Package: MultiscaleDTM (via r-universe)

October 5, 2024

Title Multi-Scale Geomorphometric Terrain Attributes

Version 0.8.3

Description Calculates multi-scale geomorphometric terrain attributes from regularly gridded digital terrain models using a variable focal windows size (Ilich et al. (2023) <doi:10.1111/tgis.13067>).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

BugReports https://github.com/ailich/MultiscaleDTM/issues

Depends terra (>= 1.7.46)

SystemRequirements C++17

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, raster, dplyr, shiny, rgl, stats, utils

Suggests knitr, rmarkdown, tmap, colorRamps, cowplot, magick

URL https://ailich.github.io/MultiscaleDTM/,
https://github.com/ailich/MultiscaleDTM

VignetteBuilder knitr

NeedsCompilation yes

Author Alexander Ilich [aut, cre] (<https://orcid.org/0000-0003-1758-8499>), Vincent Lecours [aut], Benjamin Misiuk [aut], Steven Murawski [aut]

Maintainer Alexander Ilich <ailich@usf.edu>

Repository CRAN

Date/Publication 2024-01-22 21:02:46 UTC

Contents

`AdjSD` *Calculates standard deviation of bathymetry (a measure of rugosity) adjusted for slope*

Description

Calculates standard deviation of bathymetry (a measure of rugosity). Using a sliding rectangular window a plane is fit to the data and the standard deviation of the residuals is calculated (Ilich et al., 2023)

Usage

```
AdjSD(
  r,
  w = c(3, 3),
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
```

```
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer in a projected coordinate system where map units match elevation/depth units
`w`	A vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number.
`na.rm`	A logical indicating whether or not to remove NA values before calculations
`include_scale`	logical indicating whether to append window size to the layer names (default = FALSE)
`filename`	character Output filename.
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Value

a SpatRaster or RasterLayer of adjusted rugosity

References

Ilich, A. R., Misiuk, B., Lecours, V., & Murawski, S. A. (2023). MultiscaleDTM: An open-source R package for multiscale geomorphometric analysis. Transactions in GIS, 27(4). https://doi.org/10.1111/tgis.13067

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
adjsd<- AdjSD(r, w=c(5,5), na.rm = TRUE)
plot(adjsd)
```

---

annulus_window *Creates annulus focal window*

Description

Creates annulus focal window around central pixel.

Usage

```
annulus_window(radius, unit, resolution)
```

Arguments

`radius`	radius of inner annulus c(inner,outer). Inner radius must be less than or equal to outer radius.
`unit`	unit for radius. Either "cell" (number of cells, the default) or "map" for map units (e.g. meters).
`resolution`	resolution of intended raster layer (one number or a vector of length 2). Only necessary if unit= "map"

Value

a matrix of 1's and NA's showing which cells to include and exclude respectively in focal calculations. It also contains attributes attributes 'unit', 'scale', and 'shape'.

BPI — *Calculates Bathymetric Position Index*

Description

Calculates Bathymetric Position Index (BPI). BPI is a measure of relative position that calculates the difference between the value of the focal cell and the mean of cells contained within an annulus shaped neighborhood. Positive values indicate local highs (i.e. peaks) and negative values indicate local lows (i.e. depressions). BPI can be expressed in units of the input DTM raster or can standardized relative to the local topography by dividing by the standard deviation or range of included elevation values in the focal window. BPI calls the function RelPos internally which serves as a general purpose and more flexible function for calculating relative position.

Usage

```
BPI(
  r,
  w,
  stand = "none",
  unit = "cell",
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	r DTM as a SpatRaster or RasterLayer.
`w`	Vector of length 2 specifying c(inner, outer) radii of the annulus in "cell" or "map" units or a focal weights matrix created by MultiscaleDTM::annulus_window. Inner radius must be less than or equal to outer radius. There is no default size.

stand	Standardization method. Either "none" (the default), "range" or "sd" indicating whether the relative position should be standardized by dividing by the standard deviation or range of included values in the focal window. If stand is 'none' the layer name will be "bpi", otherwise it will be "sbpi" to indicate that the layer has been standardized.
unit	Unit for w if it is a vector (default is unit="cell"). If w is a matrix, unit is ignored and extracted directly from w.
na.rm	Logical indicating whether or not to remove NA values before calculations.
include_scale	Logical indicating whether to append window size to the layer names (default = FALSE) or a character vector specifying the name you would like to append or a number specifying the number of significant digits. If include_scale = TRUE the appended scale will be the inner and outer radius. If unit="map" then window size will have "MU" after the number indicating that the number represents the scale in map units (note units can be extracted from w created with MultiscaleDTM::circle_window and MultiscaleDTM::annulus_window).
filename	Character output filename.
overwrite	Logical. If TRUE, filename is overwritten (default is FALSE).
wopt	List with named options for writing files as in writeRaster.

Value

A SpatRaster or RasterLayer.

References

Lundblad, E.R., Wright, D.J., Miller, J., Larkin, E.M., Rinehart, R., Naar, D.F., Donahue, B.T., Anderson, S.M., Battista, T., 2006. A benthic terrain classification scheme for American Samoa. Marine Geodesy 29, 89–111. https://doi.org/10.1080/01490410600738021

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
bpi<- BPI(r, w = c(2,4), stand= "none", unit = "cell", na.rm = TRUE)
plot(bpi)
```

---

circle_window *Creates circular focal window*

Description

Creates circular focal window around central pixel.

Usage

```
circle_window(radius, unit, resolution, return_dismat = FALSE)
```

Arguments

`radius`	radius of circular window
`unit`	unit for radius. Either "cell" (number of cells) or "map" for map units (e.g. meters).
`resolution`	resolution of intended raster layer (one number or a vector of length 2). Only necessary if unit= "map"
`return_dismat`	logical, if TRUE return a matrix of distances from focal cell instead of a matrix to pass to terra::focal.

Value

a matrix of 1's and NA's showing which cells to include and exclude respectively in focal calculations, or if return_dismat=TRUE, a matrix indicating the distance from the focal cell. It also contains attributes attributes 'unit', 'scale', and 'shape' if return_dismat=FALSE, and if return_dismat=TRUE the attribute 'unit'.

`classify_features_ff` *Helper function factory to classify morphometric features*

Description

Helper function factory to classify morphometric features according to a modified version of Wood 1996 page 120

Usage

```
classify_features_ff(slope_tolerance = 1, curvature_tolerance = 1e-04)
```

Arguments

`slope_tolerance`
: Slope tolerance that defines a 'flat' surface (degrees; default is 1.0). Relevant for the features layer.

`curvature_tolerance`
: Curvature tolerance that defines 'planar' surface (default is 0.0001). Relevant for the features layer.

Value

A function that can be passed to raster::overlay to classify morphometric features

References

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

DMV *Calculates Difference from Mean Value (DMV)*

Description

Calculates Difference from Mean Value (DMV). DMV is a measure of relative position that calculates the difference between the value of the focal cell and the mean of all cells in a rectangular or circular neighborhood. Positive values indicate local highs (i.e. peaks) and negative values indicate local lows (i.e. depressions). DMV can be expressed in units of the input DTM raster or can standardized relative to the local topography by dividing by the standard deviation or range of elevation values in the focal window. DMV calls the function RelPos internally which serves as a general purpose and more flexible function for calculating relative position.

Usage

```
DMV(
  r,
  w = dplyr::case_when(tolower(shape) == "rectangle" ~ 3, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "cell") ~ 1, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "map") ~ max(terra::res(r))),
  shape = "rectangle",
  stand = "none",
  unit = "cell",
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer.
`w`	For a "rectangle" focal window, a vector of length 2 containing odd numbers specifying dimensions where the first number is the number of rows and the second is the number of columns (or a single number if the number of rows and columns is equal). For a "circle" shaped focal window, a single integer representing the radius in "cell" or "map" units or a focal weights matrix created by MultiscaleDTM::circle_window.
`shape`	Character representing the shape of the focal window. Either "rectangle" (default) or "circle".
`stand`	Standardization method. Either "none" (the default), "range" or "sd" indicating whether the TPI should be standardized by dividing by the standard deviation or range of included values in the focal window. If stand is 'none' the layer name will be "dmv", otherwise it will be "sdmv" to indicate that the layer has been standardized.

unit
: Unit for w if shape is 'circle' and it is a vector (default is unit="cell"). For circular windows specified with a matrix, unit is ignored and extracted directly from w. For rectangular and custom focal windows set unit='cell' or set unit to NA/NULL.

na.rm
: Logical indicating whether or not to remove NA values before calculations.

include_scale
: Logical indicating whether to append window size to the layer names (default = FALSE) or a character vector specifying the name you would like to append or a number specifying the number of significant digits. If include_scale = TRUE the number of rows and number of columns will be appended for rectangular windows. For circular windows it will be a single number representing the radius. If unit="map" then window size will have "MU" after the number indicating that the number represents the scale in map units (note units can be extracted from w created with MultiscaleDTM::circle_window).

filename
: Character output filename.

overwrite
: Logical. If TRUE, filename is overwritten (default is FALSE).

wopt
: List with named options for writing files as in writeRaster.

Value

a SpatRaster or RasterLayer.

References

Lecours, V., Devillers, R., Simms, A.E., Lucieer, V.L., Brown, C.J., 2017. Towards a Framework for Terrain Attribute Selection in Environmental Studies. Environmental Modelling & Software 89, 19-30. https://doi.org/10.1016/j.envsoft.2016.11.027 Wilson, J.P., Gallant, J.C. (Eds.), 2000. Terrain Analysis: Principles and Applications. John Wiley & Sons, Inc.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
dmv<- DMV(r, w=c(5,5), shape= "rectangle", stand="range", na.rm = TRUE)
plot(dmv)
```

---

erupt *Create georeferenced version of R's built in volcano dataset*

Description

Create georeferenced version of R's built in volcano dataset. Useful dataset for generating quick examples.

Usage

```
erupt()
```

Value

SpatRaster

Examples

```
r<- erupt()
```

---

explore_terrain *Interactive Shiny app to look at terrain attributes*

Description

Interactive Shiny app to look at terrain attributes based on a surface fit using a Wood/Evans Quadratic Equation: Z =ax^2+by^2+cxy+dx+ey(+f)

Usage

```
explore_terrain()
```

Value

No return value, launches Shiny app.

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

---

kmax *Calculate max curvature*

Description

Calculate max curvature, kmax, from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
kmax(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

---

kmean *Calculate mean curvature*

Description

Calculate mean curvature, kmean, from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
kmean(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

kmin *Calculate min curvature*

Description

Calculate min curvature, kmin, from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
kmin(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

knc *Calculate normal contour curvature*

Description

Calculate normal contour curvature (kn)c, which is the principal representative of the plan curvature group based on regression coefficients from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
knc(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

---

kns *Calculate normal slope line curvature*

Description

Calculate normal slope line curvature (kn)s, which is the principal representative of the profile curvature group based on regression coefficients from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
kns(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

ku — *Calculate unsphericity curvature*

Description

Calculate unsphericity curvature, ku, from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
ku(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

outlier_filter — *Helper function to filter outliers from regression parameters using interquartile range*

Description

Helper function to filter outliers from regression parameters using interquartile range

Usage

```
outlier_filter(params, outlier_quantile, wopt = list())
```

Arguments

`params`	regression parameters for fitted surface
`outlier_quantile`	A numeric vector of length two or three. If two numbers are used it specifies the lower (Q1) and upper (Q2) quantiles used for determining the interquantile range (IQR). These values should be between 0 and 1 with Q2 > Q1. An optional third number can be used to specify a the size of a regular sample to be taken which can be useful if the full dataset is too large to fit in memory. Values are considered outliers if they are less than Q1-(100*IQR) or greater than Q2+(100*IQR), where IQR=Q2-Q1.
`wopt`	list with named options for writing files as in writeRaster

---

Qfit — *Calculates multiscale slope, aspect, curvature, and morphometric features using a local quadratic fit*

Description

Calculates multiscale slope, aspect, curvature, and morphometric features of a DTM over a sliding rectangular window using a local quadratic fit to the surface (Evans, 1980; Wood, 1996).

Usage

```
Qfit(
  r,
  w = c(3, 3),
  unit = "degrees",
 metrics = c("elev", "qslope", "qaspect", "qeastness", "qnorthness", "profc", "planc",
    "twistc", "meanc", "maxc", "minc", "features"),
  slope_tolerance = 1,
  curvature_tolerance = 1e-04,
  outlier_quantile = c(0.01, 0.99),
  na.rm = FALSE,
  force_center = FALSE,
  include_scale = FALSE,
  mask_aspect = TRUE,
  return_params = FALSE,
  as_derivs = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r` DTM as a SpatRaster (terra) or RasterLayer (raster) in a projected coordinate system where map units match elevation/depth units (up is assumed to be north for calculations of aspect, northness, and eastness).

`w` Vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number. Default is 3x3.

`unit` "degrees" or "radians".

`metrics` Character vector specifying which terrain attributes to return. The default is to return all available metrics, c("elev", "qslope", "qaspect", "qeastness", "qnorthness", "profc", "planc", "twistc", "meanc", "maxc", "minc", "features"). Slope, aspect, eastness, and northness are preceded with a 'q' to differentiate them from the measures calculated by SlpAsp() where the 'q' indicates that a quadratic surface was used for the calculation. 'elev' is the predicted elevation at the central cell (i.e. the intercept term of the regression) and is only relevant when force_center=FALSE. 'profc' is the profile curvature, 'planc' is the plan curvature, 'meanc' is the mean curvature, 'minc' is minimum curvature, and 'features' are morphometric features. See details.

`slope_tolerance`
Slope tolerance that defines a 'flat' surface (degrees; default = 1.0). Relevant for the features layer.

`curvature_tolerance`
Curvature tolerance that defines 'planar' surface (default = 0.0001). Relevant for the features layer.

`outlier_quantile`
A numeric vector of length two or three. If two numbers are used it specifies the lower (Q1) and upper (Q2) quantiles used for determining the interquantile range (IQR). These values should be between 0 and 1 with Q2 > Q1. An optional third number can be used to specify a the size of a regular sample to be taken which can be useful if the full dataset is too large to fit in memory. Values are considered outliers and replaced with NA if they are less than Q1-(100*IQR) or greater than Q2+(100*IQR), where IQR=Q2-Q1. The outlier filter is performed on the results of the regression parameters ('a'-'e' and 'elev') prior to calculation of subsequent terrain attributes. Note that c(0,1) will skip the outlier filtering step and can speed up computations. The default is c(0.01,0.99).

`na.rm` Logical indicating whether or not to remove NA values before calculations.

`force_center` Logical specifying whether the constrain the model through the central cell of the focal window

`include_scale` Logical indicating whether to append window size to the layer names (default = FALSE).

`mask_aspect` Logical. If TRUE (default), aspect will be set to NA and northness and eastness will be set to 0 when slope = 0. If FALSE, aspect is set to 270 degrees or 3*pi/2 radians ((-pi/2)- atan2(0,0)+2*pi) and northness and eastness will be calculated from this.

`return_params` Logical indicating whether to return Wood/Evans regression parameters (default = FALSE).

`as_derivs`	Logical indicating whether parameters should be formatted as partial derivatives instead of regression coefficients (default = FALSE) (Minár et al., 2020).
`filename`	character Output filename. Can be a single filename, or as many filenames as there are layers to write a file for each layer
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Details

This function calculates slope, aspect, eastness, northness, profile curvature, plan curvature, mean curvature, twisting curvature, maximum curvature, minimum curvature, morphometric features, and a smoothed version of the elevation surface using a quadratic surface fit from Z = aX^2+bY^2+cXY+dX+eY+f, where Z is the elevation or depth values, X and Y are the xy coordinates relative to the central cell in the focal window, and a-f are parameters to be estimated (Evans, 1980; Minár et al. 2020; Wood, 1996). For aspect, 0 degrees represents north (or if rotated, the direction that increases as you go up rows in your data) and increases clockwise. For calculations of northness (cos(asp)) and eastness (sin(asp)), up in the y direction is assumed to be north, and if this is not true for your data (e.g. you are using a rotated coordinate system), you must adjust accordingly. All curvature formulas are adapted from Minár et al 2020. Therefore all curvatures are measured in units of 1/length (e.g. m^-1) except twisting curvature which is measured in radians/length (i.e. change in angle per unit distance), and we adopt a geographic sign convention where convex is positive and concave is negative (i.e., hills are considered convex with positive. Naming convention for curvatures is not consistent across the literature, however Minár et al (2020) has suggested a framework in which the reported measures of curvature translate to profile curvature = (kn)s, plan curvature = (kn)c, twisting curvature (Tg)c, mean curvature = kmean, maximum curvature = kmax, minimum curvature = kmin. For morphometric features cross-sectional curvature (zcc) was replaced by planc (kn)c, z''min was replaced by kmax, and z''max was replaced by kmin as these are more robust ways to measures the same types of curvature (Minár et al., 2020). Additionally, the planar feature from Wood (1996) was split into planar flat and slope depending on whether the slope threshold is exceeded or not.

Value

a SpatRaster (terra) or RasterStack/RasterLayer (raster)

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

Wilson, M.F., O'Connell, B., Brown, C., Guinan, J.C., Grehan, A.J., 2007. Multiscale Terrain Analysis of Multibeam Bathymetry Data for Habitat Mapping on the Continental Slope. Marine Geodesy 30, 3-35. https://doi.org/10.1080/01490410701295962

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
qmetrics<- Qfit(r, w = c(5,5), unit = "degrees", na.rm = TRUE)
plot(qmetrics)

# To get only the regression coefficients, set "metrics=c()" and "return_params=TRUE"
reg_coefs<- Qfit(r, w = c(5,5), metrics=c(), unit = "degrees", na.rm = TRUE, return_params=TRUE)
plot(reg_coefs)
```

RelPos *Calculates Relative Position of a focal cell*

Description

Calculates the relative position of a focal cell, which represents whether an area is a local high or low. Relative position is the value of the focal cell minus the value of a reference elevation (often the mean of included values in the focal window but see "fun" argument). Positive values indicate local highs (i.e. peaks) and negative values indicate local lows (i.e. depressions). Relative Position can be expressed in units of the input DTM raster or can standardized relative to the local topography by dividing by the standard deviation or range of included elevation values in the focal window.

Usage

```
RelPos(
  r,
  w = dplyr::case_when(tolower(shape) == "rectangle" ~ 3, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "cell") ~ 1, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "map") ~ max(terra::res(r))),
  shape = "rectangle",
  stand = "none",
  exclude_center = FALSE,
  unit = "cell",
  fun = "mean",
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

r	DTM as a SpatRaster or RasterLayer.

- `w` For a "rectangle" focal window, a vector of length 2 containing odd numbers specifying dimensions where the first number is the number of rows and the second is the number of columns (or a single number if the number of rows and columns is equal). For a "circle" shaped focal window, a single integer representing the radius in "cell" or "map" units or a focal weights matrix created by MultiscaleDTM::circle_window. The default radius is 1 cell if unit= "cell" or the maximum of the x and y cell resolution if unit="map". For an "annulus" shaped focal window, a vector of length 2 specifying c(inner, outer) radii of the annulus in "cell" or "map" units or a focal weights matrix created by MultiscaleDTM::annulus_window. Inner radius must be less than or equal to outer radius. There is no default size for an annulus window. If a "custom" focal window shape is used, w must be a focal weights matrix with 1's for included values and NAs for excluded values.
- `shape` Character representing the shape of the focal window. Either "rectangle" (default), "circle", or "annulus", or "custom". If a "custom" shape is used, w must be a focal weights matrix.
- `stand` Standardization method. Either "none" (the default), "range" or "sd" indicating whether the relative position should be standardized by dividing by the standard deviation or range of included values in the focal window. If stand is 'none' the layer name will be "rpos", otherwise it will be "srpos" to indicate that the layer has been standardized.
- `exclude_center` Logical indicating whether to exclude the central value from focal calculations (Default=FALSE). Use FALSE for DMV and TRUE for TPI. Note, if a focal weights matrix is supplied to w, setting exclude_center=TRUE will overwrite the center value of w to NA, but setting exclude_center=FALSE will not overwrite the central value to be 1.
- `unit` Unit for w if shape is 'circle' or 'annulus' and it is a vector (default is unit="cell"). For circular and annulus shaped windows specified with a matrix, unit is ignored and extracted directly from w. For rectangular and custom focal windows set unit='cell' or set unit to NA/NULL.
- `fun` Function to apply to included values to determine the reference elevation. Accepted values are "mean","median", "min", and "max". The default is "mean"
- `na.rm` Logical indicating whether or not to remove NA values before calculations.
- `include_scale` Logical indicating whether to append window size to the layer names (default = FALSE) or a character vector specifying the name you would like to append or a number specifying the number of significant digits. If include_scale = TRUE the number of rows and number of columns will be appended for rectangular or custom windows. For circular windows it will be a single number representing the radius. For annulus windows it will be the inner and outer radius. If unit="map" then window size will have "MU" after the number indicating that the number represents the scale in map units (note units can be extracted from w created with MultiscaleDTM::circle_window and MultiscaleDTM::annulus_window).
- `filename` Character output filename.
- `overwrite` Logical. If TRUE, filename is overwritten (default is FALSE).
- `wopt` List with named options for writing files as in writeRaster.

Value

A SpatRaster or RasterLayer.

References

Lecours, V., Devillers, R., Simms, A.E., Lucieer, V.L., Brown, C.J., 2017. Towards a Framework for Terrain Attribute Selection in Environmental Studies. Environmental Modelling & Software 89, 19-30. https://doi.org/10.1016/j.envsoft.2016.11.027

Lundblad, E.R., Wright, D.J., Miller, J., Larkin, E.M., Rinehart, R., Naar, D.F., Donahue, B.T., Anderson, S.M., Battista, T., 2006. A benthic terrain classification scheme for American Samoa. Marine Geodesy 29, 89–111. https://doi.org/10.1080/01490410600738021

Weiss, A., 2001. Topographic Position and Landforms Analysis. Presented at the ESRI user conference, San Diego, CA.

Wilson, J.P., Gallant, J.C. (Eds.), 2000. Terrain Analysis: Principles and Applications. John Wiley & Sons, Inc.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
rpos<- RelPos(r, w = c(5,5), shape= "rectangle", exclude_center = TRUE, na.rm = TRUE)
plot(rpos)
```

RIE *Calculates Roughness Index-Elevation*

Description

Calculates Roughness Index-Elevation. This is the standard deviation of residual topography in a focal window where residual topography is calculated as the focal pixel minus the focal mean.

Usage

```
RIE(
  r,
  w = c(3, 3),
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer
`w`	A vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number. Default is 3x3.
`na.rm`	A logical indicating whether or not to remove NA values before calculation of SD
`include_scale`	logical indicating whether to append window size to the layer names (default = FALSE)
`filename`	character Output filename.
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Details

Note the original paper by Cavalli et al (2008) uses a fixed 5x5 window and uses 25 as the denominator indicating use of the population standard deviation. This implementation provides a flexible window size and istead calculates the sample standard deviation which uses a denominator of n-1.

Value

a SpatRaster or RasterLayer

References

Cavalli, M., Tarolli, P., Marchi, L., Dalla Fontana, G., 2008. The effectiveness of airborne LiDAR data in the recognition of channel-bed morphology. CATENA 73, 249–260. https://doi.org/10.1016/j.catena.2007.11.001

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
rie<- RIE(r, w=c(5,5), na.rm = TRUE)
plot(rie)
```

SAPA — *Calculates surface area to planar area rugosity*

Description

Calculates surface area (Jenness, 2004) to planar area rugosity and by default corrects planar area for slope using the arc-chord ratio (Du Preez, 2015). Additionally, the method has been modified to allow for calculations at multiple different window sizes (see details and Ilich et al. (2023)).

Usage

```
SAPA(
  r = NULL,
  w = 1,
  slope_correction = TRUE,
  na.rm = FALSE,
  include_scale = FALSE,
  slope_layer = NULL,
  sa_layer = NULL,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer in a projected coordinate system where map units match elevation/depth units (Not used if both slope_layer and sa_layer are specified).
`w`	A single number or a vector of length 2 (row, column) specifying the dimensions of the rectangular window over which surface area will be summed. Window size must be an odd number. 1 refers to "native" scale and surface area and planar area will be calculated per cell (the traditional implementation).
`slope_correction`	Whether to use the arc-chord ratio to correct planar area for slope (default is TRUE)
`na.rm`	logical indicating whether to remove/account for NAs in calculations. If FALSE any calculations involving NA will be NA. If TRUE, NA values will be removed and accounted for.
`include_scale`	logical indicating whether to append window size to the layer names (default = FALSE)
`slope_layer`	Optionally specify an appropriate slope layer IN RADIANS to use. If not supplied, it will be calculated using the SlpAsp function based on Misiuk et al (2021). The slope layer should have a window size that is 2 larger than the w specified for SAPA.
`sa_layer`	Optionally specify a surface area raster that contains the surface area on a per cell level. This can be calculated with the SurfaceArea function. If calculating SAPA at multiple scales it will be more efficient to supply this so that it does not need to be calculated every time.
`filename`	character Output filename.
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Details

Planar area is calculated as the x_dis * y_dis if uncorrected for slope and (x_dis * y_dis)/cos(slope) if corrected for slope. When w=1, this is called "native" scale and is equivalent to what is presented in Du Preez (2015) and available in the ArcGIS Benthic Terrain Modeller add-on. In this case operations are performed on a per cell basis where x_dis is the resolution of the raster in the x direction (left/right) and y_dis is the resolution of the raster in the y direction (up/down) and slope is calculated using the Horn (1981) method. To expand this to multiple scales of analysis, at w > 1 slope is calculated based on Misiuk et al (2021) which provides a modification of the Horn method to extend the matric to multiple spatial scales. Planar area is calculated the same way as for w=1 except that now x_dis is the x resolution of the raster * the number of columns in the focal window, and y_dis is y resolution of the raster * the number of rows. For w > 1, surface area is calculated as the sum of surface areas within the focal window. Although the (modified) Horn slope is used by default to be consistent with Du Preez (2015), slope calculated using a different algorithm (e.g. Wood 1996) could be supplied using the slope_layer argument. Additionally, a slope raster can be supplied if you have already calculated it and do not wish to recalculate it. However, be careful to supply a slope layer measured in radians and calculated at the relevant scale (2 larger than the w of SAPA).

Value

a SpatRaster or RasterLayer

References

Du Preez, C., 2015. A new arc–chord ratio (ACR) rugosity index for quantifying three-dimensional landscape structural complexity. Landscape Ecol 30, 181–192. https://doi.org/10.1007/s10980-014-0118-8

Horn, B.K., 1981. Hill Shading and the Reflectance Map. Proceedings of the IEEE 69, 14-47.

Ilich, A. R., Misiuk, B., Lecours, V., & Murawski, S. A. (2023). MultiscaleDTM: An open-source R package for multiscale geomorphometric analysis. Transactions in GIS, 27(4). https://doi.org/10.1111/tgis.13067

Jenness, J.S., 2004. Calculating landscape surface area from digital elevation models. Wildlife Society Bulletin 32, 829-839.

Misiuk, B., Lecours, V., Dolan, M.F.J., Robert, K., 2021. Evaluating the Suitability of Multi-Scale Terrain Attribute Calculation Approaches for Seabed Mapping Applications. Marine Geodesy 44, 327-385. https://doi.org/10.1080/01490419.2021.1925789

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
sapa<- SAPA(r, w=c(5,5), slope_correction = TRUE)
plot(sapa)
```

`SlpAsp` *Multiscale Slope and Aspect*

Description

Calculates multiscale slope and aspect based on the slope.k/aspect.k algorithm from Misiuk et al (2021) which extends classical formulations of slope restricted to a 3x3 window to multiple scales. The code from Misiuk et al (2021) was modified to allow for rectangular rather than only square windows.

Usage

```
SlpAsp(
  r,
  w = c(3, 3),
  unit = "degrees",
  method = "queen",
  metrics = c("slope", "aspect", "eastness", "northness"),
  na.rm = FALSE,
  include_scale = FALSE,
  mask_aspect = TRUE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer in a projected coordinate system where map units match elevation/depth units
`w`	A vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number.
`unit`	"degrees" or "radians"
`method`	"queen" or "rook", indicating how many neighboring cells to use to compute slope for any cell. queen uses 8 neighbors (up, down, left, right, and diagonals) and rook uses 4 (up, down, left, right). Alternatively, instead of "queen" or "rook", method can be specified as 8 and 4 respectively.
`metrics`	a character string or vector of character strings of which terrain attributes to return. Default is to return all available attributes which are c("slope", "aspect", "eastness", "northness").
`na.rm`	Logical indicating whether or not to remove NA values before calculations. Only applicable if method is "queen" or "8".
`include_scale`	logical indicating whether to append window size to the layer names (default = FALSE)
`mask_aspect`	A logical. When mask_aspect is TRUE (the default), if slope evaluates to 0, aspect will be set to NA and both eastness and northness will be set to 0. When mask_aspect is FALSE, when slope is 0 aspect will be pi/2 radians or 90 degrees which is the behavior of raster::terrain, and northness and eastness will be calculated from that.

filename	character Output filename. Can be a single filename, or as many filenames as there are layers to write a file for each layer
overwrite	logical. If TRUE, filename is overwritten (default is FALSE).
wopt	list with named options for writing files as in writeRaster

Details

When method="rook", slope and aspect are computed according to Fleming and Hoffer (1979) and Ritter (1987). When method="queen", slope and aspect are computed according to Horn (1981). These are the standard slope algorithms found in many GIS packages but are traditionally restricted to a 3 x 3 window size. Misiuk et al (2021) extended these classical formulations to multiple window sizes. This function modifies the code from Misiuk et al (2021) to allow for rectangular rather than only square windows and also added aspect.

Value

a SpatRaster or RasterStack of slope and/or aspect (and components of aspect)

References

Fleming, M.D., Hoffer, R.M., 1979. Machine processing of landsat MSS data and DMA topographic data for forest cover type mapping (No. LARS Technical Report 062879). Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.

Horn, B.K., 1981. Hill Shading and the Reflectance Map. Proceedings of the IEEE 69, 14-47.

Misiuk, B., Lecours, V., Dolan, M.F.J., Robert, K., 2021. Evaluating the Suitability of Multi-Scale Terrain Attribute Calculation Approaches for Seabed Mapping Applications. Marine Geodesy 44, 327-385. https://doi.org/10.1080/01490419.2021.1925789

Ritter, P., 1987. A vector-based slope and aspect generation algorithm. Photogrammetric Engineering and Remote Sensing 53, 1109-1111.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
slp_asp<- SlpAsp(r = r, w = c(5,5), unit = "degrees",
method = "queen", metrics = c("slope", "aspect",
"eastness", "northness"))
plot(slp_asp)
```

`SurfaceArea` *Calculates surface area of a DTM*

Description

Calculates surface area on a per cell basis of a DTM based on Jenness, 2004.

Usage

```
SurfaceArea(
  r,
  na.rm = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer in a projected coordinate system where map units match elevation/depth units
`na.rm`	Logical indicating whether to remove NAs from calculations. When FALSE, the sum of the eight triangles is calculated. When TRUE, the mean of the created triangles is calculated and multiplied by 8 to scale it to the proper area.
`filename`	character Output filename.
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Value

a SpatRaster or RasterLayer

References

Jenness, J.S., 2004. Calculating landscape surface area from digital elevation models. Wildlife Society Bulletin 32, 829-839.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
sa<- SurfaceArea(r)
plot(sa)
```

tgc *Calculate contour geodesic torsion*

Description

Calculate contour geodesic torsion (tg)c, which is the principal representative of the twisting curvature group based on regression coefficients from the equation Z =ax^2+by^2+cxy+dx+ey(+f).

Usage

```
tgc(a, b, c, d, e)
```

Arguments

a	regression coefficient
b	regression coefficient
c	regression coefficient
d	regression coefficient
e	regression coefficient

References

Evans, I.S., 1980. An integrated system of terrain analysis and slope mapping. Zeitschrift f¨ur Geomorphologic Suppl-Bd 36, 274–295.

Wood, J., 1996. The geomorphological characterisation of digital elevation models (Ph.D.). University of Leicester.

Minár, J., Evans, I.S., Jenčo, M., 2020. A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modelling and prediction. Earth-Science Reviews 211, 103414. https://doi.org/10.1016/j.earscirev.2020.103414

TPI *Calculates Topographic Position Index*

Description

Calculates Topographic Position Index (TPI). TPI is a measure of relative position that calculates the the difference between the value of the focal cell and the mean of mean of the surrounding cells (i.e. local mean but excluding the value of the focal cell).Positive values indicate local highs (i.e. peaks) and negative values indicate local lows (i.e. depressions). TPI can be expressed in units of the input DTM raster or can standardized relative to the local topography by dividing by the standard deviation or range of included elevation values in the focal window.

Usage

```
TPI(
  r,
  w = dplyr::case_when(tolower(shape) == "rectangle" ~ 3, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "cell") ~ 1, tolower(shape) == "circle" &
    isTRUE(tolower(unit) == "map") ~ max(terra::res(r))),
  shape = "rectangle",
  stand = "none",
  unit = "cell",
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer. TPI calls the function RelPos internally which serves as a general purpose and more flexible function for calculating relative position.
`w`	For a "rectangle" focal window, a vector of length 2 containing odd numbers specifying dimensions where the first number is the number of rows and the second is the number of columns (or a single number if the number of rows and columns is equal). For a "circle" shaped focal window, a single integer representing the radius in "cell" or "map" units or a focal weights matrix created by MultiscaleDTM::circle_window.
`shape`	Character representing the shape of the focal window. Either "rectangle" (default) or "circle".
`stand`	Standardization method. Either "none" (the default), "range" or "sd" indicating whether the TPI should be standardized by dividing by the standard deviation or range of included values in the focal window. If stand is 'none' the layer name will be "tpi", otherwise it will be "stpi" to indicate that the layer has been standardized.
`unit`	Unit for w if shape is 'circle' and it is a vector (default is unit="cell"). For circular windows specified with a matrix, unit is ignored and extracted directly from w. For rectangular and custom focal windows set unit='cell' or set unit to NA/NULL.
`na.rm`	Logical indicating whether or not to remove NA values before calculations.
`include_scale`	Logical indicating whether to append window size to the layer names (default = FALSE) or a character vector specifying the name you would like to append or a number specifying the number of significant digits. If include_scale = TRUE the number of rows and number of columns will be appended for rectangular windows. For circular windows it will be a single number representing the radius. If unit="map" then window size will have "MU" after the number indicating that the number represents the scale in map units (note units can be extracted from w created with MultiscaleDTM::circle_window).

filename	Character output filename.
overwrite	Logical. If TRUE, filename is overwritten (default is FALSE).
wopt	List with named options for writing files as in writeRaster.

Value

SpatRaster or RasterLayer.

References

Weiss, A., 2001. Topographic Position and Landforms Analysis. Presented at the ESRI user conference, San Diego, CA.

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
tpi<- TPI(r, w=c(5,5), shape="rectangle", stand="none", na.rm = TRUE)
plot(tpi)
```

VRM — *Implementation of the Sappington et al., (2007) vector ruggedness measure*

Description

Implementation of the Sappington et al., (2007) vector ruggedness measure, modified from Evans (2021).

Usage

```
VRM(
  r,
  w = c(3, 3),
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`r`	DTM as a SpatRaster or RasterLayer
`w`	A vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number. Default is 3x3.
`na.rm`	A logical indicating whether or not to remove NA values before calculations. See details for more information.
`include_scale`	logical indicating whether to append window size to the layer names (default = FALSE)
`filename`	character Output filename.
`overwrite`	logical. If TRUE, filename is overwritten (default is FALSE).
`wopt`	list with named options for writing files as in writeRaster

Details

If the crs is cartesian, when na.rm=TRUE, NA's will be removed from the slope/aspect calculations. When the crs is lat/lon, na.rm=TRUE will not affect the calculation of slope/aspect as terra::terrain will be used since it can calculate slope and aspect for spherical geometry but it does not support na.rm. In both cases when na.rm=TRUE, the x, y, and z components will be summed with na.rm=TRUE, and the N used in the denominator of the VRM equation will be the number of non-NA cells in the window rather than the total number of cells.

Value

a RasterLayer

References

Evans JS (2021). spatialEco. R package version 1.3-6, https://github.com/jeffreyevans/spatialEco.

Sappington, J.M., Longshore, K.M., Thompson, D.B., 2007. Quantifying Landscape Ruggedness for Animal Habitat Analysis: A Case Study Using Bighorn Sheep in the Mojave Desert. The Journal of Wildlife Management 71, 1419-1426. https://doi.org/10.2193/2005-723

Examples

```
r<- rast(volcano, extent= ext(2667400, 2667400 +
ncol(volcano)*10, 6478700, 6478700 + nrow(volcano)*10),
crs = "EPSG:27200")
vrm<- VRM(r, w=c(5,5), na.rm = TRUE)
plot(vrm)
```

Index