Package 'mkde'

May 8, 2026

Type Package

Title 2D and 3D Movement-Based Kernel Density Estimates (MKDEs)

Version 0.4

Date 2024-11-26

Description Provides functions to compute and visualize movement-based kernel density estimates (MKDEs) for animal utilization distributions in 2 or 3 spatial dimensions.

LazyLoad yes

Depends R (>= 3.5.0)

Imports Rcpp (>= 0.9.6), sf, terra, stars

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-12-11 21:30:02 UTC

Author Jeff A. Tracey [aut],
James Sheppard [aut],
Jun Zhu [aut],
Amit Chourasia [aut],
Glenn Lockwood [aut],
Matthew Wang [aut],
Robert N. Fisher [aut],
Robert Sinkovits [aut, cre]

Maintainer Robert Sinkovits <`sinkovit@sdsc.edu`>

Contents

`mkde-package` *Movement-based kernel density estimates (MKDEs) in 2 or 3 spatial dimensions.*

Description

The mkde package enables animal space use to be estimated in three dimensions (3D) using data collected from biotelemetry tracking devices. This package addresses a recognized need in modeling animal space use (Belant et al. 2012) wherein researchers have been limited by the lack of 3D home range estimators. Animal space use can be characterized by the (x, y) spatial dimensions as well as a third z-dimension representing altitude, elevation, or depth for flying, terrestrial, or aquatic species, respectively. Although many biotelemetry devices record 3D location data with x, y, and z coordinates from tracked animals, the third z coordinate is typically not integrated into studies of animal spatial use. The mkde package enables users to visually explore the 3D MKDE volumes of animals to more intuitively understand how they are spatially related to the environmental covariates and bounding layers within their ranges, such as bathymetry or topography.

The mkde package builds on previous work on the Brownian bridge approach for estimating animal utilization distributions (Horne et al. 2007). This method, in contrast to location-based KDEs, integrates kernels over time along a movement path interpolated between observed locations. Benhamou distinguished location-based kernel density estimators (LKDE) from movement-based kernel density estimators (MKDE), which includes Brownian bridge and biased random walk models. MKDEs account for time between consecutively observations in the estimator, do not requiring independent samples from the UD, and thus more realistically represent the space used by an animal.

The user inputs animal location data typically obtained by a Global Positioning System (GPS) or Very High Frequency (VHF) device in which each observation includes an x-coordinate, a y-coordinate,a z-coordinate, and time. The observed locations are assumed to be subject to observation error and are normal random variables. The observation error variances are either provided by the manufacturers of the telemetry equipment or estimated from field trials, e.g., Hansen and Riggs (2008). Often, an animal's movement is limited in the z-dimension. For example, avian species are generally bounded below by the earth's surface, whereas marine animals are bounded below by the sea floor and above by the water's surface. Package functions allow the mkde user to bound the density in the z-dimension by a(x,y) and b(x,y) with a constant or a 2D raster.

The mkde package provides a 2.5D approach for computing home range area that essentially uses a 2D MKDE draped over a 2D elevation raster. The bias is corrected by calculating and summing the surface area of each cell of the elevation raster that falls within a desired probability contour of the 2D MKDE. An algorithm developed by Jenness (2004, 2014) is used to compute the surface area of each raster cell. This method uses the cell center coordinates and elevations of the focal cell and its eight neighboring cells to construct eight triangular facets within the focal cell. The area of each facet is calculated using Heron's formula and then summed to obtain the surface area for the focal cell.

Numerous functions are provided to write output files in various formats (VTK, XDMF, ASCII) for use in other GIS and 3D Visualization applications.

Details

Package:	mkde
Type:	Package
Version:	1.0
Date:	2011-08-23
License:	GPL-2
LazyLoad:	yes

Author(s)

Jeff A. Tracey (US Geological Survey, San Diego Field Station, Western Ecological Research Center)
James Sheppard (San Diego Zoo Institute for Conservation Research)

Jun Zhu (Department of Statistics and Department of Entomology, University of Wisconsin – Madison)
Robert Sinkovits (San Diego Supercomputer Center)
Amit Chourasia (San Diego Supercomputer Center)
Glenn Lockwood (San Diego Supercomputer Center)
Maintainer: Jeff A. Tracey <jatracey@usgs.gov, jeff.a.tracey@gmail.com>

References

Tracey, J. A., Sheppard, J., Zhu, J., Wei, F., Swaisgood, R. R. and Fisher, R. N. (2014) Movement-Based Estimation and Visualization of Space Use in 3D for Wildlife Ecology and Conservation. PLoS ONE 9(7): e101205. doi: 10.1371/journal.pone.0101205
Tracy, J. A., Sheppard, J. Lockwood, G., Chourasia, A., Tatineni, M., Fisher, R. N., and Sinkovits, R. (2014) Efficient 3D Movement-Based Kernel Density Estimator and Application to Wildlife Ecology. XSEDE 14 Conference Proceedings, Article No. 14. doi: 10.1145/2616498.2616522
Belant, J. L., Millspaugh, J. J., Martin, J. A. & Gitzen, R. A. (2012). Multi-dimensional space use: the final frontier. Frontiers in Ecology & Environment 10, 11-12.
Benhamou, S. (2011). Dynamic Approach to Space and Habitat Use Based on biased random bridges. PLoS ONE 6.
Hansen, M. C., & Riggs, R. A. (2008). Accuracy, precision, and observation rates of global positioning system telemetry collars. The Journal of Wildlife Management, 72(2), 518-526.
Horne, J. S., Garton, E. O., Krone, S. M., Lewis, J. S. (2007). Analyzing animal movements using Brownian bridges. Ecology 88, 2354-2363.
Jenness J. S. (2004) Calculating landscape surface area from digital elevation models. Wildlife Society Bulletin 32: 829-839.
Jenness, J. S. (2014) Calculating landscape surface area from unprojected digital elevation models. In preparation.

`computeAreaRaster` *Calculate cell areas from elevation raster.*

Description

A lower-level function for calculating a cell area matrix (2D array) from an elevation matrix. The area is based on the surface area of the terrain in the cell. The area matrix is then used in calculating areas of 2.5D MKDES.

Usage

```
computeAreaRaster(RelevMatrix, RcellSize)
```

Arguments

`RelevMatrix`	A 2D array with elevation values.
`RcellSize`	Size of the cells. It is assumed to be the same in the x and y dimensions.

Details

This is a wrapper function for C++ function that calculates the surface area of each raster cell given the cell elevations. It is not intended to be used directly; instead, the user should call `initializeAreaRaster` on an MKDE object.

Value

A 2D matrix of cell surface areas.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

References

Jenness J.S. (2004) Calculating landscape surface area from digital elevation models. Wildlife Society Bulletin 32: 829-839.
Jenness, J.S. (2014) Calculating landscape surface area from unprojected digital elevation models. In preparation.

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem))
area.rast <- computeAreaRaster(as.matrix(condordem, wide=TRUE), cell.sz)
```

`computeContourValues` *Find thresholds for contour intervals*

Description

Find the cell or voxel probabilities that correspond to user-specified probability contours

Usage

```
computeContourValues(mkde.obj, prob)
```

Arguments

`mkde.obj`	An MKDE object with density initialized
`prob`	Probabilities (i.e. proportions) for desired contours of the MKDE

Details

This function computes threshold cell or voxel probability values corresponding to contours for specified proportions of the utilization distribution. Note that the arugment `prob` specifies the cumulative probability of the cells or voxels within the contour corresponding to the cell or voxel threshold probability. The cell or voxel threshold probabilities may be orders of magnitude smaller than the cumulative probabilities provided in the `prob` argument.

Value

A data frame with the probabilities given in the prob argument and corresponding thresholds in the MKDE

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
data(condor)

# Find min/max coordinates and add buffer
xmax = max(condor$x) + 1000
xmin = min(condor$x) - 1000
ymax = max(condor$y) + 1000
ymin = min(condor$y) - 1000

# Calculate grid dimensions
xrange <- xmax - xmin
yrange <- ymax - ymin
cell.sz = 150
nx <- as.integer(xrange/cell.sz)
ny <- as.integer(yrange/cell.sz)

mv.dat <- initializeMovementData(condor$t, condor$x, condor$y, sig2obs=25.0, t.max=185.0)
mkde.obj <- initializeMKDE2D(xmin, cell.sz, nx, ymin, cell.sz, ny)
dens.res <- initializeDensity(mkde.obj, mv.dat)

mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat
my.quantiles <- c(0.95, 0.75, 0.50)
res <- computeContourValues(mkde.obj, my.quantiles)
print(res)
```

`computeSizeMKDE` *Compute the area or volume of an MKDE.*

Description

For 2D MKDEs, this function computes the area. For a 2.5D MKDE it computes the area based on the cell areas computed from an elevation raster. For a 3D MKDE, it computes the volume.

Usage

```
computeSizeMKDE(mkde.obj, prob)
```

Arguments

`mkde.obj`	MKDE list object with density estimate calculated.
`prob`	Probabilities for the desired contours.

Details

For a 2D or 2.5D MKDE list object, areas within the contours specified by quant are calculated. For a 3D MKDE, the volumes within the contours are calculated.

Value

A vector with the areas or volumes is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
data(condor)

condor <- condor[1:20,] # simply to make example run more quickly
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
```

```
nx <- ncol(condordem120)
ny <- nrow(condordem120)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem120), na.rm=TRUE), cell.sz, 25)

# note: we use a raster coarse integration time step so the
# example runs faster
dens.res <- initializeDensity(mkde.obj, mv.dat, integration.step=10.0)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat

my.quantiles <- c(0.95, 0.75, 0.50)
res <- computeContourValues(mkde.obj, my.quantiles)
res$volume <- computeSizeMKDE(mkde.obj, my.quantiles)
print(res)
```

`condor` *California condor locations*

Description

A data frame containing California condor location data

Usage

`data(condor)`

Format

A data frame with 421 observations on the following 4 variables.

`time`	Elapsed time in minutes
`x`	x-coordinate (UTM easting in meters)
`y`	y-coordinate (UTM northing in meters)
`z`	z-coordinate (height above sea level in meters)

Details

GPS location data acquired from a wild California condor (Gymnogyps californianus) tracked by San Diego Zoo Global around its reintroduction site in Baja California, Mexico during November 2012 (Sheppard et al. 2013).

Source

James Sheppard, San Diego Zoo Institute for Conservation Research

References

Sheppard, J.K., Walenski, M., Wallace, M.P., Velazco, J.J.V., Porras, C., & Swaisgood, R.R. (2013). Hierarchical dominance structure in reintroduced California condors: correlates, consequences, and dynamics. Behavioral Ecology and Sociobiology. 67: 1-12.

Examples

```
data(condor)
head(condor, 30)
```

`condordem` *A RasterLayer containing a digital elevation model (DEM) for the condor data*

Description

Digital elevation model of the terrain within the condor's range with a cell resolution of 30 meters (ASTER GDEM version 2, Tachikawa et. al. 2011).

Format

A raster package RasterLayer object with 749 rows, 623 columns, and a cell size of 30 by 30 meters. Each cell contains an elevation in meters.

Source

URL: http://asterweb.jpl.nasa.gov/gdem.asp

References

Tachikawa, T., Hato, M., Kaku, M. & Iwasaki, A. (2011). The characteristics of ASTER GDEM version 2, IGARSS.

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)
plot(condordem)
```

`condordem120` *A RasterLayer containing a digital elevation model (DEM) for the condor data*

Description

Digital elevation model of the terrain within the condor's range with a cell resolution of 120 meters (ASTER GDEM version 2, Tachikawa et. al. 2011).

Format

A raster package RasterLayer object with 188 rows, 156 columns, and a cell size of 120 by 120 meters. Each cell contains an elevation in meters.

Source

URL: http://asterweb.jpl.nasa.gov/gdem.asp

References

Tachikawa, T., Hato, M., Kaku, M. & Iwasaki, A. (2011). The characteristics of ASTER GDEM version 2, IGARSS.

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
plot(condordem120)
```

`deselectLocationsOutsideBounds` *Flag 3D locations with out-of-bounds z-coordinates*

Description

For a 3D MKDE object and 3D location data, the z-coordinates in the location data are checked to make sure they are within the lower and upper bounds specified in the MKDE list object.

Usage

```
deselectLocationsOutsideBounds(move.dat, mkde.obj)
```

Arguments

`move.dat`	A move data object created with `initializeMovementData`
`mkde.obj`	An MKDE object created with `initialize2DMKDE` or `initialize3DMKDE`

Details

If a 2D or 2.5D MDKE object is passed as an argument, no change is made to the movement data list object. If a 3D MKDE list object is passed as an argument, the z-coordinates in the movement data are checked to determine if they are in range. If they are not, the corresponding value in move.dat$use.obs is set to FALSE. Note that this function is not called within `initialzeDensity`. If you want to exclude locations because they are outside of the allowed range in the z-dimension, this function must be used before computing the density.

Value

An updated move data list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)

cell.sz <- mean(res(condordem))
ext <- ext(condordem)
nx <- ncol(condordem)
ny <- nrow(condordem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem), na.rm=TRUE), 30.0, 100)
mkde.obj <- setMinimumZfromRaster(mkde.obj, condordem)
mv.dat <- deselectLocationsOutsideBounds(mv.dat, mkde.obj)
```

`deselectNonMovementSteps` *Flag non-movements so they are excluded from MKDE estimation*

Description

This function deselects move steps where the probability that the initial and final location are the same location is greater than or equal to a user-defined threshold probability

Usage

```
deselectNonMovementSteps(move.dat, p)
```

Arguments

`move.dat`	A move data object created with `initializeMovementData`
`p`	The threshold probability

Details

If the probability that the initial and final location are the same location is greater than or equal to a user-defined threshold probability, the corresponding value in move.dat$use.obs is set to FALSE. Note that this function is not called within `initialzeDensity`. If you want to exclude locations because they the initial location in a non-movement step, this function must be used before computing the density.

Value

An updated move data list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

mv.dat <- deselectNonMovementSteps(mv.dat, 0.05)
```

`dugong` *Dugong locations*

Description

A data frame containing dugong location data

Usage

```
data(dugong)
```

Format

A data frame with 426 observations on the following 4 variables.

`time` Elapsed time in minutes

`x` x-coordinate (UTM easting in meters)

`y` y-coordinate (UTM northing in meters)

`z` z-coordinate (Depth in meters)

Details

GPS location data acquired from a wild dugong (Dugong dugon) tracked by James Cook University in Hervey Bay, Australia during August 2004 (Sheppard et. al. 2010)

Source

James Sheppard, San Diego Zoo Institute for Conservation Research

References

Sheppard, J. Marsh, H., Jones, R.E., & Lawler, I.R. (2010). Dugong habitat use in relation to seagrass nutrients, tides, and diel cycles. Marine Mammal Science 26, 855-879.

Examples

```
data(dugong)
head(dugong, 30)
```

`dugongdem` *A RasterLayer containing bathymetry for the dugong data*

Description

A 10-meter resolution bathymetric surface of the core seagrass habitat in depth below the mean sea level within the dugong's range (Sheppard et. al. 2007).

Format

A raster package RasterLayer object with 220 rows, 485 columns, and a cell size of 10 by 10 meters. Each cell contains a depth at low tide in meters.

Source

James Sheppard, San Diego Zoo Institute for Conservation Research

References

Sheppard, J. K., Lawler, I. R., & Marsh, H. (2007). Seagrass as pasture for seacows: Landscape-level dugong habitat evaluation. Estuarine, Coastal and Shelf Science, 71(1), 117-132.

Examples

```
library(terra)
fpath <- system.file("extdata", "dugongdem.RDS", package="mkde")
dugongdem <- terra::readRDS(fpath)
plot(dugongdem)
```

`estVarMKDE` *Estimate move variances for 3D MKDE.*

Description

Estimates the move variance for the (x, y) and z dimensions and altitude for a Brownian bridge MKDE.

Usage

```
estVarMKDE(move.dat)
```

Arguments

`move.dat`	A move data object created with `initializeMovementData`

Details

Computes maximum-likelihood estimates for move variances. If the MKDE is 2D or 2.5D, only the variance for the xy-dimension is estimated. If the MKDE is 3D, a separate move variance for the z-dimension is also estimated.

Value

A move data object updated with move variances

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

mv.dat <- estVarMKDE(mv.dat)
cat(paste("Move variance in xy-dimensions = ",
mv.dat$sig2xy[1], "\n", sep=""))
cat(paste("Move variance in z-dimension = ",
mv.dat$sig2z[1], "\n", sep=""))
```

`initializeAreaRaster` *Initialize an area raster for a 2.5D MKDE.*

Description

Initialize the surface area for a 2.5D MKDE.

Usage

```
initializeAreaRaster(mkde.obj)
```

Arguments

`mkde.obj`	An MKDE object created using `initialize2DMKDE`

Details

After creating the MKDE object and setting the lower bounds in the z-dimension using `setMinimumZfromRaster`, this function computes the surface area of each raster cell and sets the dimension of the MKDE object to 2.5.

Value

Returns a 2.5 MKDE object with an initialized area raster.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
cell.sz <- mean(res(pandadem))
ext <- ext(pandadem)
nx <- ncol(pandadem)
ny <- nrow(pandadem)
mkde.obj <- initializeMKDE2D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz, ny)

mkde.obj <- setMinimumZfromRaster(mkde.obj, pandadem)
mkde.obj <- initializeAreaRaster(mkde.obj)
```

`initializeDensity` *Calculate raster of density values for MKDE.*

Description

Given a movement data object (list) and MKDE object (list), estimate the movement variance parameters and update them in the movement data object and then compute the density for a 2D, 2.5D, or 3D MKDE.

Usage

```
initializeDensity(mkde.obj, move.dat, integration.step, d.thresh=1e-25)
```

Arguments

`mkde.obj`	An MKDE object created with `initialize2DMKDE` or `initialize3DMKDE`
`move.dat`	A move data object created with `initializeMovementData`
`integration.step`	A time step used for numerical integration over the movement trajectory
`d.thresh`	The value of the kernel below which its contibrution to the overal density is considered negligible

Details

This function computes the density for a 2D, 2.5D, or 3D MKDE. If the move variance parameters have not been estimated, they will be prior to computing the density. The integration time step should be much less than the maximum time step allowed between observed animal locations. For a 2.5D MKDE, if the area has been calculated with `initializeAreaRaster`, then the cell probabilities are weighted by the proportion of the total surface area represented by the cell.

Value

A list is returned with two elements: `move.dat` and `mkde.obj`. The first is an updated move data object and the second is an updated MKDE object.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Robert Sinkovits, PhD
San Diego Supercomputer Center
`<sinkovit@sdsc.edu >`
Glenn Lockwood, PhD
San Diego Supercomputer Center
`<glock@sdsc.edu>`

Examples

```
library(terra)
data(condor)
condor <- condor[1:20,] # simply to make example run more quickly
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
nx <- ncol(condordem120)
ny <- nrow(condordem120)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem120), na.rm=TRUE), cell.sz, 25)

# note: we use a raster coarse integration time step so the
# example runs faster
dens.res <- initializeDensity(mkde.obj, mv.dat, integration.step=10.0)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat
```

`initializeMKDE2D` *Set up a 2D MKDE object.*

Description

Define the spatial extent and resolution of a 2D MKDE and create an 2D MKDE list object for use in other functions in the package.

Usage

```
initializeMKDE2D(xLL, xCellSize, nX, yLL, yCellSize, nY)
```

Arguments

`xLL`	Lower bounds of the grid in the x-dimension
`xCellSize`	Cell size in the x-dimension
`nX`	Number of cells in the x-dimension
`yLL`	Lower bounds of the grid in the y-dimension
`yCellSize`	Cell size in the y-dimension
`nY`	Number of cells in the y-dimension

Details

It is strongly recommended that the same value is used for `xCellSize` and `yCellSize`. The grid should be defined so that it covers the area that the animal used, plus a sufficient buffer so that the density is negligable beyond the grid.

Value

A list representing an MKDE object is returned with the following elements:

`dimension`	The dimension of the MKDE; that is, 2.
`x`	A grid of points along the x-axis where the cell centers occur.
`y`	A grid of points along the y-axis where the cell centers occur.
`z`	A grid of points along the z-axis where the cell centers occur. For a 2D MKDE z = NA.
`z.min`	A 2D array representing the lower bounds of space in the z-dimension at each x and y coordinate. Defaults to -Inf.
`z.max`	A 2D array representing the upper bounds of space in the z-dimension at each x and y coordinate. Defaults to Inf.
`nx`	Number of cells in the x-dimension.
`ny`	Number of cells in the y-dimension.
`nz`	Number of cells in the z-dimension. For a 2D MKDE nz = 1.
`d`	A 2D array with dimensions (nx, ny) that stores the density. The elements are initialized to NA.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
cell.sz <- mean(res(pandadem))
ext <- ext(pandadem)
nx <- ncol(pandadem)
ny <- nrow(pandadem)
mkde.obj <- initializeMKDE2D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz, ny)
```

`initializeMKDE3D` *Set up a 3D MKDE object.*

Description

Define the spatial extent and resolution of a 3D MKDE and create an 3D MKDE list object for use in other functions in the package.

Usage

```
initializeMKDE3D(xLL, xCellSize, nX, yLL, yCellSize, nY,
zLL, zCellSize, nZ)
```

Arguments

`xLL`	Lower bounds of the grid in the x-dimension
`xCellSize`	Cell size in the x-dimension
`nX`	Number of cells in the x-dimension
`yLL`	Lower bounds of the grid in the y-dimension
`yCellSize`	Cell size in the y-dimension
`nY`	Number of cells in the y-dimension
`zLL`	Lower bounds of the grid in the z-dimension
`zCellSize`	Cell size in the z-dimension
`nZ`	Number of cells in the z-dimension

Details

It is strongly recommended that the same value is used for `xCellSize` and `yCellSize`. The grid should be defined so that it covers the volume that the animal used, plus a sufficient buffer so that the density is negligable beyond the grid.

Value

A list representing an MKDE object is returned with the following elements:

`dimension`	The dimension of the MKDE; that is, 3.
`x`	A grid of points along the x-axis where the voxel centers occur.
`y`	A grid of points along the y-axis where the voxel centers occur.
`z`	A grid of points along the z-axis where the voxel centers occur.
`z.min`	A 2D array representing the lower bounds of space in the z-dimension at each x and y coordinate. Defaults to a constant value based on the arguments zLL, zCellSize,and nZ.
`z.max`	A 2D array representing the upper bounds of space in the z-dimension at each x and y coordinate. Defaults to a constant value based on the arguments zLL, zCellSize,and nZ.
`nx`	Number of cells in the x-dimension.
`ny`	Number of cells in the y-dimension.
`nz`	Number of cells in the z-dimension.
`d`	A 3D array with dimensions (nx, ny, nz) that stores the density. The elements are initialized toNA.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem))
ext <- ext(condordem)
nx <- ncol(condordem)
ny <- nrow(condordem)
zmin <- min(values(condordem), na.rm=TRUE)
nz <- 30
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, zmin, cell.sz, nz)
```

`initializeMovementData` *Initialize a movement data list*

Description

This function sets up the movement data for use in other functions in the package.

Usage

```
initializeMovementData(t.obs, x.obs, y.obs, z.obs=NULL, sig2obs=0.0,
sig2obs.z=NA, t.max=max(diff(t.obs), na.rm=TRUE))
```

Arguments

`t.obs`	A vector of times at which the animal locations were observed
`x.obs`	A vector of x-coordinates at which the animal locations were observed
`y.obs`	A vector of y-coordinates at which the animal locations were observed
`z.obs`	A vector of z-coordinates at which the animal locations were observed
`sig2obs`	Location error variance in the xy-dimensions
`sig2obs.z`	Location error variance in the z-dimension
`t.max`	The maximum time allowed between locations if the movement step is to be used in computing the density

Details

If only 2D or 2.5D MKDEs are to be calculated, then z.obs and sig2obs.z do not have to be provided.

Value

A move data object, in the form of a list, is returned.

`dimension`	The spatial dimension of the movement data. If z-coordinates are passed, the dimension will be 3; otherwise, the dimension will be 2.
`t.obs`	Times of the animal locations.
`x.obs`	x-coordinates of the animal locations.
`y.obs`	y-coordinates of the animal locations.
`z.obs`	z-coordinates of the animal locations.
`a.obs`	Altitude of the animal; that is, its z-coordinate minus the lower bound on the z-coordinate at the corresponding xy-coordinates of the animal location.
`use.obs`	A logical array that indicates whether each location should be used in the MKDE calculations. By default these values are set to TRUE.
`t.max`	The maximum time allowed between locations if the movement step is to be used in computing the density

`sig2xy`	A vector of movemenet variance paramters for the xy-dimensions, with NAs as placeholders. The functions estVarMKDE is provided to estimate these parameters, or they can be set manually to allow for different movement variances for each move step.
`sig2z`	A vector of movemenet variance paramters for the z-dimension, with NAs as placeholders. The functions estVarMKDE is provided to estimate these parameters, or they can be set manually to allow for different movement variances for each move step.
`sig2obs`	A vector of variances for the location observation error in the xy-dimensions. If only one value is provided for the variance parameters, the value is repeated to form a vector with one element for each location. Otherwise, a vector of location error variances can be passed to allow for different errors for each observed location.
`sig2obs.z`	A vector of variances for the location observation error in the z-dimension. If only one value is provided for the variance parameters, the value is repeated to form a vector with one element for each location. Otherwise, a vector of location error variances can be passed to allow for different errors for each observed location.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x,
condor$y, z.obs=condor$z, 65.0, 25.0, sig2obs.z=81.0)
```

`mkde2Dgrid` *Movement-based kernel density estimate (MKDE) in 2D using Rcpp*

Description

Provides a function for 2-dimensional MKDEs.

Usage

```
mkde2Dgrid(mkde.obj, move.dat, t.step, d.thresh)
```

Arguments

`mkde.obj`	A 2D or 2.5D MKDE object
`move.dat`	A move data object
`t.step`	An integration time step
`d.thresh`	A kernel density threshold

Details

This is lower-level function that call the C++ function. for estimating the movement-based density in 2D. In practice, users should call `initializeDensity`.
The argument d.thresh is a univariate probability density beyond which the kernel contribution to the overall MKDE is assumed to be negligible. Usually this is set at a very small value and is used to prevent calculations from being performed in cells to which the kernel makes a negligible contribution.

Value

An array whose elements are the estimated utilization probabilities for each cell.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Robert Sinkovits, PhD
San Diego Supercomputer Center
`<sinkovit@sdsc.edu >`
Glenn Lockwood, PhD
San Diego Supercomputer Center
`<glock@sdsc.edu>`
Jun Zhu, PhD
University of Wisconsin-Madison
`<jzhu@stat.wisc.edu>`

Examples

```
data(panda)

# Find min/max coordinates and add buffer
xmax = max(panda$x) + 100
xmin = min(panda$x) - 100
ymax = max(panda$y) + 100
ymin = min(panda$y) - 100

# Calculate grid dimensions
xrange <- xmax - xmin
```

```
yrange <- ymax - ymin
cell.sz = 30
nx <- as.integer(xrange/cell.sz)
ny <- as.integer(yrange/cell.sz)

mv.dat <- initializeMovementData(panda$time, panda$x, panda$y, t.max=185.0, sig2obs=25.0)
if (all(is.na(mv.dat$sig2xy))) {
  mv.dat <- estVarMKDE(mv.dat)
}

mkde.obj <- initializeMKDE2D(xmin, cell.sz, nx, ymin, cell.sz, ny)
res <- mkde2Dgrid(mkde.obj, mv.dat, 10.0, 1e-20)
```

`mkde2Dinteraction` *Probability of 2D spatial-temporal interaction.*

Description

Probability of 2D spatial-temporal interaction.

Usage

```
mkde2Dinteraction(mkde.obj, move.dat0, move.dat1, t.step, d.thresh)
```

Arguments

`mkde.obj`	An MKDE object created with `initialize2DMKDE`
`move.dat0`	A move data object for the first individual created with `initializeMovementData`
`move.dat1`	A move data object for the second individual created with `initializeMovementData`
`t.step`	A time step used for numerical integration over the movement trajectory
`d.thresh`	The value of the kernel below which its contibrution to the overal density is considered negligible

Details

This function assumes that the two individual animals were observed at the same times. The cell values returned in the mkde.obj can be summed to obtain a global measure of spatio-temporal interaction.

Value

Returns a list with the following elements:

`mkde.obj`	An updated MKDE object containing the cell-level Bhattacharyya coefficients
`move.dat0`	A move data object for the first individuals with updated variance parameters
`move.dat1`	A move data object for the second individuals with updated variance parameters

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Jun Zhu, PhD
University of Wisconsin-Madison
`<jzhu@stat.wisc.edu>`

Examples

```
library(terra)
data(panda)
mv.dat0 <- initializeMovementData(panda$time, panda$x, panda$y,
sig2obs=25.0, t.max=185.0)

n <- nrow(panda)
v <- 20.0 # increase from 0 to increase difference
mv.dat1 <- initializeMovementData(panda$time, panda$x+rnorm(n, 0, v),
panda$y+rnorm(n, 0, v), sig2obs=25.0, t.max=185.0)

fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
cell.sz <- mean(res(pandadem))
ext <- ext(pandadem)
nx <- ncol(pandadem)
ny <- nrow(pandadem)
mkde.obj <- initializeMKDE2D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny)

res <- mkde2Dinteraction(mkde.obj, mv.dat0, mv.dat1, 10.0, 1e-20)
mkde.obj <- res$mkde.obj
mv.dat0 <- res$move.dat0
mv.dat1 <- res$move.dat1
```

`mkde3Dgrid` *Movement-based kernel density estimate (MKDE) in 3D using Rcpp*

Description

Provides a function for 3-dimensional MKDEs.

Usage

```
mkde3Dgrid(mkde.obj, move.dat, t.step, d.thresh)
```

Arguments

`mkde.obj`	A 3D MKDE object
`move.dat`	A move data object
`t.step`	An integration time step
`d.thresh`	A kernel density threshold

Details

This is lower-level function that call the C++ function. for estimating the movement-based density in 3D. In practice, users should call `initializeDensity`.
The argument `d.thresh` is a univariate probability density beyond which the kernel contribution to the overall MKDE is assumed to be negligible. Usually this is set at a very small value and is used to prevent calculations from being performed in cells to which the kernel makes a negligible contribution.

Value

An array whose elements are the estimated utilization probabilities for each voxel.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Robert Sinkovits, PhD
San Diego Supercomputer Center
`<sinkovit@sdsc.edu >`
Glenn Lockwood, PhD
San Diego Supercomputer Center
`<glock@sdsc.edu>`
Jun Zhu, PhD
University of Wisconsin-Madison
`<jzhu@stat.wisc.edu>`

Examples

```
data(condor)

# Find min/max coordinates and add buffer
xmax = max(condor$x) + 1000
xmin = min(condor$x) - 1000
ymax = max(condor$y) + 1000
ymin = min(condor$y) - 1000

# Calculate grid dimensions
xrange <- xmax - xmin
```

```
yrange <- ymax - ymin
cell.sz = 600
nx <- as.integer(xrange/cell.sz)
ny <- as.integer(yrange/cell.sz)
nz <- ceiling(3000.0/cell.sz)

mv.dat <- initializeMovementData(condor$time, condor$x, condor$y, z.obs=condor$z, t.max=185.0,
sig2obs=25.0, sig2obs.z=81.0)
if (all(is.na(mv.dat$sig2xy))) {
  mv.dat <- estVarMKDE(mv.dat)
}

mkde.obj <- initializeMKDE3D(xmin, cell.sz, nx, ymin, cell.sz, ny, 0.0, cell.sz, nz)
res <- mkde3Dgrid(mkde.obj, mv.dat, 5.0, 1e-20)
```

`mkde3Dinteraction` *Probability of 3D spatial-temporal interaction.*

Description

Metric of 3D spatial-temporal interaction.

Usage

```
mkde3Dinteraction(mkde.obj, move.dat0, move.dat1, t.step, d.thresh)
```

Arguments

`mkde.obj`	An MKDE object created with `initialize3DMKDE`
`move.dat0`	A move data object for the first individual created with `initializeMovementData`
`move.dat1`	A move data object for the second individual created with `initializeMovementData`
`t.step`	A time step used for numerical integration over the movement trajectory
`d.thresh`	The value of the kernel below which its contibrution to the overal density is considered negligible

Details

This function assumes that the two individual animals were observed at the same times. The voxel values returned in the mkde.obj can be summed to obtain a global measure of spatio-temporal interaction.

Value

Returns a list with the following elements:

`mkde.obj`	An updated MKDE object containing the voxel-level Bhattacharyya coefficients
`move.dat0`	A move data object for the first individuals with updated variance parameters
`move.dat1`	A move data object for the second individuals with updated variance parameters

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
<jatracey@usgs.gov>
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
<jsheppard@sandiegozoo.org>
Jun Zhu, PhD
University of Wisconsin-Madison
<jzhu@stat.wisc.edu>

Examples

```
library(terra)
data(condor)
condor <- condor[1:4,] # simply to make example run more quickly
mv.dat0 <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)
n <- nrow(condor)
v <- 20.0 # increase from 0 to increase difference between move trajectories
vz <- 5.0
mv.dat1 <- initializeMovementData(condor$time, condor$x+rnorm(n, 0, v),
condor$y+rnorm(n, 0, v), z.obs=condor$z+rnorm(n, 0, vz), sig2obs=25.0,
sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
# next two lines reduce extent of 2D space to speed execution of example
tmp <- ext(c(range(condor$x) + c(-100, 100), range(condor$y) + c(-100, 100)))
condordem120 <- crop(condordem120, tmp)
cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
nx <- ncol(condordem120)
ny <- nrow(condordem120)
nz <- ceiling(3000.0/cell.sz)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, 0.0, cell.sz, nz)

res <- mkde3Dinteraction(mkde.obj, mv.dat0, mv.dat1, 10.0, 1e-20)
mkde.obj <- res$mkde.obj
mv.dat0 <- res$move.dat0
mv.dat1 <- res$move.dat1
```

`mkdeToTerra` *MKDE to RasterLayer or RasterStack*

Description

Converts an MKDE into a RasterLayer or RasterStack so that raster package functions can be used.

Usage

```
mkdeToTerra(mkde.obj)
```

Arguments

`mkde.obj`	A 2D, 2.5D, or 3D MKDE object created with `initialize2DMKDE` or `initialize3DMKDE` and density initialized with `initializeDensity`

Details

This method converts the density array in the MKDE oject to an object of a class from the raster package. This allows the functions in the raster package to be used with the MKDEs.

Value

If the MKDE is 2D or 2.5D, a RasterLayer object is returned. If the MKDE is 3D, a RasterStack is returned with one layer in the stack per level.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)

# set up MKDE object
fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
cell.sz <- mean(res(pandadem))
ext <- ext(pandadem)
nx <- ncol(pandadem)
ny <- nrow(pandadem)
mkde.obj <- initializeMKDE2D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz, ny)

# set up movement data
data(panda)
mv.dat <- initializeMovementData(panda$time, panda$x, panda$y,
sig2obs=25.0, t.max=185.0)

# estimate density
dens.res <- initializeDensity(mkde.obj, mv.dat)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat
mkde.rst <- mkdeToTerra(mkde.obj)
plot(mkde.rst)
```

`panda` *Giant panda locations*

Description

A data frame containing giant panda location data

Usage

```
data(panda)
```

Format

A data frame with 147 observations on the following 4 variables.

`time`	Elapsed time in minutes
`x`	x-coordinate (UTM easting in meters)
`y`	y-coordinate (UTM northing in meters)
`z`	z-coordinate (height above sea level in meters)

Details

GPS location data acquired from a wild giant panda (Ailuropoda melanoleuca) tracked by the Chinese Academy of Sciences and San Diego Zoo Global in Foping National Nature Reserve, China during August 2008 (Zhang et. al. 2012).

Source

Giant panda (Ailuropoda melanoleuca) biotelemetry data was collected by Fuwen Wei and others of the Key Laboratory of Animal Ecology and Conservation Biology, Institute of Zoology, Chinese Academy of Science, People's Republic of China in collaboration with the San Diego Zoo Institute for Conservation Research. Giant panda research was funded by the National Natural Science Foundation of China (31230011), Wildlife Experimental Platform of Chinese Academy of Sciences, and San Diego Zoo Global.

References

Zhang, Z., Sheppard, J.K., Swaisgood, R.R., Wang, G., Nie, Y., Wei, W., Zhao, N. & Wei, F. (2014). Ecological Scale and Seasonal Heterogeneity in the Spatial Behaviors of Giant Pandas. Integrative Zoology, 9: 46-60.

Examples

```
data(panda)
head(panda, 30)
```

`pandadem` *A RasterLayer containing a digital elevation model (DEM) for the panda data*

Description

Digital elevation model of the terrain within the panda's range with a cell resolution of 30 meters (ASTER GDEM version 2, Tachikawa et. al. 2011).

Format

A raster package RasterLayer object with 58 rows, 78 columns, and a cell size of 30 by 30 meters. Each cell contains an elevation in meters.

Source

URL: http://asterweb.jpl.nasa.gov/gdem.asp

References

Tachikawa, T., Hato, M., Kaku, M. & Iwasaki, A. (2011). The characteristics of ASTER GDEM version 2, IGARSS.

Examples

```
library(terra)
fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
plot(pandadem)
```

`plotMKDE` *Make an image plot of an MKDE.*

Description

Makes an image plot of an MKDE. If the MKDE is 3D, the contours will be based on the entire MKDE, but only one level indexed by z.index will be plotted.

Usage

```
plotMKDE(mkde.obj, z.index=1, probs=c(0.99, 0.95, 0.90, 0.75, 0.5, 0.0),
cmap=rev(rainbow(length(probs)-1)), add=FALSE, ...)
```

Arguments

`mkde.obj`	A 2D, 2.5D, or 3D MKDE object created with `initialize2DMKDE` or `initialize3DMKDE` and density initialized with `initializeDensity`
`z.index`	Index for the z-dimension of the density if a 3D MKDE
`probs`	Probabilities for image contours.
`cmap`	Color map for image plot.
`add`	FALSE to make a new plot, TRUE to add to existing plot
`...`	Additional graphical parameters.

Details

A plot of the density for the specified level in the z-dimension (which should be 1, the defaul value, for a 2D or 2.5D MKDE) is generated.

Value

No value is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
<jatracey@usgs.gov>
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
<jsheppard@sandiegozoo.org>

Examples

```
library(terra)
fpath <- system.file("extdata", "pandadem.RDS", package="mkde")
pandadem <- terra::readRDS(fpath)
cell.sz <- 0.25*mean(res(pandadem))
ext <- ext(pandadem)
nx <- 4*ncol(pandadem)
ny <- 4*nrow(pandadem)
mkde.obj <- initializeMKDE2D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz, ny)

# set up movement data
data(panda)
mv.dat <- initializeMovementData(panda$time, panda$x, panda$y,
sig2obs=25.0, t.max=185.0)

# estimate density
dens.res <- initializeDensity(mkde.obj, mv.dat)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat
plotMKDE(mkde.obj)
```

`setMaximumZfromConstant` *Initialize maximum allowable z-axis coordinates to a constant value.*

Description

Set the upper bounds in the z-dimension for each location in the x and y dimensions to a constant value.

Usage

```
setMaximumZfromConstant(mkde.obj, val)
```

Arguments

`mkde.obj`	2D or 3D MKDE object created with `initialize3DMKDE` or `initialize3DMKDE`, respectively
`val`	The value at which the upper bound should be set for all locations in the x and y dimensions.

Details

Obviously, the upper bound must be greater than the lower bound.

Value

An updated MKDE list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "dugongdem.RDS", package="mkde")
dugongdem <- terra::readRDS(fpath)
cell.sz <- mean(res(dugongdem))
ext <- ext(dugongdem)
nx <- ncol(dugongdem)
ny <- nrow(dugongdem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(dugongdem), na.rm=TRUE), 50.0, 15)
mkde.obj <- setMaximumZfromConstant(mkde.obj, 100.0)
```

`setMaximumZfromRaster` *Initialize maximum z-axis value from a raster*

Description

Set the upper bounds in the z-dimension for each location in the x and y dimensions from a raster.

Usage

```
setMaximumZfromRaster(mkde.obj, spat.raster)
```

Arguments

`mkde.obj`	2D or 3D MKDE object created with `initialize3DMKDE` or `initialize3DMKDE`, respectively
`spat.raster`	A RasterLayer object representing the lower bounds of the space the animal may occupy in the z-dimension.

Details

This function sets the upper bounds of the space the animal may occupy in the z-dimension. For example, the `ascii.raster.filez` argument may represent a raster for elevation for subterranean animals, or other surface.

Value

An updated MKDE list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "dugongdem.RDS", package="mkde")
dugongdem <- terra::readRDS(fpath)
cell.sz <- mean(res(dugongdem))
ext <- ext(dugongdem)
nx <- ncol(dugongdem)
ny <- nrow(dugongdem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(dugongdem), na.rm=TRUE), 50.0, 15)
mkde.obj <- setMaximumZfromRaster(mkde.obj, dugongdem)
```

`setMinimumZfromConstant` *Set minimum z-axis value to a constant.*

Description

Set the lower bounds in the z-dimension for each location in the x and y dimensions to a constant value.

Usage

```
setMinimumZfromConstant(mkde.obj, val)
```

Arguments

`mkde.obj`	2D or 3D MKDE object created with `initialize3DMKDE` or `initialize3DMKDE`, respectively
`val`	The value at which the lower bound should be set for all locations in the x and y dimensions.

Details

Obviously, the lower bound must be less than the upper bound.

Value

An updated MKDE list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "dugongdem.RDS", package="mkde")
dugongdem <- terra::readRDS(fpath)
cell.sz <- mean(res(dugongdem))
ext <- ext(dugongdem)
nx <- ncol(dugongdem)
ny <- nrow(dugongdem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(dugongdem), na.rm=TRUE), 50.0, 15)
mkde.obj <- setMinimumZfromConstant(mkde.obj, -20.0)
```

`setMinimumZfromRaster` *Set minimum z-axis values from a raster*

Description

Set the lower bounds in the z-dimension for each location in the x and y dimensions from a raster.

Usage

```
setMinimumZfromRaster(mkde.obj, spat.raster)
```

Arguments

`mkde.obj`	A 2D or 3D MKDE object created with `initialize3DMKDE` or `initialize3DMKDE`, respectively
`spat.raster`	A RasterLayer object representing the lower bounds of the space the animal may occupy in the z-dimension.

Details

This function sets the lower bounds of the space the animal may occupy in the z-dimension. For example, the `ascii.raster.file` argument may represent a raster for elevation, depth of the sea floor, or other surface.

Value

An updated MKDE list object is returned.

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
fpath <- system.file("extdata", "dugongdem.RDS", package="mkde")
dugongdem <- terra::readRDS(fpath)
cell.sz <- mean(res(dugongdem))
ext <- ext(dugongdem)
nx <- ncol(dugongdem)
ny <- nrow(dugongdem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(dugongdem), na.rm=TRUE), 50.0, 15)
mkde.obj <- setMinimumZfromRaster(mkde.obj, dugongdem-20.0)
```

`writeInterpolatedPathVTK` *Write path to VTK.*

Description

Write the interpolated move path to a VTK file.

Usage

```
writeInterpolatedPathVTK(move.dat, mkde.obj, description, filename, control)
```

Arguments

`move.dat`	A move data object created with `initializeMovementData`
`mkde.obj`	An MKDE object created with `initialize3DMKDE`
`description`	A text description for the file header
`filename`	A string for the path and file name
`control`	A list for finer control

Details

Writes 3D lines between observed locations for move steps. Move steps where the initial location i has a value of FALSE for `move.dat$use.obs[i]` are omitted. Currently, the list argument control only has three elements: (1) “method” with a default value of “linear”, (2) method.par which is a list of method parameters, and (3) z.scale used to scale the z-coordinates. Only the `z.scale` should be set by the user at this time.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem))
ext <- ext(condordem)
nx <- ncol(condordem)
ny <- nrow(condordem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem), na.rm=TRUE), 30.0, 100)

writeInterpolatedPathVTK(mv.dat, mkde.obj,
"Example California condor move steps", "condor_path.vtk")

# Clean up files
unlink("condor_path.vtk")
```

`writeObservedLocationVTK`

Write observed locations to VTK.

Description

Write the observed points to a VTK file.

Usage

```
writeObservedLocationVTK(move.dat, mkde.obj, description, filename, control)
```

Arguments

`move.dat`	A move data object created with `initializeMovementData`
`mkde.obj`	An MKDE object created with `initialize3DMKDE`
`description`	A text description for the file header
`filename`	A string for the path and file name
`control`	A list for finer control

Details

Writes 3D points for observed locations for move steps. Currently, the list argument control only has one element `z.scale` used to scale the z-coordinates.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
data(condor)
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem))
ext <- ext(condordem)
nx <- ncol(condordem)
ny <- nrow(condordem)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem), na.rm=TRUE), 30.0, 100)

writeObservedLocationVTK(mv.dat, mkde.obj,
"Example California condor locations", "condor_locations.vtk")

# Clean up files
unlink("condor_locations.vtk")
```

`writeRasterToVTK` *Write a 2D raster to XDMF XML wrapper and binary data file.*

Description

Write the raster to a XDMF files.

Usage

```
writeRasterToVTK(elev, r.rst, g.rst, b.rst, descr, fname)
```

Arguments

`elev`	A RasterLayer object
`r.rst`	A RasterLayer object for red
`g.rst`	A RasterLayer object for green
`b.rst`	A RasterLayer object for blue
`descr`	String description to be added to header of VTK file
`fname`	The path and base file name for output HDF5 files

Details

This function writes a raster to VTK format. The raster is colored according to the RGB values in `r.rst`, `g.rst`, and `b.rst`, respectively. The RGB balues must be an interger from 0 to 255.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Amit Chourasia, MS
San Diego Supercomputer Center
`<amit@sdsc.edu>`

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
elev.val <- values(condordem120)
elev.min <- min(elev.val, na.rm=TRUE)
elev.max <- max(elev.val, na.rm=TRUE)

# make a color lookup table
cmap <- data.frame(value=c(0.0, 0.25, 0.5, 0.75, 1.0),
  R=c(150, 179, 205, 192, 252),
  G=c(224, 204, 205, 183, 243),
  B=c(94, 147, 168, 147, 226))

cmap$value <- cmap$value*(elev.max - elev.min) + elev.min
# red
f.R <- approxfun(cmap$value, cmap$R)
elev.r <- rast(condordem120)
values(elev.r) <- round(f.R(elev.val))
```

```
# green
f.G <- approxfun(cmap$value, cmap$G)
elev.g <- rast(condordem120)
values(elev.g) <- round(f.G(elev.val))
# blue
f.B <- approxfun(cmap$value, cmap$B)
elev.b <- rast(condordem120)
values(elev.b) <- round(f.B(elev.val))
writeRasterToVTK(condordem120, elev.r, elev.g, elev.b, "Elevation for
California condor Example", "condor_dem.vtk")

# Clean up files
unlink("condor_dem.vtk")
```

`writeRasterToXDMF` *Write a 2D raster to XDMF XML wrapper and binary data file.*

Description

Write the raster to a XDMF files.

Usage

```
writeRasterToXDMF(rast, fname, nodat="NA")
```

Arguments

`rast`	A RasterLayer object
`fname`	The path and base file name for output HDF5 files
`nodat`	A no data character string that will be written in place of no data values.

Details

This function writes XDMF XML wrapper and binary data file.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Amit Chourasia, MS
San Diego Supercomputer Center
`<amit@sdsc.edu>`

Examples

```
library(terra)
fpath <- system.file("extdata", "condordem.RDS", package="mkde")
condordem <- terra::readRDS(fpath)

# Save as XDMF (notice no file extension in file name)
writeRasterToXDMF(condordem, "condor_dem")

# Clean up files
unlink("condor_dem.dat")
unlink("condor_dem.xdmf")
```

`writeToGRASS` *Write MKDE to a GRASS GIS 3D ASCII raster file.*

Description

Write the MKDE to a VTK file.

Usage

```
writeToGRASS(mkde.obj, fname, nodat="NA", cumprob=FALSE)
```

Arguments

`mkde.obj`	3D MKDE object created with `initialize3DMKDE` and density initialized with `initializeDensity`
`fname`	The patch and file name for output VTK file
`nodat`	A no data character string that will be written in place of no data values.
`cumprob`	Indicate whether to write the voxel probabilities of cumulative probabilities.

Details

This function writes a GRASS GIS ASCII raster file that can be imported using the `r3.in.ascii` function.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
<jatracey@usgs.gov>
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
<jsheppard@sandiegozoo.org>

Examples

```
library(terra)
data(condor)
condor <- condor[1:20,] # simply to make example run more quickly
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
nx <- ncol(condordem120)
ny <- nrow(condordem120)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem120), na.rm=TRUE), cell.sz, 25)

# note: we use a raster coarse integration time step so the
# example runs faster
dens.res <- initializeDensity(mkde.obj, mv.dat, integration.step=10.0)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat

# Write file
writeToGRASS(mkde.obj, "ascii3d.txt")

# Clean up files
unlink("ascii3d.txt")
```

`writeToVTK` *Write MKDE to an ASCII Visualization Tool Kit (VTK) structured grid file.*

Description

Write the MKDE to a VTK file.

Usage

```
writeToVTK(mkde.obj, fname, description="3D MKDE", cumprob=FALSE)
```

Arguments

`mkde.obj`	3D MKDE object created with `initialize3DMKDE` and density initialized with `initializeDensity`
`fname`	The patch and file name for output VTK file
`description`	A character string with a brief description that will be placed in the header of the VTK file
`cumprob`	Indicate whether to write the voxel probabilities of cumulative probabilities.

Details

This function writes a VTK structured grid file for a 3D MKDE that can be used in 3D visualization tools such as MayaVI or ParaView

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`

Examples

```
library(terra)
data(condor)
condor <- condor[1:20,] # simply to make example run more quickly
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)

fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
# next two lines reduce extent of 2D space to speed execution of example
tmp <- ext(c(range(condor$x) + c(-100, 100), range(condor$y) + c(-100, 100)))
condordem120 <- crop(condordem120, tmp)

cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
nx <- ncol(condordem120)
ny <- nrow(condordem120)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem120), na.rm=TRUE), cell.sz, 25)

# note: we use a raster coarse integration time step so the
# example runs faster
dens.res <- initializeDensity(mkde.obj, mv.dat, integration.step=10.0)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat

# Save as VTK file
writeToVTK(mkde.obj, "condor_3dMKDE.vtk",
description="Example California condor 3D MKDE")

# Clean up files
unlink("condor_3dMKDE.vtk")
```

`writeToXDMF` *Write 3D MKDE to XDMF XML wrapper and binary data file.*

Description

Write the MKDE to a XDMF files.

Usage

```
writeToXDMF(mkde.obj, fname, nodat="NA", cumprob=FALSE)
```

Arguments

`mkde.obj`	3D MKDE object created with `initialize3DMKDE` and density initialized with `initializeDensity`
`fname`	The path and base file name for output XDMF files
`nodat`	A no data character string that will be written in place of no data values.
`cumprob`	Indicate whether to write the voxel probabilities of cumulative probabilities.

Details

This function writes XDMF XML wrapper and binary data file.

Value

No value is returned

Author(s)

Jeff A. Tracey, PhD
USGS Western Ecological Research Center, San Diego Field Station
`<jatracey@usgs.gov>`
James Sheppard, PhD
San Diego Zoo Institute for Conservation Research
`<jsheppard@sandiegozoo.org>`
Amit Chourasia, MS
San Diego Supercomputer Center
`<amit@sdsc.edu>`

Examples

```
library(terra)
data(condor)
condor <- condor[1:20,] # simply to make example run more quickly
mv.dat <- initializeMovementData(condor$time, condor$x, condor$y,
z.obs=condor$z, sig2obs=25.0, sig2obs.z=81.0, t.max=65.0)
```

```
fpath <- system.file("extdata", "condordem120.RDS", package="mkde")
condordem120 <- terra::readRDS(fpath)
cell.sz <- mean(res(condordem120))
ext <- ext(condordem120)
nx <- ncol(condordem120)
ny <- nrow(condordem120)
mkde.obj <- initializeMKDE3D(ext$xmin, cell.sz, nx, ext$ymin, cell.sz,
ny, min(values(condordem120), na.rm=TRUE), cell.sz, 25)

# note: we use a raster coarse integration time step so the
# example runs faster
dens.res <- initializeDensity(mkde.obj, mv.dat, integration.step=10.0)
mkde.obj <- dens.res$mkde.obj
mv.dat <- dens.res$move.dat

# Write file (note no file extension!)
writeToXDMF(mkde.obj, "condor_3dMKDE")

# Clean up files
unlink("condor_3dMKDE.dat")
unlink("condor_3dMKDE.xdmf")
```

Index