Package ‘rKIN’

June 12, 2023

Type Package
Title (Kernel) Isotope Niche Estimation
Version 1.0.0
Date 2023-06-12

URL https://github.com/salbeke/rKIN

BugReports https://github.com/salbeke/rKIN/issues

Depends R (>= 3.0), ks, sf
Imports ggplot2, MASS, RColorBrewer, randomcolorR, shades, dplyr

Maintainer Shannon E Albeke <salbeke@uwyo.edu>

Description Applies methods used to estimate animal hom range, but
instead of geospatial coordinates, we use isotopic coordinates. The estimation
methods include: 1) 2-dimensional bivariate normal kernel utilization density
estimator, 2) bivariate normal ellipse estimator, and 3) minimum convex polygon
estimator, all applied to stable isotope data. Additionally, functions to
determine niche area, polygon overlap between groups and levels (confidence
contours) and plotting capabilities.

License GPL (>= 3)

LazyData TRUE

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Shannon E Albeke [aut, cre]

Repository CRAN

Date/Publication 2023-06-12 19:20:02 UTC
**Description**

A simple wrapper for the ks::Hbcv function.

**Usage**

```r
bw_hbcv(x)
```

**Arguments**

- `x` 2d matrix of data values.

**Value**

A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

**Author(s)**

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming
bw_hlscv

Examples

data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1",]
# Calculate the bandwidth
bw_hbcv(as.matrix(spec1[, c("Ave_C", "Ave_N"))])

---

bw_hlscv

Least-squares cross-validation bandwidth matrix selector for multivariate data.

Description

A simple wrapper for the ks::Hlscv function.

Usage

bw_hlscv(x)

Arguments

x

2d matrix of data values.

Value

A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples

data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1",]
# Calculate the bandwidth
bw_hlscv(as.matrix(spec1[, c("Ave_C", "Ave_N"))])
bw_hnm

*Normal mixture bandwidth.*

**Description**

A simple wrapper for the `ks::Hnm` function.

**Usage**

```r
bw_hnm(x)
```

**Arguments**

- `x`  
  2d matrix of data values.

**Value**

A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

**Author(s)**

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

**Examples**

```r
data("rodents")
# Subset the data for a single species
spec1 <- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hnm(as.matrix(spec1[, c("Ave_C", "Ave_N")]))
```

----------

bw_hns

*Normal scale bandwidth using ks::Hns function.*

**Description**

A simple wrapper for the `ks::Hns` function.

**Usage**

```r
bw_hns(x)
```

**Arguments**

- `x`  
  2d matrix of data values.
**bw_hpi**

**Value**
A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

**Author(s)**
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

**Examples**
```r
data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hns(as.matrix(spec1[, c("Ave_C", "Ave_N"))])
```

---

**bw_hpi**

Default Plug-in bandwidth selector using ks::Hpi function.

**Description**
A simple wrapper for the ks::Hpi function.

**Usage**
```
bw_hpi(x)
```

**Arguments**

- `x` 2d matrix of data values.

**Value**
A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

**Author(s)**
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

**Examples**
```r
data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hpi(as.matrix(spec1[, c("Ave_C", "Ave_N"))])
```
**bw_hscv**

**Smoothed cross-validation bandwidth selector.**

**Description**

A simple wrapper for the ks::Hscv function.

**Usage**

```
bw_hscv(x)
```

**Arguments**

- `x`: 2d matrix of data values.

**Value**

A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

**Author(s)**

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

**Examples**

```r
data("rodents")
# Subset the data for a single species
spec1 <- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hscv(as.matrix(spec1[, c("Ave_C", "Ave_N")]))
```

---

**bw_hucv**

**Least-squares cross-validation bandwidth matrix selector for multivariate data.**

**Description**

A simple wrapper for the ks::Hucv function.

**Usage**

```
bw_hucv(x)
```

**Arguments**

- `x`: 2d matrix of data values.
Value
A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

Author(s)
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples
```r
data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hucv(as.matrix(spec1[, c("Ave_C", "Ave_N")]))
```

Description
A simple wrapper for the MASS::bandwidth.nrd function. Divides values by 4 to match the scale of ks methods

Usage
```
bw_ref(x)
```

Arguments
- `x` 2d matrix of data values.

Value
A numeric vector of estimated x and y bandwidths. Must subset your data if you wish to obtain group specific bandwidths.

Author(s)
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples
```r
data("rodents")
# Subset the data for a single species
spec1<- rodents[rodents$Species == "Species1", ]
# Calculate the bandwidth
bw_hucv(as.matrix(spec1[, c("Ave_C", "Ave_N")]))
```
### calcOverlap

**Calculate Percent Overlap of Isotopic Niche Space**

**Description**
Calculates the percent of polygon overlap between each group and level.

**Usage**
calcOverlap(estObj)

**Arguments**
estObj  List object of class estObj containing returned sf data frames from estimating functions estKIN, etc.

**Value**
A data.frame containing the percent of the polygon overlap for each group and level. Rows are the 1st input polygon, columns are the 2nd input, the returned area of overlap is divided by the area of the 1st polygon (row).

**Author(s)**
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

**Examples**
```r
library(rKIN)
data("rodents")
#estimate niche overlap between 2 species using kernel UD
test.kin <- estKIN(data=rodents, x="Ave_C", y="Ave_N", group="Species", levels=c(50, 75, 95), scaler=2)
#determine polygon overlap for all polygons
dat.olp <- calcOverlap(test.kin)
```

### createSPDF

**Internal helper function**

**Description**
Create empty sf data frame with estObj schema

**Usage**
createSPDF()
estEllipse

Value
An empty sf data frame object matching the expected schema of the estKIN function.

Author(s)
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

estEllipse Estimate Bivariate Normal Ellipse Isotope Niche

Description
Calculates the Bivariate Normal Ellipse Polygon for isotopic values at multiple confidence levels. Returns a list of sf data frames, each list item representing the grouping variable (i.e. species).

Usage
estEllipse(data, x, y, group, levels = c(50, 75, 95), smallSamp = FALSE)

Arguments
data data.frame object containing columns of isotopic values and grouping variables
x character giving the column name of the x coordinates
y character giving the column name of the y coordinates
group character giving the column name of the grouping variable (i.e. species)
levels Numeric vector of desired percent levels (e.g. c(10, 50, 90). Should not be less than 1 or greater than 100)
smallSamp logical value indicating whether to override minimum number of samples. Currently 10 samples are required.

Value
A list of sf data frames, each list item representing the grouping variable.

Author(s)
Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples
library(rKIN)
data("rodents")
#estimate niche overlap between 2 species using bivariate ellipse
test.elp<- estEllipse(data=rodents, x="Ave_C", y="Ave_N", group="Species", levels=c(50, 75, 95))
#determine polygon overlap for all polygons
plotKin(test.elp, scaler=2, title="Ellipse Estimates", xlab="Ave_C", ylab="Ave_N")
Description

Calculates the 2D kernel for isotopic values at multiple confidence levels. Returns a list of sf data frames, each list item representing the grouping variable (i.e. species).

Usage

```r
estKIN(
  data, 
  x, 
  y, 
  h = "ref", 
  hval = NULL, 
  group, 
  levels = c(50, 75, 95), 
  scaler = 10, 
  smallSamp = FALSE 
)
```

Arguments

data  data.frame object containing columns of isotopic values and grouping variables
x  character giving the column name of the x coordinates
y  character giving the column name of the y coordinates
h  character describing the bandwidth estimator method. Default = "ref". See Details for more information.

hval  numeric vector of length 2 describing the bandwidth in x and y directions. Default = NULL

hval  numeric vector of length 2 describing the bandwidth in x and y directions. Default = NULL

group  character giving the column name of the grouping variable (i.e. species)
levels  Numeric vector of desired percent levels (e.g. c(10, 50, 90). Should not be less than 1 or greater than 99)

scaler  numeric value to expand the min/max x and y values. This assists with error given smaller sample sizes. Default value = 10

smallSamp  logical value indicating whether to override minimum number of samples. Currently 10 samples are required.

Details

For the h argument there are 8 different bandwidth estimation options ("hns", "hpi", "hscv", "hlscv", "hbcv", "hnm", "hucv", "ref"). "ref" = The default MASS::kde2d bandwidth method. The remaining options are obtained from the 'ks' package with the default method being "hpi". For all ks package methods, the default values are accepted and only the x and y values are passed to the
bivariate bandwidth estimating functions. For all bandwidth estimation methods, reducing the data
to an individual group will provide the same bandwidths as used during rKIN estimation.

* hpi - Default Plug-in bandwidth selector using ks::Hpi function. Values can be obtained using
  bw_hpi(). * hns - Normal scale bandwidth using ks::Hns function. Values can be obtained using
  bw_hns(). * hscv - Smoothed cross-validation bandwidth selector. Values can be obtained using
  bw_hscv(). * hlscv - Least-squares cross-validation bandwidth matrix selector for multivariate
data. Values can be obtained using bw_hlscv(). * hbcv - Biased cross-validation bandwidth matrix
selector for bivariate data. Values can be obtained using bw_hbcv(). * hnm - Normal mixture
bandwidth. Values can be obtained using bw_hnm(). * hucv - Least-squares cross-validation band-
width matrix selector for multivariate data. Values can be obtained using bw_hucv(). * ref - Uses
MASS::bandwidth.nrd for both x and y separately, dividing values by 4 to match the scale of ks
methods. Values can be obtained using bw_ref(). See MASS:kde2d() for details (i.e. the function
divides the values by 4).

Value

A class rKIN object containing a list of sf data frames, each list item representing the grouping
variable.

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples

```r
library(rKIN)
data("rodents")
#estimate niche overlap between 2 species using kernel UD
test.kin <- estKIN(data=rodents, x="Ave_C", y="Ave_N", group="Species",
  levels=c(50, 75, 95), scaler=2)
#determine polygon overlap for all polygons
plotKIN(test.kin, scaler=2, title="Kernel Estimates", xlab="Ave_C", ylab="Ave_N")
```

Description

Calculates the Minimum Convex Polygon for isotopic values at multiple confidence levels. Returns
a list of sf data frames, each list item representing the grouping variable (i.e. species).

Usage

```r
estMCP(data, x, y, group, levels = c(50, 75, 95), smallSamp = FALSE)
```
Arguments

data  data.frame object containing columns of isotopic values and grouping variables
x       character giving the column name of the x coordinates
y       character giving the column name of the y coordinates
group   character giving the column name of the grouping variable (i.e. species)
levels  Numeric vector of desired percent levels (e.g. c(10, 50, 90). Should not be less than 1 or greater than 100)
smallSamp  logical value indicating whether to override minimum number of samples. Currently 10 samples are required.

Value

A list of sf data frames, each list item representing the grouping variable.

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples

library(rKIN)
data("rodents")
# estimate niche overlap between 2 species using minimum convex polygons
test.mcp <- estMCP(data=rodents, x="Ave_C", y="Ave_N", group="Species",
                   levels=c(50, 75, 95))
# determine polygon overlap for all polygons
plotKIN(test.mcp, scaler=2, title="Minimum Convex Hull Estimates", xlab="Ave_C", ylab="Ave_N")

Description

This is a helper function that creates a sequence of points on a circle of radius \( r \) as a resolution determined by \( n \). This function was directly borrowed from SIBER package (Intended for generating various SIBER ellipses). It is not intended for direct calling. NB not an exported function.

Usage

genCircle(n = 100, r)

Arguments

n  the number of points to create around the circle. Defaults to 100.
r  the radius of the circle to create.
getArea

Value

A 2 x n matrix of x and y coordinates of points on a circle.

Description

Extracts the polygon area for an rKIN object for each group and level.

Usage

getArea(estObj)

Arguments

estObj List object created from estKIN, estMCP or estEllipse functions

Value

A data.frame() of polygon areas.

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples

data(“rodents”)  #estimate niche overlap between 2 species using kernel UD
test.kin<- estKIN(data=rodents, x=”Ave_C”, y=”Ave_N”, group=”Species”,
levels=c(50, 75, 95), scaler=2)  #determine polygon sizes/areas
kin.area<- getArea(test.kin)
getColors

Create a list of colors for plotKIN function

Description

The list of colors were obtained from Colorbrewer2.org using single hue. This is run within the function plotKIN()

Usage

getColors(groups, levels, colors = NULL)

Arguments

- groups: The number of groups within grouping variable (i.e. species)
- levels: The number of confidence intervals provided by the user
- colors: Character vector of hex codes representing desired colors

Value

A character vector of RGB colors

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

getKernelThreshold

Miscellaneous functions to complete kernel 2D estimates: Get contour threshold values

Description

Obtains the quantile threshold levels for a vector of probabilities from a kernel density estimate.

Usage

getKernelThreshold(x, levels = c(50, 75, 95))

Arguments

- x: Numeric vector of probabilities from a kernel density estimate
- levels: Numeric vector of desired percent levels (e.g. c(10, 50, 90). Should not be less than 1 or greater than 99)
**Value**

A list of threshold values for each percent.

**Author(s)**

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

---

**plotKIN**

*Plotting function for rKIN polygons*

---

**Description**

Using ggplot2 methods, simultaneously plot all of the groups and levels of niche space.

**Usage**

```r
plotKIN(
  estObj,
  scaler = 1,
  alpha = 0.3,
  title = "",
  xlab = "x",
  ylab = "y",
  xmin = NULL,
  xmax = NULL,
  ymin = NULL,
  ymax = NULL,
  colors = NULL
)
```

**Arguments**

- `estObj` list object created from estKIN, estMCP or estEllipse functions
- `scaler` numeric value indicating number of isotopic units to expand the x and y axes of the plot. Default is 1.
- `alpha` numeric value between 0 and 1, representing the amount of transparency of each polygon. 0 is transparent, 1 is opaque.
- `title` character string for a plot title.
- `xlab` character or expression string for the x-axis label.
- `ylab` character or expression string for the y-axis label.
- `xmin` default is NULL, numeric value of user specified minimum x axis value
- `xmax` default is NULL, numeric value of user specified maximum x axis value
- `ymin` default is NULL, numeric value of user specified minimum y axis value
- `ymax` default is NULL, numeric value of user specified maximum y axis value
- `colors` default is NULL, character vector of hex codes representing colors for plot
Value

A plot of all groups and levels.

Author(s)

Shannon E. Albeke, Wyoming Geographic Information Science Center, University of Wyoming

Examples

```r
library(rKIN)
data("rodents")
#estimate niche overlap between 2 species using kernel UD
test.kin<- estKIN(data=rodents, x="Ave_C", y="Ave_N", group="Species",
levels=c(50, 75, 95), scaler=2)
#determine polygon overlap for all polygons
plotKIN(test.kin, scaler = 1, title = "Kernel Estimates",
   xlab = expression({delta}^13*C~ (\u2030/\u2030/Var_1/Var_2)),
   ylab = expression({delta}^15*N~ (\u2030/\u2030/Var_1/Var_2)))
```

rKIN

rKIN: A package for computing isotopic niche space

Description

The rKIN This package applies methods used to estimate animal home range, but instead of geospatial coordinates, we use isotopic coordinates. The estimation methods include: 1) 2-dimensional bivariate normal kernel utilization density estimator with multiple bandwidth estimation methods, 2) bivariate normal ellipse estimator, and 3) minimum convex polygon estimator, all applied to stable isotope data. Additionally, functions to determine niche area, polygon overlap between groups and levels (confidence contours) and plotting capabilities.

rKIN functions

The rKIN functions: estKIN, estEllipse, estMCP, plot.kin, getArea, calcOverlap

rodents

Isotopic data from rodent blood samples.

Description

A dataset containing the individual Species, Habitat sampled, and Percent delta C and N.

Usage

rodents
Format

A data frame with 530 rows and 4 variables:

Species  Generic species used as a grouping variable
Habitat  Habitat in which the individual was captured
Ave_C   Averaged delta 13C present within the blood sample
Ave_N   Averaged delta 15N present within the blood sample ...

Source

http://www.uwyo.edu/zoology/people/bendavid.html
Index

* datasets
  rodents, 16

bw_hbcv, 2
bw_hlscv, 3
bw_hnm, 4
bw_hns, 4
bw_hpi, 5
bw_hscv, 6
bw_hucv, 6
bw_ref, 7

calcOverlap, 8
createSPDF, 8

estEllipse, 9
estKIN, 10
estMCP, 11

genCircle, 12
getArea, 13
getColors, 14
getKernelThreshold, 14

plotKIN, 15

rKIN, 16
rodents, 16