Package: topoDistance (via r-universe)

November 5, 2024

Type Package

Title Calculating Topographic Paths and Distances

Version 1.0.2

Description A toolkit for calculating topographic distances and
identifying and plotting topographic paths. Topographic
distances can be calculated along shortest topographic paths
(Wang (2009) <doi:10.1111/j.1365-294X.2009.04338.x>), weighted
topographic paths (Zhan et al. (1993)
<doi:10.1007/3-540-57207-4_29>), and topographic least cost
paths (Wang and Summers (2010)
<doi:10.1111/j.1365-294X.2009.04465.x>). Functions can map
topographic paths on colored or hill shade maps and plot
topographic cross sections (elevation profiles) for the paths.

Depends R (>= 3.1.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports igraph, gdistance, plotly, raster, RColorBrewer, scales, sp

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libglpk-dev make
libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev

Repository https://ianjwang.r-universe.dev

RemoteUrl https://github.com/ianjwang/topodistance

RemoteRef HEAD

RemoteSha 11d7f8650385b5485461f899b4932a8c6da9cba0

Contents

---

topoDist   *Topographic distances and paths*

Description

Calculates shortest topographic distances and paths

Usage

```
topoDist(DEM, pts, directions = 8, paths = FALSE, zweight = 1)
```

Arguments

DEM	A RasterLayer for digital elevation model (DEM) data.
pts	A SpatialPoints object or two-column matrix with xy coordinates for the geographic points from which to calculate pairwise distances and paths.
directions	numeric (default = 8). The number of directions for movement between cells, either 4 or 8.
paths	logical. Default is FALSE, in which case only topographic distances are calculated. If TRUE, topographic paths are also identified.
zweight	numeric (default = 1). The weight to be applied to the elevation (z) distances relative to the horizontal (xy) distances.

Details

If paths = FALSE, the function will return a matrix of pairwise topographic distances between the specified points. If paths = TRUE, the function will return a list with two items: (1) the matrix of pairwise topographic distances, and (2) a SpatialLines object containing the topographic paths.

Value

Matrix of topographic distances (if paths = FALSE), or a list containing a matrix of topographic distances and the topographic paths as an object of class SpatialLines (if paths = TRUE).

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
    c(-119.5566, 37.72474,
    -119.4718, 37.76078))
topoDist(Yosemite$DEM, xy, paths = TRUE)
```

---

topoDistance	*topoDistance: A package for calculating topographic paths and distances*

Description

The topoDistance package provides functions for calculating topographic distances and identifying and plotting topographic paths. Topographic distances can be calculated along shortest topographic paths or topographic least cost paths. Functions can map topographic paths on colored or hillshade maps and plot topographic cross sections (elevation profiles) for the paths.

Details

Unlike the topographically-corrected distances calculated by some GIS software, which just adjust for elevational changes along a straight-line path between points, topoDistance calculates the distance along the shortest topographic path between points, which is more likely to realistically reflect biological movement on a topographically complex landscape.

Topographic distances are calculated as the hypotenuse of the horizontal and vertical distances between cells on an elevation raster. These distances are assignd to the weights of vertices between the nodes for each cell on a landscape graph, and functions from the gdistance and igraph packages are used to find the shortest path between nodes. For topographic least cost paths, resistance distance weights are multiplied by the topographic distance weights to get topographically corrected least cost path distances.

To see the preferable citation of the package, type citation("topoDistance").

---

topoLCP	*Topographic least cost paths and distances*

Description

Calculates topographic least cost distances and paths

Usage

```
topoLCP(DEM, costSurface, pts, directions = 8, paths = FALSE,
  zweight = 1)
```

Arguments

`DEM`	A RasterLayer for digital elevation model (DEM) data.
`costSurface`	A RasterLayer for the conductance (inverse of resistance) values for each cell.
`pts`	A SpatialPoints object or two-column matrix with xy coordinates for the geographic points from which to calculate pairwise distances and paths.
`directions`	numeric (default = 8). The number of directions for movement between cells, either 4 or 8.
`paths`	logical. Default is FALSE, in which case only topographic distances are calculated. If TRUE, topographic paths are also identified.
`zweight`	numeric (default = 1). The weight to be applied to the elevation (z) distances relative to the horizontal (xy) distances.

Details

The values of the raster for costSurface should be conductance values rather than resistance values. These can be calculating by taking the inverse of resistance values.

Value

Matrix of topographic distances (if paths = FALSE), or a list containing a matrix of topographic distances and the topographic paths as an object of class SpatialLines (if paths = TRUE).

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
   c(-119.5566, 37.72474,
   -119.4718, 37.76078))
topoLCP(Yosemite$DEM, Yosemite$SDM, xy, paths = TRUE)
```

---

topoPathMap   *Map of topographic paths*

Description

Plots a map of topographic paths on a landscape layer

Usage

```
topoPathMap(DEM, pts, topoPaths, type = "hillshade",
  costSurface = NULL, costColors = NULL, pathWidth = 2,
  pathColor = "darkred", alpha = 0.65, angle = 45, direction = 0,
  cex = 2, bg = "gray", col = "black", pch = 21, ...)
```

Arguments

DEM	A RasterLayer for digital elevation model (DEM) data.
pts	A SpatialPoints object or two-column matrix with xy coordinates for the geographic points from which to calculate pairwise distances and paths.
topoPaths	A SpatialLines object containing the topographic paths to be plotted.
type	character (default = "hillshade"). Type of map on which to plot topographic paths: "hillshade", "terrain", or "topo".
costSurface	(optional) A RasterLayer for the conductance (inverse of resistance) values for each cell.
costColors	(optional) A function that takes an integer argument (the required number of colors) and returns a character vector of colors (see rgb) interpolating the given sequence (similar to heat.colors or terrain.colors), such as the one returned by colorRampPalette.
pathWidth	numeric (default = 2). Width for drawing path line.
pathColor	character (default = "darkred"). Color for drawing path line.
alpha	numeric (default = 0.65). Alpha transparency for drawing path line.
angle	numeric (default = 45). Angle of lighting for hillshade maps, only (type = "hillshade").
direction	numeric (default = 0). Direction of lighting for hillshade maps, only (type = "hillshade").
cex	numeric (optional). Point size.
bg	character (optional). Background color for spatial points.
col	character (optional). Color for spatial points.
pch	numeric (optional). Shape of spatial points.
...	character, logical, or numeric (optional). Additional arguments to be passed to the plot function.

Details

The objects supplied for the DEM and pts arguments (and, optionally, costSurface) are generally those used to calculate the topographic paths using topoDist or topoLCP.

For the type argument, choosing "hillshade" will plot a shaded relief map, "terrain" will plot a map with terrain colors, and "topo" will plot a map with topo colors.

Optional arguments can be supplied to control the size (cex), shape (pch), and color (bg and col) of the points.

Value

Plot of topographic paths

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
   c(-119.5566, 37.72474,
   -119.4718, 37.76078))
YosPaths <- topoDist(Yosemite$DEM, xy, paths = TRUE)
topoPathMap(Yosemite$DEM, xy, topoPaths = YosPaths)
```

---

topoPaths   *Identify shortest topographic paths*

Description

Identify shortest topographic paths

Usage

```
topoPaths(t.dist, pts)
```

Arguments

t.dist   A TransitionLayer object.

pts   A SpatialPoints object for the geographic points from which to calculate pair-
                wise distances and paths.

Details

This function identifies shortest topographic paths from a topographic TransitionLayer. It does not
need to be called separately from the topoDist and topoLCP functions.

Value

An object of class SpatialLines

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
   c(-119.5566, 37.72474,
   -119.4718, 37.76078))
xy <- sp::SpatialPoints(xy)
topoTL <- topoSurface(Yosemite$DEM)
topoPaths(topoTL, xy)
```

---

topoProfile   *Elevation profiles of topographic paths*

Description

Plots topographic cross sections (elevation profiles) of topographic paths

Usage

```
topoProfile(DEM, topoPaths, pts = 100, type = "base",
  singlePlot = FALSE, rows = NULL, cols = NULL, limits = NULL,
  legendx = "bottomright")
```

Arguments

DEM	A RasterLayer for digital elevation model (DEM) data.
topoPaths	A SpatialLines object containing the topographic paths to be plotted.
pts	numeric (default = 100). The number of elevation points to plot along each path.
type	character (default = "base"). Type of plotting: "base" or "plotly".
singlePlot	logical (default = FALSE). If TRUE, profiles will be drawn on a single plot.
rows	numeric (optional). Number of rows for plot layout (if singlePlot = FALSE)
cols	numeric (optional). Number of columns for plot layout (if singlePlot = FALSE)
limits	numeric vector (optional). A vector with the lower and upper limits for the y-axis (elevation).
legendx	character (default = "bottomright"). Position for the legend.

Details

For the type argument, choosing "base" will use base R plotting, and choosing "plotly" will draw an interactive plot with the plotly package.

If rows, cols, or limits are not specified, the most even arrangement will be detected and used.

Value

Plot of elevation profiles

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
   c(-119.5566, 37.7247,
     -119.4718, 37.7608))
YosPaths <- topoDist(Yosemite$DEM, xy, paths = TRUE)
topoProfile(Yosemite$DEM, topoPaths = YosPaths)
```

| topoSurface | *Topographic distance surface* |

Description

Generates a TransitionLayer object for topographic distance from a RasterLayer

Usage

```
topoSurface(DEM, conductance = TRUE, directions = 8, zweight = 1)
```

Arguments

DEM   A RasterLayer for digital elevation model (DEM) data.

conductance   logical (default = TRUE). If FALSE, resistance values are returned. If TRUE,
               conductance values (1/resistance) are returned.

directions   numeric (default = 8). Directions of allowable movement between raster cells
               (4 or 8).

zweight   numeric (default = 1). The weight to be applied to the elevation (z) distances
               relative to the horizontal (xy) distances.

Details

This function generates a TransitionLayer from a DEM, which is used by the topoDist and topoLCP
functions. It does not need to be called separately from the topoDist and topoLCP functions.

Value

TransitionLayer

Examples

```
YosTL <- topoSurface(Yosemite$DEM)
```

| topoWeightedDist | *Weighted topographic distances and paths* |

Description

Calculates weighted topographic distances and paths

Usage

```
topoWeightedDist(DEM, pts, directions = 8, paths = FALSE,
  hFunction = NULL, vFunction = NULL)
```

Arguments

`DEM`	A RasterLayer for digital elevation model (DEM) data; should be in a projected coordinate system.
`pts`	A SpatialPoints object or two-column matrix with xy coordinates for the geographic points from which to calculate pairwise distances and paths.
`directions`	numeric (default = 8). The number of directions for movement between cells, either 4 or 8.
`paths`	logical. Default is FALSE, in which case only topographic distances are calculated. If TRUE, topographic paths are also identified.
`hFunction`	character or function (default = NULL). A function describing the cost of changing aspect angle.
`vFunction`	character or function (default = NULL). A function describing the cost of movement along an incline.

Details

The hFunction argument can be set to "exponential" or "linear" to use standard functions for the cost of aspect angle changes. If providing a custom function instead, the equation should be a function of one variable, angle (in radians). The vFunction argument can be set to "exponential" or "quadratic" to use standard functions for the cost of movement along an incline. If providing a custom function instead, The equation should be a function of one variable, slope. If paths = FALSE, the function will return a matrix of pairwise topographic distances between the specified points. If paths = TRUE, the function will return a list with two items: (1) the matrix of pairwise topographic distances, and (2) a SpatialLines object containing the topographic paths.

Value

Matrix of transport costs (if paths = FALSE), or a list containing a matrix of transport costs and paths as an object of class SpatialLines (if paths = TRUE).

Examples

```
xy <- matrix(ncol = 2, byrow = TRUE,
             c(-119.5566, 37.7247,
               -119.4718, 37.7608))
topoWeightedDist(Yosemite$DEM, xy, vFunction = "exponential")
```

Yosemite	*Spatial data for western fence lizards in Yosemite*

Description

A dataset containing a RasterStack with two RasterLayer objects, a digital elevation model (DEM) for part of Yosemite National Park (USA) and a species distribution model (SDM) for the western fence lizard (Sceloporus occidentalis).

Usage

```
data(Yosemite)
```

Format

A RasterStack

Examples

```
data(Yosemite)
```

Index