the shapefile layers to the view extent (may speed up drawing time)
`date.bar`	The height of the lower section of the plot to devote to the time bar, in inches. To hide the time bar completely, set `date.bar=0`.
`date.bar.bins`	The target number of bins (tick marks + 1) on the time bar (integer)
`col.db`	A single color value for the date bar axes / tick labels, character
`cex.axis.db`	Character expansion factor for the labels on the date bar axis.
`beep`	Beep when one, T/F
`report.time`	Show the time taken when done, T/F
`status`	Show progress bar and status messages

Value

A list with information about each *.mov file created. Each element of the list is another list with two elements: fn (the full filename) and dim (a two-element numeric vector with the frame width and height). If no *.mov file(s) were created, returns NULL.

Note

To create the animation, two and only two of the following parameters must be passed: `duration`, `fps`, and `skip`. The third parameter will be computed. To include every frame, pass `fps`, set `skip=1`, and leave `duration` out.

Larger values for `fps` will result in the animation running 'faster'. Values between 10 and 20 often work well; beyond 30 fps the eye can't keep up with the motion Note if you pass values for `fps` and `duration`, an appropriate value for `skip` will be computed but the final duration of the animation may not be exactly equal to `duration` because only interger values of `skip` are allowed.

One will normally want to run the function a few times without actually encoding to tweak the frame layout (e.g., where the date label and legend appear). To see what a frame in the output will *approximately* look like, set `screen.test=TRUE`. Once the screen sample looks good, next set `max.frames=3`, `tmp.dir="."` (or another folder), `create.mov=FALSE`, and `tmp.files.delete=FALSE`.

This will generate a few sample frames in PNG files and not delete them so you can inspect them. Once these look good, create the full animation by setting `create.mov=TRUE`.

If frame.method is 'auto', the script will use time-based frames when multiple individuals are being animated simultaneously, and location-based frames otherwise.

`duration` (the duration of the animation in seconds) should not be confused with `frame.rtd` which is the real-time duration of a single frame in seconds (e.g., `frame.rtd=3600` means each frame will represent 1 hour).

Passing a positive value for `clean.multihull.frames` (default value is 4) enables time-shifting hulls if needed to prevent two hulls appearing in one frame and no hulls in the previous or next frame (which produces jerky playback when animated). The maximum amount of time a hull can be shifted is calculated by `frame.rtd / clean.multihull.frames`. Thus for example if `frame.rtd=7200` (two hours), and `clean.multihull.frames=4`, then if a frame has two hulls appearing in it and the parent point of one of those hulls lies within 30 minutes (2 hours / 4) of the beginning or end of the frame, *and* there is no hull in the adjacent frame, that hull will be moved to the earlier / later frame. To disable this effect, set `clean.multihull.frames = 0`.

If `date.bar` is too small or too large, you might get a 'margins too large' error. Try values around 1, or hide the date bar completely by setting `date.bar=0`.

The output animation is encoded in QuickTime format. The Quicktime file is encoded using the 'animation' codec, a lossless format that 'scrubs' well (i.e., you can drag the scroll bar to view frame by frame). This requires installing the open source encoding program ffmpeg. ffmpeg is a command line program that Linux and Windows users can download from http://ffmpeg.org/download.html. Windows users should save the ffmpeg.exe file to the working directory or a directory on the path environment variable (e.g., c:\windows). Mac users can download ffmpegX from http://ffmpegx.com/download.html but this has not been tested (pass name to `ffmpeg`).

If ffmpeg is not available, you can still use this function to generate the individual frames and then use another utility (e.g., ImageMagick, Quicktime Pro) to combine the frames into a video file. For best results use a 'lossless' compression method in the encoding program. To create the individual frames only, set `tmp.dir="."` (the working directory) and `tmp.files.delete=FALSE`.

If `fn.mov.exists = "auto.increment"`, a two-digit number will be appended to the *.mov filename to avoid overwriting an existing file

---

lhs.exp.shp   *Export a LoCoH-hullset to shapefile*

Description

Export hulls, hull parent points, and/or isopleths from a locoh-hullset object to ESRI's shapefile format

Usage

```
lhs.exp.shp(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, hpp = FALSE, hulls = FALSE, iso = FALSE,
  iso.idx = NULL, iso.metric = NULL, ellipses = FALSE, dr = FALSE,
  dr.idx = NULL, allpts = FALSE, nn = FALSE, dir = ".",
  file.base = "", file.base.auto = TRUE, avl.file = NULL, status = TRUE,
  show.time = TRUE, hm = "all", anv = NULL, hsp = NULL,
  metadata = TRUE)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of individuals to export
k	The k value of hullsets to export
r	The r value of hullsets to export
a	The a value of hullsets to export
s	The s value of hullsets to export
hs.names	The name(s) of saved hullsets to export
hpp	Export hull parent points (as a point shapefile). T/F
hulls	Export hulls (as a polygon shapefile). T/F
iso	Export isopleths (as a polygon shapefile). T/F
iso.idx	Numeric vector of the indices of the isopleths that will be exported (acts a filter)
iso.metric	Hull sort metric(s) for the isopleths that will be exported (acts a filter). Character vector.
ellipses	Export ellipses (as a polygon shapefile). T/F
dr	Export directional routes. T/F
dr.idx	Numeric vector of the indices of the isopleths that will be exported (acts a filter)
allpts	Export all points (as a point shapefile). T/F
nn	Export nearest neighbors (as a multipoint shapefile). T/F
dir	The directory where the shapefiles will be placed (use "." for the working directory, and "~" for the user directory)
file.base	The base of the file name without any extension (the script will append to this)
file.base.auto	Whether to automatically generate the file name base from the id, s-value, and k-a-r value. Ignored if file.base is passed. T/F
avl.file	The name of an existing avl (ArcView legend) file
status	Show messages. T/F
show.time	Report time for script to complete. T/F
hm	The name of hull metric(s) that will be included in the attribute table. Default is 'all'. Applicable only to hulls and hull parent points. Character vector or comma separated string.
anv	The name of ancillary variables(s) of the parent point that will be included in the attribute table. Default is NULL. Applicable only to hulls and hull parent points. Character vector or comma separated string.
hsp	A list containing one hull scatterplot object with regions saved (i.e., object returned by [lhs.plot.scatter](#)), or the index of a hull scatter plot saved in the hullset (see [lhs.hsp.add](#)). If passed, a column for the region number will be included in the attribute table. Applies only to hulls and hull parent points.
metadata	Export field descriptions in a meta data file (not yet supported)

Note

Filenames for the resulting shapefiles are automatically generated, but you can specify the directory with the `dir` argument and the base of the file name with `file.base`. The constructed filenames will consist of a base followed by the name of the hullset and a suffix indicating the type of feature saved (e.g., '.hulls', '.pts', ".iso", etc.). If a shapefile already exists with the same name, a unique filename will be constructed by appending a two-digit numeric extension.

Most of the time, exporting hull parent points (`hpp=T`) is the same as exporting all points (`allpts=T`), however in some cases not all points have enough neighbors to make a hull so they wouldn't be included in exported of hull parent points. Only hull parent points will have the corresponding hull metrics included in the attribute table.

`avl.file`, if passed, is the name of an ArcView 3.x legend file that contains symbology info. If a value is passed, the script will make a copy of avl.file with the same base name as the shapefile, so that symbology will automatically created in ArcMap. `avl.file` is presumed to be relative to the working folder.

See Also

[lhs.exp.csv](), [hulls](), [isopleths]()

---

lhs.filter.anv	*Define subsets of hulls based on an ancillary variable*

Description

Define subsets of hulls based on an ancillary variable

Usage

```
lhs.filter.anv(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, anv.var = NULL, anv.val = NULL, label = NULL,
  col = NULL, status = TRUE)
```

Arguments

lhs	A [LoCoH-hullset]() object
id	The name(s) of the individual(s) to include. Character vector or comma-delimited character.
k	The k value of hullsets to include
r	The r value of hullsets to include
a	The a value of hullsets to include
s	The s value of hullsets to include
hs.names	The name(s) of saved hullsets to include
anv.var	The name of a single ancillary variable in the LoCoH-hullset object
anv.val	The value(s) of the ancillary variable that define each group
label	Character vector for the labels for each subset. If omitted the value of the ancillary variable will be used
col	Vector of color values (one per group). If omitted colors drawn from a rainbow pallete will be used
status	Show status messages. T/F

Details

This will return a list that defines subsets of hulls based on the hull parent point's value of an ancillary variable. This can be passed to several functions, including `plot.locoh.lhs` (future), and `lhs.plot.scatter`, to create plots of subsets of hulls.

`anv.val` is a vector of values of `anv.var` that will be used to define the group(s). For a hull to be included in a group, its value of `anv.var` must equal one of the values in `anv.val`. If `anv.val` is omitted, one group will be created for each unique value of `anv.var`. Defining groups based on a range of values (lower and upper limits) is not yet supported, but you could create a new ancillary variable that classifies ranges of values into discrete groups.

Note that this function can only return subsets for *one* hullset. This means that either `lhs` must contain a single hullset, or other parameters (e.g., `id`, `hs.names`, `k`, `r`, and/or `a`) are passed to select one and only one of the hullsets in `lhs`.

Value

A list that defines subsets of hulls, with one element for each region in `hsp`. Each element is a list with three elements:

idx indices of the hull parent points

label the label for the subset

col the color

---

lhs.filter.hsp   *Define subsets of hulls based on the location of the parent point in hull scatterplot space*

Description

Define subsets of hulls based on the location of the parent point in hull scatterplot space

Usage

```
lhs.filter.hsp(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, hsp = NULL, reg.idx = NULL, label = NULL,
  col = NULL)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of the individual(s) to include in the plot. Character vector or comma-delimited character.
k	The k value of hullsets to create isopleths for
r	The r value of hullsets to create isopleths for
a	The a value of hullsets to create isopleths for
s	The s value of hullsets to create isopleths for
hs.names	The name(s) of saved hullsets to create isopleths for

hsp	Either the index of a hull scatterplot(s) saved in lhs (use `summary.locoh.lhs` to which how many hsp objects have been saved), or a list of objects of class `locoh.hsp` (i.e., the return value of `lhs.plot.scatter`.
reg.idx	A numeric vector of the indices of the regions in `hsp` to include in the filter
label	Character vector of the labels to use for each subset. If not passed, the saved label values in `hsp` will be used
col	Vector of color values (one per region). If omitted the colors saved in `hsp` will be used

Details

This will return a list that defines subsets of hulls grouped according to which manually-digitized region in scatterplot space the hull parent point falls. This can be passed to several functions, including `plot.locoh.lhs` (future), and `lhs.plot.scatter`, to create plots of subsets of hulls.

The scatterplot space, including the manually defined regions, are saved in a `locoh.hsp` object. Typically hsp will be created using the `lhs.plot.scatter` function. hsp objects can also be saved in the lhs object (see `lhs.hsp.add`.

Note that this function can only return subsets for *one* hullset and *one* hsp object. This means that either lhs must contain a single hullset, or other parameters (e.g., id, hs.names, k, r, and/or a) are passed to select one and only one of the hullsets in lhs.

Value

A list that defines subsets of hulls, with one element for each region in hsp. Each element is a list with three elements:

idx indices of the hull parent points

label the label of the region of hsp

col the color of the region in hsp

---

lhs.hsp.add	*Adds a hull scatterplot to a LoCoH-hullset object*

Description

Adds a hull scatterplot to a LoCoH-hullset object

Usage

```
lhs.hsp.add(lhs, hsp.lst)
```

Arguments

lhs	A LoCoH-hullset object
hsp.lst	A list of object(s) of class locoh.hsp

Note

hsp objects are typically created with `lhs.plot.scatter` function. They can then be 'saved' in a LoCoH-hullset object with this function.

Examples

```
## Display a scatter plot and manually draw four regions on it
## hsp <- lhs.plot.scatter(lhs, x="area", y="par", regions=4)

## Save hull scatter plot as part of the hullset
## lhs <- lhs.hsp.add(lhs, hsp.lst=hsp)
## summary(lhs)
```

---

lhs.hsp.del   *Delete a hull scatterplot from a LoCoH-hullset object*

Description

Deletes a saved hull scatterplot from a LoCoH-hullset object

Usage

```
lhs.hsp.del(lhs, hsp.idx = "all", status = TRUE)
```

Arguments

lhs	A LoCoH-hullset object
hsp.idx	The index(s) of the hullsets to delete. A numeric vector or 'all'
status	Show messages, T/F

Value

A LoCoH-hullset object

Note

Use the summary() command to see the indices of saved hull scatterplots

---

lhs.ids.get   *Get the ids in a hullsets*

Description

Get the ids in a hullsets

Usage

```
lhs.ids.get(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, hs.idx = NULL)
```

Arguments

lhs	A [LoCoH-hullset](#) object
id	The name(s) of individuals to extract
k	The k value of hullsets to extract
r	The r value of hullsets to extract
a	The a value of hullsets to extract
s	The s value of hullsets to extract
hs.names	The name(s) of saved hullsets to extract
hs.idx	The indices of saved hullsets to extract

Details

This function returns the ids of specified hullsets

Value

A character vector of ids

See Also

[lhs.select](#)

---

lhs.iso.add   *Compute isopleths*

Description

Adds isopleth(s) to a LoCoH-hullset object

Usage

```
lhs.iso.add(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, sort.metric = c("auto", hm.expr(names.only = T, desc = F,
  print = F))[1], iso.levels = c(0.1, 0.25, 0.5, 0.75, 0.95),
  iso.method = c("pt.quantiles", "hm.vals")[1], iso.cap.method = c(">=",
  "<=")[1], scale.iso.levels.to.hm.vals = (iso.method == "hm.vals" &&
  max(iso.levels <= 1)), subset.metric = NULL, subset.vals = NULL,
  allow.gpc = TRUE, sliver_check = TRUE, beep = FALSE, status = TRUE,
  ...)
```

Arguments

lhs	A LoCoH-hullset object
id	The id(s) of the hullsets to create isopleths for
k	The k value of hullsets to create isopleths for
r	The r value of hullsets to create isopleths for
a	The a value of hullsets to create isopleths for
s	The s value of hullsets to create isopleths for

`hs.names`	The name(s) of saved hullsets to create isopleths for
`sort.metric`	The name of a hull metric that will be used to sort the hulls prior to merging into isopleths
`iso.levels`	A numeric vector of the levels of the isopleths. See details.
`iso.method`	The method used to define isopleths. Default is "pt.quantiles" which defines isopleths as containing a quantile of points (e.g., the 0.1th isopleth contains 10% of the points). Can also be "hm.vals", in which case the isopleth level represents not a proportion of points enclosed but a value of the hull metric (e.g., the 0.1 isopleth contains hulls whose hull metric is <= 0.1)
`iso.cap.method`	A character object specifying how isopleths defined by proportions of enclosed points will be 'capped'. Ignored when `iso.method="hm.vals"`. See details.
`scale.iso.levels.to.hm.vals`	
	Whether to linearly scale iso.levels (presumed to be 0..1) to the minimum and maximum of the sort.metric value. This is used in conjunction iso.method='hm.vals', to produce isopleths as the aggregation of hulls whose sort.metric value is >= i% of the maximum, where i is the isopleth level. T/F
`subset.metric`	The name of a hull metric that will be used to create subsets of isopleths
`subset.vals`	A two-column data frame (or matrix) containing the lower and upper bounds of the hull metric in subset.metric, from which subsets of hulls will be extract for separate isopleth construction. Ignored if subset.metric is not provided. Can also be a character string in the form of "equal intervals, n", or "quantitles, n" where n is the number of strata desired (shorthand versions of these commands are "ei,n" and "q,n" where n is a number). The script will compute the break points either spread equally between the minimum and maximum subset metric value (equal interval) or so an equal number of hulls is in each strata
`allow.gpc`	Allow functions from the gpclib package if the functions from rgeos fail. See details. T/F
`sliver_check`	Whether to check for and delete slivers in isopleths. See details. T/F
`beep`	Beep when done. T/F
`status`	Show status messages. T/F
`...`	Additional auxillary parameters for the hull sort metric

Details

This function creates isopleth(s) for a LoCoH-hullset object. This involves sorting hulls by one of the hull metrics, and then cummulatively unioning them together until a certain level is reached at which point the union is saved as an isopleth. By default, hulls will be sorted according to the area (for k-method) or number of enclosed points (r- and a-methods), producing density isopleths (i.e., utilization distributions).

When `iso.method=="pt.quantiles"` (the default), isopleths will be defined by the proportion of points they enclose (given in `iso.levels`). Thus for example the 50% isopleth contains 50% of the total points. The default isopleth levels are 0.15, 0.25, ..., 0.95. To get an isopleth that encloses all points in the dataset, `iso.levels` must include '1'.

When `iso.method=="hm.vals"`, isopleths will be defined by the value of the hull metrics (provided in `iso.levels`). So if for example `sort.metric="scg.nn.mean"` (average speed of all points identified as nearest neighbors), and `scg.nn.mean` varied from 0.01 to 3.2, the 0.6 isopleth would be the union of all hulls whose average nearest neighbor speed was 0.6 or less. If you would like hulls to be identified by hull metric values, but don't know the range of hull metric values, you can set isopleth levels between 0..1 and set `scale.iso.levels.to.hm.vals=TRUE`.

By default, isopleths defined by quantiles of enclosed points are 'capped' as the smallest number of progressively unioned hulls that enclose a number of points equal to or greater than the isopleth level. For example, if there are 1000 points in the dataset, the 50% isopleth would be the smallest union of hulls that encloses 500 or more points. Alternately, when iso.cap.method = "<-", the isopleth will be the largest number of hulls which enclose up to but not more than the isopleth level. The actual number of points enclosed by each isopleth is included in the data table for the isopleth (which you can view by setting iso.details=TRUE when calling summary.locoh.lhs).

The hull metric used for sorting must already have been computed. Several hull metrics (e.g., hull area, number of enclosed points) are 'automatically' computed when a hullset is initially generated (see lxy.lhs). Other hull metrics must be created separately with functions such as lhs.ellipses.add and lhs.visit.add. You can use the summary.locoh.lhs function to see which isopleths have been saved in a LoCoH-hullset object.

Hulls are unioned using the gUnion and gUnaryUnion functions from the rgeos package, which in general is very fast. These functions occasionally fail when lines are too close together or there are other topological conditions. This is not a problem with the data, but an inherent limitation of the algorithms / processing. If allow.gpc=TRUE, functions from the gpclib package will be used as backup. If sliver_check=TRUE, isopleths will checked for 'slivers'. Slivers (as used here) refer to polygons or holes with <3 unique nodes. Slivers can be caused by reductions in numeric precision or rounding errors, and are not uncommon.

Value

A LoCoH-hullset object

See Also

isopleths

Examples

```
# Create 0.5 and 0.95 isopleths. By not specifying the sort.metric,
# density isopleths will be created by default, with hulls sorted
# by area (k-method) or number of enclosed points (r and a method)

# lhs <- lhs.iso.add(lhs, iso.levels=c(0.5, 0.95))

# Compute hull metrics for a 24 hour inter-visit gap, then produce isopleths with hulls sorted by
# the number of separate visits (e.g., visitation)
# lhs <- lhs.visit.add(lhs, ivg=3600.24)
# lhs <- lhs.iso.add(lhs, sort.metric="nsv", ivg=3600*24)
```

---

lhs.iso.del   *Delete an isopleth*

Description

Delete isopleth(s) from a LoCoH-hullset object

Usage

```
lhs.iso.del(lhs, iso.idx = "all")
```

Arguments

`lhs`	A LoCoH-hullset object
`iso.idx`	The index(s) of isopleths to delete (numeric vector or 'all')

Value

A LoCoH-hullset object

Note

To see the indices of isopleths, run `summary.locoh.lhs`

---

lhs.merge   *Merge LoCoH-hullsets*

Description

Merge LoCoH-hullsets

Usage

```
lhs.merge(..., check.class = TRUE)
```

Arguments

`...`	Two or more LoCoH-hullset objects
`check.class`	Whether to enforce that all objects merged belong to class locoh.lhs (T/F)

Value

A LoCoH-hullset object

Note

Hullsets in different objects can be merged together, but the hullsets themselves remain separate. If duplicate hullsets are detected (e.g., same id, same mode, same parameter value), an error message will appear.

lhs.pep.add	*Add hull metrics for proportion of enclosed points by each hull for an ancillary variable*

Description

Computes proportion of enclosed points of a second set of points

Usage

```
lhs.pep.add(lhs, pep.var, pep.val = NULL, npep = TRUE, status = TRUE)
```

Arguments

lhs	A [LoCoH-hullset](#) object
pep.var	Name(s) of ancillary variable(s) saved with the LoCoH-hullset.
pep.val	Vector of value(s) for which the percentage of enclosed points will be calculated. If NULL, all unique values of pep.var will be used.
npep	Whether to compute the normalized proportion of enclosed points hull metric (normalized by the proportion of points in the entire dataset), T/F
status	Show messages, T/F

Value

A LoCoH-hullset object

Note

The proportion of enclosed points is a hull metric that measures relative association among N individuals when you have movement data for multiple individuals simultaneous over the same tim period. The general idea is to create hulls for all of the locations combined (ignoring which location was for each individual), then for each hull look at the proportion of enclosed points for each individual. If the individuals ignored each other, one would expect the proportion of enclosed points for any given hull to be approximately equal to the proportions of each individual in the entire dataset. Deviations from this random mixing null model reflect places where the individuals did not mix evenly (e.g., one dominated).

Do compute pep, all points must have the same id (beause you create hulls with the combined dataset) with the original id values saved as an ancillary variable (see [lxy.id.new](#)). Create hulls as you normally would using the 'a' or 'k' method, although it would be recommneded to omit time (let s=0) because the time difference between locations means different things for different pairs of points depending on whether they are from the same or different individuals. Pass the name of the ancillary variable that contains the original ids as pep.var and the original id value(s) as pep.val (if pep.val is omitted the pep metric will be computed for all id values found in pep.var. *npep* (normalized proportion of enclosed points) normalizes pep by the proportion of that individual in the entire dataset, such that *npep=1* means the individual was in the hull in the same proportion as it was in the entire dataset. '

See Also

[lxy.id.new](#)

| lhs.plot.isoarea | *Plot the area of isopleths as a function of the k/a/r parameter* |

Description

Creates a plot of the area of isopleths in a LoCoH-hullset area

Usage

```
lhs.plot.isoarea(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, sort.metric = NULL, iso.idx = NULL,
  figs.per.page = 1, legend = TRUE, title = NULL, title.show = TRUE,
  subtitle = TRUE, mar = c(3.3, 3.2, if (title.show) (if (subtitle) 3.2 else
  2.3) else 0.5, 0.5), mgp = c(2, 0.7, 0), png.fn = NULL, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.pointsize = 12 + (png.width - 480)/80, png.fn.pre = NULL,
  png.fn.suf = NULL, png.overwrite = TRUE, panel.num = NULL,
  panel.num.inside.plot = !title.show, bg = "white", legend.space = if
  (legend) 0.05 else 0, ...)
```

Arguments

lhs	A LoCoH-hullset object
id	The id(s) of the individual(s) to include in the plot
k	A k-value for the number of nearest neighbors around each point to include in the plot
r	A r-value for the number of nearest neighbors around each point to include in the plot
a	A a-value for the number of nearest neighbors around each point to include in the plot
s	The s value(s) of nearest neighbor sets to include in the plot. If NULL, all values will be used
hs.names	The name(s) of saved hullsets to analyze
sort.metric	The name(s) of isopleth sort metrics to include in the plot
iso.idx	The indices of isopleths to include in the plot
figs.per.page	Number of plots per page
legend	Whether to include a legend. T/F.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F.
subtitle	Whether to add a subtitle to the automatically constructed title (when title=NULL, otherwise ignored)
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
png.fn	A filename for a PNG file

png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
png.fn.pre	A prefix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.fn.suf	A suffix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F.
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. Ignored if panel.num is NULL. T/F
bg	Background color
legend.space	The amount of additional space on the lower end of the x-axis to make room for the legend. Expressed as a proportion of the range of the x-axis values
...	Additional parameters that will be passed to the plot function

---

lhs.plot.isoear   *Plot the isopleth edge:area ratio as a function of k/a/r*

Description

Creates a plot of the ratio of edge to area for isopleths

Usage

```
lhs.plot.isoear(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, sort.metric = NULL, iso.idx = NULL,
  figs.per.page = 1, legend = TRUE, title = NULL, title.show = TRUE,
  subtitle = TRUE, mar = c(3.3, 3.2, if (title.show) (if (subtitle) 3.2 else
  2.3) else 0.5, 0.5), mgp = c(2, 0.7, 0), png.fn = NULL, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.pointsize = 12 + (png.width - 480)/80, png.fn.pre = NULL,
  png.fn.suf = NULL, png.overwrite = TRUE, panel.num = NULL,
  panel.num.inside.plot = !title.show, bg = "white", legend.space = if
  (legend) 0.05 else 0, ...)
```

Arguments

lhs	A LoCoH-hullset object
id	The id(s) of the individual(s) to include in the plot
k	A k-value for the number of nearest neighbors around each point to include in the plot
r	A r-value for the number of nearest neighbors around each point to include in the plot
a	A a-value for the number of nearest neighbors around each point to include in the plot
s	The s value(s) of nearest neighbor sets to include in the plot. If NULL, all values will be used
hs.names	The name(s) of saved hullsets to analyze
sort.metric	The name(s) of isopleth sort metrics to include in the plot
iso.idx	The indices of isopleths to include in the plot
figs.per.page	Number of plots per page
legend	Whether to include a legend. T/F.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F.
subtitle	Whether to add a subtitle to the automatically constructed title (when title=NULL, otherwise ignored)
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
png.fn	A filename for a PNG file
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
png.fn.pre	A prefix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.fn.suf	A suffix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F.
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. Ignored if panel.num is NULL. T/F
bg	Background color
legend.space	The amount of additional space on the lower end of the x-axis to make room for the legend. Expressed as a proportion of the range of the x-axis values
...	Additional parameters that will be passed to the [plot](#) function

---

lhs.plot.revisit *Plot distribution of revisit times*

Description

Plots a histogram of the revisit times a LoCoH-hullset area

Usage

```
lhs.plot.revisit(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, ta.min = "auto", ta.max = NULL,
  breaks = 40, xaxis.vals = NULL, ta.min.auto.tau = 4, col = "gray80",
  figs.per.page = 1, title = NULL, title.show = TRUE, subtitle = TRUE,
  mar = c(2.8, 3.2, if (title.show) (if (subtitle) 3.2 else 2.3) else 0.5,
  0.5), mgp = c(1, 0.7, 0), png.fn = NULL, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.pointsize = 12 + (png.width - 480)/80, png.fn.pre = NULL,
  png.fn.suf = NULL, png.overwrite = TRUE, panel.num = NULL,
  panel.num.inside.plot = !title.show, bg = "white", legend.space = if
  (legend) 0.05 else 0, ...)
```

Arguments

lhs	A LoCoH-hullset object
id	The id(s) of the individual(s) to include in the plot
k	A k-value for the number of nearest neighbors around each point to include in the plot
r	A r-value for the number of nearest neighbors around each point to include in the plot
a	A a-value for the number of nearest neighbors around each point to include in the plot
s	The s value(s) of nearest neighbor sets to include in the plot. If NULL, all values will be used
hs.names	The name(s) of saved hullsets to plot
ta.min	The minimum time away (in seconds) to include on the histogram , can also be 'auto'
ta.max	The maximum time away (in seconds) to include. If NULL then no upper limit will be imposed
breaks	A value for 'breaks' that will be passed to the hist function (see help function for breaks)
xaxis.vals	A numeric vector of time-away values (in seconds) that will be labeled on the x-axis.
ta.min.auto.tau	
	The minimum time away to include in 'auto' expressed as the number of median sampling intervals. Ignored if ta.min is not 'auto'.
col	A color value for the histogram bars
figs.per.page	Number of plots per page

`title`	The title to be displayed. Character. If NULL a title will be constructed.
`title.show`	Whether to show the title. T/F.
`subtitle`	Whether to add a subtitle to the automatically constructed title (when `title=NULL`, otherwise ignored)
`mar`	The plot margins. A four item numeric vector
`mgp`	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
`png.fn`	A filename for a PNG file
`png.dir`	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
`png.dir.make`	Whether to create png.dir if it doesn't exist. T/F
`png.width`	The width of the PNG image
`png.height`	The height of the PNG image
`png.pointsize`	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
`png.fn.pre`	A prefix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
`png.fn.suf`	A suffix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
`png.overwrite`	Whether to overwrite an existing PNG file if it exists. T/F.
`panel.num`	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
`panel.num.inside.plot`	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. Ignored if panel.num is NULL. T/F
`bg`	A color value for the plot background
`legend.space`	The amount of additional space on the lower end of the x-axis to make room for the legend. Expressed as a proportion of the range of the x-axis values
`...`	Additional parameters that will be passed to the [hist](#) function
`legend`	Whether to include a legend. T/F.

Details

This function will plot a histogram of the revisitation times for a hullset. Revisitation time is simply the time between points enclosed in a hull. Basically, all points enclosed by each hull are examined and their time intervals computed. This may help reveal where there are natural temporal cycles in revisitation, for example if you see a spike in revisitation around 24 hours there may be daily revisitation pattern in some of the hulls. You can specify the minimum amount of time-away to show in the histogram (e.g., if enclosed points that are separated by the median sampling interval are of little interest), as well as the maximum time-away period. Note that revision metrics do *not* have to be computed for the hullset for the histogram to be computed.

Value

A list of objects of class histogram (one for each hullset plotted)

See Also

[lhs.visit.add](#)

lhs.plot.scatter	*Create scatterplot of hull metrics*

Description

Multi-purpose scatterplot function for the hull metrics in a LoCoH-hullset

Usage

```
lhs.plot.scatter(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, x.axis = NULL, y.axis = NULL,
  limx = NULL, limy = NULL, trans.x = NULL, trans.y = NULL,
  jiggle.x = "auto", jiggle.y = "auto", filter = NULL,
  filter.label.in.subtitle = TRUE, filter.sampsize.in.subtitle = TRUE,
  filter.col.use = TRUE, filter.axes.uniform = TRUE, title = NULL,
  title.show = TRUE, title.hs.name.include = TRUE, title.two.id = FALSE,
  title.axes = TRUE, col = c("gray50", "spiral")[1], bg = NULL,
  mar = c(4, 3.2, if (title.show) (if (title.hs.name.include) 3.9 else 3.2)
  else 0.5, 0.5), mgp = c(2.1, 0.8, 0), lo.save = TRUE,
  lo.margins.set = TRUE, lo.colors.set = TRUE, lo.bg.as.box = FALSE,
  sat.base = NULL, val.base = NULL, hue.offset = NULL,
  center.method = c("bbox", "mean")[2], cex = 0.6, type = c("p", "l",
  "b")[1], ufat = NULL, figs.per.page = 1, add = FALSE, regions = NULL,
  prompt.labels = TRUE, hsp = NULL, hsp.reg.col = TRUE,
  hsp.reg.out = FALSE, hsp.reg.lbl = FALSE, hsp.override = TRUE,
  png.fn = NULL, png.dir = NULL, png.dir.make = TRUE, png.width = 800,
  png.height = png.width, png.pointsize = 12 + (png.width - 480)/80,
  png.fn.pre = NULL, png.fn.mid = NULL, png.fn.suf = NULL,
  png.exists = c("overwrite", "skip", "abort")[1], status = TRUE,
  desc = c(NONE <- 0, BOTTOM <- 1, TOP <- 3)[ifelse(figs.per.page == 1, 2,
  1)], cex.desc = 0.8, col.desc = NULL, panel.num = NULL,
  panel.num.inside.plot = !title.show, hmap = NULL,
  hmap.in.subtitle = TRUE, check.ap.value.in.hmparams = TRUE, ...)
```

Arguments

lhs	A LoCoH-hullset object
id	The name(s) of the individual(s) to include in the plot. Character vector or comma-delimited character.
k	The k value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
r	The r value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
a	The a value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
s	The s value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
hs.names	The name(s) of saved hullsets to include in the plot.
x.axis	The name of a hull metric (see `hm.expr`)

y.axis	The name of a hull metric
limx	The lower and upper limits of the x-axis. Two-element numeric vector.
limy	The lower and upper limits of the y-axis. Two-element numeric vector.
trans.x	The name of a function that will be used to transform the x-axis values. Can be any R function (e.g., "log", "sqrt"), also "square" and "cube"
trans.y	The name of a function that will be used to transform the y-axis values
jiggle.x	"auto" or a numeric value which will be used to add a normally distributed stochastic value (mean 0, sd=jiggle.x) to the x-axis values for better visualization of the number of points in each group (helpful when the x-values take on discrete values)
jiggle.y	"auto" or a numeric value which will be used to add a normally distributed stochastic value (mean 0, sd=jiggle.y) to the y-axis values
filter	NULL or a list of filter parameters. See details.
filter.label.in.subtitle	
	Whether to add the filter label to the plot subtitle. Ignored if title is passed or filter is NULL. T/F.
filter.sampsize.in.subtitle	
	Whether to add the filter sample size to the plot subtitle. Ignored if title is passed or filter is NULL. T/F.
filter.col.use	Whether the points will be displayed with the color(s) saved in the filter (overriding the 'col' parameter). Ignored if filter is NULL. T/F.
filter.axes.uniform	
	Whether the upper and lower limits of the axes for each plot will be set as the limit for the entire combined dataset. Ignored if filter is NULL or values are passed for limx or limy. T/F.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F.
title.hs.name.include	
	Whether to include the hullset name as part of the title. Ignored if title is passed. T/F.
title.two.id	An ad-hoc way to construct a title consisting of the ids of the hullset and hs2
title.axes	Whether to include the names of the hull metrics on the axes as part of the subtitle. Ignored if title is passed. T/F.
col	A single color value, or a vector of color values of the same length as the number of hulls, or "spiral" (see additional parameters below). Ignored if filter.col.use=T or something is passed for hsp (in which case col will be extracted from hsp).
bg	Background color
mar	The plot margins. A four item numeric vector.
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector.
lo.save	Whether to save and reset the plot device margin settings (some wrapper functions that call this function don't want device settings reset). T/F.
lo.margins.set	Whether to save and reset the plot device margin settings (some wrapper functions that call this function don't want device settings reset). T/F.
lo.colors.set	Whether to set the background, foreground color on the plot device. T/F.

lo.bg.as.box	Whether to display the background color as a box on the plot (needed by some wrapper function that want the axes and margins to be a different color). This overrides the setting of lo.colors.set. T/F.
sat.base	A number between 0 and 1 for the base staturation (how much of the color is present) of the center color (s=0.4 gives pastels), ramping up to 1 (full color) at edges.
val.base	A number between 0 and 1 controlling how much black is in the center color (0 is total black), ramping out to no blackness at the edges
hue.offset	A number between 0 and 2*pi for the color wheel rotation (in radians), which controls the color of points directly to the right of the center.
center.method	Determines how the center of the color will be computed. "bbox" = the center of the bounding box of the full range of points; "mean" = the mean of the data splot
cex	Expansion factor for the points on the scatterplot
type	The type of plot: 'p'=points only, 'l'=line, 'b'=both
ufat	Whether to substitute user-friendly axis titles. T/F.
figs.per.page	The number of plots per page
add	Whether to add to the current plot device. T/F.
regions	Determines whether the user will be prompted to draw regions (polygons) on the scatterplot window with the mouse, that will be saved as part of the scatterplot objecr returned. Values can be 1) a number whereby the user will be prompted to create N regions with randomly assigned colors, or 2) a vector of color values in which case the regions will be assigned those colors.
prompt.labels	Whether the user should be prompted to also enter labels for each of the regions drawn. Ignored if regions = NULL. T/F.
hsp	Either the index(s) of a hull scatterplot(s) saved in lhs (use summary() command to which how many hsp objects have been saved), or a list of objects of class locoh.hsp. When passed, the parameters in hsp will be used to create the scatterplot
hsp.reg.col	Whether to use the colors assigned to the regions in hsp. Ignored if hsp is NULL. T/F.
hsp.reg.out	Whether to display the outline of the regions in hsp. Ignored if hsp is NULL. T/F.
hsp.reg.lbl	Whether to display the labels of the regions in hsp. Ignored if hsp is NULL. T/F.
hsp.override	Whether the parameters in hsp should override other parameters passed. Ignored if hsp is NULL. T/F.
png.fn	The path and name of the PNG file to create (instead of displaying in a plot window).
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed.
png.dir.make	Whether to create png.dir if it doesn't exist. T/F.
png.width	The width of the PNG image. Ignored if png.fn is passed.
png.height	The height of the PNG image. Ignored if png.fn is passed.
png.pointsize	The pointsize (in pixels) for the PNG image (increase to make labels appear larger). Equivalent to the height or width of a character in pixels.

png.fn.pre	A prefix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.fn.mid	A mid-fix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.fn.suf	A suffix that will be used in the construction of the PNG filename. Ignored if png.fn is passed.
png.exists	What to do if a PNG with the same filename already exists: "overwrite", "skip", or "abort"
status	Whether to show messages. T/F.
desc	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
cex.desc	The expansion factor for the descriptive text. Numeric value.
col.desc	The color of the descriptive text. Color value.
panel.num	A number or letter to display in the upper left hand corner of the plot, used when the plot will be part of a multi-frame graphic (as in publications). Character.
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself (as opposed to the title area). T/F.
hmap	A named list of hull metric auxillary parameters, the name of each list element is the name of the variable.
hmap.in.subtitle	
	Include the hmap value(s) in the plot subtitle. T/F
check.ap.value.in.hmparams	
	Check to make sure that hull metrics have been computed for the hull metric auxillary parameter values passed (either as separate arguments or in hmap)
...	Other parameters, including any auxillary parameters required by certain hull metrics

Details

This function has two main purposes. 1) To make scatterplot graphics either in a plot window or PNG file 2) To create a list of objects of class "locoh.hsp", which are basically a collection of the parameters that were used to create the scatterplot including any manually drawn regions created by the user (see regions).

Note that hull metrics must already have been computed. Several hull metrics are 'automatically' computed when the hullset is defined (e.g., hull area, number of enclosed points). Other hull metrics must be created separately with functions such as lhs.ellipses.add and lhs.visit.add. Auxillary parameters required for hull metrics can be passed either as individual parameters (e.g., ivg=3600*12) or as a list element (e.g., hmap=list(ivg=86400)).

filter, if passed, will create scatterplots for subsets of hulls. For this to work, filter must be a list whose elements are named lists with the following elements: *idx* = a vector of the indices of the hulls (in the hulls SpatialPolygonsDataFrame), *label* = a label for the subset, and *col* = a color value. lhs.filter.hsp can be used to create a filter list based on manually drawn regions of a hull scatterplot, and lhs.filter.anv can be used to create a filter based on ranges of values of an ancillary variable. The arguments filter.sampsize.in.subtitle, filter.col.use, and filter.axes.uniform control how the subsets are treated in the scatterplot(s).

Value

If png.fn or png.dir is passed, the plots are exported to PNG file(s) and the function returns a list of file names and image dimensions (in pixels). Otherwise returns a named list whose element(s) are of class *locoh.hsp*. This list of *locoh.hsp* objects can then be 'permanently' saved in the LoCoH-hullset object using `lhs.hsp.add`, and/or fed into other functions such as `plot.locoh.lhs` (to symbolize hull points).

See Also

`lhs.plot.scatter.auto`, `hm.expr`, `lhs.hsp.add`, `lhs.filter.hsp`, `lhs.filter.anv`

---

lhs.plot.scatter.auto   *Generate multiple scatterplots of hull metrics*

Description

This is a wrapper function for `lhs.plot.scatter` that creates scatterplots for pairs of hull metrics. It can quickly generate several few dozen scatterplots for the purpose of visually looking for novel associations between hull metrics.

Usage

```
lhs.plot.scatter.auto(hs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, x.metrics = c("auto", "all")[1],
  y.metrics = c("auto", "all")[1], metrics.exclude = c("bearing",
  "sgf.nn.mean", "sgf.enc.mean"), exclude.same.pair = TRUE,
  exclude.reverse.pair = TRUE, png.dir = NULL, png.dir.make = TRUE,
  png.width = 800, png.height = png.width, png.pointsize = 12 + (png.width
  - 480)/80, png.fn.pre = NULL, png.fn.mid = NULL, png.fn.suf = NULL,
  png.exists = c("skip", "overwrite")[1], progress.bar = TRUE,
  test.sample = NULL, enumerate.pairs.only = FALSE,
  new.plot.window = TRUE, hmap = NULL, ...)
```

Arguments

hs	A LoCoH-hullset object
id	The names of the individual(s) to include in the plot. Character vector or comma-delimited character.
k	The k value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
r	The r value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
a	The a value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
s	The s value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
hs.names	The name(s) of saved hullsets to include in the plot.
x.metrics	The name(s) of hull metric(s) to plot on the x-axis. Can also be "auto" for the most common ones, or "all"

y.metrics	The name(s) of hull metric(s) to plot on the y-axis. Can also be "auto" for the most common ones, or "all"
metrics.exclude	
	The name(s) of hull metrics to exclude from both the x and y axes
exclude.same.pair	
	Whether to exclude a hull metric being plotted against itself. T/F
exclude.reverse.pair	
	Whether to exclude opposite pairs of metrics. T/F
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed.
png.dir.make	Whether to create png.dir if it doesn't exist. T/F.
png.width	The width of the PNG image. Ignored if png.fn is passed.
png.height	The height of the PNG image. Ignored if png.fn is passed.
png.pointsize	The pointsize (in pixels) for the PNG image (increase to make labels appear larger). Equivalent to the height or width of a character in pixels.
png.fn.pre	A prefix that will be used in the construction of the PNG filename
png.fn.mid	A mid-fix that will be used in the construction of the PNG filename
png.fn.suf	A suffix that will be used in the construction of the PNG filename
png.exists	What to do if a PNG with the same filename already exists: "overwrite" or "skip"
progress.bar	Display a progress bar. T/F
test.sample	An optional number of randomly selected pairs of hull metrics to generate for testing purposes. Or NULL.
enumerate.pairs.only	
	Whether to only return a two-column data frame of the names of hull metrics. T/F
new.plot.window	
	Whether to start a new plot window and turn recording on. T/F
hmap	A named list of hull metric auxillary parameters (for hull metrics that involve additional parameters such as an intervisit gap period). These can also be passed on their own (e.g., ivg=3600*24)
...	Additional parameters that will be passed to the [hist](#) function

Value

A data frame containing the pairs of hull metrics

Note

This function generates a list of pairs of hull metrics, and then calls [lhs.plot.scatter](#) repeatedly to generate the histograms. This is a quick way to visualize the relationships between hull metrics.

If `new.plot.window` is TRUE (default), a new plot window will be opened and plot recording turned on so you can flip through the histograms using the PgUp and PgDn keys. You may also pass a value for `png.dir`) which will create a bunch of PNG files which you can then view them as a slideshow.

See Also

[lhs.plot.scatter](#), [hm.expr](#)

---

lhs.save   *Save a LoCoH-hullset object to disk*

Description

Saves a LoCoH-hullset object to disk, constructing a filename if needed that reflects the contents

Usage

```
lhs.save(lhs, file = NULL, dir = ".", suf = NULL, compress = TRUE,
  auto.num.files = TRUE, width = 2, save.as = NULL)
```

Arguments

lhs	A LoCoH-hullset object
file	Optional filename. If omitted a sensible filename will be constructed
dir	Directory for the output file (relative to the working directory), ignored if `file` passed
suf	A suffix that will be used in the construction of the filename, ignored if `file` is passed
compress	Compress file. T/F
auto.num.files	Use a number as part of the constructed filename. T/F
width	The number of digits of the auto-number token in the filename (ignored if `auto.num.files=F`)
save.as	The name of the object when saved (default is the same as the original)

See Also

lxy.save

---

lhs.select   *Select hullsets*

Description

Select hullsets

Usage

```
lhs.select(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, hs.idx = NULL, selection = NULL, status = TRUE)
```

Arguments

lhs	A [LoCoH-hullset](#) object
id	The name(s) of individuals to extract
k	The k value of hullsets to extract
r	The r value of hullsets to extract
a	The a value of hullsets to extract
s	The s value of hullsets to extract
hs.names	The name(s) of saved hullsets to extract
hs.idx	The indices of saved hullsets to extract
selection	A locoh.selection object, see Details
status	Show status messages (T/F)

Details

This function returns a subset of hullsets based on the id value, k, a, r, or s. You may also select hullsets based on hullset names, indices, or a hullset selection object. A *locoh.selection* object is a named list, with one element for each id, containing one or more a, k, or r value(s).

See the `lhs.selection` function in the tlocoh.dev package for details.

Value

A LoCoH-hullset object

---

lhs.visit.add   *Add time use hull metrics to a LoCoH-hullset object*

Description

Computes visitation and duration hull metrics based on inter-visit gap value(s)

Usage

```
lhs.visit.add(lhs, ivg = NULL, check_samp_int = TRUE, status = TRUE)
```

Arguments

lhs	A LoCoH-hullset object
ivg	Value(s) for inter-visit gap (in seconds) (numeric vector)
check_samp_int	Whether or not to check whether `ivg` is greater than the median sampling interval. T/F.
status	Show status messages. T/F.

Value

A LoCoH-hullset object

Note

The inter-visit gap is the period of time (in seconds) which must pass before another occurrence in the hull can be considered a separate visit. Occurences in the hull are considered a separate visit only if the animal was absent from the hull for a period of time >= ivg.

For each ivg value, the function examines each hull and computes the number of separate visits to the hull (visitation, hull metric name = "nsv") and the mean number of locations per visit (visit duration, hull metric name = "mnlv")

---

locoh.lhs   *Class for a hullsets*

Description

Class for a collection of hullsets and associated objects including hull metrics and isopleths

Details

An object of class `locoh.lhs` is a list (S3) containing one or more sets of hulls (local MCPs) for a set of points. One hull is saved for each point if the method used to create the hulls returned at least two nearest neighbors for each point, otherwise the number of hulls might be less than the number of points. In addition to hulls (saved as a SpatialPolygonsDataFrame), LoCoH-hullset object may also contain:

- the original points and their attribute table
- an index of the points a) used to construct the hull and b) enclosed by each hull
- hull metrics
- isopleths (progressive unions of hulls) as both vector (SpatialPolygonsDataFrame) and raster (utilzation distribution) formats.
- saved parameters for scatterplots of hull metrics.

In code examples, objects of class `locoh.lhs` are often noted by a 'lhs' suffix, and functions that are designed to work on LoCoH-hullset objects typically start with 'lhs'.

For a complete description of the data structure, see the vignette *T-LoCoH Data Classes*.

See Also

`lxy.lhs`, `summary.locoh.lhs`, `plot.locoh.lhs`, `lhs.merge`, `lhs.iso.add`, `lhs.save`, `lhs.exp.shp`

`locoh.lxy`   *Class for location data*

Description

Data class for locations with associated dates, nearest neighbor info, parameters for a random walk null model, and assorted meta-data

Details

An object of class `locoh.lxy` is a list containing a series of point locations and ancillary variables that go with those locations (e.g., time stamps, point IDs, animal IDs). These items are bundled together and 'ready to go' as inputs into T-LoCoH functions. The benefits of bundling all of the objects related to a set of point locations in a single list object includes:

- cleaning and error checking only has to be done once when the lxy object is created (e.g., with `xyt.lxy`)

- having all of the ancillary variables together simplifies the task of passing parameters to other functions as well as saving/retrieving your work to disk

- the nearest neighbor lookup-table (which can take a long time to compute) can be reused

- locations for multiple individuals can be saved in one object

In code examples, objects of class `locoh.lxy` are often indicated by a 'lxy' suffix, and functions that are designed to work on LoCoH-xy objects typically start with 'lxy'.

For a complete description of the data structure, see the vignette *T-LoCoH Data Classes*.

See Also

`xyt.lxy`, `lxy.proj.add`, `lxy.repair`, `lxy.subset`, `lxy.merge`, `lxy.exp.csv`, `lxy.exp.kml`, `lxy.exp.shp`, `move.lxy`

`lxy.amin.add`   *Finds the value of a such that p percent of points are a nearest neighbor for at least one hull*

Description

Finds the value of a such that p percent of points are a nearest neighbor for at least one hull

Usage

```
lxy.amin.add(lxy, id = NULL, s = NULL, ptp = 0.98, nnn = 2,
  prec = NULL, max.iter = 20, status = TRUE)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The id value(s) to analyze. If NULL all ids will be used.
s	Value(s) for the s term in the time-scaled-distance equation for point-to-point distance
ptp	The proportion of total points that should be a nearest neighbor for at least one hull (0..1]
nnn	The minimum number of nearest neighbors each point should have (can pass a vector of several values)
prec	A numeric value in map units to which the value of 'a' will be found.
max.iter	The maximum number of iterations to try to get within prec of the minimum value
status	Show messages, T/F

Details

This function finds the value of 'a' (within a specified threshhold prec) such that the proportion ptp of all points will be a nearest neighbor for at least one hull. This value of 'a' is intended to be a reasonable lower bound for a home range that includes as many points as desired, but minimizes areas where the individual was not observed. This assumes that duplicate points will be offset by a random amount when creating hullset(s).

If no value of prec is passed, it will default to one-half of the median step legnth of the entire dataset if timestamps are present. If timestamps are not present, it will use the median non-zero nearest neighbor distance for the entire dataset.

Note that the value of 'a' such that ptp of points are nearest neighbors does not mean that ptp points are enclosed. Points can be enclosed by hulls that are not a nearest neighbor of any hull parent point.

---

lxy.anv.add	*Add ancillary variable(s) to a LoCoH-xy object*

Description

Add ancillary variable(s) to a LoCoH-xy object

Usage

```
lxy.anv.add(lxy, anv, anv.desc = NULL, overwrite = FALSE)
```

Arguments

lxy	A [LoCoH-xy](#) object
anv	A vector, named list or data frame with the same number of values as points
anv.desc	A character vector of descriptions
overwrite	Whether to overwrite existing variables with the same names (T/F)

Value

A [LoCoH-xy](#) object

---

lxy.exp.csv   *Export coordinates to a csv file*

Description

Exports coordinates and date-stamps from a LoCoH-xy object to a csv file

Usage

```
lxy.exp.csv(lxy, dir = ".", file = NULL, id = NULL,
  file.overwrite = FALSE, quote = FALSE)
```

Arguments

lxy	A LoCoH-xy object
dir	The output directory
file	Filename (with extension .csv)
id	The id(s) of the locations to export
file.overwrite	Overwrite existing file, T/F
quote	Delimit strings in quotes, T/F

Value

A data frame containing the coordinates and attribute values of lxy

Note

This function will export the id value (i.e., animal name) point-id value (ptid), coordinates, and date stamp (if exists) of a LoCoH-xy object to a csv file

If file is not passed, a filename will be automatically constructed.

---

lxy.exp.kml   *Export a lxy object to kml (Google Earth)*

Description

Export a lxy object to a format that can be animated in Google Earth

Usage

```
lxy.exp.kml(lxy, file, id = NULL, skip = 1, overwrite = TRUE,
  compress = TRUE, pt.scale = 0.5, col = NULL, show.path = TRUE,
  path.col = NULL, path.opaque = 80, path.lwd = 3)
```

Arguments

`lxy`	A LoCoH-xy object
`file`	The name of the kml file to be created (relative to the working folder). The .kml extension will be added if needed.
`id`	The id(s) of the individual(s) to include in the plot
`skip`	The skip factor to use to reduce the number of points exported (i.e., every nth point exported where n=skip)
`overwrite`	Whether to overwrite an existing kml file (T/F)
`compress`	Whether to copmress the kml file and create a kmz file (T/F)
`pt.scale`	The relative size of the placemark symbols
`col`	The point colors (one per id)
`show.path`	Whether to include the point path in the kml file (T/F)
`path.col`	The path colors (one per id), ignored if `show.path=FALSE`
`path.opaque`	The opacity of the path (0=transparent, 255=fully opaque), ignored if `show.path=FALSE`
`path.lwd`	The width of the path, ignored if `show.path=FALSE`

Value

The file name of the kml/kmz file

Note

This function will export the locations in a LoCoH-xy object to a kml file that can be opened in Google Earth for animation. The kml format is not optimized for large datasets, but this works reasonably well for moderate datasets (e.g., <1000). To create animations of larger datasets, use a `skip` value > 1 or see `lxy.exp.mov`.

Adapated from plotKML package.

See Also

`lxy.exp.mov`

`lxy.exp.mov` *Create an animation from a LoCoH-xy object*

Description

Create an animation from a LoCoH-xy object

Usage

```
lxy.exp.mov(lxy, id = NULL, all.ids.at.once = TRUE,
  all.ids.col.unique = all.ids.at.once, all.ids.col = NULL,
  all.ids.legend = c("bottomright", "bottom", "bottomleft", "left", "topleft",
  "top", "topright", "right", "center")[5], all.ids.legend.cex = 0.8,
  dt.start = NULL, dt.end = NULL, frame.method = c("auto", "time",
  "location")[1], frame.rtd = "auto", xlim = NULL, ylim = NULL,
  dt.label = TRUE, dt.label.x = NULL, dt.label.y = NULL,
  dt.label.col = "black", dt.label.bg = "gray90", title = NULL,
  title.show = TRUE, axes.show = TRUE, axes.ticks = axes.show,
  axes.titles = FALSE, mar.map = c(0.7 + (if (axes.ticks) 0.9 else 0) + (if
  (axes.titles) 1.3 else 0), 0.5 + (if (axes.ticks) 0.9 else 0) + (if
  (axes.titles) 1.3 else 0), if (title.show) 2.1 else 0.5, 0.5),
  mgp.map = c(0.4 + if (axes.ticks) 1.2 else 0, 0.4, 0),
  col.xys.active = "red", col.xys.background = "gray80",
  cex.xys.active = 1.5, cex.xys.background = 0.5, tz.local = NULL,
  tz.local.check = TRUE, col.by.hour.of.day = FALSE,
  col.hod = colorRampPalette(colors()[c(24, 30, 553, 121, 26, 121, 553, 30,
  24)])(24), width = if (screen.test) 7 else 608, height = NULL,
  max.frames = NULL, png.pointsize = 16 + (width - 480)/80,
  screen.test = FALSE, tmp.dir = NULL, tmp.files.delete = TRUE,
  prompt.continue = TRUE, fn.mov = NULL, fn.mov.dir = getwd(),
  fn.mov.exists = c("auto.increment", "overwrite", "stop", "ask")[1],
  duration = NULL, fps = NULL, skip = NULL, ffmpeg = NULL,
  create.mov = TRUE, show.cmd = FALSE, fmt = c("mov", "mp4")[1],
  info.only = TRUE, shp.csv = NULL, layers = NULL, tiff.fn = NULL,
  tiff.bands = c(3, 2, 1), tiff.col = gray(0:255/255), tiff.pct = FALSE,
  tiff.buff = 0, tiff.fill.plot = TRUE, bg2png = !is.null(layers),
  crop.layers.to.extent = TRUE, date.bar = 0.85, date.bar.bins = 12,
  col.db = "darkblue", cex.axis.db = 0.7, beep = FALSE,
  report.time = TRUE, status = TRUE)
```

Arguments

`lxy`	A [LoCoH-xy](#) object
`id`	The id value(s) to be on the plot
`all.ids.at.once`	
	Display all the individual ids simultaneously. If False, an animation will be created for each id. T/F.
`all.ids.col.unique`	
	Whether to use unique colors for each individual when plotting multiple individuals simultaneously, T/F
`all.ids.col`	A named list of color values; the element names must match the name(s) of the ids
`all.ids.legend`	Where to place a legend showing the color of each id (for an animation showing the movement of multiple individuals simultaneously): 'bottomright', 'bottom', 'bottomleft', 'left', 'topleft', 'top', 'topright', 'right', or 'center'. May also be NULL, in which case the legend will not be displayed. Ignored if `ids.legend.bln`=FALSE or `all.ids.col.unique` = FALSE.
`all.ids.legend.cex`	
	The character expansion factor for the id legend. See parameter `all.ids.legend` above.

dt.start	The starting date-time. An object of class POSIXt or one that can be coerced to POSIXt. If NULL, the earliest date-time in the series will be used.
dt.end	The ending date-time. An object of class POSIXt or one that can be coerced to POSIXt. If NULL, the last date-time in the series will be used.
frame.method	How each frame should be defined temporally, "time" - each frame represents a fixed amount of time, "location" each frame is a point in the series
frame.rtd	The real-time duration of each frame (in seconds). If "auto" (default), the lowest median sampling frequency will be used
xlim	A two-element numeric vector for the range of the x-axis (in map units)
ylim	A two-element numeric vector for the range of the y-axis (in map units)
dt.label	Add a label for the date of the frame
dt.label.x	The x-coordinate for the date label (ignored if dt.label=FALSE)
dt.label.y	The y-coordinate for the date label (ignored if dt.label=FALSE)
dt.label.col	A color value/name for the date label (ignored if dt.label=FALSE)
dt.label.bg	The background color for the date label. Set to NA for a transparent background. (ignored if dt.label=FALSE)
title	A title for the map
title.show	Whether to show the title. T/F.
axes.show	Whether to show the axes (ticks, labels and titles). Can be over-written by axes.ticks and axes.titles. T/F.
axes.ticks	Whether to show the tick marks and labels on the axes. T/F.
axes.titles	Whether to show axes titles. T/F.
mar.map	Margin settings for the map, see [par](#)
mgp.map	Locations of the axes elements for the map, see [par](#)
col.xys.active	Color of the active point. Ignored if col.by.hour.of.day = TRUE.
col.xys.background	Color of the non-active points. To hide non-active points, set this to NA.
cex.xys.active	The character expansion factor of the active point
cex.xys.background	The character expansion factor of non-active points
tz.local	The name of the time zone of the study area
tz.local.check	Check whether tz.local is a valid timezone name (not implemented) T/F.
col.by.hour.of.day	Whether to color the active point by the hour of day (i.e., dark colors at night, orange for day time locations). T/F
col.hod	A vector of 24 color values used to symbolize the color of the active point by the hour of day. Ignored if col.by.hour.of.day = FALSE.
width	The width of each frame in pixels (if screen.test=FALSE) or inches (if screen.test=TRUE).
height	The height of each frame in pixels (if screen.test=FALSE) or inches (if screen.test=TRUE).
max.frames	The maximum number of frames to produce.
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
screen.test	Create up to three sample frame(s) on the screen (instead of PNG files)

`tmp.dir`	A directory where temporary PNG files for each frame will be created, character.
`tmp.files.delete`	
	Delete the temporary PNG files when done, T/F
`prompt.continue`	
	Whether to present a summary of the encoding settings and get user confirmation before continuing, T/F
`fn.mov`	The path and filename of output file (either *.mov or *.mp4). If NULL a filename will be automatically generated
`fn.mov.dir`	The directory where the animation will be saved (ignored if a value for `fn.mov` is passed)
`fn.mov.exists`	What to do if the animation file already exists: "auto.increment", "overwrite", "stop", or "ask".
`duration`	The desired duration of the animation (in seconds)
`fps`	A numeric value for frames per second
`skip`	Output every nth frame. To include every frame set skip=1. Integer.
`ffmpeg`	The name of the ffmpeg executable. If NULL (the default), a default file name will be used. See notes.
`create.mov`	Whether to actually create the mov file. Set to FALSE preview a few frames without actually encoding them.
`show.cmd`	Whether to display the ffmpeg command line that stitches the images into an animation
`fmt`	Video format: `'mov'` (Quicktime animation codec) or `'mp4'` (h.264)
`info.only`	Only return info
`shp.csv`	The path and filename of a csv file that contains information about shapefiles, including layer names, file, and symbology.
`layers`	The name(s) of layers in shp.csv to display in the background. Will be displayed using the symbology in shp.csv. Character vector or comma delimited string.
`tiff.fn`	The path and name of a GeoTIFF file (e.g., satellite image) that will be displayed in the background. See notes.
`tiff.bands`	A vector of threee integers corresponding to the bands of the GeoTIFF image that will be mapped to the red, green and blue color guns respectively.
`tiff.col`	A vector of color values for plotting single-band images in the background. Ignored if using three bands.
`tiff.pct`	Whether or to convert the GeoTIFF to an indexed 256 color RGB image, which may speed up drawing. T/F.
`tiff.buff`	A numeric buffer distance in map units that the range of the plot will be expanded so the points are not right on the edge of the GeoTIFF.
`tiff.fill.plot`	Whether to fill the entire plot area with the GeoTIFF. T/F.
`bg2png`	Save the plot background elements as a static raster image (to improve speed), ignored if `screen.test=TRUE`
`crop.layers.to.extent`	
	Whether to crop the shapefile layers to the view extent (may speed up drawing time)
`date.bar`	The height of the lower section of the plot to devote to the time bar, in inches. To hide the time bar completely, set `date.bar=0`.

date.bar.bins	The target number of bins (tick marks + 1) on the time bar (integer)
col.db	A single color value for the date bar axes / tick labels, character
cex.axis.db	Character expansion factor for the labels on the date bar axis.
beep	Beep when done, T/F
report.time	Show the time taken when done, T/F
status	Show progress bar and status messages

Value

A list with information about each file created. Each element of the list is another list with two elements: fn (the full filename) and dim (a two-element numeric vector with the frame width and height). If no animation file(s) were created, returns NULL.

Note

This function creates an animation from a LoCoH-xy object. There are three general steps in the workflow: 1) design the frame layout, 2) choose values for the speed and duration of the animation, and 3) create the animation.

One will normally want to run the function a few times without actually encoding to tweak the frame design (e.g., where the date label and legend appear). To see what a frame in the output will approximately look like, set screen.test=TRUE. See Appendix VI of the T-LoCoH tutorial for further details.

If you have locations for all hours of the day, you can color the active point by the hour-of-day by setting col.by.hour.of.day=TRUE. This can help you visualize daily patterns in movement, assuming that the time-stamps of each point are in local time. Locations at night will appear dark, sunrise and sunset reddish, and mid-day points blue (these colors can be adjusted by passing a different value for col.hod.

To create the animation, two and only two of the following parameters must be passed: duration, fps, and skip. The third parameter will be computed based on the other two. To include every frame, pass fps, set skip=1, and leave duration out.

Larger values for fps will result in the animation running 'faster'. Values between 10 and 20 often work well; beyond 30 fps the eye can't keep up with the motion Note if you pass values for fps and duration, an appropriate value for skip will be computed but the final duration of the animation may not be exactly equal to duration because only interger values of skip are allowed.

tmp.dir="." (or another folder), create.mov=FALSE, and tmp.files.delete=FALSE. This will generate a few sample frames as PNG files but not delete them so you can inspect them using an image viewer. Once these look good, create the full animation by setting create.mov=TRUE and max.frames=NULL.

If frame.method is 'auto', the script will use time-based frames when multiple individuals are being animated simultaneously, and location-based frames otherwise.

duration (the duration of the animation in seconds) should not be confused with frame.rtd which is the real-time duration of a single frame in seconds (e.g., frame.rtd=3600 means each frame will represent 1 hour).

If date.bar is too small or too large, you might get a 'margins too large' error. Try values around 1, or hide the date bar completely by setting date.bar=0.

The output animation is encoded in QuickTime format or mp4. The Quicktime file is encoded using Quicktime's 'animation' codec, a lossless format that 'scrubs' well (i.e., you can drag the scroll bar to view frame by frame). The mp4 settings use the h.264 compression algorithm with a constant rate factor of 20 (good quality). In both cases, encoding requires installing the open

source encoding program ffmpeg. ffmpeg is a command line program that can be downloaded from <http://ffmpeg.org/download.html> (select the package for your OS). After downloading, the ffmpeg executable file should be saved in the current working directory, or a directory on the PATH environment variable (e.g., on windows c:\windows). You may also pass the path and name to ffmpeg using the ffmpeg argument.

If ffmpeg is not available, you can still use this function to generate the individual frames and then use another utility (e.g., ImageMagick, Quicktime Pro) to combine the frames into a video file. For best results use a 'lossless' compression method in the encoding program. To create the individual frames only for encoding with another utility, set tmp.dir="." (the working directory) and tmp.files.delete=FALSE. Passing show.cmd=TRUE will display the ffmpeg command line that will generate animation, which can be useful if you want to do some additional editing to the frames prior to encoding.

If fn.mov.exists = "auto.increment", a two-digit number will be appended to the filename to avoid overwriting an existing file

---

lxy.exp.shp   *Export a LoCoH-xy to shapefile*

Description

Export a LoCoH-xy object to point shapefile

Usage

```
lxy.exp.shp(lxy, id = NULL, dir = ".", file.base = "", anv = "all",
  idsSeparate = FALSE, autoName = c("long", "short", "none")[1],
  status = TRUE)
```

Arguments

lxy	A LoCoH-xy object
id	The name(s) of individuals to export. If NULL all individuals will be exported
dir	The directory where the shapefiles will be placed (use "." for the working directory, and "~" for the home directory)
file.base	The base of the file name without any extension (the script will append to this)
anv	The name of ancillary variables(s) to be exported in the attribute table. Character vector or comma separated string.
idsSeparate	Whether to export the locations for each individual in separate shapefiles. T/F
autoName	The rule for constructing a filename: short (consisting of the id(s) only), long (id + num points), or none
status	Show messages. T/F

Details

Filenames are automatically generated, but the user can specify the directory and a base (file.base). If a shapefile already exists, a unique filename will be constructed using two-digit numeric extension. For shorter filenames, pass autoName="short".

Use the anv argument to specify which ancillary variables (i.e., attribute fields) should be exported. To not export any ancillary variables, pass an empty string.

Only the individual locations and their ancillary values are exported. The nearest neighbor table is not exported.

Value

A character vector of the shapefile(s) created

See Also

[lxy.exp.kml](), [lxy.exp.mov]()

---

lxy.gridanv.add   *Add ancillary variable(s) to a LoCoH-xy object from a raster*

Description

Add ancillary variable(s) to a LoCoH-xy object from a raster

Usage

```
lxy.gridanv.add(lxy, band = 1, dtfn = NULL, fn = NULL, anv.name = NULL,
  anv.desc = NULL, date.match = c("closest", "before")[1])
```

Arguments

lxy	[LoCoH-xy]() object
band	Numeric vector of which band(s) to read
dtfn	A two-column data frame containing a date and filename
fn	Filename of a raster
anv.name	The name(s) of the ancillary variable(s) (one per band)
anv.desc	A character vector of descriptions
date.match	How to match dates when dtfn is used: 'closest' or 'before'. See note.

Value

A [LoCoH-xy]() object

Note

This function will create new ancillary variables and fill it with the value of the pixel at each point. The image should be in the same coordinate system as the points. Current the script does not check this. Any points that fall beyond the edges of the image will be given 'NA' values. If an ancillary variable with the same name already exists, it will be overwritten.

The image / raster specified by fn be in a format supported by the readGDAL function in the rdgal package. Multi-band formats are support. If the image has multiple bands, you can specify which band(s) to read using the band parameter. A separate ancillary variable will be created for each band. You can provide name(s) for the ancillary variable(s) with the anv.name parameter.

You can also pass a time-series of images, and the function will use the image that is the closest match in time for each location. To do this, pass a two column data frame to the parameter dtfn. The first column of dtfn must be a date (one of the POSIXt date-time classes), and the second column must be a valid raster filename. If each image is a time composite, (e.g., two-week NDVI), the date should either be the mid-point or the beginning date. Two options for time matching are available: closest = select the image whose date is closest in time, and before = pick the closest image that comes before the location.

```
lxy.id.new                      Change the id value(s) in a LoCoH-xy object
```

Description

Changes the id values in a LoCoH-xy object, with option to save the 'old' id values as an ancillary variable

Usage

```
lxy.id.new(lxy, id.new, save.old.id = NULL, dup.dt = c("check", "offset",
  "ignore")[1], dup.dt.offset = 5, dt.int.round.to = 0.1,
  tau.diff.max = 0.02, status = TRUE)
```

Arguments

`lxy`	A [LoCoH-xy](#) object
`id.new`	New value(s) for the id (i.e., name of the individual(s)). Can be either a single character object 'broken_tooth', or factor or character vector of the same length as the number of points
`save.old.id`	Whether to save the old ids as an ancillary variable. Can be T/F, or the name of a column in the ancillary variables data frame, see Details
`dup.dt`	What to do if duplicate dt values are encountered in the merged object, see details
`dup.dt.offset` `dt.int.round.to`	The maximum offset in seconds to offset duplicate dt values, used only if `dup.dt = "offset"`
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
`tau.diff.max`	The maximum deviation from tau (the median delta.t of the entire dataset), expressed as a proportion of tau, that time difference between two points must fall for the distance between those two points to be included in the calculation of the median step length
`status`	Show status messages. T/F

Details

This function can be used to change the id value for each point. Note that the id value is the name of the individual animal or device, as opposed to ptid which is a unique numeric integer for each point.

`tau.diff.max` exists to eliminate the inclusion of temporal outliers in the computation of the median step length. The time difference betwen points must be withint `tau.diff.max` of tau for that pair of points to be included in step length calculation.

One can also use this function to change the id (name of the individual), rebuild the rw.params or table of sampling frequencies

Value

A [LoCoH-xy](#) object containing the original xys, dt, ptid and ancillary values for each point. Other items and attributes (including the comment) will be recreated or set to NULL

---

lxy.identify.aoi   *Identify the points that fall within a bounding box*

Description

Identify the points that fall within a bounding box

Usage

```
lxy.identify.aoi(lxy, aoi)
```

Arguments

lxy   A LoCoH-xy object

aoi   Area-of-interest, a two-column xy data frame containing the upper left point and
                  the lower right point in the first and second rows respectively.

Value

A dataframe containing the ptid values and indices of the points from lxy that fall within aoi

Note

This function can be used to identify specific points on a plot. For example, you see a cluster of
points on a locoh-xy plot, and want to know when those points were recorded, and so on. To do
this, you can use the aoi function to define an area of interest box on the plot window, and then use
lxy.identify.aoi to grab the indices and ptid values for that cluster of points. You can then use
this information as a parameter in lxy.subset function to create a locoh-xy object containing just
that cluster of points.

See Also

aoi, plot.locoh.lxy, lxy.subset

---

lxy.lhs   *Create a LoCoH-hullset object from a LoCoH-xy object*

Description

Creates a LoCoH-hullset object (class "locoh.lhs") containing one or more sets of hulls.

Usage

```
lxy.lhs(lxy, id = NULL, s = 0, a = NULL, r = NULL, k = NULL,
  kmin = 0, anv.copy = TRUE, decimal.places = 1, offset.dups = 1,
  velocity.metrics = TRUE, ud = NULL, iso.add = FALSE,
  iso.levels = c(0.1, 0.25, 0.5, 0.75, 0.95), sliver_check = TRUE,
  pbo.style = 3, beep = FALSE, status = TRUE, save.hulls = TRUE,
  save.enc.pts = TRUE)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The id value(s) of lxy for which hullsets will be created. If NULL all ids will be used.
s	Value(s) for the s term in the time-scaled-distance equation for point-to-point distance. When s=0 time is excluded and TSD is equivalent to Euclidean distance. Numeric vector or comma-separated string. Can also be 'all'
a	Value(s) for the adaptive method. Numeric vector or comma-separated string. a also be the results of the [auto.a](#) function, see notes.
r	Value(s) for the r method. Numeric vector or comma separate list of values.
k	Value(s) for the k method. Numeric vector or comma separate list of values.
kmin	A minimum number of nearest neighbors for each parent point regardless of the method used. Set k=2 to ensure that every point is part of a hull. Integer.
anv.copy	Copy the ancillary variables data frame (if exists), T/F
decimal.places	The number of decimal places for the k/r/a parameter value used when constructing the name of a hullset. Note this has no effect on hull creation or the precision of values used, only the number of digits that appear in the hullset name.
offset.dups	A number of map units to randomly offset duplicate points. Set offset.dups=0 to ignore duplicate points
velocity.metrics	
	Compute the velocity hull metrics
ud	Deprecated (no longer used). Use iso.add instead
iso.add	Whether to also create density isopleths, T/F
iso.levels	Isopleth levels (see also [lhs.iso.add](#)), numeric vector. Ignored if |codeiso.add=FALSE.
sliver_check	Whether to check for and delete slivers in isopleths. See details. T/F
pbo.style	Progress bar style (see pbapply package)
beep	Beep when done. T/F
status	Show messages. T/F
save.hulls	Whether to save the hulls. T/F
save.enc.pts	Whether to save the enclosed points. T/F

Details

This function creates a [LoCoH-hullset](#) object from a [LoCoH-xy](#) object. Other functions allow you to do things with hullsets, including computing additional hull metrics, constructing isopleths, directional routes, generating scatterplots, exporting, etc. Note that before you use this function, nearest neighbors must have already been identified and saved in the input [LoCoH-xy](#) object (see [lxy.nn.add](#)).

If iso.add=TRUE, after the hulls are created the function will create density isopleths using the default sort order (area for *k-method*, number of enclosed points for the *a* and *r-methods*). You can control which isopleth levels are created using the iso.levels argument. If sliver_check=TRUE, isopleths will checked for 'slivers'. Slivers (as used here) refer to polygons or holes with <3 unique nodes. Slivers can be caused by reductions in numeric precision or rounding errors, and are not uncommon.

When working with a large dataset where memory limts may affect performance, you can choose to not save the hulls or enclosed points by setting save.hulls=FALSE and save.enc.pts=FALSE. See the vignette on working with large datasets.

Value

An object of class [locoh.lhs](#)

See Also

[xyt.lxy](#) for creating LoCoH-xy objects [lhs.iso.add](#) for adding isopleths [lhs.ellipses.add](#) for computing bounding ellipses [lhs.visit.add](#) for computing time-use metrics [lhs.plot.scatter](#) for creating scatterplots of hull metrics [plot.locoh.lhs](#) for plotting a LoCoH-hullset [hulls](#) for extracting hulls as a SpatialPolygonsDataFrame [lhs.exp.shp](#) to export hulls as shapefile

---

lxy.merge	*Merges LoCoH-xy object*

Description

Merges LoCoH-xy objects together

Usage

```
lxy.merge(..., pts.flds = "all", save.ptid = FALSE, new.id = NULL,
   save.old.id = NULL, dt.int.round.to = 0.1, tau.diff.max = 0.02)
```

Arguments

...	[LoCoH-xy](#) objects
pts.flds	A character vector of column names to include in the merged object, can also be "all" or NULL
save.ptid	If duplicate ptid values are encounted in the merged object, whether to save the original ptid values as an ancillary variable, T/F
new.id	A new id value for the merged object
save.old.id	A boolean or the name of a ancillary variable where the old id values will be saved
dt.int.round.to	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
tau.diff.max	The maximum deviation from tau (the median delta.t of the entire dataset), expressed as a proportion of tau, that time difference between two points must fall for the distance between those two points to be included in the calculation of the median step length

Details

The coordinate system of the LoCoH-xy objects being merged must be the same. Also each lxy object must have the same data columns present, e.g., dt, col, ancillary variables, etc. To merge a subset of the attribute fields, use the `pts.flds` parameter.

Because LoCoH-xy object must have unique ptid values for each location, if the merged object contains duplicate ptid values new one will be created. If save.ptid=TRUE, the 'old' ptid values will be saved as `ptid.orig`.

The merged object will no have nearest neighbors table and the random walk parameters will be recomputed.

See Also

[lxy.id.new](lxy.id.new)

---

lxy.nn.add   *Identifies nearest neighbors for a LoCoH-xy object*

Description

Identifies nearest neighbors for a LoCoH-xy object

Usage

```
lxy.nn.add(lxy, id = NULL, ptid = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, kmin = 0, ptsh = NULL, ptsh.idx = 1,
  nn.exists = c("append", "replace", "skip")[1], time.term = c("vmax",
  "dif")[1], FNN.algorithm = c("cover_tree", "kd_tree", "VR", "CR")[2],
  dec.places = 1, ra.init.samp.size = 30, ra.init.quant = 0.8,
  tct = 0.05, beep = FALSE, status = TRUE)
```

Arguments

lxy	A LoCoH-xy object
id	The name(s) of individuals to analyze
ptid	A vector of ptid indicating which points should have nearest neighbors identified (can be used to speed up processing during testing)
k	Value for the fixed-k method (identify k nearest neighbors of each point)
r	Value for the fixed-r method (identify all points within radius r)
a	Value for the a method (identify all points within cummulative distance a). Can also be a data frame of parameters returned by the auto.a function.
s	The value for s for the time-scaled distance metric
kmin	A minimum number of neighbors to identify for each point
ptsh	The desired proportion of time-selected hulls [0..1], may also be 'all' if the s-ptsh map has already been computed. See details.
ptsh.idx	The index of the saved ptsh-s table. See also summary.locoh.lhs
nn.exists	What to do if nearest neighbors have already been saved: "append", "replace", or "skip"
time.term	The space-time transformation to use in the TSD distance metric: 'vmax' for the maximum velocity transformation (default) or 'dif' for the diffusion transformation
FNN.algorithm	The algorithm to be used in the get.knnx() function in the package FNN
dec.places	The number of decimal places that rmax and amax should be rounded to
ra.init.samp.size	
	How many randomly selected points to use to come up with the first guess for how many neighbors to find for each point to satisify a (a-method only)
ra.init.quant	The proportion of randomly selected to points to use for the a value when coming up with an initial guess at the number of neighbors to find for each point to satisify a (a-method only)

tct	Temporal continuity threshhold used when a=auto.a
beep	Beep when done (T/F)
status	Show status messages (T/F)

Details

When s > 0, the 'distance' (as defined by the time-scaled distance metric') between two points is a function of their separation in time as well as space.

If `ptsh` is provided, the script will find a value of 's' such that the proportion of hulls which are time-selected (e.g, consecutive points in time) is equal to `ptsh`. In this case, no value for s should be passed. If `ptsh="all"` and the proportion of time selected hulls has already been computed, those values of 's' will be used.

Value

A LoCoH-xy object object with saved information about each points' nearest neighbors

See Also

`summary.locoh.lxy` to see what nearest neighbors have already been identified `auto.a` `lxy.amin.add` `lxy.ptsh.add`

Examples

```
## Not run:
# lxy <- lxy.nn.add(lxy, k=10, s=0.01)

## End(Not run)
```

---

lxy.plot.freq	*Plots of sampling frequency*

Description

Creates plots of the temporal extent and sampling frequency of a LoCoH-xy object

Usage

```
lxy.plot.freq(lxy, id = NULL, deltat.by.date = F, samp.by.date = F,
  cp = F, cp.min = 0, cp.max = 1, time.unit = c("auto", "sec", "min",
  "hour", "day", "week", "month", "year")[1], xlim = NULL, ylim = NULL,
  ylim.deltat.by.date = c(0, 0.999), sbd.y.jiggle = 0.1, sbd.vline = NULL,
  sbd.vline.trunc.units.to = c("secs", "mins", "hours", "days")[4],
  cex.pts = 0.3, desc = c(0, 1, 3)[2], cex.desc = 0.8,
  col.desc = "darkgreen", title = NULL, title.show = TRUE,
  subtitle = "auto", mar = c(2.8, 2.8, if (title.show) 2.8 else 0.5, 0.5),
  mgp = c(1.8, 0.5, 0), figs.per.page = NULL, status = TRUE,
  panel.num = NULL, panel.num.inside.plot = !title.show, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.overwrite = TRUE, png.pointsize = 12 + (png.width - 480)/80, ...)
```

Arguments

`lxy`	A LoCoH-xy object
`id`	The id(s) of the individual(s) to include in the plot
`deltat.by.date`	Produce a plot of the sampling interval by date. T/F
`samp.by.date`	Produce a scatterplot showing the number of samples by date. T/F
`cp`	Produce a plot of the cummulative proportion of sampling frequencies. T/F
`cp.min`	The minimum sampling frequency (expressed as a proportion of tau) shown on the plot
`cp.max`	The maximum sampling frequency (expressed as a proportion of tau) shown on the plot
`time.unit`	A unit of time that will be used on the y-axis when plotting frequency by date
`xlim`	The range for the x-axis. Can be a two-element vector (min, max), NULL (range will be computed from the data), or "same" (same range will be used for all individuals).
`ylim`	The range for the y-axis.
`ylim.deltat.by.date`	
	A two-element numeric vector with values 0..1. Will interpret these values as the first and last percentiles of delta.t to plot
`sbd.y.jiggle`	An amount to 'jiggle' the y values to better see point density
`sbd.vline`	An interval of time (in seconds) for which vertical lines will be included on the sampling interval by date plot, to help see the dates
`sbd.vline.trunc.units.to`	
	A character "secs", "mins", "hours" or "days" specifying where to draw the first vertical lines. quantiles of delta.t when setting the range of ylim when `deltat.by.date=TRUE`
`cex.pts`	The expansion factor for points on the plot
`desc`	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
`cex.desc`	The expansion factor for the descriptive text. Numeric value.
`col.desc`	The color of the descriptive text. Color value.
`title`	The title to be displayed. Character. If NULL a title will be constructed.
`title.show`	Whether to show the title. T/F
`subtitle`	The plot subtitle. If 'auto' a default subtitle will be used, or pass an empty string `''` to omit. Character
`mar`	The plot margins. A four item numeric vector
`mgp`	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
`figs.per.page`	The number of plots per page.
`status`	Show status messages. T/F
`panel.num`	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
`panel.num.inside.plot`	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F

png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
...	Additional parameters that will be passed to the [plot](#) function

Value

A list of lists, one for each plot containing the filename (NULL if no png made), the image dimensions (NULL), descriptive text, and id

Note

Plotting the cummulative percentage of sampling frequencies (codecp=TRUE) can help determine a threshhold below which pairs of points should be considered to be part of the same 'burst' (see [lxy.thin.bursts](#)). For example, if most points are sampled 1 hour apart, but there are a few bursts of points that were only 2 minutes apart, the cumuulative percentage of sampling frequencies will reveal a cluster of sampling intervals around 2 minutes and the majority around 1 hour.

Plotting the sampling frequency over time can help identify gaps in the data as well as frequencies for harmonising sampling frequency

See Also

[lxy.thin.bursts](#), [lxy.thin.byfreq](#)

---

lxy.plot.mtdr	*Plot distribution of the ratio the maximum theoretical distance ratio for nearest neighbors*

Description

Plot distributions of the ratio of maximum theoretical distance times 's' over TSD for all unique pairs of nearest neighbors

Usage

```
lxy.plot.mtdr(lxy, s = NULL, k = NULL, a = NULL, r = NULL,
  type = c("mtd.tsd", "mtd.ed")[1], offset.dups = 1, id = NULL,
  show.samp.size = TRUE, outline = FALSE, desc = c(0, 1, 3)[2],
  cex.desc = 0.8, col.desc = "darkgreen", mar = c(3, 3, if (title.show) 3
  else 0.7, 0.5), mgp = c(1.9, 0.5, 0), figs.per.page = NULL,
  title = NULL, title.show = TRUE, title.obj.name = FALSE,
  panel.num = NULL, panel.num.inside.plot = !title.show,
  no.sci.notation = FALSE, png.dir = NULL, png.dir.make = TRUE,
  png.width = 800, png.height = png.width, png.overwrite = TRUE,
  png.pointsize = 12 + (png.width - 480)/80, status = TRUE, ...)
```

Arguments

lxy	A [LoCoH-xy](#) object
s	The s value(s) of nearest neighbor sets to include in the plot. If NULL, all values will be used
k	A k-value for the number of nearest neighbors around each point to include in the plot
a	A a-value for the number of nearest neighbors around each point to include in the plot
r	A r-value for the number of nearest neighbors around each point to include in the plot
type	Which ratio to plot: maximum theoretical distance over TSD, or maximum theoretical distance over Euclidean distance
offset.dups	An amount in map units for which pairs of points with the same location but different times will be offset so that a TSD or Euclidean distance can be calculated (see also `lxy.lhs`)
id	The id(s) of the individual(s) to include in the plot
show.samp.size	Whether to display the sample size of the number of unique pairs of points for each value of s, T/F
outline	Show outliers in the box plots T/F
desc	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
cex.desc	The expansion factor for the descriptive text. Numeric value.
col.desc	The color of the descriptive text. Color value.
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
figs.per.page	The number of plots per page.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F
title.obj.name	Whether to add the name of the lxy object to the plot title (ignored if `title` is passed). T/F
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F
no.sci.notation	
	Whether to avoid the use of scientific notation on labels on the x-axis. T/F
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F

png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
status	Display status messages. T/F
...	Additional parameters that will be passed to the [plot](#) function

Details

This function plots the ratio of the maximum theoretical distance the individual could have traveled (multiplied by 's') over the total TSD distance metric for all nearests neighbors for the specified value of k/a/r for different values of s. This ratio will of course never be greater than one because the maximum theoretical distance is one of the terms in TSD. The purpose of this function is to see the effect of s on the relative contribution of the time term in the TSD metric.

This function requires that nearest neighbors already be computed. To identify nearest neighbors for a range of 's' values, use `lxy.ptsh.add` (see examples).

Value

A list of plots created, with one element per id, and each element consisting of another list with elements for the filename (or NULL), filename dimensions in pixels (or NULL), the MTD ratios (as a list with one element for each value of s), descriptive text, and the k/a/r value

Examples

```
## Not run:
## Identify a range of 's' values from space-selection to time-selection,
## and find 10 nearest neighbors for each value of 's'
lxy <- lxy.ptsh(lxy, nn=TRUE)
## Plot the ratio of maximum theoretical distance travel to TSD
lxy.plot.mtdr(lxy)

## End(Not run)
```

---

lxy.plot.pt2ctr   *Plot distance to centroid over time*

Description

Produces a plot of the distance of each point to the centroid over time to see 'natural' periodicities in the data

Usage

```
lxy.plot.pt2ctr(lxy, id = NULL, idx = NULL, vline = NULL,
  vline.trunc.units.to = c("secs", "mins", "hours", "days")[4],
  figs.per.page = NULL, ...)
```

Arguments

`lxy`	A [LoCoH-xy](#) object
`id`	The id value(s) to include on the plot
`idx`	An optional vector of indices of points that the plot will be restricted to. See notes
`vline`	An interval of time (in seconds) for the vertical grid lines
`vline.trunc.units.to`	
	A character value: "secs", "mins", "hours", or "days". Determines where the first vertical grid lines will appear
`figs.per.page`	The number of figures per page
`...`	Additional parameters that will be passed to the [plot](#) function

Note

The purpose of this graph is to help see the 'natural' movement cycles in a dataset, when no *a priori* information exists that helps identify the time scale of interest. In T-locoh, defining the timescale of interest helps to select a value for `s`.

To 'zoom in' to a section of the timeline, pass a value for `idx`. For example, if there are 5000 points in the dataset

If a value is passed for vline, the first vertical line will coincide with a multiple of `vline.trunc.units.to`. For example, if `vline=3600*24`(24 hours), and `vline.trunc.units.to="days"`, the first vertical line will be the beginning of a day regardless of when the point series started. This can be helpful for seeing cycles of a specific frequency.

See Also

[lxy.plot.sfinder](#)

---

lxy.plot.ptsh   *Plots the proportion of time-selected hulls over 's'*

Description

Plots the proportion of time-selected hulls over 's'

Usage

```
lxy.plot.ptsh(lxy, id = NULL, ptsh.idx = NULL, use.nn = FALSE, k = NULL,
  r = NULL, a = NULL, slim = NULL, boxplot = FALSE, overlay = FALSE,
  desc = c(0, 1, 3)[2], cex.desc = 0.8, col.desc = "darkgreen",
  title = NULL, title.show = TRUE, legend = c("none", "bottomright",
  "bottom", "bottomleft", "left", "topleft", "top", "topright", "right",
  "center")[2], mar = c(3, 3, if (title.show) 2.8 else 0.7, 0.5),
  mgp = c(1.8, 0.5, 0), figs.per.page = NULL, panel.num = NULL,
  panel.num.inside.plot = !title.show, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.overwrite = TRUE, png.pointsize = 12 + (png.width - 480)/80,
  lo.save = TRUE, ...)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The name(s) of individuals to plot
ptsh.idx	The index number of the saved ptsh table to use (ignored if use.nn=TRUE)
use.nn	Whether to create the plot based on nearest neighbor sets (as opposed to saved ptsh tables, see Details). T/F.
k	Value for the k method if creating the plot based on nearest neighbor tables (ignored if use.nn=FALSE)
r	Value for the r method if creating the plot based on nearest neighbor tables (ignored if use.nn=FALSE)
a	Value for the a method if creating the plot based on nearest neighbor tables (ignored if use.nn=FALSE)
slim	The lower and upper bounds for s, two-element numeric vector
boxplot	Display boxplots of the range of s-values for each target ptsh for multiple individuals. T/F
overlay	Overlay curves for multiple individuals. T/F
desc	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
cex.desc	The expansion factor for the descriptive text. Numeric value.
col.desc	The color of the descriptive text. Color value.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F.
legend	A character object specifying where to put the legend (ignored when use.nn=TRUE)
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
figs.per.page	The number of plots per page.
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F
png.dir	The directory for a PNG file (filename will be constructed automatically).
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
lo.save	Change layout
...	Additional parameters that will be passed to the [plot](#) function

Details

This function will plot the proportion of total hulls that are 'time-selected'. Time-selected means all of the nearest neighbors are were sampled sequentially in time. This is one end of the spectrum as far as nearest neighbor identification goes, the other being space-selected (i.e., time has no bearing). The s parameter in the TSD equation determines the degree to which time plays a role in the point-to-point 'distance'. When s=0, time plays no role in TSD and TSD is equivalent to Euclidean distance. As s increases, time plays a bigger and bigger role until eventually nearest neighbor selection is equivalent to selection points based only on their separation in time.

In order to plot the proportion of time-selected hulls (ptsh), nearest neighbors must have already been identified. This can be done in one of two ways. The `lxy.ptsh.add` function will compute ptsh for different values of s using a random sample of points (to save time), automatically picking values of s such that ptsh is close to target values provided by the user. Alternately, one can identify nearest neighbors for different values of s using `lxy.nn.add` and then run this function with use.nn=TRUE. The main difference between `lxy.ptsh.add` and `lxy.nn.add` is that `lxy.ptsh.add` finds nearest neighbors for a random sample of points, and doesn't actually save the nearest neighbor information for individual points, whereas `lxy.nn.add` identifies and saves nearest neighbor information for each and every point.

Value

A list of lists, one for each plot containing the filename (NULL if no png made), the image dimensions (or NULL), the descriptive text, the id, and a matrix of the values

See Also

`lxy.ptsh.add`, `lxy.nn.add`

Examples

```
# data(toni.lxy)
# toni.lxy <- lxy.ptsh.add(toni.lxy)
# lxy.plot.ptsh(toni.lxy)
```

---

lxy.plot.sfinder   *Plot distributions of 's' such that the vmax/diffusion term in TSD are*
                          equivalent to the actual displacement

---

Description

This will find the value of s that will result in the time term of TSD being approximately equal to the distance term

Usage

```
lxy.plot.sfinder(lxy, id = NULL, delta.t = "auto", delta.t.auto.n = 8,
  delta.t.err = 0.01, outline = FALSE, desc = c(0, 1, 3)[2],
  cex.desc = 0.8, col.desc = "darkgreen", time.term = c("vmax", "dif")[1],
  mar = c(3, 3, if (title.show) 2.8 else 0.7, 0.5), mgp = c(1.8, 0.5, 0),
  figs.per.page = NULL, title = NULL, title.show = TRUE,
  panel.num = NULL, panel.num.inside.plot = !title.show, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.overwrite = TRUE, png.pointsize = 12 + (png.width - 480)/80, ...)
```

Arguments

`lxy`	A LoCoH-xy object
`id`	The id(s) of the individual(s) to include in the plot
`delta.t`	numeric vector of delta.t values (in seconds) for which a s-values will be computed. Can also be 'auto' in which case `delta.t.auto.n` delta.t values will be computed ranging from 3 * tau to the 80th percent point-to-point interval for the entire dataset
`delta.t.auto.n`	The number of delta-ts to use when delta.t="auto"
`delta.t.err`	The proportion of delta.t within which the interval between two points must fall to be included in the plot
`outline`	Show outliers in the box plots T/F
`desc`	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
`cex.desc`	The expansion factor for the descriptive text. Numeric value.
`col.desc`	The color of the descriptive text. Color value.
`time.term`	The type of time term to use ('vmax' is computed as deltaT * maximum possible velocity, while 'dif' computes Gauusian diffusion distance)
`mar`	The plot margins. A four item numeric vector
`mgp`	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
`figs.per.page`	The number of plots per page.
`title`	The title to be displayed. Character. If NULL a title will be constructed.
`title.show`	Whether to show the title. T/F.
`panel.num`	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
`panel.num.inside.plot`	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F
`png.dir`	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
`png.dir.make`	Whether to create png.dir if it doesn't exist. T/F
`png.width`	The width of the PNG image
`png.height`	The height of the PNG image
`png.overwrite`	Whether to overwrite an existing PNG file if it exists. T/F
`png.pointsize`	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
`...`	Additional parameters that will be passed to the [plot](#) function

Value

A list of lists, one for each plot containing the filename (NULL if no png made), the image dimensions (or NULL), the descriptive text, the id, and a list of the svals

```
lxy.plot.tspan                    Plot distributions of the time span of nearest neighbors
```

Description

Plot distributions of the time span of nearest neighbors as a proportion of the median sampling interval

Usage

```
lxy.plot.tspan(lxy, s = NULL, k = NULL, a = NULL, r = NULL, id = NULL,
    type = c("hist", "boxplot", "mean")[1], outline = FALSE, breaks = 20,
    desc = c(0, 1, 3)[2], cex.desc = 0.8, col.desc = "darkgreen",
    col.hist = "gray80", mar = c(if (type == "hist") 1.5 else 3, 3, if (type
    == "hist") 2 else if (title.show) 3 else 0.5, 0.5), mgp = c(1.9, 0.5, 0),
    figs.per.page = NULL, title = NULL, title.show = TRUE,
    title.obj.name = FALSE, panel.num = NULL,
    panel.num.inside.plot = !title.show, panel.num.cex = 2,
    no.sci.notation = FALSE, png.dir = NULL, png.dir.make = TRUE,
    png.width = 800, png.height = png.width, png.overwrite = TRUE,
    png.pointsize = 12 + (png.width - 480)/80, status = TRUE, ...)
```

Arguments

lxy	A [LoCoH-xy](#) object
s	The s value(s) of nearest neighbor sets to include in the plot. If NULL, all values will be used
k	A k-value for the number of nearest neighbors around each point to include in the plot
a	A a-value for the number of nearest neighbors around each point to include in the plot
r	A r-value for the number of nearest neighbors around each point to include in the plot
id	The id(s) of the individual(s) to include in the plot
type	The type of plot for the time span: 'hist', 'boxplot', or 'mean'
outline	Show outliers in the box plots T/F
breaks	The breaks parameter for a histogram, see [hist](#)
desc	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top.
cex.desc	The expansion factor for the descriptive text. Numeric value.
col.desc	The color of the descriptive text. Color value.
col.hist	The color of the histogram bars. Color value.
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
figs.per.page	The number of plots per page.

title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F
title.obj.name	Whether to add the name of the lxy object to the plot title (ignored if `title` is passed). T/F
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. Ignored if panel.num is NULL. T/F
panel.num.cex	The expansion factor for the panel number. Ignored if panel.num is NULL.
no.sci.notation	
	Whether to avoid the use of scientific notation on labels on the x-axis. T/F
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
status	Show progress bar and status messages. T/F
...	Additional parameters that will be passed to the [plot](#) function

Value

A list of plots created, with one element per id, and each element consisting of another list with elements for the filename (or NULL), filename dimensions in pixels (or NULL), the timespan values (as a list with one element for each value of s), descriptive text, and the k/a/r value

---

lxy.proj.add	*Specify the projection system for a LoCoH-xy object*

Description

Specify the projection system for a LoCoH-xy object

Usage

```
lxy.proj.add(lxy, proj4string = CRS(as.character(NA)))
```

Arguments

lxy	[LoCoH-xy](#) object
proj4string	An object of class "CRS", projection string

Details

This function will add projection information to a [LoCoH-xy](#) object. Note this does not reproject coordinates, it only adds or changes the projection information. Any existing projection information will be overwritten.

Value

A [LoCoH-xy](#) object

See Also

[CRS-class](#) for arguments accepted by the [CRS](#) function.

Examples

```
# lxy <- lxy.proj.add(lxy, CRS("+proj=utm +south +zone=34"))
```

---

lxy.ptsh.add   *Compute s vs. proportion-time-selected hulls table*

Description

Identify the values of s that result in proportion p of hulls being time-selected

Usage

```
lxy.ptsh.add(lxy, id = NULL, k = 10, n = 200, samp.idx = NULL,
  sinit = 0.005, ptsh.target = 1:9/10, ptsh.max = 0.98, ptsh.buf = 0.01,
  max.iter = 15, max.loops = 15, max.reached = c("warning", "stop")[1],
  time.term = c("vmax", "dif")[1], FNN.algorithm = c("cover_tree",
  "kd_tree", "VR", "CR")[2], use.nn = FALSE, plotme = TRUE, save = TRUE,
  nn.add = FALSE, use.pb.n = 200, ptsh.exists = c("replace", "append")[2],
  beep = FALSE, status = TRUE)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The name(s) of individuals to analyze
k	Value for the k-method that will be used to create hulls for the sample points (and then used to find the proportion of time-selected hulls)
n	Number of randomly selected points to use as samples (ignored if samp.idx is passed)
samp.idx	A vector of indices of the points to use as samples
sinit	The initial value of *s* that will be used to start the iteration to find that value of *s* that produces a proportion of time selected hulls >= ptsh.max
ptsh.target	A vector of values 0..1 that will serve as the target *ptsh* values. For each ptsh target, script will try to find a value of *s* that generates the targeted proportion of time selected hulls (within ptsh.buf)

ptsh.max	The highest value of ptsh for which a value of *s* will be computed. It is generally recommended that this be less than 1.
ptsh.buf	The level of accuracy or precison to which the found values of 's' return the targetted ptsh.
max.iter	The maximum number of times the script will double sinit in an effort to find the value of s that produces ptsh.max
max.loops	The maximum number of intermediate values of s the script will try when 'zooming' in on the target ptsh levels
max.reached	What to do if the maximum number of iterations is reached, warning or stop
time.term	The space-time transformation to use in the TSD distance metric: 'vmax' for the maximum velocity transformation (default) or 'dif' for the diffusion transformation.
FNN.algorithm	The algorithm to be used in the get.knnx() function in the package FNN
use.nn	Whether to use saved nearest neighbor sets for the ptsh computations. T/F
plotme	Whether to plot the ptsh vs. s curve. T/F
save	Whether to save the ptsh-s table in the LoCoH-xy object. T/F
nn.add	Whether to also identify and save nearest neighbor sets for all points for *k*=k and the values of *s* identified. T/F
use.pb.n	The sample size (i.e., n) above which a progress bar will be displayed
ptsh.exists	What to do if there is already a ptsh table in the LoCoH-xy object: 'replace' or 'append'
beep	Whether to beep when done. T/F
status	Display status messages. T/F

Details

When *s*=0, nearest neighbors are space-selected (i.e., point-to-point distance is merely Euclidean distance). For large values of *s*, nearest neighbors are effectively time-selected (e.g., temporally contiguous). One approach to selecting a value of *s* between these two ends of the spectrum is to pick a *s* value that returns an intermediate proportion of time-selected-hulls representing the desired balance between space and time selection (which in turn is shaped by the research question / objective). This function will find the value(s) of *s* that produce a desired proportion of time-selected-hulls using a randomly-selected subset of points. A general rule of thumb is to pick a value of *s* that results in a proportion of time-selected hulls from 0.4 to 0.6.

The script iteratively tries a variety of *s* values until it finds one that produces the desired proportion of time-selected hulls within a tolerance of ptsh.buf. If for example the vector of target values ptsh.target includes 0.5, and ptsh.buf = 0.01, the script will try to find a value of *s* that results in a proportion of time-selected hulls between 0.49 and 0.51.

See Also

[lxy.nn.add](lxy.nn.add)

lxy.repair   *Repair a LoCoH-xy object*

Description

Recreates a LoCoH-xy object

Usage

```
lxy.repair(lxy, fix.dup.ptid = FALSE, dt.int.round.to = 0.1,
  tau.diff.max = 0.02)
```

Arguments

lxy	A LoCoH-xy object
fix.dup.ptid	Generate new ptid value(s) if there are any duplicates
dt.int.round.to	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
tau.diff.max	The maximum deviation from tau (the median delta.t of the entire dataset), expressed as a proportion of tau, that time difference between two points must fall for the distance between those two points to be included in the calculation of the median step length

Details

This will return a new lxy object containing the original xys, dt, and ptid from lxy. All other items and attributes (including the comment) will be recreated or set to NULL

Set `fix.dup.ptid=TRUE` to automatically generate new ptid values if any duplcates are detected. If `fix.dup.ptid=FALSE` and duplicate ptid values are detected, an error message will result.

`tau.diff.max` exists to eliminate the inclusion of temporal outliers in the computation of the median step length. The time difference betwen points must be withint `tau.diff.max` of tau for that pair of points to be included in step length calculation.

Value

A LoCoH-xy object

See Also

`xyt.lxy`, `lxy.id.new`, `lxy.merge`

`lxy.reproject`   *Project or reproject coordinates*

Description

Project or reproject coordinates of a LoCoH-xy object

Usage

```
lxy.reproject(lxy, proj, dt.int.round.to = 0.1, tau.diff.max = 0.02,
  status = TRUE)
```

Arguments

`lxy`	A LoCoH-xy object
`proj`	Projection object of class `CRS-class`
`dt.int.round.to`	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
`tau.diff.max`	The maximum deviation from tau (expressed as a proportion of tau) that a point-to-point time difference must fall within for the point-to-point distance to be included in the calculation of the median step length
`status`	Show status messages (T/F)

Details

This function will reproject the locations to a new coordinate system. In order for this to work,

- The coordinate system of the current locations must be recorded. To see the current coordinate system, run `summary.locoh.lxy`. To add projection information (which you need to get from the source of the data), use `lxy.proj.add`.

- Argument `proj4string` which is an object of class `CRS-class` containing information about the new projection system. A good place to find the proj4 strings for a projection system is `http://www.spatialreference.org`.

Note that not all coordinate systems can be projected into all other coordinate systems. If you get an error message like *'non finite transformation detected'*, the coordinate system you are trying to project into may not be compatible with the existing coordinate system.

See Also

`lxy.proj.add`, `summary.locoh.lxy`, `CRS`

Examples

```
## Not run:
lxy.utm <- lxy.reproject(lxy, CRS("+proj=utm +north +zone=18 +ellps=WGS84"))

## End(Not run)
```

---

lxy.save   *Save a LoCoH-xy object to disk*

Description

Saves a LoCoH-xy object to disk, constructing a file name if needed that reflects the contents

Usage

```
lxy.save(lxy, file = NULL, save.as = NULL, dir = ".", suf = NULL,
  compress = TRUE, auto.num.files = TRUE, width = 2)
```

Arguments

lxy	A [LoCoH-xy](#) object
file	Optional file name
save.as	The name of the saved object, if NULL the original object name will be used
dir	Directory for the output file (relative to the working directory), ignored if `file` passed
suf	A suffix that will be used in the construction of the file name, ignored if `file` is passed
compress	Compress file (T/F)
auto.num.files	Use a numeric token as part of the constructed file name to get a unique file name(T/F)
width	The number of digits in auto.num.files

See Also

[xyt.lxy](#), [lhs.save](#)

---

lxy.subset   *Take a subset of a LoCoH-xy object*

Description

Returns a subset of a LoCoH-xy object based on the id, ptid, idx, or dates

Usage

```
lxy.subset(lxy, id = NULL, ptid = NULL, idx = NULL, dt.start = NULL,
  dt.end = NULL, dt.int.round.to = 0.1, tau.diff.max = 0.02)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The id value(s) to include in the subset
ptid	A vector of ptid values for the subset of points
idx	A vector of indices for the subset of points
dt.start	A starting date for the subset. POSIXct object (or something that can be coered to POSIXct)
dt.end	An ending date for the subset. POSIXct object (or something that can be coered to POSIXct)
dt.int.round.to	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
tau.diff.max	The maximum deviation from tau (expressed as a proportion of tau) that a point-to-point time difference must fall within for the point-to-point distance to be included in the calculation of the median step length

Value

A [LoCoH-xy](#) object

See Also

[xyt.lxy](#), [lxy.repair](#), [lxy.thin.byfreq](#), [lxy.thin.bursts](#), [lxy.merge](#)

---

lxy.thin.bursts	*Thins out the 'bursts' in a GPS dataset*

Description

Thin out closely-timed bursts of locations

Usage

```
lxy.thin.bursts(lxy, id = NULL, thresh = NULL, replace = c("mean",
  "median")[2], info.only = FALSE, dt.int.round.to = 0.1,
  tau.diff.max = 0.02, status = TRUE)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The id value(s) to be thinned
thresh	The threshhold for delta.t below which a pair of points is considered to be part of a burst, expressed either as a proportion of the median sampling frequency (0..1) or an absolute unit of time (in seconds)
replace	The burst replacement method ('mean' or 'median')
info.only	Show information about number of bursts (only)

dt.int.round.to

The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)

tau.diff.max   The maximum deviation from tau (the median delta.t of the entire dataset), expressed as a proportion of tau, that time difference between two points must fall for the distance between those two points to be included in the calculation of the median step length

status   Show status messages. T/F

Value

a LoCoH-xy object

Note

Many GPS devices have a feature to save 'bursts' of points close together in time (relative to the dominant sampling frequency) The 'burst' feature should not be confused with point averaging, whereby a GPS device internally averages locations for a period of time (e.g., 2 minutes) but saves a single location.

This function can be used to thin out 'bursts' of locations, when they are an artifact of data collection and therefore not desirable. Each group of points in a burst is replaced with a single point.

thresh is a numeric value for identifying which points should be considered part of a burst. When thresh < 1, it is taken to be a proportion of the median sampling frequency (0..1), and any pair of points that are recorded closer in time are considered part of a 'burst', When thresh > 1, it is taken to be absolute unit of time (in seconds).

To identify whether there are bursts in a LoCoH-xy dataset, and the sampling frequency of those bursts (i.e., the value

The two replacement methods are /codereplace="mean", in which case a burst of locations is replaced by a single point consisting of the spatial and temporal average of the burst, or /codereplace="median" in which case a burst of locations is replaced by the location at the temporal median of the burst. If /codereplace="mean", then any ancillary variables for the 'new' points will be set to NA.

Because this function deletes points, the nearest neighbor lookup table of the LoCoH-xy object (if any) will be deleted, and the parameters for random walk model will be recomputed.

See Also

lxy.plot.freq, lxy.thin.byfreq

---

lxy.thin.byfreq   *Delete locations to harmonize the sampling frequency and time dura-*
                         tion

Description

Standardize the sampling frequency and duration across individuals in a LoCoH-xy object by deleting points

Usage

```
lxy.thin.byfreq(lxy, id = NULL, trim.ends = TRUE, dt.start = NULL,
  dt.end = NULL, byfreq = TRUE, samp.freq = "lcm", lcm.round = 120,
  lcm.max.iter = 300, status = TRUE, dt.int.round.to = 0.1,
  tau.diff.max = 0.02)
```

Arguments

lxy	A [LoCoH-xy](#) object
id	The id value(s) to be harmonized
trim.ends	Truncate points from either end of the timeline to achieve a common time window, T/F
dt.start	The starting date-time that all individual trajectories will be truncated to. If NULL, the first date-time that all points have in common will be used.
dt.end	The end date-time that all individual trajectories will be truncated to. If NULL, the last date-time that all points have in common will be used.
byfreq	Delete points to achieve a common sampling frequency (samp.freq), T/F
samp.freq	The common time step for the output (in seconds). Can also be set to "lcm", in which case the least common multiple of the median time step of each individual will be computed
lcm.round	When samp.freq="lcm", the median time step for each individual will be rounded to the nearest interval of lcm.round (in seconds)
lcm.max.iter	The maximum number of iterations to be used in the algorithm that finds the least common multiple of the median time steps
status	Show messages, T/F
dt.int.round.to	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
tau.diff.max	The maximum deviation from tau (the median delta.t of the entire dataset), expressed as a proportion of tau, that time difference between two points must fall for the distance between those two points to be included in the calculation of the median step length

Value

a [LoCoH-xy](#) object

Note

This function processes a [LoCoH-xy](#) object that contains movement data for several individuals, and removes points such that the output contains a fixed start and end date for each individual, as well as an approximately uniform sampling frequency (time step).

Before using this function, you should clean your data of all abnormally short time intervals (e.g., bursts). See `lxy.thin.bursts`.

If you know the time interval the data was *supposed* to be sampled (e.g., every 20 minutes), you should pass that value for samp.freq (expressed in seconds). If samp.freq="lcm", the function will automatically compute the common time step for the individuals by taking the least common

multiple of the median time steps of each individual. You can deal with noise by rounding the median sampling interval to the value of `lcm.round` (expressed in seconds).

The function `lxy.plot.freq` can help you see the 'actual' sampling intervals in the data (set by.date=TRUE).

Because this function deletes points, the nearest-neighbors lookup table of the LoCoH-xy object (if any) will be deleted.

See Also

`xyt.lxy`, `lxy.plot.freq`, `lxy.thin.bursts`

---

matperim   *matperim*

Description

matperim

Usage

```
matperim(mat)
```

Arguments

mat   2-column matrix representing the xy coordinates of a closed polygon or line
                   segment

Value

The length of the perimeter

---

move.lxy   *Converts a Move object to a LoCoH-xy object*

Description

Converts a Move or MoveStack object from the move package to a LoCoH-xy object

Usage

```
move.lxy(move.obj, use.utm = FALSE, xcoord = NULL, ycoord = NULL,
  proj = NULL, anv.flds = NULL, ptid = NULL, del.dup.xyt = TRUE,
  dup.dt.check = TRUE, dt.int.round.to = 0.1, tau.diff.max = 0.02,
  req.id = TRUE, warn.latlong = TRUE, status = TRUE)
```

Arguments

move.obj	A object of class Move or MoveStack.
use.utm	Import UTM coordinates if found *and* all points fall within the same UTM zone (T/F).
xcoord	The name of a field in the data table for the x-coordinates. If NULL, the default coordinates will be used.
ycoord	The name of a field in the data table for the y-coordinates. If NULL, the default coordinates will be used.
proj	Projection object of class [CRS-class](). Used *only* to specify the coordinate system when the coordinates being imported are coming from the data table
anv.flds	The name of field(s) in the Move object data table that will be imported as ancillary variables. Can also be 'all'.
ptid	The name of a numeric field in the Move object data table containing unique numeric values for each point. If NULL new point id values will be created.
del.dup.xyt	Whether to delete duplicate rows with the same x, y, dt, and id value (T/F).
dup.dt.check	Whether to check to make sure there are no duplicate date values for the same id.
dt.int.round.to	
	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps).
tau.diff.max	The maximum deviation from tau (expressed as a proportion of tau) that a point-to-point time difference must fall within for the point-to-point distance to be included in the calculation of the median step length.
req.id	Require a value for id (T/F).
warn.latlong	Show a warning message if it detects coordinates may be in latitude-longitude (T/F).
status	Show status messages (T/F).

Value

A [LoCoH-xy]() object

Note

Move and MoveStack are formal classes from the *Move* package for storage of movement data for a single (Move) or multiple (MoveStack) indviduals. move.lxy will convert these objects to [LoCoH-xy]() objects. To import the attribute data associated with each point (termed ancillary variables in T-LoCoH), set anv.flds to the names of the field(s) you wish to import (or anv.flds="all"). Note that the Move and MoveStack classes store a lot more metadata than LoCoH-xy objects, including unused records, which are not supported by the [LoCoH-xy]() data class and are not imported.

See Also

[xyt.lxy]()

Examples

```
## Not run:
require(move)
leroy.move <- move(system.file("extdata","leroy.csv.gz",package="move"))
class(leroy.move)
leroy.lxy <- move.lxy(leroy.move, anv.flds=c("ground.speed", "heading"))
summary(leroy.lxy)

## End(Not run)
```

---

mvee   *mvee*

Description

Computes the minimum volume enclosing ellipsoid around a set of points using the Khachiyan Algorithm. In two dimensions, this is equivalent to the bounding ellipse.

Usage

```
mvee(xy = NULL, tolerance = 0.005, plotme = FALSE, max.iter = 500,
  shiftxy = TRUE, no.ellipse.val = 1, checks = TRUE)
```

Arguments

xy	a two-column data frame containing x and y coordinates. If NULL then a random sample set of 10 points will be generated
tolerance	a tolerance value
plotme	Plot the points and ellipse (T/F)
max.iter	Maximum number of iterations before giving up. If the script tries this number of iterations but still can't get to the tolerance value, it displays an error message and returns NULL
shiftxy	Apply a shift to the coordinates to make them smaller and speed up the matrix calculations, then reverse the shift to the center point of the resulting ellipoid (T/F)
no.ellipse.val	Determines what the function returns if a mvee can not be found (e.g., if the points are colinear). The default value 1 means a NULL will be returned. A value of 2 means a list of NAs will be returned which is more useful if these values will be stored in a data frame.
checks	Check xy object for the correct data type and remove any duplicate points. T/F

Value

A list containing the "center form" matrix equation of the ellipse. i.e. a 2x2 matrix "A" and a 2x1 vector "C" representing the center of the ellipse such that: $(x - C)' A (x - C) <= 1$ List elements include A - 2x2 matrix C - 2x1 vector of the center coordinates elps.axes.lngth - 2x1 vector whose elements are one-half the lengths of the major and minor axes (i.e., variables a and b in the standard ellipse equation) alpha - angle of rotation

Note

Adapted by Andy Lyons from Matlab code by Nima Moshtagh. Copyright (c) 2009, Nima Mosh-
tagh http://www.mathworks.com/matlabcentral/fileexchange/9542 http://www.mathworks.com/matlabcentral/fileexchange/
http://stackoverflow.com/questions/1768197/bounding-ellipse

---

n2z   *Convert NULL to a zero*

Description

Convert NULL to a zero

Usage

```
n2z(x)
```

Arguments

x   Input object

---

plot.locoh.lhs   *Plot a LoCoH-hullset object*

Description

Multi-purpose plotting function for a LoCoH-hullset object

Usage

```
## S3 method for class 'locoh.lhs'
plot(x, lhs, id = NULL, k = NULL, r = NULL,
  a = NULL, s = NULL, hs.names = NULL, iso = FALSE, rast = FALSE,
  hulls = FALSE, hpp = FALSE, dr = FALSE, nn = FALSE,
  ellipses = FALSE, allpts = FALSE, ptid = NULL, ptid.highlight = TRUE,
  add = FALSE, aoi = NULL, iso.idx = NULL, iso.sort.metric = NULL,
  iso.legend = TRUE, legend.space = 0.25, dr.metric = NULL,
  dr.thresh.val = NULL, dr.thresh.type = NULL, dr.smooth = NULL,
  lwd.dr = 2, pch.allpts = 16, cex.nn = 2, cex.hpp = 0.6,
  cex.allpts = 0.5, cex.pp = 2, cex.axis = 0.8, cex.legend = 0.8,
  col.hpp = "gray50", col.hpp.na = "yellow", col.hulls.border = "gray50",
  col.hulls.fill = NA, col.ellipses = "red", col.allpts = c("auto",
  "gray")[1], col.nn = "black", col.nn.pp = "blue", col.iso.fill = 1,
  col.iso.opacity = 1, col.iso.border = NA, col.iso.scale = TRUE,
  col.dr = "red", hpp.classify = c("none", "hsp", hm.expr(names.only = TRUE,
  print = FALSE, desc = FALSE))[1], hpp.classify.bins = 10,
  hpp.classify.chop = 0.01, hpp.classify.legend = TRUE,
  hpp.classify.common.scale.discrete = TRUE, col.ramp = c("gray10,gray90",
  "rainbow")[1], col.ramp.bins = 10, hsp = NULL, record = FALSE,
  figs.per.page = 1, same.axes.4all = NULL, overlay = FALSE,
```

```
col.overlay.byhullset = FALSE, ufat = TRUE, ufipt = TRUE,
axes.show = TRUE, axes.ticks = axes.show, axes.titles = axes.show,
title = NULL, title.show = TRUE, title.inc = c("title", "hs.name", "id",
"kar", "features", "hpp.classify", "hmap", "ptid")[if (is.null(title)) 3:4
else 1], subtitle.inc = c("title", "hs.name", "id", "kar", "features",
"hpp.classify", "hmap", "ptid")[if (is.null(title)) 5:8 else 0], mar = c(if
(axes.titles || axes.ticks) 3.3 else 0.5, if (axes.titles || axes.ticks) 3.2
else 0.5, if (title.show) 3.2 else 0.5, 0.5), mgp = c(2, 0.7, 0),
lo.save = TRUE, lo.margins.set = TRUE, desc = c(NONE <- 0, BOTTOM <- 1,
TOP <- 3)[ifelse(figs.per.page == 1, 2, 1)], cex.desc = 0.8,
col.desc = "darkgreen", gmap = c("none", "roadmap", "satellite", "hybrid",
"terrain")[1], gmap.one4all = TRUE, tiff.fn = NULL, tiff.pct = FALSE,
tiff.bands = c(4, 3, 2), tiff.col = gray(0:255/255), tiff.buff = 0,
tiff.fill.plot = TRUE, shp.csv = NULL, layers = NULL, png.fn = NULL,
png.dir = NULL, png.dir.make = TRUE, png.fn.pre = NULL,
png.fn.mid = NULL, png.fn.suf = NULL, png.fn.incld.hs.name = TRUE,
png.each.plot.separate = TRUE, png.width = 800, png.height = png.width,
png.pointsize = 12 + (png.width - 480)/80, png.overwrite = TRUE,
status = TRUE, panel.num = NULL, panel.num.inside.plot = !title.show,
hmap = NULL, iso.level = NULL, xlim = NULL, ylim = NULL,
check.ap.value.in.hmparams = TRUE, hme = hm.expr(names.only = FALSE), ...)
```

Arguments

x	A [LoCoH-hullset](#) object
lhs	Deprecated, use x instead
id	The names of the individual(s) to include in the plot.
k	The k value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
r	The r value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
a	The a value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
s	The s value(s) of the hullset(s) to include in the plot. Numeric vector or comma-delimited character object.
hs.names	The name(s) of saved hullsets to include in the plot
iso	Whether to display isopleths. T/F.
rast	Whether to display rasterized isopleths. T/F.
hulls	Whether to display hulls. T/F.
hpp	Whether to display hull parent-points. T/F.
dr	Whether to display directional routes. T/F.
nn	Whether to display nearest neighbors (requires that a value for ptid is passed and nearest neighbors were saved when creating the hullset object). T/F.
ellipses	Whether to display bounding ellipses (requires that ellipses were saved when computing ellipses for the hullset). T/F.
allpts	Whether to display all points. T/F.

ptid	One or more ptid (point id) values. A separate plot will be drawn 'zoomed in' to each point indicated. Can also be "auto", in which case ptid will be selected at random. Used primarily to inspect the hull, nearest neighbors, and/or ellipse for specific hulls.
ptid.highlight	Whether to highlight the point specified by ptid. T/F.
add	Whether to add to the existing plot. T/F.
aoi	An area-of-interest object (e.g., box), used to 'zoom in' to specific parts of the plot. aoi objects may be created by the function aoi().
iso.idx	The index(s) of the isopleths to plot. Use the summary() function to see the indices of the isopleths.
iso.sort.metric	
	The sort metric(s) of the isopleths that will be displayed. Character.
iso.legend	Whether to include a legend for the isopleths.
legend.space	An expansion factor for the x-axis that will be used to make room for the legend.
dr.metric	The name of the metric for the directional routes to be displayed (acts to filter on which directional routes are displayed). See lhs.dr.add
dr.thresh.val	The threshhold value for the directional routes to be displayed (acts to filter on which directional routes are displayed). See lhs.dr.add
dr.thresh.type	The threshhold type for the directional routes to be displayed (acts to filter on which directional routes are displayed). See lhs.dr.add
dr.smooth	The smoothing factor for the directional routes to be displayed (acts to filter on which directional routes are displayed). See lhs.dr.add
lwd.dr	The line width of directional routes.
pch.allpts	The plot character for all points
cex.nn	The expansion factor for nearest neighbor points.
cex.hpp	The expansion factor for hull parent-points.
cex.allpts	The expansion factor for all points.
cex.pp	The expansion factor for the parent point.
cex.axis	The expansion factor for axis labels.
cex.legend	The expansion factor for the legend.
col.hpp	The color of hull parent points. Either a single color value or a vector of color values of the same length as the number of hulls.
col.hpp.na	The color of hull parent points that have no value of the metric specified by hpp.classify. Single color value.
col.hulls.border	
	The outline color of hulls. Either a single color value or a vector of color values of the same length as the number of hulls.
col.hulls.fill	The fill color of hulls.
col.ellipses	The outline color of bounding ellipses. Either a single color value or a vector of color values of the same length as the number of hulls.
col.allpts	The color of all points. Single color value.
col.nn	The color of nearest neighbors. Single color value.
col.nn.pp	The color of parent-points. Used only when ptid is passed. Single color value.

`col.iso.fill`	A number corresponding to a preset color ramp that will be used to display the isopleths. 1=red to blue, 2=yellow to red, 3=blue to red, 4=red to yellow. Alternately, a character vector of color values of the same length as the number of isopleth levels.
`col.iso.opacity`	
	A number 0..1 for the opacity of isopleth fill color, where 1 is opaque and 0 is transparent
`col.iso.border`	Color value of the isopleth border. Color value (use NA for no border).
`col.iso.scale`	When plotting a subset of isopleths (by ecc or par), whether to scale the isopleth colors to the values of the subset. Not being used.
`col.dr`	The color of directional routes.
`hpp.classify`	How to classify hull parent points. May be "none", the name of a hull metric, or "hsp" (hull scatterplot)
`hpp.classify.bins`	
	The number of equal-interval bins of the hull metric value to create when classifying hull parent-points by a hull metric.
`hpp.classify.chop`	
	The proportion of hull parent-points at the tails of the distribution to remove when calculating the bins.
`hpp.classify.legend`	
	Whether to include a legend for the classification of hull parent-points. T/F.
`hpp.classify.common.scale.discrete`	
	Whether to classify hull parent points using a common scale when there are multiple maps produced. Ignored if hpp.classify is 'none' or 'hsp'. T/F.
`col.ramp`	The first and last values of the color ramp to use when classifying hull parent-points by a hull metric. Two-item vector with color values.
`col.ramp.bins`	The number of bins in the color ramp when classifying hull parent-points by a hull-metric
`hsp`	The hull scatterplot object to use when displaying hull parent points by a hull scatterplot. May be a hull scatterplot object or the index of a hull scatterplot saved in the hullset.
`record`	Whether to open a new plot window and turn on recording. T/F.
`figs.per.page`	The number of plots per page.
`same.axes.4all`	Whether to use the same axes ranges for all plots. T/F.
`overlay`	Whether to overlay the plots. T/F.
`col.overlay.byhullset`	
	Whether to use a different color for each hullset when overlay=TRUE. Affects only isopleths, hulls and ellipses. T/F.
`ufat`	Whether to use user-friendly-axis-titles. T/F.
`ufipt`	Whether to use user-friendly-isopleth-plot-titles. T/F.
`axes.show`	Whether to show the axes. T/F.
`axes.ticks`	Whether to show the tick marks and labels on the axes. T/F.
`axes.titles`	Whether to show axes titles. T/F.
`title`	The title to be displayed. Character. If NULL a title will be constructed.
`title.show`	Whether to show the title. T/F.

title.inc	The element(s) to include in the title (ignored if `title` is passed or `overlay=TRUE`)
subtitle.inc	The element(s) to include in the subtitle (ignored if `title` is passed or `overlay=TRUE`)
mar	The plot margins. A four item numeric vector.
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector.
lo.save	Whether to save and reset the plot device margin settings (some wrapper functions that call this function don't want device settings reset). T/F.
lo.margins.set	Whether to save and reset the plot device margin settings (some wrapper functions that call this function don't want device settings reset). T/F.
desc	Which side to display automatically generated desciptive text (e.g. caption). 0=none, 1=bottom, 3=top. Ignored if `overlay=TRUE`.
cex.desc	The expansion factor for the descriptive text. Numeric value.
col.desc	The color of the descriptive text. Color value.
gmap	The name of a background image that will be downloaded from Google: `"none"`, `"roadmap"`, `"satellite"`, `"hybrid"`, or `"terrain"`. May also be a object of type `locoh.gmap`, see Notes.
gmap.one4all	Whether to download a single background image for all ids. T/F
tiff.fn	The path and name of a GeoTIFF file (e.g., satellite image) that will be displayed in the background. See notes.
tiff.pct	Whether or to convert the GeoTIFF to an indexed 256 color RGB image, which may speed up drawing. T/F.
tiff.bands	A vector of exactly one (for a single band image) or exactly three integers corresponding to the bands of the GeoTIFF image that will be mapped to the red, green and blue color guns respectively,. Ignored if `tiff.fn` only contains one band.
tiff.col	A vector of color values for plotting single-band images in the background. Ignored if using three bands.
tiff.buff	A numeric buffer distance that the range of the plot will be expanded so the points are not right on the edge of the GeoTIFF.
tiff.fill.plot	Whether to fill the entire plot area with the GeoTIFF. T/F.
shp.csv	The path and filename of a csv file that contains information about shapefiles, including layer names, file, and symbology.
layers	The name(s) of layers in shp.csv to display in the background. Will be displayed using the symbology in shp.csv. Character vector or comma delimited string
png.fn	The path and name of the PNG file to create (instead of displaying in a plot window)
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.fn.pre	A prefix that will be used in the construction of the PNG filename. Ignored if png.fn is passed
png.fn.mid	A mid-fix that will be used in the construction of the PNG filename. Ignored if png.fn is passed
png.fn.suf	A suffix that will be used in the construction of the PNG filename. Ignored if png.fn is passed

png.fn.incld.hs.name

               Whether to include the hullset name as part of the PNG filename. (T/F). Ignored if png.fn is passed

png.each.plot.separate

               Whether to make each plot in a separate plot / PNG. T/F. Ignored if png.fn is passed

png.width	The width of the PNG image. Ignored if png.fn is passed
png.height	The height of the PNG image. Ignored if png.fn is passed
png.pointsize	The pointsize (in pixels) for the PNG image (increase to make labels appear larger). Equivalent to the height or width of a character in pixels.
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
status	Whether to show messages. T/F
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character

panel.num.inside.plot

               Whether to display panel.num inside the plot area itself, as opposed to the title area

hmap	A named list of hull metric auxillary parameters, the name of each list element is the name of the variable
iso.level	The isopleth levels to plot, numeric vector
xlim	The lower and upper limit of the x-axis, two-element numeric vector
ylim	The lower and upper limit of the y-axis, two-element numeric vector

check.ap.value.in.hmparams

               Whether to check if hull metrics exist for the auxillary parameters passed. T/F

hme	A list of hull metric expressions
...	Other parameters, including any auxillary parameters required by certain hull metrics

Value

If png.fn or png.dir is passed, a list object where each element is a three-element list of the properties of the PNG file(s) created: $fn, $dim, $desc. Otherwise, returns a vector of desc

Note

This is a multi-purpose plotting function for LoCoH-hullset objects. You specify which objects in the hullset to include on the plot by passing parameters.

All hullsets have hulls and hull parent points that can be plotted (e.g., codehulls=TRUE, hpp=TRUE. All hullsets also have the original locations saved and these can be added to the plot by setting allpts=TRUE. Depending how the hullset was created (see lxy.lhs), some hullsets may also have the nearest neighbors for each parent point saved, which can also be plotted (nn=TRUE).

Hullsets for which have bounding ellipses (lhs.ellipses.add) or directional routes (lhs.dr.add) have been computed can have these elements plotted by setting ellipses=TRUE and dr=TRUE respectively. If isopleths have been constructed (lhs.iso.add), these can be plotted by setting iso=TRUE. If you only want to plot some of the isopleths, you can specify which one(s) to plot with the parameters iso.idx (the index(s) of the isopleths to plot, see summary.locoh.lhs) or iso.sort.metric.

The plot title can be manually set with the `title` parameter. If no value for `title` is passed, a title will be constructed. You can define what elements will go in the title and subtitle (i.e., the second line of the title) with the `title.inc` and `subtitle.inc` parameters. To omit the plot title completely, set `title.show=FALSE`.

To display an image from Google in the background, set gmap to `"roadmap"`, `"satellite"`, `"hybrid"`, or `"terrain"`. This requires an internet connection. When creating plots of hullsets from multiple individuals, gmap.one4all determines whether a single background image is used for all individuals (faster to download but could result in blurry backgrounds if the individuals occupy different parts of the landscape), or a separate image is used for each individual. In the later case, You may also set gmap to an object of type `locoh.gmap`, so the image(s) don't have to be downloaded each time. See `lhs.gmap` (in the tlocoh.dev package). `tiff.fn`, `tiff.pct`, `tiff.buff`, `tiff.bands`, `tiff.col`, and `tiff.fill.plot` control the display of a GeoTIFF image in the plot background. The GeoTIFF image must be georeferenced in the same coordinate system as the locoh-hullset object, and the pixel values must be 'prestretched' for display. `tiff.bands` controls which bands in the TIFF file will be displayed using the red, green, and blue color guns if using three bands. In a Landsat TM image, for example, the first four bands are blue, green, red, and infrared. To display a TM GeoTIFF image as 'natural colors', you would set `tiff.bands=c(3,2,1)`. If `tiff.pct=T`, the script will create a indexed 256-color version of the image, which may result in quicker drawing time particularly if several plots are being drawn. If using a single-band, for example a DEM or classified image, the colors are set by `tiff.col`. `tiff.buff` can be used to expand the range of values on the x and y axis so that you see a bit of the background image beyond the extent of the points.

---

plot.locoh.lxy   *Plot LoCoH-xy object*

Description

Multi-purpose plotting function for a LoCoH-xy object

Usage

```
## S3 method for class 'locoh.lxy'
plot(x, lxy, id = NULL, cex = 0.8, show.start = TRUE,
  show.end = TRUE, col = c("auto", "gray80")[1], connect.dots = TRUE,
  overlay = FALSE, status = TRUE, title = NULL, title.show = TRUE,
  axes.show = TRUE, axes.titles = axes.show, axes.ticks = axes.show,
  mar = c(if (axes.titles || axes.ticks) 3.3 else 0.5, if (axes.titles ||
  axes.ticks) 3.2 else 0.5, if (title.show) 3.2 else 0.5, 0.5), mgp = c(2,
  0.7, 0), lo.save = TRUE, panel.num = NULL,
  panel.num.inside.plot = !title.show, png.fn = NULL, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.overwrite = TRUE, png.pointsize = 12 + (png.width - 480)/80,
  tiff.fn = NULL, tiff.bands = c(3, 2, 1), tiff.col = gray(0:255/255),
  tiff.pct = FALSE, tiff.buff = 0, tiff.fill.plot = TRUE, layers = NULL,
  shp.csv = NULL, xlim = NULL, ylim = NULL, legend = NULL,
  add = FALSE, ...)
```

Arguments

x	A [LoCoH-xy](#) object
lxy	Deprecated, use x instead
id	A vector of the id value(s) to plot
cex	Character expansion factor for the points
show.start	Whether to highlight the starting location (if time stamps are present) in green. T/F.
show.end	Whether to highlight the end location (if time stamps are present) in red. T/F.
col	A single value or vector of color values. Can also be 'auto' in which case the colors saved in lxy will be used (which are rainbow by default).
connect.dots	Whether to draw line segments between consecutive locations. T/F.
overlay	Whether to overlay the plots of all individuals in lxy on one pair of axes (map). T/F.
status	Display status messages. T/F.
title	The title to be displayed. Character. If NULL a title will be constructed.
title.show	Whether to show the title. T/F.
axes.show	Whether to show the axes. T/F.
axes.titles	Whether to show axes titles. T/F.
axes.ticks	Whether to show the tick marks and labels on the axes. T/F.
mar	The plot margins. A four item numeric vector
mgp	The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector
lo.save	Whether to save and reset the plot device margin settings (some wrapper functions that call this function don't want device settings reset). T/F.
panel.num	A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character
panel.num.inside.plot	
	Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F
png.fn	The path and name of the PNG file to create (instead of displaying in a plot window)
png.dir	The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed
png.dir.make	Whether to create png.dir if it doesn't exist. T/F
png.width	The width of the PNG image
png.height	The height of the PNG image
png.overwrite	Whether to overwrite an existing PNG file if it exists. T/F
png.pointsize	The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger)
tiff.fn	The path and name of a GeoTIFF file (e.g., satellite image) that will be displayed in the background. See notes.
tiff.bands	A vector of threee integers corresponding to the bands of the GeoTIFF image that will be mapped to the red, green and blue color guns respectively.

`tiff.col`	A vector of color values for plotting single-band images in the background. Ignored if using three bands.
`tiff.pct`	Whether or to convert the GeoTIFF to an indexed 256 color RGB image, which may speed up drawing. T/F.
`tiff.buff`	A numeric buffer distance in map units that the range of the plot will be expanded so the points are not right on the edge of the GeoTIFF.
`tiff.fill.plot`	Whether to fill the entire plot area with the GeoTIFF. T/F.
`layers`	The name(s) of layers in shp.csv to display in the background. Will be displayed using the symbology in shp.csv. Character vector or comma delimited string
`shp.csv`	The path and filename of a csv file that contains information about shapefiles, including layer names, file, and symbology.
`xlim`	The lower and upper limit of the x-axis, two-element numeric vector
`ylim`	The lower and upper limit of the y-axis, two-element numeric vector
`legend`	One of the following keywords specifying where to put a legend when overlaying the locations of multiple individuals: `bottomright`, `bottom`, `bottomleft`, `left`, `topleft`, `top`, `topright`, `right` or `center`
`add`	Whether to add to an existing plot. T/F.
`...`	Additional parameters that will be passed to the [plot](#) function

Details

This is a multi-purpose plotting function for a LoCoH-xy object, with several custom arguments useful for plotting a movement trajectory. You may also extract the SpatialPointsDataFrame element of the LoCoH-xy object directly by referencing the `pts` element (e.g., `x$pts`, where `x` is a LoCoH-xy object).

See Also

Vignette on T-LoCoH data classes, Tutorial Vignette (for details on displyaing GIS layers and raster images in the background)

---

secs.fmt	*Format a number of seconds as minutes, hours or days*

Description

Returns a formated version of a time value

Usage

```
secs.fmt(secsVal, round.pos = 1)
```

Arguments

`secsVal`	a time value (in seconds)
`round.pos`	The number of decimal places in the formatted string

Value

a character string

Note

This function only works with a single value. To format a multiple time values stored in a vector, use sapply or lapply

Based on the value of secsVal, the formatted version will be converted to days, hours, minutes, or seconds

---

square	*square*

Description

square

Usage

```
square(x)
```

Arguments

x	Input object

---

strSplitAtChar	*Split a character object into multiple lines*

Description

Split a character object at a specific character to wrap to multiple lines for plotting

Usage

```
strSplitAtChar(object, size, char = ".", separator = paste("\n", char, sep
 = ""), ...)
```

Arguments

object	Input character vector (should be length 1)
size	The maximum number of characters in one piece
char	The character to split the input string at
separator	The character to use as a separator in the returned object
...	Other arguments (unused)

Details

This will take a long character object and split it into pieces at character char such that the length of each piece is <= size. If separator is passed, the individual pieces will be concatenated using the separator character and returned as a character vector of length 1. Otherwise each piece will be returned as a separate element of a character vector.

---

strTrim   *Trim leading and/or trailing blanks from a character object*

Description

Trim leading and/or trailing blanks from a character object

Usage

```
strTrim(str, side = "both")
```

Arguments

str	Input string
side	Which side(s) of the input string to trim: 'left', 'right', or 'both'

---

summary.locoh.lhs   *Show summary of a LoCoH-hullset*

Description

Displays a summary of a LoCoH-hullset object

Usage

```
## S3 method for class 'locoh.lhs'
summary(object, lhs, file = "", id = NULL, k = NULL,
  r = NULL, a = NULL, s = NULL, hs.names = NULL, iso.details = FALSE,
  hsp.details = FALSE, desc = FALSE, compact = FALSE, ...)
```

Arguments

object	A [LoCoH-hullset](#) object
lhs	Deprecated, use object instead
file	A file name
id	The name(s) of individuals to summarize
k	The k value of hullsets to summarize
r	The r value of hullsets to summarize
a	The a value of hullsets to summarize
s	The s value of hullsets to summarize
hs.names	The name(s) of hullsets to summarize
iso.details	Display details of the isopleths. T/F
hsp.details	Display details about saved hull scatterplots. T/F
desc	Display the hullset description. T/F
compact	Use a compact format. T/F
...	Other arguments

---

summary.locoh.lxy *Show a summary of a LoCoH-xy object*

Description

Prints a summary of a locoh xy object (set of locations)

Usage

```
## S3 method for class 'locoh.lxy'
summary(object, lxy, file = "", id = NULL,
  dt.int = FALSE, round.coords = 1, ptsh = FALSE, ...)
```

Arguments

object	A [LoCoH-xy](#) object
lxy	Deprecated, use `object` instead
file	A file name where the results will be saved
id	The name of an individual(s)
dt.int	Whether to show a summary of the sampling interval
round.coords	The number of digits to display for the coordinates of the spatial extent
ptsh	Show table of 's' and proportion of time selected hulls values (if available)
...	Other arguments

See Also

[lxy.plot.freq](#), [lxy.ptsh.add](#)

---

textContrastColor *Choose a white or black plot background for a particular color of text*

Description

Choose a white or black plot background for a particular color of text

Usage

```
textContrastColor(x, ...)
```

Arguments

x	A color value
...	Other arguments (not used)

---

toni   *Tracking data for one buffalo in South Africa*

Description

This dataframe contains the locations from a GPS collar fitted to a buffalo in Kruger National Park, South Africa in 2005-06.

Usage

```
data(toni)
```

Format

A dataframe with 6371 rows. Columns are id, lat, long, and timestamp (in UTC). Datum presumed to be WGS84.

Source

MoveBank <http://www.movebank.org>

Name: Kruger African Buffalo, GPS tracking, South Africa

Acknowledgements: Collection of Kruger Park Buffalo data funded by NSF Grant DEB-0090323 to Wayne M. Getz

Principal Investigator Name: Paul Cross

Examples

```
data(toni)
head(toni)
plot(toni[,2:3])
```

---

tsd.zvals   *Space-time transformation term for TSD*

Description

Returns the values for the "time axis" of the TSD distance metric

Usage

```
tsd.zvals(delta.t, sVal, type, d.bar = NULL, tau = NULL, vmax = NULL)
```

Arguments

`delta.t`	is the time difference in seconds
`sVal`	is the value of S
`type`	The type of space-time transformation: 'vamx' for the maximum velocity transformation, or 'dif' for diffusion
`d.bar`	is the median step length (for the entire dataset)
`tau`	is the median sampling interval (for the entire dataset)
`vmax`	The maximum velocity parameter (used only when `type='vmax'`)

`vectorize.parameter`   *Convert a comma-separated string to a vector*

Description

Convert a comma-separated string to a vector

Usage

```
vectorize.parameter(x, n2z = FALSE, type = c("numeric", "character")[1],
  sort.res = TRUE)
```

Arguments

`x`	The vector or character string to be evaluated
`n2z`	Convert null values to zero or an empty string (depending on `type`
`type`	The data type: `numeric` or `character`
`sort.res`	Whether to sort the result. T/F

Value

A vector

`xyt.lxy`   *Create a LoCoH-xy object*

Description

Creates a [LoCoH-xy](#) object from a set of location data

Usage

```
xyt.lxy(xy, dt = NULL, tz = NULL, id = NULL, ptid = NULL,
  proj4string = CRS(as.character(NA)), anv = NULL, anv.desc = NULL,
  col = NULL, del.dup.xyt = TRUE, dup.dt.check = TRUE,
  show.dup.dt = FALSE, dt.int.round.to = 0.1, tau.diff.max = 0.02,
  req.id = TRUE, warn.latlong = TRUE, show.bad.timestamps = FALSE,
  status = TRUE)
```

Arguments

xy	A two-column matrix or data frame containing the xy coordinates of the points
dt	Optional vector of date-time values (either POSIXct objects or objects that can be coerced to POSIXct)
tz	The name of the time zone that will be assigned if not explicit in dt.
id	Optional character vector or factor containing the name(s) of the individual(s) of each location.
ptid	Optional integer vector of point id values
proj4string	Projection string object of class [CRS-class](#)
anv	Optional ancillary variables for each point (data frame with same number of records as xy)
anv.desc	Optional character vector with descriptions of the ancillary variables (in the same order as they appear in anv)
col	Optional vector of color values (one for each point), or a single color value
del.dup.xyt	Whether to delete duplicate rows with the same x, y, dt, and id value. (T/F)
dup.dt.check	Whether to check to make sure there are no duplicate date values for the same id
show.dup.dt	Whether to show duplicate time stamps (if found) (T/F)
dt.int.round.to	The proportion of the median sampling frequency that time intervals will be rounded to when computing the frequency table of sampling intervals (no change is made to the time stamps)
tau.diff.max	The maximum deviation from the median sampling interval (tau), expressed as a proportion of the median sampling interval, see details
req.id	Require a value for id (T/F)
warn.latlong	Show a warning message if coordinates appear to be in geographic coordinates (T/F)
show.bad.timestamps	If TRUE, will print any timestamps that didn't successfully convert to a POSIXct object (T/F)
status	Show status messages (T/F)

Details

At a minimum, a [LoCoH-xy](#) object contains a set of points. It can also contain date-time values for each point, the name of the individual(s) associated with each point, and a nearest nieghbors lookup table.

Importing date-time values into R is often a painful process. You may want to try exporting your date-time values from your spreadsheet or database as a text field that is formatted in a style that R will recognize as a valid time. To see a date-time format that R will be able to convert to a date- time object (class *POSIXct*), type Sys.time() at the console. Once you've noted the date-time format recognized by R, try to format the date field in your spreadsheet or database in a similar manner, then export the values in a text format (e.g., csv). For example in MS Access, you can use the format() function as part of a query to format a date field, e.g. format(PointDate,"yyyy-mm-dd hh:nn:ss"). In Excel, you can go to Format - Cells and give the date-time cells a custom format yyyy-mm-dd hh:mm:ss. See the Appendices of the T-LoCoH tutorial for more examples.

The time zone paramter `tz` is optional. If `tz` is omitted, R will try to get the timezone from dt, and if there is no timezone specified set it to UTC. Valid timezone names are to some extent OS specific. To see valid timezone names, you may run `OlsonNames()`. If dt is of class POSIXct (which stores the time zone) and a different value for `tz` is passed, a prompt will ask whether date values should be converted.

By default, if the function finds two or more locations with the same time stamp for the same id (individual), an error will be triggered. This usually results from a data processing error (e.g., duplication of a row), or a conversion problem or rounding issue in the time stamp values. To see which records have duplicate time stamps, pass `show.dup.dt=TRUE`. To disable the checking for duplicate time stamps, pass `dup.dt.check=TRUE`. Duplicate locations are allowed, as are duplicate time stamps for different individuals.

`ptid` is an optional vector of numeric id values (i.e., primary key) for each location. When present, point id values of parent points will be saved in the data table of hulls. This enables linking the outputs of tlocoh (hulls and hull metrics) with other data and/or other software tools. See also [lhs.exp.csv](lhs.exp.csv) and [lhs.exp.shp](lhs.exp.shp). ptid should not be confused with id, which is a character vector of the name(s) of the individuals in the dataset.

`tau.diff.max` controls which consecutive pairs of points will be used to compute the median step length (d-bar) and the maximum observed speed (v-max). Pairs of points whose sampling interval (i.e., time difference) deviates from the median by more than `tau.diff.max` will be excluded from this computation (presumably because there was a drop-out in the data, and the distance between those pairs of locations does not represent the characteristic movement pattern). However for data that have a wide distribution of sampling intervals (e.g., manually collected radio telemetry locations), this filtering may result in an insufficient number of pairs. In this case, increase `tau.diff.max` or set it to zero to disable filtering completely.

Other variables, including environmental variables or biometric variables can be brought in using the anv parameter. These variables can then be used in hull metrics.

Value

A object of class locoh.lxy

See Also

[lxy.nn.add](lxy.nn.add), lxy.repair, lxy.subset

Examples

```
## Not run:
#Create an unbounded random walk as a sample dataset
n <- 500; stepsize <- 5
theta <- runif(n, min=0, max=2*pi)
start <- c(100,100)
xy <- data.frame(x=start[1]+cumsum(stepsize * cos(theta)), y=start[2]+cumsum(stepsize*sin(theta)))
plot(xy, pch=20, type="b", lty=1)
timestamps <- Sys.time() + 3600 * (1:n-1)

#Combine the xy locations and timestamps into a LoCoH-xy object
lxy <- xyt.lxy(xy=xy, dt=timestamps, id="broken_tooth")
summary(lxy)

## End(Not run)
```

Index