Package ‘movecost’

June 5, 2020

Title Calculation of Accumulated Cost Surface and Least-Cost Paths Related to Human Movement Across the Landscape

Version 0.5

Description Provides the facility to calculate non-isotropic accumulated cost surface and least-cost paths using a number of human-movement-related cost functions that can be selected by the user. It just requires a Digital Terrain Model, a start location and (optionally) destination locations. See Alberti (2019) <doi:10.1016/j.softx.2019.100331>.

Depends R (>= 3.4.0)

Imports gdistance (>= 1.2-2), raster (>= 2.8-4), rgdal (>= 1.3-6), rgeos (>= 0.4-2), sp (>= 1.3-1)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Gianmarco Alberti [aut, cre]

Maintainer Gianmarco Alberti <gianmarcoalberti@gmail.com>

Repository CRAN

Date/Publication 2020-06-05 15:10:03 UTC

R topics documented:

`destin.loc` *Dataset: locations on the volcano Maunga Whau (Auckland, New Zealand)*

Description

A SpatialPointsDataFrame representing spots on the volcano Maunga Whau (Auckland, New Zealand), to be used as destination locations for least-cost paths calculation.

Usage

```
data(destin.loc)
```

Format

SpatialPointsDataFrame

`movecost` *R function for calculating accumulated anisotropic cost of movement across the terrain and least-cost paths from an origin*

Description

The function provides the facility to calculate the anisotropic accumulated cost of movement around a starting location and to optionally calculate least-cost path(s) toward one or multiple destinations. It implements different cost estimations related to human movement across the landscape. The function takes as input a Digital Terrain Model ('RasterLayer' class) and a point feature ('SpatialPointsDataFrame' class), the latter representing the starting location, i.e. the location from which the accumulated cost is calculated.

Usage

```
movecost(
  dtm,
  origin,
  destin = NULL,
  funct = "t",
  time = "h",
  outp = "r",
  move = 16,
  sl.crit = 10,
  W = 70,
  L = 0,
  N = 1,
```

```
  V = 1.2,
  breaks = NULL,
  cont.lab = TRUE,
  destin.lab = TRUE,
  cex.breaks = 0.6,
  cex.lcp.lab = 0.6,
  oneplot = TRUE,
  export = FALSE
)
```

Arguments

`dtm`	digital terrain model (RasterLayer class).
`origin`	location from which the walking time is computed (SpatialPointsDataFrame class).
`destin`	location(s) to which least-cost path(s) is calculated (SpatialPointsDataFrame class).
`funct`	cost function to be used: **t** (default) uses the on-path Tobler's hiking function; **tofp** uses the off-path Tobler's hiking function; **mt** uses the Marquez-Perez et al.'s modified Tobler's function; **icmonp** uses the Irmischer-Clarke's modified Tobler's hiking function (male, on-path); **icmoffp** uses the Irmischer-Clarke's modified Tobler's hiking function (male, off-path); **icfonp** uses the Irmischer-Clarke's modified Tobler's hiking function (female, on-path); **icfoffp** uses the Irmischer-Clarke's modified Tobler's hiking function (female, off-path); **ug** uses the Uriarte Gonzalez's slope-dependant walking-time cost function; **alb** uses the Alberti's Tobler hiking function modified for animal foraging excursions; **ree** uses the relative energetic expenditure cost function; **hrz** uses the Herzog's metabolic cost function; **wcs** uses the wheeled-vehicle critical slope cost function; **p** uses the Pandolf et al.'s metabolic energy expenditure cost function; **vl** uses the Van Leusen's metabolic energy expenditure cost function; **ls** uses the Llobera-Sluckin's metabolic energy expenditure cost function (see Details).
`time`	time-unit expressed by the accumulated raster and by the isolines if Tobler's and Tobler-related cost functions are used; 'h' for hour, 'm' for minutes.
`outp`	type of output: 'raster' or 'contours' (see Details).
`move`	number of directions in which cells are connected: 4 (rook's case), 8 (queen's case), 16 (knight and one-cell queen moves; default).
`sl.crit`	critical slope (in percent), typically in the range 8-16 (10 by default) (used by the wheeled-vehicle cost function; see Details).

`W`	walker's body weight (in Kg; 70 by default; used by the Pandolf's and Van Leusen's cost function; see Details).
`L`	carried load weight (in Kg; 0 by default; used by the Pandolf's and Van Leusen's cost function; see Details).
`N`	coefficient representing ease of movement (1 by default) (used by the Pandolf's and Van Leusen's cost function; see Details).
`V`	speed in m/s (1.2 by default) (used by the Pandolf's and Van Leusen's cost function; see Details).
`breaks`	isolines interval; if no value is supplied, the interval is set by default to 1/10 of the range of values of the accumulated cost surface.
`cont.lab`	if set to TRUE (default) display the labels of the contours over the accumulated cost surface.
`destin.lab`	if set to TRUE (default) display the label(s) indicating the cost at the destination location(s).
`cex.breaks`	set the size of the time labels used in the isochrones plot (0.6 by default).
`cex.lcp.lab`	set the size of the labels used in least-cost path(s) plot (0.6 by default).
`oneplot`	TRUE (default) or FALSE if the user wants or does not want the plots displayed in a single window.
`export`	TRUE or FALSE (default) if the user wants or does not want the outputs to be exported; if TRUE, the accumulated cost surface will be exported as a GeoTiff file, while the isolines and the least-cost path(s) will be exported as shapefile; all the exported files will bear a suffix corresponding to the cost function selected by the user.

Details

If the parameter 'destin' is fed with a dataset representing destination location(s) ('SpatialPoints-DataFrame' class), the function also calculate least-cost path(s) plotted on the input DTM; the length of each path will be saved under the variable 'length' stored in the 'LCPs' dataset ('SpatialLines' class) returned by the function. The red dot(s) representing the destination location(s) will be labelled with numeric values representing the cost value at the location(s). The cost value will be also appended to the updated destination dataset returned by the function and storing a new variable named 'cost'.

The function builds on functions out of Jacob van Etten's 'gdistance' package. Under the hood, movecost() calculates the slope as rise over run, following the procedure described by van Etten, "R Package gdistance: Distances and Routes on Geographical Grids" in Journal of Statistical Software 76(13), 2017, pp. 14-15. The number of directions in which cells are connected in the cost calculation can be set to 4 (rook's case), 8 (queen's case), or 16 (knight and one-cell queen moves) using the 'move' parameter (see 'Arguments').

Besides citing this package, you may want to refer to: **Alberti (2019) <doi:10.1016/j.softx.2019.100331>**.

The following cost functions are implemented (**x[adj]** stands for slope as rise/run calculated for adjacent cells):

Tobler's hiking function (on-path) (speed in kmh):

$6 * exp(-3.5 * abs(x[adj] + 0.05))$

Tobler's hiking function (off-path) (speed in kmh):

$(6 * exp(-3.5 * abs(x[adj] + 0.05))) * 0.6$

as per Tobler's indication, the off-path walking speed is reduced by 0.6.

Marquez-Perez et al.'s modified Tobler hiking function (speed in kmh):

$4.8 * exp(-5.3 * abs((x[adj] * 0.7) + 0.03))$

modified version of the Tobler's hiking function as proposed by Joaquin Marquez-Parez, Ismael Vallejo-Villalta & Jose I. Alvarez-Francoso (2017), "Estimated travel time for walking trails in natural areas", Geografisk Tidsskrift-Danish Journal of Geography, 117:1, 53-62, DOI: 10.1080/00167223.2017.1316212.

Irmischer-Clarke's modified Tobler hiking function (male, on-path):

$(0.11 + exp(-(abs(x[adj]) * 100 + 5)^2/(2 * 30)^2)) * 3.6$

modified version of the Tobler's function as proposed for (male) on-path hiking by Irmischer, I. J., & Clarke, K. C. (2018). Measuring and modeling the speed of human navigation. Cartography and Geographic Information Science, 45(2), 177-186. https://doi.org/10.1080/15230406.2017.1292150. **Note**: all the all the Irmischer-Clarke's functions are originally express speed in m/s; they have been reshaped (multiplied by 3.6) to turn m/s into km/h for consistency with the other Tobler-related cost functions; slope is in percent.

Irmischer-Clarke's modified Tobler hiking function (male, off-path):

$(0.11 + 0.67 * exp(-(abs(x[adj]) * 100 + 2)^2/(2 * 30)^2)) * 3.6$

Irmischer-Clarke's modified Tobler hiking function (female, on-path):

$(0.95 * (0.11 + exp(-(abs(x[adj]) * 100 + 5)^2/(2 * 30^2)))) * 3.6$

Irmischer-Clarke's modified Tobler hiking function (female, off-path):

$(0.95 * (0.11 + 0.67 * exp(-(abs(x[adj]) * 100 + 2)^2/(2 * 30^2)))) * 3.6$

Uriarte Gonzalez's slope-dependant walking-time cost function:

$1/(0.0277 * (abs(x[adj]) * 100) + 0.6115)$

proposed by Uriarte Gonzalez; **see**: Chapa Brunet, T., Garcia, J., Mayoral Herrera, V., & Uriarte Gonzalez, A. (2008). GIS landscape models for the study of preindustrial settlement patterns in Mediterranean areas. In Geoinformation Technologies for Geo-Cultural Landscapes (pp. 255-273). CRC Press. https://doi.org/10.1201/9780203881613.ch12.
The cost function is originally expressed in seconds; for the purpose of its implementation in this function, it is the reciprocal of time (1/T) that is used in order to eventually get T/1. Unlike the original cost function, here the pixel resolution is not taken into account since 'gdistance' takes care of the cells' dimension when calculating accumulated costs.

Alberti's Tobler hiking function modified for animal foraging excursions:

$(6 * exp(-3.5 * abs(x[adj] + 0.05))) * 0.25$

proposed by Gianmarco Alberti; **see**: Locating potential pastoral foraging routes in Malta through the use of Geographic Information System (in press).
The Tobler's function has been rescaled to fit animal walking speed during foraging excursions. The distribution of the latter, as empirical data show, turns out to be right-skewed and to vary along a continuum. It ranges from very low speed values (corresponding to slow grazing activities grazing while walking) to comparatively higher values (up to about 4.0 km/h) corresponding to travels without grazing (directional travel toward feeding stations). The function consider 1.5 km/h as the average flock speed, which roughly corresponds to the average speed recorded in some studies, and can be considered the typical speed of flocks during excursions in which grazing takes place while walking (typical form of grazing in most situations). Tobler's hiking function has been rescaled by a factor of 0.25 to represent the walking pace of a flock instead of humans. The factor corresponds to the ratio between the flock average speed (1.5 km/h) and the maximum human walking speed (about 6.0 km/h) on a favourable slope.

Relative energetic expenditure cost function:

$1/(tan((atan(abs(x[adj])) * 180/pi) * pi/180)/tan(1 * pi/180))$

slope-based cost function expressing change in potential energy expenditure; **see** Conolly, J., & Lake, M. (2006). Geographic Information Systems in Archaeology. Cambridge: Cambridge University Press, p. 220; **see also** Newhard, J. M. L., Levine, N. S., & Phebus, A. D. (2014). The development of integrated terrestrial and marine pathways in the Argo-Saronic region, Greece. Cartography and Geographic Information Science, 41(4), 379-390, with references to studies that use this function; **see also** ten Bruggencate, R. E., Stup, J. P., Milne, S. B., Stenton, D. R., Park, R. W., & Fayek, M. (2016). A human-centered GIS approach to modeling mobility on southern Baffin Island, Nunavut, Canada. Journal of Field Archaeology, 41(6), 684-698. https://doi.org/10.1080/00934690.2016.1234897.

Herzog's metabolic cost function in J/(kg*m):

$1/((1337.8 * abs(x[adj])^6) + (278.19 * abs(x[adj])^5) - (517.39 * abs(x[adj])^4) - (78.199 * abs(x[adj])^3) + (93.419 * abs(x[adj])^2) + (19.825 * abs(x[adj])) + 1.64)$

see Herzog, I. (2016). Potential and Limits of Optimal Path Analysis. In A. Bevan & M. Lake (Eds.), Computational Approaches to Archaeological Spaces (pp. 179-211). New York: Routledge.

Wheeled-vehicle critical slope cost function:

$$1/(1 + ((abs(x[adj]) * 100)/sl.crit)^2)$$

where $sl.crit$ (=critical slope, in percent) is "the transition where switchbacks become more effective than direct uphill or downhill paths" and typically is in the range 8-16; **see** Herzog, I. (2016). Potential and Limits of Optimal Path Analysis. In A. Bevan & M. Lake (Eds.), Computational Approaches to Archaeological Spaces (pp. 179-211). New York: Routledge.

Pandolf et al.'s metabolic energy expenditure cost function (in Watts):

$$1/(1.5*W+2.0*(W+L)*(L/W)^2+N*(W+L)*(1.5*V^2+0.35*V*(abs(x[adj])*100)))$$

where W is the walker's body weight (Kg), L is the carried load (in Kg), V is the velocity in m/s, N is a coefficient representing ease of movement on the terrain.

As for the latter, suggested values available in literature are: Asphalt/blacktop=1.0; Dirt road=1.1; Grass=1.1; Light brush=1.2; Heavy brush=1.5; Swampy bog=1.8; Loose sand=2.1; Hard-packed snow=1.6; Ploughed field=1.3; **see** de Gruchy, M., Caswell, E., & Edwards, J. (2017). Velocity-Based Terrain Coefficients for Time-Based Models of Human Movement. Internet Archaeology, 45(45). https://doi.org/10.11141/ia.45.4.

For this cost function, **see** Pandolf, K. B., Givoni, B., & Goldman, R. F. (1977). Predicting energy expenditure with loads while standing or walking very slowly. Journal of Applied Physiology, 43(4), 577-581. https://doi.org/10.1152/jappl.1977.43.4.577.

For the use of this cost function in a case study, **see** Rademaker, K., Reid, D. A., & Bromley, G. R. M. (2012). Connecting the Dots: Least Cost Analysis, Paleogeography, and the Search for Paleoindian Sites in Southern Highland Peru. In D. A. White & S. L. Surface-Evans (Eds.), Least Cost Analysis of Social Landscapes. Archaeological Case Studies (pp. 32-45). University of Utah Press; **see also** Herzog, I. (2013). Least-cost Paths - Some Methodological Issues, Internet Archaeology 36 (http://intarch.ac.uk/journal/issue36/index.html) with references.

Note: in the returned charts, the cost is transposed from Watts to Megawatts (see, e.g., Rademaker et al 2012 cited above).

Van Leusen's metabolic energy expenditure cost function (in Watts):

$$1/(1.5*W+2.0*(W+L)*(L/W)^2+N*(W+L)*(1.5*V^2+0.35*V*abs(x[adj])*100)+10))$$

which modifies the Pandolf et al.'s equation; **see** Van Leusen, P. M. (2002). Pattern to process: methodological investigations into the formation and interpretation of spatial patterns in archaeological landscapes. University of Groningen.
Note that, as per Herzog, I. (2013). Least-cost Paths - Some Methodological Issues, Internet Archaeology 36 (http://intarch.ac.uk/journal/issue36/index.html) and unlike Van Leusen (2002), in the above equation slope is expressed in percent and speed in m/s; also, in the last bit of the equantion, 10 replaces the value of 6 used by Van Leusen (as per Herzog 2013).
Note: in the returned charts, the cost is transposed from Watts to Megawatts.

Llobera-Sluckin's metabolic energy expenditure cost function (in KJ/m):

$1/(2.635 + (17.37 * abs(x[adj])) + (42.37 * abs(x[adj])^2) - (21.43 * abs(x[adj])^3) + (14.93 * abs(x[adj])^4))$

for which **see** Llobera M.-Sluckin T.J. (2007). Zigzagging: Theoretical insights on climbing strategies, Journal of Theoretical Biology 249, 206-217.

Note that the walking-speed-related cost functions listed above are used as they are, while the other functions are reciprocated. This is done since "gdistance works with conductivity rather than the more usual approach using costs"; therefore "we need inverse cost functions" (Nakoinz-Knitter (2016). "Modelling Human Behaviour in Landscapes". New York: Springer, p. 183). As a consequence, if we want to estimate time, we have to use the walking-speed functions as they are since the final accumulated values will correspond to the reciprocal of speed, i.e. pace. In the other cases, we have to use 1/cost-function to eventually get cost-function/1.

When using the Tobler-related cost functions, the time unit can be selected by the user setting the 'time' parameter to 'h' (hour) or to 'm' (minutes).

In general, the user can also select which type of visualization the function has to produce; this is achieved setting the 'outp' parameter to either 'r' (=raster) or to 'c' (=contours). The former will produce a raster with a colour scale and contour lines representing the accumulated cost surface; the latter parameter will only produce contour lines.

The contour lines' interval is set using the parameter 'breaks'; if no value is passed to the parameter, the interval will be set by default to 1/10 of the range of values of the accumulated cost surface.

Value

The function returns a list storing the following components

- accumulated.cost.raster: raster representing the accumualted cost ('RasterLayer' class)
- isolines: contour lines derived from the accumulated cost surface ('SpatialLinesDataFrame' class)
- LCPs: estimated least-cost paths ('SpatialLines' class)
- LCPs$length: length of each least-cost path (units depend on the unit used in the input DTM)

- dest.loc.w.cost: copy of the input destination location(s) dataset with a new variable ('cost') added

Examples

```
# load a sample Digital Terrain Model
volc <- raster::raster(system.file("external/maungawhau.grd", package="gdistance"))

# load a sample start location on the above DTM
data(volc.loc)

# load the sample destination locations on the above DTM
data(destin.loc)

# calculate walking-time isochrones based on the on-path Tobler's hiking function,
# setting the time unit to hours and the isochrones interval to 0.05 hour;
# also, since destination locations are provided,
# least-cost paths from the origin to the destination locations will be calculated
# and plotted; 8-directions move is used
result <- movecost(dtm=volc,origin=volc.loc, destin=destin.loc, move=8, breaks=0.05)

#same as above, but using 16-directions move (which is the default value)
result <- movecost(dtm=volc,origin=volc.loc, destin=destin.loc, move=16, breaks=0.05)
```

---

volc.loc — *Dataset: location on the volcano Maunga Whau (Auckland, New Zealand)*

Description

A SpatialPointsDataFrame representing a spot on the volcano Maunga Whau (Auckland, New Zealand).

Usage

```
data(volc.loc)
```

Format

SpatialPointsDataFrame

Index