Package ‘divraster’

May 26, 2025

Title Diversity Metrics Calculations for Rasterized Data

Version 1.2.1

Maintainer Flávio M. M. Mota <flaviomoc@gmail.com>

Description Alpha and beta diversity for taxonomic (TD), functional (FD), and phylogenetic (PD) dimensions based on rasters. Spatial and temporal beta diversity can be partitioned into replacement and richness difference components. It also calculates standardized effect size for FD and PD alpha diversity and the average individual traits across multilayer rasters. The layers of the raster represent species, while the cells represent communities. Methods details can be found at Cardoso et al. 2022 <https://CRAN.R-project.org/package=BAT> and Heming et al. 2023 <https://CRAN.R-project.org/package=SESraster>.

License GPL (>= 3)

URL https://github.com/flaviomoc/divraster,
https://flaviomoc.github.io/divraster/

BugReports https://github.com/flaviomoc/divraster/issues

Imports BAT, SESraster, terra, utils

Suggests ape, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Flávio M. M. Mota [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-0308-7151>),
Neander Marcel Heming [aut] (ORCID:
<https://orcid.org/0000-0003-2461-5045>),
Gabriela Alves-Ferreira [aut] (ORCID:
<https://orcid.org/0000-0001-5661-3381>)

Repository CRAN

Date/Publication 2025-05-26 21:50:02 UTC

Contents

area.calc *Calculate SpatRaster Layer Areas and Overlap Areas*

Description

Calculates the total area for each layer (e.g., species) within a SpatRaster object. Optionally, it can also compute the overlapping areas between the primary SpatRaster (x) and one or two additional single-layer SpatRaster objects (y and z). Results are returned as a data.frame and can optionally be saved to a CSV file.

Usage

```
area.calc(x, y = NULL, z = NULL, filename = "", unit = "km", cellsize = NULL)
```

Arguments

x	A SpatRaster object for which the area of each layer will be calculated. This SpatRaster can have one or multiple layers.
y	An optional SpatRaster object with a **single layer**. If provided, the overlapping area between each layer in x and this y raster will be calculated. It should have the same extent and resolution as x.
z	An optional SpatRaster object with a **single layer**. If provided, the overlapping area between each layer in x and this z raster, as well as the three-way overlap (x, y, and z), will be calculated. Requires y to also be provided. It should have the same extent and resolution as x.
filename	Character string. If provided (e.g., "results.csv"), the resulting data frame will be saved to a CSV file with this name. If not provided, results are returned only to the R session.

`unit` Character string specifying the unit of measurement for area calculations. Defaults to "km" (kilometers). Other options include "ha" (hectares), "m" (meters), etc.

`cellsize` Numeric. An optional value specifying the cell size (area of a single cell) to be used for calculations. If `NULL` (default), the function will automatically determine the cell size from the input raster `x`.

Value

A `data.frame` with the following columns:

- **Layer**: Name of each layer from the input `SpatRaster x`.
- **Area**: The calculated area for each layer in `x` (e.g., total species range area).
- **Overlap_Area_Y** (optional): If `y` is provided, the area where the `x` layer and `y` raster both have a value of 1 (overlap).
- **Overlap_Area_Z** (optional): If `z` is provided, the area where the `x` layer and `z` raster both have a value of 1 (overlap).
- **Overlap_Area_All** (optional): If both `y` and `z` are provided, the area where the `x` layer, `y` raster, and `z` raster all have a value of 1 (triple overlap).

Areas are reported in the specified `unit`.

Examples

```
library(terra)

# Load example rasters for demonstration
# Ensure these files are present in your package's inst/extdata folder
bin_rast <- terra::rast(system.file("extdata", "ref.tif", package = "divraster"))

# Example 1: Calculate area for 'bin_rast' only
area_only <- area.calc(bin_rast)
area_only
```

`differ.rast` *Calculate Absolute or Percentage Difference Between SpatRaster Objects*

Description

Computes the difference between two `SpatRaster` objects, either as an absolute value or as a percentage of change relative to the first raster (`r1`). This function is commonly used to assess changes in spatial patterns, such as shifts in species richness or environmental variables over time or between scenarios.

Usage

```
differ.rast(r1, r2, perc = TRUE, filename = "")
```

Arguments

r1	A SpatRaster object representing the baseline or initial values. Can have one or multiple layers.
r2	A SpatRaster object representing the future or comparison values. Must have the same dimensions, resolution, CRS, and number of layers as r1.
perc	Logical (default is TRUE). If TRUE, the percentage of change relative to r1 is calculated: ((r2 - r1) / r1) * 100. If FALSE, the absolute difference (r2 - r1) is returned.
filename	Character string. Optional path and filename to save the resulting SpatRaster. Supported formats are those recognized by terra::writeRaster (e.g., ".tif", ".grd"). If provided, the SpatRaster will be saved to this file.

Details

This function performs a cell-wise subtraction (r2 - r1).

- For **percentage difference**, the formula used is ((r2 - r1) / r1) * 100. Cells where r1 is NA or 0 will result in NA in the output SpatRaster for percentage calculations, to avoid division by zero or meaningless percentages.
- It is crucial that r1 and r2 are aligned spatially (same extent, resolution, and Coordinate Reference System - CRS) and have the same number of layers, with corresponding layers representing the same variable or species.

Value

A SpatRaster object containing the calculated differences.

- If perc = TRUE, the layer name will be "Percentage_Difference".
- If perc = FALSE, the layer name will be "Absolute_Difference".

The output SpatRaster will have the same dimensions, resolution, and CRS as the input rasters.

Examples

```
library(terra)

# Load rasters
rich1 <- terra::rast(system.file("extdata", "rich_ref.tif",
package = "divraster"))
rich2 <- terra::rast(system.file("extdata", "rich_fut.tif",
package = "divraster"))

# Calculate absolute difference in richness
abs_diff_rast <- differ.rast(rich1, rich2, perc = FALSE)
abs_diff_rast
```

```
plot(abs_diff_rast, main = "Absolute Difference in Richness")

# Calculate percentage difference in richness
perc_diff_rast <- differ.rast(rich1, rich2, perc = TRUE)
perc_diff_rast
plot(perc_diff_rast, main = "Percentage Difference in Richness")
```

`inputs_chk` *Check if objects are valid*

Description

Check if objects are valid

Usage

```
inputs_chk(bin1, bin2, tree)
```

Arguments

`bin1` A SpatRaster with presence-absence data (0 or 1) for a set of species.

`bin2` A SpatRaster with presence-absence data (0 or 1) for a set of species. Species names in ’bin2’ and ’bin1’ must match!

`tree` It can be a ’data.frame’ with species traits or a ’phylo’ with a rooted phylogenetic tree. Species names in ’tree’, ’bin1’, and ’bin2’ must match!

Value

Either a success message or an error.

`load.data` *Load data adapted from Mota et al. (2022), Tobias et al. (2022), and Jetz et al. (2014)*

Description

Load data adapted from Mota et al. (2022), Tobias et al. (2022), and Jetz et al. (2014)

Usage

```
load.data()
```

Value

A list with binary maps of species for the reference and future climate scenarios, species traits, and a rooted phylogenetic tree for the species. The species names across these objects must match!

References

Mota, F. M. M. et al. 2022. Climate change is expected to restructure forest frugivorous bird communities in a biodiversity hot-point within the Atlantic Forest. - Diversity and Distributions 28: 2886–2897.

Tobias, J. A. et al. 2022. AVONET: morphological, ecological and geographical data for all birds. - Ecology Letters 25: 581–597.

Jetz, W. et al. 2014. Global Distribution and Conservation of Evolutionary Distinctness in Birds. - Current Biology 24: 919–930.

Examples

```
data <- load.data()
data
```

`spat.alpha` *Alpha calculation for raster*

Description

Calculates alpha diversity for taxonomic (TD), functional (FD), and phylogenetic (PD) dimensions. Adapted from `alpha`

Usage

```
spat.alpha(bin, tree, cores = 1, filename = "", ...)
```

Arguments

`bin`	A SpatRaster with presence-absence data (0 or 1) for a set of species.
`tree`	It can be a 'data.frame' with species traits or a 'phylo' with a rooted phylogenetic tree. Species names in 'tree' and 'bin' must match!
`cores`	A positive integer. If cores > 1, a 'parallel' package cluster with that many cores is created and used.
`filename`	Character. Save results if a name is provided.
`...`	Additional arguments to be passed passed down from a calling function.

Details

Alpha calculations use a tree-based approach for TD, FD, and PD (Cardoso et al. 2014). In the FD calculation, a species traits matrix is transformed into a distance matrix and clustered to create a regional dendrogram (i.e. a dendrogram with all species in the raster stack), from which the total branch length is calculated. When computing FD for each community (i.e. raster cell), the regional dendrogram is subsetted to create a local dendrogram that includes only the species present in the local community. The branch lengths connecting these species are then summed to represent the functional relationships of the locally present species (Petchey and Gaston, 2002, 2006). Similarly, in PD, the cumulative branch lengths connecting species within a community indicate their shared phylogenetic relationships (Faith, 1992). Alpha TD can also be visualized using a tree diagram, where each species is directly connected to the root by an edge of unit length, reflecting the number of different taxa in the community (i.e. species richness) since all taxa are at the same level (Cardoso et al. 2014).

Value

A SpatRaster with alpha result.

References

Cardoso, P. et al. 2014. Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. - Journal of Biogeography 41: 749–761.

Faith, D. P. 1992. Conservation evaluation and phylogenetic diversity. - Biological Conservation 61: 1–10.

Petchey, O. L. and Gaston, K. J. 2002. Functional diversity (FD), species richness and community composition. - Ecology Letters 5: 402–411.

Rodrigues, A. S. L. and Gaston, K. J. 2002. Maximising phylogenetic diversity in the selection of networks of conservation areas. - Biological Conservation 105: 103–111.

Examples

```
library(terra)
bin1 <- terra::rast(system.file("extdata", "ref.tif",
package = "divraster"))
traits <- read.csv(system.file("extdata", "traits.csv",
package = "divraster"), row.names = 1)
tree <- ape::read.tree(system.file("extdata", "tree.tre",
package = "divraster"))
spat.alpha(bin1)
spat.alpha(bin1, traits)
spat.alpha(bin1, tree)
```

spat.alpha.vec *Alpha calculation for vector*

Description

Alpha calculation for vector

Usage

```
spat.alpha.vec(x, tree, resu, ...)
```

Arguments

`x` A numeric vector with presence-absence data (0 or 1) for a set of species.

`tree` It can be a data frame with species traits or a phylogenetic tree.

`resu` Numeric. A vector to store results.

`...` Additional arguments to be passed passed down from a calling function.

Value

A vector with alpha result.

`spat.alpha2` *Alternative Method to Calculate Alpha Taxonomic Diversity*

Description

Calculates the alpha taxonomic diversity, specifically **species richness**, for each cell in a `SpatRaster` object containing species presence-absence data. This function provides a straightforward method to sum the number of species present in each grid cell.

Usage

```
spat.alpha2(bin, cores = 1, filename = "")
```

Arguments

`bin` A `SpatRaster` object with multiple layers, where each layer represents a species and cell values are binary (0 for absence, 1 for presence). Species names should correspond to layer names (e.g., `names(bin)`).

`cores` A positive integer (default is 1). If `cores > 1`, a parallel processing cluster is created using the `parallel` package to speed up calculations across raster cells.

`filename` Character string. Optional path and filename to save the resulting `SpatRaster`. Supported formats are those recognized by `terra::writeRaster` (e.g., ".tif", ".grd"). If provided, the `SpatRaster` will be saved to this file.

Details

This function calculates species richness by summing the presence (value 1) of all species across layers for each individual raster cell. It is an alternative to `spat.alpha()` when only Taxonomic Diversity (TD) is required, offering a more direct and potentially faster computation for this specific metric. `NA` values in input cells are ignored during the sum calculation.

Value

A `SpatRaster` object with a single layer named "Richness". Each cell in this `SpatRaster` contains the calculated species richness (number of species present). The output `SpatRaster` will have the same dimensions, resolution, and CRS as the input `bin`.

Examples

```
library(terra)

# Load an example SpatRaster with binary presence-absence data
bin_rast <- terra::rast(system.file("extdata", "ref.tif", package = "divraster"))

# Calculate species richness (alpha taxonomic diversity)
richness_map <- spat.alpha2(bin_rast)
richness_map

# Plot the resulting richness map
plot(richness_map, main = "Species Richness Map")
```

`spat.beta` *Spatial beta diversity for raster*

Description

Calculates spatial beta diversity for taxonomic (TD), functional (FD), and phylogenetic (PD) dimensions. See `raster.beta`.

Usage

```
spat.beta(x, tree, filename = "", func = "jaccard", abund = FALSE, ...)
```

Arguments

`x` A SpatRaster with presence-absence data (0 or 1) for a set of species. (This maps to `layers` in `BAT::raster.beta`).

`tree` It can be a 'data.frame' with species traits or a 'phylo' with a rooted phylogenetic tree. Species names in 'tree' and 'x' must match!

`filename` Character. Save results if a name is provided.

`func` Character. Distance function for beta diversity calculation. Defaults to "jaccard". Passed to `BAT::beta`.

`abund` Logical. Whether to use abundance data (TRUE) or presence-absence (FALSE). Defaults to FALSE. Passed to `BAT::beta`.

`...` Additional arguments to be passed to internal functions within `BAT::raster.beta` (e.g., `BAT::beta`). Note: `BAT::raster.beta` does not accept a 'neighbour' argument.

Value

A SpatRaster with beta results (total, replacement, richness difference, and ratio).

Examples

```
library(terra)
bin1 <- terra::rast(system.file("extdata", "fut.tif",
package = "divraster"))
traits <- read.csv(system.file("extdata", "traits.csv",
package = "divraster"), row.names = 1)
tree <- ape::read.tree(system.file("extdata", "tree.tre",
package = "divraster"))
spat.beta(bin1)
spat.beta(bin1, traits)
spat.beta(bin1, tree)
```

`spat.rand` *Standardized Effect Size (SES) for raster*

Description

Calculates the standardized effect size for functional and phylogenetic alpha diversity. See `bootspat_str` and `bootspat_naive`

Usage

```
spat.rand(
  x,
  tree,
  aleats,
  random = c("site", "species", "both", "spat"),
  cores = 1,
  filename = "",
  ...
)
```

Arguments

`x` SpatRaster. A SpatRaster containing presence-absence data (0 or 1) for a set of species.

`tree` It can be a 'data.frame' with species traits or a 'phylo' with a rooted phylogenetic tree. Species names in 'tree' and 'x' must match!

`aleats` positive integer. A positive integer indicating how many times the calculation should be repeated.

`random` character. A character indicating the type of randomization. The currently available randomization methods are "spat", "site", "species" or "both" (site and species).

`cores` positive integer. If cores > 1, a 'parallel' package cluster with that many cores is created and used.

`filename` character. Output filename.

`...` additional arguments to be passed passed down from a calling function.

Value

SpatRaster with Mean, SD, Observed, and SES.

Examples

```
x <- terra::rast(system.file("extdata", "ref.tif",
package = "divraster"))
traits <- read.csv(system.file("extdata", "traits.csv",
package = "divraster"), row.names = 1)
tree <- ape::read.tree(system.file("extdata", "tree.tre",
package = "divraster"))
spat.rand(x, tree, 3, "site")
spat.rand(x, traits, 3, "site")
```

`spat.trait` *Average trait calculation for raster*

Description

Compute average for each trait.

Usage

```
spat.trait(x, trait, cores = 1, filename = "", ...)
```

Arguments

`x`	A SpatRaster with presence-absence data (0 or 1) for a set of species.
`trait`	A 'data.frame' with species traits. Rownames must have species names that match with 'x'!
`cores`	A positive integer. If cores > 1, a 'parallel' package cluster with that many cores is created and used.
`filename`	Character. Save results if a name is provided.
`...`	Additional arguments to be passed passed down from a calling function.

Value

SpatRaster with average traits.

Examples

```
library(terra)
bin1 <- terra::rast(system.file("extdata", "ref.tif",
package = "divraster"))
traits <- read.csv(system.file("extdata", "traits.csv",
package = "divraster"), row.names = 1)
spat.trait(bin1, traits)
```

spat.trait.vec *Average trait calculation for vector*

Description

Average trait calculation for vector

Usage

```
spat.trait.vec(x, col_trait, ...)
```

Arguments

`x`	A numeric vector with presence-absence data (0 or 1) for a set of species.
`col_trait`	A numeric vector with trait numbers.
`...`	Additional arguments to be passed passed down from a calling function.

Value

Vector of average trait.

suit.change *Species Suitability Change Between Climate Scenarios*

Description

Compares two `SpatRaster` objects, each containing species presence-absence data for multiple species under different climate scenarios (e.g., baseline vs. future). It calculates and encodes the change in habitat suitability (gain, loss, unchanged, unsuitable) for each species in each raster cell.

Usage

```
suit.change(r1, r2, filename = "")
```

Arguments

`r1`	A `SpatRaster` with multiple layers. Each layer represents a species' presence-absence data (0 for absence, 1 for presence) for the **baseline climate scenario**. Layer names should correspond to species names.
`r2`	A `SpatRaster` with multiple layers. Each layer represents a species' presence-absence data (0 for absence, 1 for presence) for the **future climate scenario**. Layer names should correspond to species names and must match those in `r1`.
`filename`	Character string. Optional path and filename to save the resulting `SpatRaster` stack. Supported formats are those recognized by `terra::writeRaster` (e.g., ".tif", ".grd"). If provided, the `SpatRaster` will be saved to this file.

Details

This function processes each species layer independently. It's crucial that both input `SpatRasters` (`r1` and `r2`) have the same extent, resolution, and the same number of layers, with corresponding layers representing the same species. The function expects binary (0 or 1) presence-absence data.

Value

A `SpatRaster` object with multiple layers, where each layer corresponds to a species from the input SpatRasters. Cell values are encoded as follows: 1 = Gain: Species absent in r1 (baseline) becomes present in r2 (future). 2 = Loss: Species present in r1 (baseline) becomes absent in r2 (future). 3 = Unchanged (Presence): Species present in both r1 and r2. 4 = Unsuitable (Both): Species absent in both r1 and r2. The dimensions, resolution, and layer names of the output raster will match those of the input r1 and r2.

Examples

```
library(terra)

# Load example rasters for baseline and future climate scenarios
r1 <- terra::rast(system.file("extdata", "ref.tif", package = "divraster"))
r2 <- terra::rast(system.file("extdata", "fut.tif", package = "divraster"))

# Calculate suitability change
change_map <- suit.change(r1, r2)
change_map
```

`temp.beta` *Temporal beta diversity calculation for raster*

Description

Calculates temporal beta diversity for taxonomic (TD), functional (FD), and phylogenetic (PD) dimensions. Adapted from `beta`

Usage

```
temp.beta(bin1, bin2, tree, filename = "", cores = 1, ...)
```

Arguments

`bin1`	A SpatRaster with presence-absence data (0 or 1) for a set of species.
`bin2`	A SpatRaster with presence-absence data (0 or 1) for a set of species. Species names in 'bin2' and 'bin1' must match!
`tree`	It can be a 'data.frame' with species traits or a 'phylo' with a rooted phylogenetic tree. Species names in 'tree', 'bin1', and 'bin2' must match!
`filename`	Character. Save results if a name is provided.

`cores` A positive integer. If cores > 1, a 'parallel' package cluster with that many cores is created and used.

`...` Additional arguments to be passed passed down from a calling function.

Details

The TD beta diversity partitioning framework we used was developed by Podani and Schmera (2011) and Carvalho et al. (2012) and expanded to PD and FD by Cardoso et al. (2014).

Value

A SpatRaster with beta results (total, replacement, richness difference, and ratio).

References

Cardoso, P. et al. 2014. Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. - Journal of Biogeography 41: 749–761.

Carvalho, J. C. et al. 2012. Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. - Global Ecology and Biogeography 21: 760–771.

Podani, J. and Schmera, D. 2011. A new conceptual and methodological framework for exploring and explaining pattern in presence - absence data. - Oikos 120: 1625–1638.

Hidasi-Neto, J. et al. 2019. Climate change will drive mammal species loss and biotic homogenization in the Cerrado Biodiversity Hotspot. - Perspectives in Ecology and Conservation 17: 57–63.

Examples

```
library(terra)
bin1 <- terra::rast(system.file("extdata", "ref.tif",
package = "divraster"))
bin2 <- terra::rast(system.file("extdata", "fut.tif",
package = "divraster"))
traits <- read.csv(system.file("extdata", "traits.csv",
package = "divraster"), row.names = 1)
tree <- ape::read.tree(system.file("extdata", "tree.tre",
package = "divraster"))
temp.beta(bin1, bin2)
temp.beta(bin1, bin2, traits)
temp.beta(bin1, bin2, tree)
```

`temp.beta.vec` *Temporal beta diversity calculation for vector*

Description

Temporal beta diversity calculation for vector

Usage

```
temp.beta.vec(x, nspp, spp, tree, resu, ...)
```

Arguments

`x`	A numeric vector with presence-absence data (0 or 1) for a set of species.
`nspp`	Numeric. Number of species.
`spp`	Character. Species name.
`tree`	It can be a data frame with species traits or a phylogenetic tree.
`resu`	Numeric. A vector to store results.
`...`	Additional arguments to be passed passed down from a calling function.

Value

A vector with beta results (total, replacement, richness difference, and ratio).

Index