Package ‘mop’

July 23, 2025

Type Package

Title Mobility Oriented-Parity Metric

Version 0.1.3

Maintainer Marlon E. Cobos <`manubio13@gmail.com`>

Date 2025-04-22

Description A set of tools to perform multiple versions of the Mobility Oriented-Parity metric. This multivariate analysis helps to characterize levels of dissimilarity between a set of conditions of reference and another set of conditions of interest. If predictive models are transferred to conditions different from those over which models were calibrated (trained), this metric helps to identify transfer conditions that differ substantially from those of calibration. These tools are implemented following principles proposed in Owens et al. (2013) <doi:10.1016/j.ecolmodel.2013.04.011>, and expanded to obtain more detailed results that aid in interpretation as in Cobos et al. (2024) <doi:10.21425/fob.17.132916>.

URL `https://github.com/marlonecobos/mop`

BugReports `https://github.com/marlonecobos/mop/issues`

Imports doSNOW (>= 1.0), foreach (>= 1.5), methods, parallel, Rcpp, snow (>= 0.4), stats, terra (>= 1.6-7), utils

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5)

LazyData true

LinkingTo Rcpp

NeedsCompilation yes

Author Marlon E. Cobos [aut, cre] (ORCID: <`https://orcid.org/0000-0002-2611-1767`>), Hannah L. Owens [aut] (ORCID: <`https://orcid.org/0000-0003-0071-1745`>),

Jorge Soberón [aut] (ORCID: <https://orcid.org/0000-0003-2160-4148>),
A. Townsend Peterson [aut] (ORCID:
<https://orcid.org/0000-0003-0243-2379>)

Repository CRAN

Date/Publication 2025-04-24 11:40:06 UTC

Contents

`mop-package` *mop: Mobility Oriented-Parity Metric*

Description

`mop` contains a set of tools to calculate the Mobility Oriented-Parity metric, which allows a user to compare a set of conditions of reference versus another set of of interest.

Details

The main goals of the MOP metric are to explore conditions in the set of interest that are non-analogous to those in the reference set, and to quantify how different conditions in the set of interest are from the reference set. The tools included here help to identify conditions outside the ranges of the reference set with greater detail than in other implementations. These tools are based on the methods proposed by Owens et al. (2013; `doi:10.1016/j.ecolmodel.2013.04.011`).

Functions in mop

`mop`, `mop_distance`, `out_range`, `match_na_raster`

Data included

`reference_matrix`, `matrix_of_interest`, `reference_layers`, `layers_of_interest`

Author(s)

Maintainer: Marlon E. Cobos `<manubio13@gmail.com>` (ORCID)

Authors:

- Hannah L. Owens `<hannah.owens@gmail.com>` (ORCID)
- Jorge Soberón `<jsoberon@ku.edu>` (ORCID)
- A. Townsend Peterson `<town@ku.edu>` (ORCID)

See Also

Useful links:

- `https://github.com/marlonecobos/mop`
- Report bugs at `https://github.com/marlonecobos/mop/issues`

`layers_of_interest` *Example of variables for a set of interest*

Description

A `SpatRaster` object representing variables in a set of interest. Variables represent future bioclimatic variables downloaded from the WorldClim database (`https://worldclim.org/`).

Format

A `SpatRaster` object.

Value

No return value. Used with function `rast` to bring raster variables to analysis.

Examples

```
layers_of_interest <- terra::rast(system.file("extdata",
                                              "layers_of_interest.tif",
                                              package = "mop"))

terra::plot(layers_of_interest)
```

`match_na_raster` *Match NA cells for all layers in SpatRaster*

Description

Option to match cells with NA values in a SpatRaster with multiple layers.

Usage

```
match_na_raster(layers)
```

Arguments

`layers`	a `SpatRaster` object containing two or more variables to be matched.

Value

A `SpatRaster` object with NA cells matching in all layers.

Examples

```
# data
layers <- terra::rast(system.file("extdata", "reference_layers.tif",
                      package = "mop"))

# add NA in some places
layers[20:24, 10:16][, 3] <- NA
terra::plot(layers)

# match NAs
matched <- match_na_raster(layers)
terra::plot(matched)
```

`matrix_of_interest` *Example of matrix with variables in a set of interest*

Description

A numeric table representing variables in a set of interest.

Usage

```
matrix_of_interest
```

Format

A matrix with 723 rows and 6 columns.

Examples

```
data("matrix_of_interest", package = "mop")

head(matrix_of_interest)
```

`mop` *Analysis of extrapolation risks using the MOP metric*

Description

Analysis to calculate the mobility-oriented parity metric and other sub-products to represent dissimilarities and non-analogous conditions when comparing a set of reference conditions (M; `m`) against another set of conditions of interest (G; `g`).

Usage

```
mop(m, g, type = "basic",  calculate_distance = FALSE,
    where_distance = "in_range", distance = "euclidean",
    scale = FALSE, center = FALSE, fix_NA = TRUE, percentage = 1,
    comp_each = 2000, tol = NULL, rescale_distance = FALSE,
    parallel = FALSE, n_cores = NULL, progress_bar = TRUE)
```

Arguments

`m`	a `SpatRaster` or matrix of variables representing a set of conditions of reference (e.g., the set of conditions in which a model was calibrated). If a matrix is used, each column represents a variable.
`g`	a `SpatRaster` or matrix of variables representing a set of conditions of interest for which dissimilarity values and non-analogous conditions will be detected (e.g., conditions in which a model is projected). Variable names must match between `m` and `g`.
`type`	`character`, type of MOP analysis to be performed. See `Details` for options.
`calculate_distance`	`logical`, whether to calculate distances (dissimilarities) between `m` and `g`. The default, FALSE, runs rapidly and does not assess dissimilarity levels.
`where_distance`	`character`, where to calculate distances, considering how conditions in `g` are positioned in comparison to the range of conditions in `m`. See `Details` for options.
`distance`	`character`, which distances are calculated, `euclidean` or `mahalanobis`. Valid if `calculate_distance` = `TRUE`.
`scale`	scaling options, `logical` or `numeric-alike` as in `scale`.
`center`	`logical` or `numeric-alike` center options as in `scale`.
`fix_NA`	`logical`, whether to fix layers so cells with NA values are the same in all layers. Setting to FALSE may save time if the rasters are big and have no NA matching problems.

`percentage`	`numeric`, percentage of `m` closest conditions used to derive mean environmental distances to each combination of conditions in `g`.
`comp_each`	`numeric`, number of combinations in `g` to be used for distance calculations at a time. Increasing this number requires more RAM.
`tol`	tolerance to detect linear dependencies when calculating Mahalanobis distances. The default, NULL, uses `.Machine$double.eps`.
`rescale_distance`	`logical`, whether or not to re-scale distances 0-1. Re-scaling prevents comparisons of dissimilarity values obtained from runs with different values of `percentage`.
`parallel`	`logical`, whether calculations should be performed in parallel using `n_cores` of the computer. Using this option will speed up the analysis but will demand more RAM.
`n_cores`	`numeric`, number of cores to be used in parallel processing. If `parallel` = `TRUE` and `n_cores` = `NULL` (all CPU cores on current host - 1) will be used.
`progress_bar`	`logical`, whether to show a progress bar.

Details

`type` options return results that differ in the detail of how non-analogous conditions are identified.

- **basic** - makes calculation as proposed by Owens et al. (2013) `doi:10.1016/j.ecolmodel.2013.04.011`.
- **simple** - calculates how many variables in the set of interest are non-analogous to those in the reference set.
- **detailed** - calculates five additional extrapolation metrics. See `mop_detailed` under `Value` below for full details.

`where_distance` options determine what values should be used to calculate dissimilarity

- **in_range** - only conditions inside `m` ranges
- **out_range** - only conditions outside `m` ranges
- **all** - all conditions

When the variables used to represent conditions have different units, scaling and centering are recommended. This step is only valid when Euclidean distances are used.

Value

A object of class `mop_results` containing:

- **summary** - a list with details of the data used in the analysis:
  - *variables* - names of variables considered.
  - *type* - type of MOP analysis performed.
  - *scale* - value according to the argument `scale`.
  - *center* - value according to the argument `center`.
  - *calculate_distance* - value according to the argument `calculate_distance`.
  - *distance* - option regarding distance used.

  - *percentage* - percentage of `m` used as reference for distance calculation.
  - *rescale_distance* - value according to the argument `rescale_distance`.
  - *fix_NA* - value according to the argument `fix_NA`.
  - *N_m* - total number of elements (cells with values or valid rows) in `m`.
  - *N_g* - total number of elements (cells with values or valid rows) in `g`.
  - *m_ranges* - matrix with ranges of variables in reference conditions (`m`).
- **mop_distances** - if `calculate_distance` = TRUE, a SpatRaster or vector with distance values for the set of interest (`g`). Higher values represent greater dissimilarity compared to the set of reference (`m`).
- **mop_basic** - a SpatRaster or vector, for the set of interest, representing conditions in which at least one of the variables is non-analogous to the set of reference. Values should be: 1 for non-analogous conditions, and NA for conditions inside the ranges of the reference set.
- **mop_simple** - a SpatRaster or vector, for the set of interest, representing how many variables in the set of interest are non-analogous to those in the reference set. NA is used for conditions inside the ranges of the reference set.
- **mop_detailed** - a list containing:
  - *interpretation_combined* - a data.frame to help identify combinations of variables in *towards_low_combined* and *towards_high_combined* that are non-analogous to `m`.
  - *towards_low_end* - a SpatRaster or matrix for all variables representing where non-analogous conditions were found towards low values of each variable.
  - *towards_high_end* - a SpatRaster or matrix for all variables representing where non-analogous conditions were found towards high values of each variable.
  - *towards_low_combined* - a SpatRaster or vector with values representing the identity of the variables found to have non-analogous conditions towards low values. If vector, interpretation requires the use of the data.frame *interpretation_combined*.
  - *towards_high_combined* - a SpatRaster or vector with values representing the identity of the variables found to have non-analogous conditions towards high values. If vector, interpretation requires the use of the data.frame *interpretation_combined*.

See Also

`mop_distance`, `out_range`

Examples

```
# data
reference_layers <- terra::rast(system.file("extdata", "reference_layers.tif",
                                            package = "mop"))

layers_of_interest <- terra::rast(system.file("extdata",
                                              "layers_of_interest.tif",
                                              package = "mop"))

# analysis
mop_res <- mop(m = reference_layers, g = layers_of_interest)

summary(mop_res)
```

`mop_distance` *MOP distance calculation*

Description

Calculates distances from each of the points of interest in `g_matrix` to a defined percentage of the reference conditions in `m_matrix`.

Usage

```
mop_distance(m_matrix, g_matrix, distance = "euclidean", percentage = 1,
             comp_each = 2000, tol = NULL, parallel = FALSE, n_cores = NULL,
             progress_bar = TRUE)
```

Arguments

`m_matrix`	matrix of variables representing the set of conditions to be used as reference. Each column represents a variable.
`g_matrix`	matrix of variables representing the set of conditions to be compared against the reference conditions (where distances are to be calculated). Each column represents a variable. Variable names must match those in `m_matrix`.
`distance`	`character`, one of two options: "euclidean" or "mahalanobis".
`percentage`	`numeric`, percentage of points of m (the closest ones) used to derive mean environmental distances to each g point.
`comp_each`	`numeric`, number of points of the g matrix to be used for distance calculations at a time (default = 2000). Increasing this number requires more RAM.
`tol`	tolerance to detect linear dependencies when calculating Mahalanobis distances. The default, NULL, uses `.Machine$double.eps`.
`parallel`	`logical`, if TRUE, calculations will be performed in parallel using `n_cores` of the computer. Using this option will speed up the analysis but will demand more RAM.
`n_cores`	`numeric`, number of cores to be used in parallel processing. Uses current host CPU cores - 1 by default.
`progress_bar`	`logical`, whether to show a progress bar for calculations. Valid when calculations are not run in parallel.

Value

A numeric vector with values of distances calculated according to parameters used.

Examples

```
# data
data("reference_matrix", package = "mop")
data("matrix_of_interest", package = "mop")

# analysis
mop_dist <- mop_distance(m_matrix = reference_matrix,
                         g_matrix = matrix_of_interest)
```

`mop_results` *Constructor of S3 objects of class mop_results*

Description

Constructor of S3 objects of class mop_results

Usage

```
new_mop_results(summary = new("list"), mop_distances = NULL,
                mop_basic = NULL, mop_simple = NULL,
                mop_detailed = new("list"))
```

Arguments

`summary`	a list with a summary of the data and parameters used in analysis. Default = empty list.
`mop_distances`	a `SpatRaster` or numeric vector of distances from the set of conditions of reference to the set of conditions of interest. Default = NULL.
`mop_basic`	a `SpatRaster` or numeric vector showing conditions in the set of interest outside the ranges in the reference set. The value `1` indicates conditions outside one or more ranges. Default = NULL.
`mop_simple`	a `SpatRaster` or numeric vector showing conditions in the set of interest outside the ranges in the reference set. Values indicate how many variables are outside reference ranges. Default = NULL.
`mop_detailed`	a list with a detailed representation of mop results in conditions outside the range of reference. Default = empty list.

Value

An object of class `mop_results`.

`out_range` *Detect values outside ranges of reference conditions*

Description

Options to identify which values in a set of conditions of interest (`g_matrix`) are outside the range of a set of conditions of reference (`m_matrix`).

Usage

```
out_range(m_matrix, g_matrix, type = "basic")
```

Arguments

`m_matrix`	matrix of variables representing the set of conditions to be used as reference. Each column represents a variable.
`g_matrix`	matrix of variables representing the set of conditions to be compared against the reference conditions (where conditions outside range are to be detected). Each column represents a variable. Variable names must match those in `m_matrix`.
`type`	`character`, type of identification to be performed. See `Details` for options.

Details

Results are produced according to `type`:

- **basic** - helps to identify conditions outside ranges, in general, one or variables are only counted as `1`. This is always returned.
- **simple** - identifies the number of variables with conditions outside ranges, for each condition of interest outside ranges, the number of non-analogous variables is returned.
- **detailed** - produces various results (including the two above):
  - *high_all* - identifies non-analogous conditions towards high values of variables, for each variable independently.
  - *low_all* - identifies non-analogous conditions towards low values of variables, for each variable independently.
  - *high_combined* - values are used to identify combinations of variables with non-analogous conditions towards high values of the variables.
  - *low_combined* - values are used to identify combinations of variables with non-analogous conditions towards low values of the variables.
  - *interpretation* - a `data.frame` to help identify which variables are considered in combined results.

Value

A list containing the ranges in `m_matrix`, results from analysis according to `type`, and table to help with interpretations. NA values represent conditions of interest inside ranges of reference conditions. See `Details`.

Examples

```
# data
data("reference_matrix", package = "mop")
data("matrix_of_interest", package = "mop")

# analysis
out <- out_range(m_matrix = reference_matrix,
                 g_matrix = matrix_of_interest)
```

`print` *Print a short version of elements in mop objects*

Description

Print a short version of elements in mop objects

Usage

```
## S3 method for class 'mop_results'
print(x, ...)
```

Arguments

`x`	object of class mop_results.
`...`	further arguments to be passed to or from other methods. Ignored in this function.

Value

A short description of objects in the console.

`reference_layers` *Example of variables for a set of reference*

Description

A `SpatRaster` object representing variables in a set of reference. Variables represent current bioclimatic variables downloaded from the WorldClim database (`https://worldclim.org/`).

Format

A `SpatRaster` object.

Value

No return value. Used with function `rast` to bring raster variables to analysis.

Examples

```
reference_layers <- terra::rast(system.file("extdata", "reference_layers.tif",
                                            package = "mop"))

terra::plot(reference_layers)
```

`reference_matrix` *Example of matrix with variables in a set of reference*

Description

A numeric table representing variables in a set of reference.

Usage

```
reference_matrix
```

Format

A matrix with 723 rows and 6 columns.

Examples

```
data("reference_matrix", package = "mop")

head(reference_matrix)
```

`summary` *Summary of attributes and results*

Description

Summary of attributes and results

Usage

```
## S3 method for class 'mop_results'
summary(object, ...)
```

Arguments

`object`	object of class mop_results.
`...`	additional arguments affecting the summary produced. Ignored in this function.

Value

A printed summary.

Index