Package 'tangles'

July 22, 2025

Type Package

Title Anonymisation of Spatial Point Patterns and Grids

Version 2.0.1

Date 2025-05-28

Description Methods for anonymisation of spatial datasets while preserving spatial structure and relationships. Original coordinates or raster geometries are transformed using randomized or predefined vertical shifts, horizontal shifts, and rotations. Compatible with point-based data in 'matrix', 'data.frame', or 'sf' formats, as well as 'terra' raster objects. Supports reversible anonymisation workflows, hash-based validation, shapefile export, and consistent tangling across related datasets using stored transformation sequences. Approach informed by the De-Identification Decision Making Framework (CM O'Keefe, S Otorepec, M Elliot, E Mackey, and K O'Hara 2017) <doi:10.4225/08/59c169433efd4>.

Depends R (>= 4.0)

Imports terra, sf, digest

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

License GPL-2

Encoding UTF-8

LazyData false

NeedsCompilation no

Author Brendan Malone [aut, cre]

Maintainer Brendan Malone <brendan.malone@csiro.au>

Repository CRAN

Date/Publication 2025-06-02 07:50:02 UTC

Contents

`detangler_data` *Example object that is output from* `tangles` *function.*

Description

This output that would be derived from the `tangles` function. It contains the instructions for anonymisation of associated point pattern or raster data, or the re-identify tangled data.

Usage

```
data(detangler_data)
```

Format

`detangler_data` is a `list` with 3 elements. The first element is a unique hash key for tracking data processing provenance. The second element contains the integer sequence of steps that were used to anonymize a given data sets. There are 3 possible anonymizing procedures: shift vertical (1), shift (horizontal), and rotate (3). The last element is a `data.frame` that encodes the values to perform either anonymization or re-identification, given the particular anonymization step used.

Note

This data is the sort of output that one would expect when the `tangles` function is used.

References

- CM O'Keefe, S Otorepec, M Elliot, E Mackey, and K O'Hara (2017) The De-Identification Decision Making Framework. CSIRO Reports EP173122 and EP175702. doi:10.4225/08/59c169433efd4

Examples

```
data(detangler_data)
detangler_data
```

detangles *Revert anonymised point patterns or raster objects*

Description

Reverses the spatial anonymisation applied by the tangles function, restoring original XY coordinates or raster data. Requires a matching detangler object containing the transformation sequence and hash key. The restoration process preserves spatial relationships and supports both point and raster inputs.

Usage

```
detangles(
  data = NULL,
  tanglerInfo = NULL,
  raster_object = FALSE,
  stub = NULL,
  hash_key = NULL,
  saveTangles = FALSE,
  exportShapefile = FALSE,
  path = NULL
)
```

Arguments

data	A 2-column matrix or data.frame of spatial coordinates, an sf POINT object, or a terra::SpatRaster object.
tanglerInfo	A detangler object returned by tangles(), containing the anonymisation steps and associated hash key.
raster_object	Logical; set to TRUE if the input is a raster object. This enables value restoration for all raster cells.
stub	Optional character string used in output file naming. Helps distinguish saved files.
hash_key	Character string representing the hash key to verify identity of the detangler. Must match tanglerInfo\$hash.
saveTangles	Logical; if TRUE, saves untangled data as .rds (and optionally raster .tif) files to path.
exportShapefile	Logical; if TRUE and input is point-based, a shapefile of the untangled coordinates is written (with undefined CRS).
path	Directory to save output files if saveTangles or exportShapefile is enabled. Defaults tempdir().

Value

A restored `data.frame` of spatial XY coordinates or a `terra::SpatRaster` object, depending on input type.

If `saveTangles = TRUE`, corresponding files are saved to `path`, including:

- `detangledXY_<stub>_<hash>.rds`
- `detangledXY_raster_<name>.tif` (for raster input)
- `detangledXY_<stub>_<hash>.shp` (for point shapefile export, if `exportShapefile = TRUE`)

Note

If the `hash_key` provided does not match the one in the detangler object, the function stops to prevent incorrect spatial restoration.

When writing shapefiles, no CRS is assigned. This ensures that untangled data does not imply real-world georeferencing but retains relative spatial relationships.

Outputs can be safely saved and shared using the embedded hash key and transformation record.

Author(s)

Brendan Malone

References

- CM O'Keefe, S Otorepec, M Elliot, E Mackey, and K O'Hara (2017) The De-Identification Decision Making Framework. CSIRO Reports EP173122 and EP175702. doi:10.4225/08/59c169433efd4

Examples

```
## EXAMPLE 1: Untangle point data.frame and export shapefile
library(sf)
set.seed(1)

# Simulate XY data
pts <- data.frame(X = runif(100), Y = runif(100))

# Anonymise
tangled <- tangles(data = pts, depth = 4, saveTangles = FALSE)

# Restore points
restored_pts <- detangles(
  data = tangled[[1]],
  tanglerInfo = tangled[[2]],
  raster_object = FALSE,
  stub = "points",
  hash_key = tangled[[2]]$hash,
  exportShapefile = TRUE
)
```

```
## EXAMPLE 2: Untangle from sf POINT input
sf_pts <- st_as_sf(pts, coords = c("X", "Y"))

# Anonymise sf object
tangled_sf <- tangles(data = sf_pts, depth = 3)

# Restore using sf input
restored_from_sf <- detangles(
  data = tangled_sf[[1]],
  tanglerInfo = tangled_sf[[2]],
  stub = "sf_restore",
  hash_key = tangled_sf[[2]]$hash,
  exportShapefile = TRUE
)

## EXAMPLE 3: Untangle raster data (terra)
library(terra)
ext_path <- system.file("extdata", package = "tangles")
rast.files <- list.files(path = ext_path, full.names = TRUE)
rasters <- terra::rast(rast.files)

# Anonymise raster
tangled_rast <- tangles(data = rasters, depth = 3, rasterdata = TRUE, raster_object = TRUE)

# Restore raster
restored_rast <- detangles(
  data = tangled_rast[[1]],
  tanglerInfo = tangled_rast[[2]],
  raster_object = TRUE,
  stub = "raster_demo",
  hash_key = tangled_rast[[2]]$hash,
  saveTangles = TRUE
)
```

`Hunter Valley subsoil pH points` *Hunter Valley subsoil pH data with environmental covariates*

Description

A `dataframe` carrying point coordinates for 506 observations of soil pH from the Lower Hunter Valley, NSW, Australia. The depth interval of the observation is 60-100cm. Together with the soil pH information is environmental covariate data that have been intersected with the point data. The environmental covariates have been sourced from a digital elevation model and Landsat 7 spectral band reflectance.

Usage

```
data(HV_subsoilpH)
```

Format

HV_subsoilpH is 506 row dataframe with the first 2 columns indicating the projected coordinate locations. The coordinate reference system is WGS 84 Zone 56. Soil pH for the 60-100cm depth interval is recorded in the following column. A suite of intersected environmental covariate data fills the remaining columns. This environmental covariate information refers to:

- Terrain_Ruggedness_Index: A quantitative measure of topographic heterogeneity. High values indicate terrain is rugged or heterogeneous.
- AACN: Difference between elevation and an interpolation of a channel network base level elevation. Knowledge of the spatial distribution of channel networks (lines) is therefore necessary for this parameter.
- Landsat_Band1: Earth surface reflectance information derived from Landsat 7 ETM+ satellite. Band 1 spectral range is .45 to .52 microns (visible blue).
- Elevation: Meters above sea level; derived from a digital elevation model.
- Hillshading: Analytic hill shading derived from digital elevation model and a fixed sun degree angle.
- Light_insolation: Measure of potential incoming solar radiation, and used as a parameter for evaluating the positional aspect effect. Derived from digital elevation model, this parameter was evaluated over the duration of a single calendar year with a 5 day time step.
- Mid_Slope_Positon: A relative slope position parameter which gives a classification of the slope position in both valley and crest positions.
- MRVBF: Multi-resolution valley bottom flatness is derived using slope and elevation to classify valley bottoms as flat, low areas (Gallant and Dowling 2003). This is accomplished through a series of neighborhood operations at progressively coarser resolutions with the goal of identifying both small and large valleys. MRVBF has been used extensively for the delineation and grading of valley floor units corresponding to areas of alluvial and colluvial deposits. High values of MRVBF indicate relatively low, flat areas of the landscape.
- NDVI: Normalized difference vegetation index. Derived from Landsat 7 data (B4-B3)/(B4+B3). High values indicate actively growing vegetation.
- TWI: A secondary land form parameter which estimates for each pixel, its tendency to accumulate water.
- Slope: Measured in degrees, is the first derivative of elevation in the direction of greatest slope.

Details

The area in question is the Hunter Wine Country Private Irrigation District (HWCPID), situated in the Lower Hunter Valley, NSW (32.83S 151.35E), and covers an area of approximately 220 km2. The HWCPID is approximately 140 km north of Sydney, NSW, Australia. Climatically, the HWCPID is situated in a temperate climatic zone, and experiences warm humid summers, and relatively cooler yet also humid winters. Rainfall is mostly uniformly distributed throughout the year. On average the HWCPID receives just over 750 mm of rainfall annually. In terms of land use, an expansive viticultural industry is situated in the area and is possibly most widespread of rural industries, followed by dry land agricultural grazing systems.

Note

This data set is used in the Use R for Digital Soil Mapping manual in the section about quantification of uncertainties.

References

- Gallant, J.C., Dowling, T.I. (2003). A multiresolution index of valley bottom flatness for mapping depositional areas. *Water Resources Research*, 39(12), 1347. doi:10.1029/2002WR001426
- Malone, B.P., Hughes, P., McBratney, A.B., Minasny, B. (2014). A model for the identification of terrons in the Lower Hunter Valley, Australia. *Geoderma Regional*, 1, 31–47. doi:10.1016/j.geodrs.2014.08.001

Examples

```
data(HV_subsoilpH)
summary(HV_subsoilpH)
```

---

hunterCovariates_sub *Environmental Covariate Rasters for a Subset of the Lower Hunter Valley, NSW*

Description

A suite of GeoTIFF rasters representing environmental covariates for a subset of the Lower Hunter Valley in New South Wales, Australia. These covariates are used in digital soil mapping and terrain analysis.

Format

The dataset consists of multiple raster layers (GeoTIFF format) located in the `inst/extdata/` directory of the package, each with a spatial resolution of 25 m × 25 m and a CRS of WGS 84 UTM Zone 56.

Available files are prefixed with `hunterCovariates_sub_` and include:

`hunterCovariates_sub_Terrain_Ruggedness_Index.tif` Topographic ruggedness index (TRI).

`hunterCovariates_sub_AACN.tif` Elevation above channel network base level.

`hunterCovariates_sub_Landsat_Band1.tif` Landsat 7 ETM+ Band 1 reflectance (0.45–0.52 µm).

`hunterCovariates_sub_Elevation.tif` Elevation in meters above sea level, derived from a DEM.

`hunterCovariates_sub_Hillshading.tif` Hillshade raster generated from the DEM using a fixed sun angle.

`hunterCovariates_sub_Light_insolation.tif` Potential solar radiation calculated over a calendar year at 5-day intervals.

`hunterCovariates_sub_Mid_Slope_Position.tif` Slope position classification for crest/valley context.

`hunterCovariates_sub_MRVBF.tif` Multi-resolution valley bottom flatness index.

`hunterCovariates_sub_NDVI.tif` Normalized Difference Vegetation Index based on Landsat 7.

`hunterCovariates_sub_TWI.tif` Topographic Wetness Index (TWI).

`hunterCovariates_sub_Slope.tif` Slope angle in degrees.

Details

The subset area corresponds to the Hunter Wine Country Private Irrigation District (HWCPID) in the Lower Hunter Valley (approx. 32.83°S, 151.35°E), located ~140 km north of Sydney. The HWCPID covers around 220 km² and supports viticulture and dryland grazing under a temperate, humid climate with ~750 mm annual rainfall.

These covariates are used in spatial prediction tasks, terrain classification, and training examples in digital soil mapping.

References

- Gallant, J.C., Dowling, T.I. (2003). A multiresolution index of valley bottom flatness for mapping depositional areas. *Water Resources Research*, 39(12), 1347. doi:10.1029/2002WR001426
- Malone, B.P., Hughes, P., McBratney, A.B., Minasny, B. (2014). A model for the identification of terrons in the Lower Hunter Valley, Australia. *Geoderma Regional*, 1, 31–47. doi:10.1016/j.geodrs.2014.08.001

Examples

```
library(terra)

# Load and plot the elevation raster
elevation <- rast(system.file("extdata/hunterCovariates_sub_Elevation.tif", package = "tangles"))
plot(elevation, main = "Hunter Valley Subset - Elevation")
```

---

tangler *Re-apply anonymisation using a stored detangler object*

Description

For a given `detangler` object (from `tangles`), spatial coordinates or raster data can be re-anonymised using the same transformation sequence. This allows consistent tangling of related datasets using a fixed spatial masking.

Usage

```
tangler(
  data = NULL,
  tanglerInfo = NULL,
  raster_object = FALSE,
  stub = NULL,
```

```
  saveTangles = FALSE,
  exportShapefile = FALSE,
  path = NULL
)
```

Arguments

data	A 2-column matrix, data.frame of coordinates, an sf POINT object, or a terra::SpatRaster object.
tanglerInfo	A detangler object returned by tangles, which encodes the anonymisation steps and hash key.
raster_object	Logical; set to TRUE if input is a raster object. This ensures the output is reconstructed as a terra::SpatRaster.
stub	Optional character string for file naming. Combined with the hash key for saved outputs.
saveTangles	Logical; if TRUE, the tangler output is saved as an .rds object, and raster layers as .tif.
exportShapefile	Logical; if TRUE and input is point-based, a shapefile of the tangled data is also exported (no CRS assigned).
path	Directory to write output files if saving is enabled. Defaults tempdir().

Value

Returns either:

- A data.frame of spatial coordinates (for point-based data), or
- A terra::SpatRaster (for raster input)

If saveTangles = TRUE, corresponding outputs are written to .rds, and optionally .tif (for raster) or shapefile (for point).

Note

This function re-applies a saved transformation sequence. It does not generate new spatial shifts.

When working with raster data, only 90°, 180°, or 270° rotations are supported. If the stored detangler contains unsupported rotation angles, the function will stop to avoid corrupting the raster structure.

Shapefiles are written without CRS metadata to preserve anonymity. Coordinates retain spatial structure but not geolocation accuracy.

Author(s)

Brendan Malone

References

- CM O'Keefe, S Otorepec, M Elliot, E Mackey, and K O'Hara (2017) The De-Identification Decision Making Framework. CSIRO Reports EP173122 and EP175702. doi:10.4225/08/59c169433efd4

Examples

```
## Example 1: Tangling a point data.frame
library(digest)
set.seed(123)
pts <- data.frame(X = runif(100), Y = runif(100))

# Anonymise original points
t1 <- tangles(data = pts, depth = 4)

# Tangling a second dataset using the same detangler
tangled_again <- tangler(
  data = pts,
  tanglerInfo = t1[[2]],
  stub = "pt_demo",
  saveTangles = TRUE,
  exportShapefile = TRUE
)

## Example 2: Tangling an sf POINT object
library(sf)
sf_pts <- st_as_sf(pts, coords = c("X", "Y"))

tangled_sf <- tangler(
  data = sf_pts,
  tanglerInfo = t1[[2]],
  stub = "sf_demo",
  saveTangles = TRUE,
  exportShapefile = TRUE
)

## Example 3: Tangling a raster using stored detangler
library(terra)
ext_path <- system.file("extdata", package = "tangles")
rast.files <- list.files(path = ext_path, full.names = TRUE)
rasters <- terra::rast(rast.files)

# Must use a detangler with 90°/180°/270° rotations for raster compatibility
t2 <- tangles(data = rasters, depth = 3, rasterdata = TRUE, raster_object = TRUE)

tangled_rast <- tangler(
  data = rasters,
  tanglerInfo = t2[[2]],
  raster_object = TRUE,
  stub = "r_demo",
  saveTangles = TRUE
)
```

tangles *Anonymise spatial point patterns and raster objects*

Description

Performs spatial anonymisation ("tangling") of coordinates through randomized transformation sequences, preserving relative spatial relationships while obscuring true locations. Three transformation types are used: X shift, Y shift, and rotation around a random origin. The sequence and parameters used for anonymisation are recorded and returned for later disentanglement.

Usage

```
tangles(data = NULL, depth = 3, rasterdata = FALSE, raster_object = FALSE,
saveTangles = FALSE, exportShapefile = FALSE, path = NULL)
```

Arguments

data	Either a two-column matrix or data.frame of coordinates, an sf POINT object, or a terra::SpatRaster object.
depth	Integer. Number of transformation steps to apply (default is 3).
rasterdata	Logical. If TRUE, rotations are limited to 90, 180, or 270 degrees to preserve axis alignment in raster data.
raster_object	Logical. Set TRUE if input is a terra::SpatRaster object. All raster values will be mapped to their transformed coordinates.
saveTangles	Logical. If TRUE, writes both the transformed data and detangler metadata to the specified path.
exportShapefile	Logical. If TRUE and input is a point-based dataset (not a raster), the anonymised coordinates are written to a shapefile. No CRS is assigned.
path	Character. Path to directory where outputs will be saved. Defaults tempdir().

Value

A list with two elements:

- The transformed coordinates (if point input) or a terra::SpatRaster (if raster input)
- A detangler list containing the transformation log (unpicker) and a unique hash

If saveTangles = TRUE, the following files are written:

- tangledXY_<hash>.rds or tangledXY_raster_<hash>.rds — the transformed data
- detangler_<hash>.rds — the metadata required for reverse transformation
- Optional: shapefile output if exportShapefile = TRUE

Note

For raster input, both `rasterdata = TRUE` and `raster_object = TRUE` are usually recommended. This ensures rotation steps align with raster grid expectations.

The detangler object is the critical output. It allows the same transformation sequence to be reversed or applied to related datasets.

Coordinate reference systems are intentionally ignored in this function. Anonymised outputs do not have spatial meaning, but retain topological properties of the input.

If writing shapefiles, no CRS is assigned and no `.prj` file is written.

Author(s)

Brendan Malone

References

- CM O'Keefe, S Otorepec, M Elliot, E Mackey, and K O'Hara (2017) The De-Identification Decision Making Framework. CSIRO Reports EP173122 and EP175702. doi:10.4225/08/59c169433efd4

Examples

```
## Example 1: Using point data.frame
library(digest)
set.seed(1)
pts <- data.frame(X = runif(100), Y = runif(100))
res <- tangles(data = pts, depth = 4)
str(res)

## Example 2: Using sf object
library(sf)
sf_pts <- st_as_sf(pts, coords = c("X", "Y"))
res_sf <- tangles(data = sf_pts, depth = 3, exportShapefile = TRUE)

## Example 3: Using terra raster
library(terra)
ext_path <- system.file("extdata", package = "tangles")
rast.files <- list.files(path = ext_path, full.names = TRUE)
rasters <- terra::rast(rast.files)
res_r <- tangles(data = rasters, depth = 3, rasterdata = TRUE, raster_object = TRUE)
str(res_r)
```

Index