

Package: mapMCDA (via r-universe)

September 16, 2026

Title Produce an epidemiological risk map by weighting multiple risk factors

Version 0.5.19

Description Use a Shiny interface to help users import multiple layers of risk factors, scale and combine them using a Multi-Criteria Decision Analysis approach to produce a risk map as an outcome.

Depends R (>= 4.0)

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

Imports deldir, geodata, geonetwork, igraph, methods, rhandsontable, shiny, shinycssloaders, shinydashboard, shinyFiles, sf, terra (>= 1.7.23), utils

Suggests covr, devtools, knitr, lintr, lwgeom, mapview, rmarkdown, roxygen2, testthat

RoxygenNote 7.3.2

URL <https://umr-astre.pages.mia.inra.fr/mapMCDA>

BugReports <https://forgemia.inra.fr/umr-astre/mapMCDA/issues>

VignetteBuilder knitr

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev make libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository <https://cirad-astre.r-universe.dev>

Date/Publication 2024-10-17 14:29:44 UTC

RemoteUrl <https://forge.inrae.fr/umr-astre/mapMCDA>

RemoteRef HEAD

RemoteSha 43b3670f1cdd6d8f429e4eb98f85bf7be143a2fb

Contents

align_layers	2
compute_weights	3
crs,geonetwork-method	4
csv_sep	4
distance_map	5
distance_to_vector	6
download_country_admin	6
epidemic_threshold	7
factor_type	8
is.reciprocal	9
layer_type	9
load_dir	10
load_layer	10
mapMCDA_app	11
mapMCDA_datasets	11
plot_riskfactor	12
rasterize_geonetwork	12
read_network	13
resolution	14
risk_layer	15
risk_plot	16
risk_table	16
risk_unit	17
rmext	18
voronoi	18
wlc	19
Index	20

align_layers	<i>Harmonise projection, resolution and extent of a list of rasters</i>
--------------	---

Description

Convert a list of rasters to a common projection, resolution and extent.

Usage

```
align_layers(x)
```

Arguments

x list of SpatRaster objects

Details

The final projection is the most common one, other than NA. If a raster has unknown projection, it is assumed to be equal to the most common one.

All rasters are resampled to the same extent and resolution as the *first* raster in the list.

Value

Another list of SpatRaster objects

compute_weights	<i>Compute MCDA weights</i>
-----------------	-----------------------------

Description

Given a square reciprocal-symmetric *ratio* matrix from an Analytic Hierarchy Process, calculate the vector of weights using the Pairwise Comparison Method (Saaty 1987, Saaty 1996, Malczewski 1999)

Usage

```
compute_weights(x)
```

Arguments

x matrix of pair-wise assessment of relative importance of risk factors.

Value

numeric vector of weights

References

Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling* 9 (3-5), 161-176.

Saaty, T. L. (1996). *Multicriteria decision making : the analytic hierarchy process : planning, priority setting, resource allocation*. RWS Publications.

Malczewski J. *GIS and multicriteria decision analysis*. Inc New York: John Wiley & Sons; 1999.

Examples

```
compute_weights(matrix(1, 5, 5))
```

crs,geonetwork-method *CRS of a geonetwork object*

Description

This should be exported in the geonetwork package.

Usage

```
## S4 method for signature 'geonetwork'
crs(x, ...)
```

Arguments

x A geonetwork object.
... Not used.

Value

A crs object.

Examples

```
cmr <- mapMCDA_datasets()
crs(cmr$mobility)
```

csv_sep *Detection of separator in csv files*

Description

Returns the separator that consistently separates the lines in the same number of fields, or fails.

Usage

```
csv_sep(x)
```

Arguments

x character. File name.

Details

Tested separators are: ",", " ", "\t", "|", ";", ":"

Value

Character representation of the field separator.

Examples

```
tf <- tempfile()
td <- data.frame(T = LETTERS[1:10], N = 1:10)
write.csv(td, tf, row.names = FALSE)
csv_sep(tf) # ","
write.csv2(td, tf, row.names = FALSE)
csv_sep(tf) # ";"
```

distance_map	<i>Compute a distance map to spatial features</i>
--------------	---

Description

Returns a distance map (in meters) to spatial features.

Usage

```
distance_map(x, boundaries, res = resolution(boundaries, max_ncells = 100))
```

Arguments

x	sf or sfc object with relevant spatial features.
boundaries	sf or sfc POLYGON object with administrative borders, or SpatRaster object from where to pick the resolution.
res	numeric. Resolution for the outcome distance map. The default divides the smallest dimension into 100 cells.

Details

Distances are computed up to the given boundaries and at given resolution.

Value

A raster map with distances in meters to the closest spatial feature.

Examples

```
cmr <- mapMCDA_datasets()
plot(distance_map(x = cmr$water_bodies, boundaries = cmr$cmr_admin3))
```

distance_to_vector	<i>Distance to vector features</i>
--------------------	------------------------------------

Description

Compute a raster file with the distances to the nearest vector feature, within the limits of a given mask.

Usage

```
distance_to_vector(x, mask)
```

Arguments

x	SpatVector object, or something convertible to.
mask	SpatRaster object

Details

The resolution of the mask determines that of the outcome.

Value

SpatRaster object. Distance map in meters (even for unprojected maps in geographical coordinates).

Examples

```
cmr <- mapMCDA_datasets()
dist_wb <- distance_to_vector(cmr$water_bodies, cmr$animal.density)
plot(dist_wb)
```

download_country_admin	<i>Download Country Administrative Borders from GADM</i>
------------------------	--

Description

Download Country Administrative Borders from GADM

Usage

```
download_country_admin(country_name, level = 3, path = tempdir())
```

Arguments

country_name	character. A valid country name.
level	integer 0:3.
path	character. Where to store the downloaded object.

Value

A sf object.

Examples

```
cmr <- download_country_admin("Cameroon")
```

epidemic_threshold	<i>Compute the Epidemic Threshold of a graph</i>
--------------------	--

Description

Weighted and unweighted Epidemic Threshold q of a graph.

Usage

```
epidemic_threshold(x, beta = 1)
```

Arguments

x	an igraph object
beta	numeric, between 0 and 1. Probability of transmission.

Details

The Epidemic Threshold q quantifies the minimal expected transmission coefficient necessary for diffusing an epidemic in a network. It is computed as the inverse of the *Potential for transmission* of the network: a measure of the expected number of nodes affected by an infectious node, which is a generalisation of the Basic Reproduction Number R_0 of an epidemic to the context of a network. It thus quantifies the potential for transmission of an infection throughout the contact network. It is computed in terms of the incoming-outgoing rates from the network's nodes:

$$R_0 = \beta \frac{\overline{k_{\text{in}} \cdot k_{\text{out}}}}{\overline{k_{\text{in}}}},$$

where β is the transmission coefficient among animals, $k_{\text{in/out}}$ are the in/out-degrees of a node and the $\bar{\cdot}$ symbol represents the average value across all nodes in the graph.

The unweighted value computed above is most appropriate for a highly infectious epidemic with high animal-prevalence on nodes, as it assumes that any contact is potentially infectious.

In the weighted formulation, $k_{in/out}$ are the weight values for the incoming/outgoing edges in each node. It is more appropriate for low-prevalence diseases, where the transmission probability is assumed proportional to the number of contacts.

The default value of 1 for the probability of transmission β implies that every infectious contact leads to transmission.

Value

a list the weighted and unweighted Potential for Transmission R_0 and its inverse, the Epidemic Threshold q . As an attribute named "sna", a data.frame with the in/out-degrees of each node and their individual contribution to R_0 .

If the input graph is unweighted, the weighted component is NULL.

References

Volkova VV, Howey R, Savill NJ, Woolhouse MEJ (2010) Sheep Movement Networks and the Transmission of Infectious Diseases. PLoS ONE 5(6): e11185. <https://doi.org/10.1371/journal.pone.0011185>

Examples

```
g <- igraph::graph_from_literal(A --- B ---C, A ---C, B ---D)
epidemic_threshold(g)

## weighted graph
igraph::E(g)$weight <- c(10, 1, 2, 5)
epidemic_threshold(g)
```

factor_type	Type of risk factor
-------------	---------------------

Description

Recognise the geographical type of an object representing a risk factor.

Usage

```
factor_type(x)
```

Arguments

x Any object.

Value

Character. Either "vector", "raster" or "network" depending on the object class. NA for unknown formats.

Examples

```
factor_type(terra::rast())
factor_type(sf::st_sf(sf::st_sfc()))
```

is.reciprocal	<i>Check reciprocal matrix</i>
---------------	--------------------------------

Description

Check whether a matrix is reciprocal. This is, whether the entrywise product of itself with its transpose (Hadamard product) is 1 for all its elements.

Usage

```
is.reciprocal(x)
```

Arguments

x square numeric matrix or number

Value

logical

Examples

```
Xf <- matrix(1:16, 4, 4)
Xr <- Xf
Xr[upper.tri(Xf)] <- 1/t(Xf)[upper.tri(Xf)]
diag(Xr) <- 1
mapMCDAs::is.reciprocal(Xf) # FALSE
mapMCDAs::is.reciprocal(Xr) # TRUE
```

layer_type	<i>Detect type of geographic information</i>
------------	--

Description

Either "vector", "raster" or "network" depending on the file extension.

Usage

```
layer_type(x)
```

Arguments

x string. A file name.

Value

character. Either "vector", "raster" or "network". NA for unknown formats.

Examples

```
layer_type("myfile.tif")
layer_type("myfile.gpkg")
```

load_dir	<i>Load all layers within a directory</i>
----------	---

Description

Load all recognised vector and raster files within a given directory.

Usage

```
load_dir(path)
```

Arguments

path String. The path to the directory where spatial layers are stored.

Value

list of layers in the form of sf or SpatRaster objects

Examples

```
pkg_layers <- system.file(path = "cartography/CMR/", package = "mapMCDA")
cmr <- load_dir(pkg_layers)
```

load_layer	<i>Load a map layer</i>
------------	-------------------------

Description

Automatically detects whether it is a vector or raster map and uses the appropriate read function.

Usage

```
load_layer(x)
```

Arguments

x File name with full path

Value

A sf object for vector maps and a SpatRaster object for raster maps.

Examples

```
(pkg_layers <- list.files(
  system.file("cartography/CMR", package = "mapMCDA"),
  full.names = TRUE
))

epiunits <- load_layer(pkg_layers[[2]])
```

mapMCDA_app

mapMCDA Shiny web interface

Description

Launches the graphical interface (GUI) of the package.

Usage

```
mapMCDA_app(bg = FALSE)
```

Arguments

bg	Logical. If TRUE, the GUI is launched in the background, leaving available the current session.
----	---

mapMCDA_datasets

Load packaged Cameroon layers into memory

Description

Loads a list of 4 geographical layers of Cameroon. Namely: the raster of animal density, the administrative borders at the third level, the national parks and the water bodies.

Usage

```
mapMCDA_datasets()
```

Details

Sources are the Gridded Livestock of the World database (ref?), the Global Administrative Boundaries database (gadm.org), and ???.

Value

list of spatial layers

Examples

```
cmr <- mapMCDA_datasets()
```

plot_riskfactor	<i>Plot a risk factor</i>
-----------------	---------------------------

Description

Plot geographical risk map of different types optionally with administrative divisions in the background.

Usage

```
plot_riskfactor(x, admin = NULL, pal = grDevices::hcl.colors(50))
```

Arguments

x	An object of class 'sf', 'SpatRaster' or 'geonetwork'.
admin	A 'sf' POLYGONS object with administrative units.
pal	Character vector. Colours. The default is grDevices::hcl.colors(50). #'

Examples

```
cmr <- mapMCDA_datasets()
plot_riskfactor(cmr$animal.density)
plot_riskfactor(cmr$mobility, admin = cmr$cmr_admin3)
plot_riskfactor(cmr$water_bodies, admin = cmr$cmr_admin3)
```

rasterize_geonetwork	<i>Rasterize a geographic network</i>
----------------------	---------------------------------------

Description

Compute a raster with the importance of the nearest network node.

Usage

```
## S4 method for signature 'geonetwork,SpatRaster'
rasterize(x, y, field = "importance", ...)

## S4 method for signature 'geonetwork,sf'
rasterize(x, y, res = resolution(y, max_ncells = 100), ...)
```

Arguments

x	a geographic network (a geonetwork object with node attributes "Lon" and "Lat" in the WGS84 reference system)
y	a SpatRaster or a sf or sfc object.
field	character. The attribute of the network nodes to be assigned to the voronoi polygon associated with each node. By default it computes the <i>importance</i> of each node in the network. But you can manually compute any other measure of interest, and pass it as a vertex attribute (see ?igraph::`vertex_attr<-`).
...	Other arguments passed on to rasterize .
res	numeric vector of length 2. Resolution in Lon/Lat of rasterisation.

Details

This assumes that the network object has node attributes "Lon" "Lat" in the WGS84 reference system.

If the graph is weighted, the weighted importance will be computed. If you want the unweighted importance, remove the weight attribute from the graph with `igraph::delete_edge_attr(x, "weight")`.

The importance of each node is computed as the relative contribution to the Potential for Transmission R_0 of the graph. I.e., R_{0k}/R_0 . See [epidemic_threshold](#).

read_network	<i>Read and interpret network data</i>
--------------	--

Description

Read a csv file with a specific format (see Details) and intepret it as network origin-destination data with possibly weighted edges.

Usage

```
read_network(x)
```

Arguments

x	character. Path of csv file
---	-----------------------------

Details

The file must be plain text with comma-separated columns and variable names in the first line. Field separators other than comma are also supported. In particular, semi-colon field separators and comma decimal separator. There must be either 6 or 7 columns in the same order and of the same types as follows:

- origin character

- destination character
- lon_orig numeric (decimal degrees, WGS84)
- lat_orig numeric (decimal degrees, WGS84)
- lon_dest numeric (decimal degrees, WGS84)
- lat_dest numeric (decimal degrees, WGS84)
- volume Optional. directed flux in some consistent unit

Variable names can be different. If strings contain spaces they must be quoted. For this reason, **quotes cannot be part of node names**. E.g. the name *Abangminko'O* is invalid and will rise an error.

Each origin-destination must be specified only once. If you have multiple animal shipments between the same origin and destination spread in several lines, you need to aggregate them into a single line.

Make sure **coordinates are consistent**. I.e. All nodes must have the same coordinates in all the lines where it appears as either source or destination.

Value

Object of classes geonetwork and igraph.

Examples

```
d <- data.frame(from = "A", to = "B", fx = 0, fy = 0, tx = 1, ty = 1)
tf <- tempfile()
write.csv(d, tf, row.names = FALSE)
read_network(tf)
```

resolution	<i>Compute a suitable resolution for a map</i>
------------	--

Description

Compute a resolution in the original map units that conforms to given criteria.

Usage

```
resolution(x, max_ncells = 100)
```

Arguments

x	a SpatRaster or sf or sfc object
max_ncells	Number of cells for the largest dimension

Value

A number. The resolution that respects the desired number of cells.

Examples

```
r <- terra::rast(nrows = 5, ncols = 11, xmin = 0, xmax = 11, ymin = 0, ymax = 5)
resolution(r, max_ncells = 10)
```

risk_layer	<i>Compute risk layer</i>
------------	---------------------------

Description

Compute a raster map from the input layer and rescale with a linear relationship.

Usage

```
risk_layer(x, boundaries, scale_target = c(0, 100))
```

Arguments

<code>x</code>	a sf, sfc, SpatRaster or igraph object
<code>boundaries</code>	a sf or sfc object, used to determine the boundaries of the computed risk layer.
<code>scale_target</code>	numeric vector of length 2. New scale.

Details

For sf objects (geometries such as point, lines or polygons), compute the `distance_map()`, which gives a SpatRaster. For igraph objects (from network data), compute a SpatRaster with the relative importance of the nearest node. For a SpatRaster mask, extend or crop to the boundaries as needed.

Finally, scale the SpatRaster outcome of any of the three input types. If you need an inverse relationship, just reverse the target scale.

Value

A SpatRaster object in the new scale.

Examples

```
ad <- mapMCDA_datasets()$animal.density
bd <- mapMCDA_datasets()$cmr_admin3
plot(ad)
plot(risk_layer(ad, bd, scale_target = c(-1, 1)))
plot(risk_layer(ad, bd, scale_target = c(1, -1)))
```

risk_plot	<i>Produce final plot of risk</i>
-----------	-----------------------------------

Description

Represent the value of risk by epidemiological unit categorised into n risk-levels

Usage

```
risk_plot(x, v, n)
```

Arguments

x	sf object. Epidemiological units.
v	Numeric. Risk values.
n	Number of risk levels.

Value

plot

Examples

```
epi_units <- mapMCDA_datasets()$cmr_admin3  
risk_plot(epi_units, seq.int(nrow(epi_units)), n = 4)
```

risk_table	<i>Produce final table of risk levels by epidemiological unit</i>
------------	---

Description

Represent the value of risk by epidemiological unit categorised into n risk-levels

Usage

```
risk_table(x, v, n)
```

Arguments

x	sf object. Epidemiological units.
v	Numeric. Risk values.
n	Number of risk levels.

Value

data.frame

Examples

```

epi_units <- mapMCDA_datasets()$cmr_admin3
risk <- runif(nrow(epi_units), 0, 100)
risk_table(epi_units, risk, n = 4)

```

risk_unit	<i>Risk by epidemiological unit</i>
-----------	-------------------------------------

Description

Compute the risk for each of the epidemiological units.

Usage

```
risk_unit(r, eu, fun = mean)
```

Arguments

r	SpatRaster object
eu	sf or sfc object of type POLYGON
fun	A summary function. By default the mean.

Details

Summarises the level of risk using a function such as the mean or median value across the epidemiological unit.

Value

numeric vector of the same length as the number of epidemiological units.

Examples

```

cmr <- mapMCDA_datasets()
cmr$cmr_admin3$rv <- risk_unit(r = cmr$animal.density, eu = cmr$cmr_admin3)
plot(cmr$cmr_admin3["rv"], pal = hcl.colors, nbreaks = 4, breaks = "quantile")

```

rmext	<i>Remove filename extension</i>
-------	----------------------------------

Description

Removes the trailing characters after and including the last point in a string.

Usage

```
rmext(x)
```

Arguments

x	Character
---	-----------

Value

character.

Examples

```
rmext("some/long.filename.extension")
```

voronoi	<i>Voronoi polygons</i>
---------	-------------------------

Description

Compute a voronoi tessellation of a map for a set of points.

Usage

```
voronoi(x, ext, eps = 1e-09)
```

Arguments

x	sf POINTS or two-column matrix or data.frame with x and y coordinates.
ext	Extent object.
eps	Numerical tolerance used in triangulation. See deldir .

Value

sf POLYGON object.

Author(s)

Borrowed from function `voronoi` from package `dismo` by R. Hijmans et al., which ultimately uses the `deldir` function by Rolf Turner.

Examples

```
pts <- data.frame(x = c(1:2, 2), y = c(1, 1:2))
region <- terra::ext(c(0, 3, 0, 3))
plot(sf::st_geometry(voronoi(pts, region)))
points(pts)
```

wlc

*Weighted Linear Combination of risk factors***Description**

Combine a list of risk layers using a weighted linear approach:

$$w_1x_1 + w_2x_2 + \dots$$

Usage

```
wlc(x, w, na.rm = TRUE)
```

Arguments

<code>x</code>	list of risk factors (SpatRaster objects)
<code>w</code>	vector of weights
<code>na.rm</code>	logical.

Details

Missing values are by default removed from the combination, yielding a result that relies on the available factors in each place. If `na.rm = FALSE`, then all pixels where at least one risk factor is missing get a missing value as the result.

Value

A SpatRaster object.

Index

[align_layers](#), [2](#)

[compute_weights](#), [3](#)
[crs, geonetwork-method](#), [4](#)
[csv_sep](#), [4](#)

[deldir](#), [18](#), [19](#)
[distance_map](#), [5](#)
[distance_to_vector](#), [6](#)
[download_country_admin](#), [6](#)

[epidemic_threshold](#), [7](#), [13](#)

[factor_type](#), [8](#)

[is.reciprocal](#), [9](#)

[layer_type](#), [9](#)
[load_dir](#), [10](#)
[load_layer](#), [10](#)

[mapMCDA_app](#), [11](#)
[mapMCDA_datasets](#), [11](#)

[plot_riskfactor](#), [12](#)

[rasterize](#), [13](#)
[rasterize, geonetwork, sf-method](#)
 ([rasterize_geonetwork](#)), [12](#)
[rasterize, geonetwork, SpatRaster-method](#)
 ([rasterize_geonetwork](#)), [12](#)
[rasterize_geonetwork](#), [12](#)
[read_network](#), [13](#)
[resolution](#), [14](#)
[risk_layer](#), [15](#)
[risk_plot](#), [16](#)
[risk_table](#), [16](#)
[risk_unit](#), [17](#)
[rmext](#), [18](#)

[voronoi](#), [18](#)

[wlc](#), [19](#)