Package: traveltime (via r-universe)

September 15, 2024

Title Calculate Travel Times Over Space

Version 0.0.0.9000

Description Calculate travel time over a friction surface from a specified set of locations.

License MIT + file LICENSE

Imports gdistance, malariaAtlas, raster, terra

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://idem-lab.r-universe.dev

RemoteUrl https://github.com/idem-lab/traveltime

RemoteRef HEAD

RemoteSha 4be069e807c08f23b540f7f4cc1358369c3595db

Contents

`calculate_travel_time` *Calculate travel time*

Description

Calculate the travel time from a set of points over a friction surface.

Usage

```
calculate_travel_time(
  friction_surface,
  points,
  filename = NULL,
  overwrite = FALSE
)
```

Arguments

`friction_surface`	A `SpatRaster` friction surface layer. See `?get_friction_surface`
`points`	A two-column `data.frame` or `tibble` with longitude (x) in the first column and latitude (y) in the second in the same coordinate reference system as `friction_surface`
`filename`	`character`. Output file name with extension suitable for `terra::writeRaster`
`overwrite`	`logical`. If `TRUE` `filename` is overwritten.

Details

Implements methods from Weiss et al. 2018, 2020 to calculate travel time from given locations over a friction surface.

Over large areas this function can require significant RAM and will be slow.

Citations: D. J. Weiss, A. Nelson, C. A. Vargas-Ruiz, K. Gligoric, S., Bavadekar, E. Gabrilovich, A. Bertozzi-Villa, J. Rozier, H. S. Gibson, T., Shekel, C. Kamath, A. Lieber, K. Schulman, Y. Shao, V. Qarkaxhija, A. K. Nandi, S. H. Keddie, S. Rumisha, P. Amratia, R. Arambepola, E. G. Chestnutt, J. J. Millar, T. L. Symons, E. Cameron, K. E. Battle, S. Bhatt, and P. W. Gething. Global maps of travel time to healthcare facilities. (2020) Nature Medicine. https://doi.org/10.1038/s41591-020-1059-1

D. J. Weiss, A. Nelson, H.S. Gibson, W. Temperley, S. Peedell, A. Lieber, M. Hancher, E. Poyart, S. Belchior, N. Fullman, B. Mappin, U. Dalrymple, J. Rozier, T.C.D. Lucas, R.E. Howes, L.S. Tusting, S.Y. Kang, E. Cameron, D. Bisanzio, K.E. Battle, S. Bhatt, and P.W. Gething. A global map of travel time to cities to assess inequalities in accessibility in 2015. (2018). Nature. doi:10.1038/nature25181.

Value

`SpatRaster`

Examples

```
ext <- matrix(
  data = c("111", "0", "112", 1),
  nrow = 2,
  ncol = 2,
  dimnames = list(
    c("x", "y"),
    c("min", "max")
   )
 )

 friction_surface <- get_friction_surface(
   surface = "motor2020",
   extent = ext
 )

 from_here <- data.frame(
   x = c(111.2, 111.9),
   y = c(0.2, 0.35)
 )

 calculate_travel_time(
  friction_surface = friction_surface,
  points = from_here
 )
```

`ext_from_terra` *Extent from* `SpatRaster` *or* `SpatVector`

Description

Formats spatial extent for use in `get_friction_surface`.

Usage

```
ext_from_terra(r)
```

Arguments

`r`	`terra::SpatRaster` or `terra::SpatVector`

Value

2x2 `matrix`

Examples

```
library(terra)
r <- terra::rast(
    extent = terra::ext(c(111, 112, 0, 1))
  )

ext_from_terra(r)
```

`ext_vect_to_matrix` *Extent vector to matrix*

Description

Extent vector to matrix

Usage

```
ext_vect_to_matrix(x)
```

Arguments

`x`	`numeric` length 4, consisting of `c(xmin, xmax, ymin, ymax)` dimensions of extent

Value

2x2 `matrix`

Examples

```
ext_vect_to_matrix(c(111,112,0, 1))
```

`get_friction_surface` *Get friction surface*

Description

Wrapper function to download friction surfaces via `malariaAtlas::getRaster`

Usage

```
get_friction_surface(
  surface = c("motor2020", "walk2020"),
  filename = NULL,
  overwrite = FALSE,
  extent = NULL
)
```

Arguments

surface	"motor2020" or "walk2020".
filename	character. File name for output layer.
overwrite	Overwrite filename if exists
extent	Spatial extent as either numeric vector specifying c(xmin, xmax, ymin, ymax), SpatExtent, SpatVector or SpatRaster (from which the extent will be taken), or 2x2 matrix (see details).

Details

Convenience wrapper to malariaAtlas::getRaster to access motorised and walking travel friction layers per Weiss et al. 2020, that adds safety to check existing files before download. Surfaces can be downloaded directly from: https://malariaatlas.org/project-resources/accessibility-to-healthcare/

surface = "motor2020" downloads "Explorer__2020_motorized_friction_surface".

surface = "walk2020" downloads "Explorer__2020_walking_only_friction_surface".

D. J. Weiss, A. Nelson, C. A. Vargas-Ruiz, K. Gligoric, S., Bavadekar, E. Gabrilovich, A. Bertozzi-Villa, J. Rozier, H. S. Gibson, T., Shekel, C. Kamath, A. Lieber, K. Schulman, Y. Shao, V. Qarkaxhija, A. K. Nandi, S. H. Keddie, S. Rumisha, P. Amratia, R. Arambepola, E. G. Chestnutt, J. J. Millar, T. L. Symons, E. Cameron, K. E. Battle, S. Bhatt, and P. W. Gething. Global maps of travel time to healthcare facilities. (2020) Nature Medicine. https://doi.org/10.1038/s41591-020-1059-1

extent is passed through is to pass to malariaAtlas::getRaster as a 2x2 matrix. If passed in as a numeric vector, SpatExtent, SpatVector, or SpatRaster, it is converted into a matrix using ext_vect_to_matrix and ext_from_terra. matrix format is as returned by sf::st_bbox() - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max")))) (use either shp OR extent; if neither is specified global raster is returned). NULL extent downloads (large) global layer.

Additional details...

Value

SpatRaster

Examples

```
ext <- matrix(
  data = c(111, 0, 112, 1),
  nrow = 2,
  ncol = 2,
  dimnames = list(
    c("x", "y"),
    c("min", "max")
   )
 )

 get_friction_surface(
```

```
  surface = "motor2020",
  extent = ext
)
```

Index