Package 'ZonationR'

April 21, 2026

Title Interface to 'Zonation' for Reproducible Prioritization Workflows

Version 1.0.1

Description An interface to 'Zonation' software, enabling users to run spatial conservation prioritization workflows in 'R'. It streamlines input preparation, execution, and post-processing, while supporting reproducibility and lowering the entry barrier for learning, teaching, and research in conservation planning. The methods implemented in 'Zonation' are described in Moilanen et al. (2022) <doi:10.1111/2041-210X.13819>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports dplyr, ggplot2, ggspatial, rlang, terra, tidyr, utils, viridis

Suggests knitr, mockery, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

URL https://github.com/thiago-cav/ZonationR, https://thiago-cav.github.io/ZonationR/

BugReports https://github.com/thiago-cav/ZonationR/issues

Config/Needs/website rmarkdown

Depends R (>= 4.1.0)

NeedsCompilation no

Author Thiago Cavalcante [aut, cre] (ORCID: <https://orcid.org/0000-0001-5357-9659>), Bruno Ribeiro [aut] (ORCID: <https://orcid.org/0000-0002-7755-6715>), Karlo Guidoni-Martins [aut] (ORCID: <https://orcid.org/0000-0002-8458-8467>), Heini Kujala [aut] (ORCID: <https://orcid.org/0000-0001-9772-3202>)

Maintainer Thiago Cavalcante <thiagocav.ferreira@gmail.com>

Repository CRAN

Date/Publication 2026-04-21 19:22:08 UTC

Contents

`check_dir_writable` *Check that an output directory is writable*

Description

This helper validates that a given directory exists and that the current R process has permission to create and delete a temporary file in it. It is intended for use before writing output files.

Usage

```
check_dir_writable(out_dir)
```

Arguments

`out_dir`	`character(1)`. Path to the directory that should be writable.

Value

Returns `TRUE` invisibly when the directory exists and is writable. The function otherwise throws an error describing the problem:

- if the directory does not exist, an error stating that it does not exist;
- if the directory is not writable, an error stating that it is not writable.

See Also

Other preflight: `check_raster_uniformity()`, `check_zonation_executable()`

Examples

```
dir <- tempdir()
check_dir_writable(dir)
```

`check_raster_uniformity` *Check raster uniformity*

Description

Verifies that all raster files in a folder are compatible with Zonation. Compatibility requires that all rasters have:

- identical resolution (cell size)
- identical extent (number of rows and columns)
- identical projection (CRS)

Usage

```
check_raster_uniformity(spp_file_dir)
```

Arguments

`spp_file_dir`	Character. Path to the folder containing raster files.

Details

Supported raster formats: `.tif`, `.tiff`, `.img`, `.asc`.

Value

Prints a message confirming uniformity. Stops with an error otherwise.

See Also

Other preflight: `check_dir_writable()`, `check_zonation_executable()`

Examples

```
withr::with_tempdir({

  r1 <- terra::rast(nrows = 5, ncols = 5)
  terra::values(r1) <- runif(terra::ncell(r1))

  r2 <- terra::rast(r1)
  terra::values(r2) <- runif(terra::ncell(r2))
```

```
  terra::writeRaster(r1, "r1.tif", overwrite = TRUE)
  terra::writeRaster(r2, "r2.tif", overwrite = TRUE)

  check_raster_uniformity(".")
})
```

`check_zonation_executable` *Check for Zonation 5 executable*

Description

This function checks if the Zonation 5 executable is available on the system. It can verify a specific path provided by the user or search common installation locations. The function works on both Windows and Linux systems.

Usage

```
check_zonation_executable(zonation_path = NULL, os = NULL)
```

Arguments

`zonation_path`	Optional character string specifying the path to check. If not provided, the function will search common installation locations.
`os`	Operating system. If NULL (default), automatically detected from the system. Can be set to "Windows" or "Linux" manually.

Details

For Windows, the function searches for `z5.exe` in:

- The provided path (if given)
- `C:/Program Files (x86)/Zonation5`
- `C:/Program Files/Zonation5`

For Linux, the function searches for `zonation5` in:

- The provided path (if given)
- Common locations like `~/Applications`, `~/bin`, `/usr/local/bin`, `/usr/bin`
- The system PATH

Value

A list with the following components:

found	Logical indicating if the executable was found
path	Character string with the path where the executable was found (or NULL)
executable	Character string with the full path to the executable (or NULL)
os	Character string indicating the detected operating system
message	Character string with a descriptive message

See Also

Other preflight: check_dir_writable(), check_raster_uniformity()

Examples

```
check_zonation_executable()
```

---

command_file *Create a Zonation command file*

Description

This function generates a command file for running Zonation and specifies the analysis options and related parameters. The file is saved with a .cmd (Windows) or .sh (Linux) suffix.

Usage

```
command_file(
  os = "os_detection",
  zonation_path,
  flags = "",
  marginal_loss_mode = "CAZ2",
  gui_activated = FALSE,
  settings_file = "settings_file.z5",
  output_dir = "output"
)
```

Arguments

os	Operating system. By default, the system is detected automatically. Alternatively, users can explicitly specify "Windows" or "Linux".
zonation_path	The specification for the path where Zonation 5 is installed.
flags	Flags that control which analysis options are used. Used to include single letter codes that switch analysis options on. Available options are: a, w, g, h, x, X, t.

`marginal_loss_mode`
: Character string specifying the marginal loss rule. Available options are "CAZ1", "CAZ2", "ABF", "CAZMAX", "LOAD", and "RAND". Default is "CAZ2".

`gui_activated`
: This parameter controls whether the Zonation Graphical User Interface (GUI) is launched when running the command file. Default is FALSE.

`settings_file`
: Character string specifying the settings file. Default is "settings_file.z5".

`output_dir`
: A character string specifying the name of the output directory where the analysis results will be saved. Default is "output".

Value

A Zonation command file containing the specified analysis options.

See Also

Other preprocessing: `feature_list()`, `settings_file()`

Examples

```
withr::with_tempdir({

  tmp_settings <- tempfile(fileext = ".z5")
  writeLines("feature list file = feature_list.txt", tmp_settings)

  command_file(
    zonation_path = "path/to/zonation",
    settings_file = tmp_settings
  )

  command_file(
    zonation_path = "path/to/zonation",
    marginal_loss_mode = "ABF",
    gui_activated = TRUE,
    settings_file = tmp_settings
  )
})
```

`cost_summary` *Summarize remaining cost at specified landscape proportions*

Description

Reads a Zonation `summary_curves.csv` file (space-separated), finds the row closest to each value in `landscape_prop`, and returns remaining cost and percentage of maximum cost at those proportions. Optionally writes the result to `dir/performance/cost_summary.csv`.

Usage

```
cost_summary(
  dir,
  output_folder_name = "output",
  landscape_prop,
  save_output = FALSE
)
```

Arguments

dir	Character. Path to the directory containing the Zonation output folder.
output_folder_name	Character. Name of the output folder inside dir. Default is "output".
landscape_prop	Numeric vector. Landscape proportions (rank values) at which to report cost (e.g. c(0.03, 0.2, 0.5)).
save_output	Logical. If TRUE, the result is written as a CSV to dir/performance/cost_summary.csv. Default FALSE.

Value

A data frame with columns:

- rank - the requested proportion
- remaining_cost - remaining cost at that rank
- percentage - remaining cost as percentage of maximum cost

See Also

Other postprocessing: coverage_distribution(), feature_curves(), feature_representation(), priority_map(), rank_similarity(), summary_curves()

Examples

```
withr::with_tempdir({

  dir.create("output")

  # simulate Zonation-style summary_curves.csv structure
  write.table(
    data.frame(
      rank = seq(0, 1, length.out = 10),
      remaining_cost = runif(10, 50, 100),
      dummy1 = runif(10),
      dummy2 = runif(10)
    ),
    file = file.path("output", "summary_curves.csv"),
    row.names = FALSE,
    col.names = TRUE,
    sep = " "
```

```
  )

  cost_summary(
    dir = ".",
    output_folder_name = "output",
    landscape_prop = c(0.2, 0.5, 0.8)
  )

})
```

`coverage_distribution` *Plot coverage distribution at a given rank*

Description

This function reads a Zonation feature curves file and plots the distribution of feature coverage values at a specified priority rank. The output is a ggplot object, which can be further customized by the user. Optionally, the plot can be saved to disk as a high-quality figure.

Usage

```
coverage_distribution(
  dir,
  output_folder_name = "output",
  target_rank,
  save_path = NULL,
  dpi = 300,
  width = 8,
  height = 6
)
```

Arguments

`dir`	Character. Path to the directory containing the Zonation output folder.
`output_folder_name`	Character. Name of the Zonation output folder inside `dir` (default: "output").
`target_rank`	Numeric. The rank value at which coverage distributions should be extracted and plotted.
`save_path`	Character. Optional file path to save the plot. The file format is inferred from the specified extension (e.g. ".png", ".pdf", ".tiff").
`dpi`	Numeric. Resolution (dots per inch) for saved figures. Default is 300.
`width`	Numeric. Width of the saved figure in inches. Default is 8.
`height`	Numeric. Height of the saved figure in inches. Default is 6.

Value

A ggplot object showing the distribution of feature coverage values at the specified priority rank.

See Also

Other postprocessing: cost_summary(), feature_curves(), feature_representation(), priority_map(), rank_similarity(), summary_curves()

Examples

```
withr::with_tempdir({

  data_path <- system.file(
    "extdata",
    "feature_curves.csv",
    package = "ZonationR"
  )

  dir.create("output")

  file.copy(
    data_path,
    file.path("output", "feature_curves.csv"),
    overwrite = TRUE
  )

  p <- coverage_distribution(
    dir = ".",
    output_folder_name = "output",
    target_rank = 0.9
  )

  print(p)

})
```

---

feature_curves *Plot feature performance curves*

Description

Reads the Zonation feature_curves.csv and features_info.csv files, standardizes the feature names, and plots representation vs priority rank for each feature. Supports a single color with transparency or the viridis palette.

Usage

```
feature_curves(
  dir,
  output_folder_name = "output",
  palette = "gray",
  alpha = 0.3,
  show_legend = FALSE,
  save_path = NULL,
  dpi = 300,
  width = 8,
  height = 6
)
```

Arguments

dir	Character. Path to the directory containing the Zonation output folder.
output_folder_name	Character. Name of the Zonation output folder (default: "output").
palette	Character. Either "gray" (default) for a single-color plot, a single color name, or "viridis" to use the viridis palette for multiple features.
alpha	Numeric. Transparency of lines (0 fully transparent, 1 fully opaque). Default is 0.3.
show_legend	Logical. If TRUE, shows legend (only applicable when palette = "viridis"). Default is FALSE.
save_path	Character. Optional path to save the plot as PNG. Default is NULL (does not save).
dpi	Numeric. Resolution (dots per inch) for saved figures. Default is 300.
width	Numeric. Width of the saved figure in inches. Default is 12.
height	Numeric. Height of the saved figure in inches. Default is 8.

Value

A `ggplot` object representing feature curves. If `save_path` is provided, also saves the plot as PNG.

See Also

Other postprocessing: `cost_summary()`, `coverage_distribution()`, `feature_representation()`, `priority_map()`, `rank_similarity()`, `summary_curves()`

Examples

```
withr::with_tempdir({

  data_path <- system.file(
    "extdata",
    package = "ZonationR"
  )
```

```
  dir.create("output")

  file.copy(
    file.path(data_path, "feature_curves.csv"),
    "output/feature_curves.csv",
    overwrite = TRUE
  )

  file.copy(
    file.path(data_path, "features_info.csv"),
    "output/features_info.csv",
    overwrite = TRUE
  )

  p1 <- feature_curves(
    dir = ".",
    output_folder_name = "output"
  )

  print(p1)

  # Plot using viridis palette with legend
  p2 <- feature_curves(
    dir = ".",
    output_folder_name = "output",
    palette = "viridis",
    show_legend = TRUE
  )

  print(p2)

})
```

`feature_list` *Create a feature list from raster files*

Description

This function generates a feature list from raster files in a specified directory, adding optional attributes to the list based on user-defined parameters. The resulting feature list is written to a text file. Supported raster formats include GeoTIFF (.tif, .tiff), ERDAS Imagine (.img), and ASCII Grid (.asc).

Usage

```
feature_list(spp_file_dir, weight = NULL, group = NULL, threshold = NULL)
```

Arguments

`spp_file_dir`	A character string specifying the directory containing the raster files.
`weight`	An optional numeric vector (`float`) to assign weights to the features in the list.
`group`	An optional integer vector (`int`) representing the output group number for each raster.
`threshold`	An optional numeric vector (`float`) representing threshold values for each raster. If thresholding is applied, values in the input raster below the threshold are set to zero. Layers that contain only zeros after thresholding are removed from the analysis.

Value

A text file containing a feature list of rasters along with any additional attributes specified by the user.

See Also

Other preprocessing: `command_file()`, `settings_file()`

Examples

```
withr::with_tempdir({

  file.create("species1.tif")
  file.create("species2.tif")

  feature_list(spp_file_dir = ".")
})
```

`feature_representation` *Calculate feature representation within an area*

Description

This function calculates the representation of each feature across raster cells, and can optionally summarize results within a specified area.

Usage

```
feature_representation(feature_layers, area_mask = NULL)
```

Arguments

feature_layers	A terra::SpatRaster with one or more layers representing feature distributions (e.g., species distributions, habitat suitability).
area_mask	Optional. A terra::SpatRaster or terra::SpatVector defining the analysis area. If a raster is provided, it should follow the Zonation analysis area mask convention, where cells with value 1 represent the area of interest and cells with value 0 or NA are excluded. If a SpatVector is provided, it will be rasterized to match the resolution and extent of feature_layers.

Value

A list with two elements:

representation_layers A terra::SpatRaster where each layer contains the fractional representation of the corresponding feature across the landscape. Each cell value represents the proportion of the global total representation of that feature occurring in that cell.

representation_in_area A named numeric vector containing the representation of each feature within the specified area_mask. If no area is provided, this element returns NULL.

See Also

Other postprocessing: cost_summary(), coverage_distribution(), feature_curves(), priority_map(), rank_similarity(), summary_curves()

Examples

```
r <- terra::rast(nrows = 10, ncols = 10)
f1 <- terra::setValues(r, runif(terra::ncell(r)))
f2 <- terra::setValues(r, runif(terra::ncell(r)))
features <- c(f1, f2)
names(features) <- c("feature_1", "feature_2")

mask <- r
terra::values(mask) <- sample(c(0,1), terra::ncell(mask), replace = TRUE)

result <- feature_representation(features, mask)
result$representation_in_area
```

---

priority_map *Plot priority ranking maps*

Description

priority_map() reads a rank map raster from a specified folder and creates a ggplot2 map. It can plot the raster as continuous values or classify it into discrete categories with custom breaks and labels.

Usage

```
priority_map(
  dir,
  output_folder_name = "output",
  breaks = NULL,
  labels = NULL,
  palette = "viridis",
  classify = FALSE,
  show_legend = TRUE,
  save_path = NULL,
  dpi = 300,
  width = 8,
  height = 6
)
```

Arguments

dir	Character. Path to the variant folder containing the output folder.
output_folder_name	Character. Name of the output folder inside dir. Default is "output".
breaks	Numeric vector. Break points for reclassifying rank values. Required only if classify = TRUE. Must have length equal to length(labels) + 1.
labels	Character vector. Category labels for each reclassification interval. Required only if classify = TRUE. Length must equal length(breaks) - 1.
palette	Character or vector of colors. If "viridis", the function uses a reversed viridis palette (option "H") for classified maps. Otherwise, a vector of colors can be provided. Default is "viridis".
classify	Logical. If TRUE, raster values are converted into classes using breaks and labels. Default is FALSE.
show_legend	Logical. Whether to display the legend. Default is TRUE.
save_path	Character. File path to save the plot. If NULL (default), the plot is not saved.
dpi	Numeric. Resolution (dots per inch) for saved figures. Default is 300.
width	Numeric. Width of the saved figure in inches. Default is 8.
height	Numeric. Height of the saved figure in inches. Default is 6.

Value

A ggplot object representing the priority map.

See Also

Other postprocessing: cost_summary(), coverage_distribution(), feature_curves(), feature_representation(), rank_similarity(), summary_curves()

Examples

```
withr::with_tempdir({

  data_path <- system.file(
    "extdata",
    package = "ZonationR"
  )

  dir.create("output")

  # copy example rankmap
  file.copy(
    file.path(data_path, "rankmap.tif"),
    "output/rankmap.tif",
    overwrite = TRUE
  )

  # ---- continuous map ----
  p1 <- priority_map(
    dir = ".",
    output_folder_name = "output",
    classify = FALSE
  )

  print(p1)

  # ---- classified map ----
  breaks <- c(0, 0.25, 0.5, 0.75, 1)
  labels <- c("Low", "Medium", "High", "Very High")

  p2 <- priority_map(
    dir = ".",
    output_folder_name = "output",
    classify = TRUE,
    breaks = breaks,
    labels = labels
  )

  print(p2)

})
```

`rank_similarity` *Calculate similarity between Zonation rank maps*

Description

This function calculates similarity between two spatial rank maps using Schoener's D or the Jaccard index.

Usage

```
rank_similarity(
  r1 = NULL,
  r2 = NULL,
  rstack = NULL,
  method = c("schoener", "jaccard"),
  threshold = NULL
)
```

Arguments

r1	A SpatRaster rank map (from the terra package). Required if rstack is not provided.
r2	A SpatRaster rank map with the same dimensions as r1. Required if rstack is not provided.
rstack	A SpatRaster with multiple layers. If provided, the function calculates pairwise coefficients between all layers.
method	Character, either "schoener" (default) or "jaccard".
threshold	Numeric, required if method = "jaccard". Cells with rank greater than or equal to this threshold are considered selected.

Value

A numeric value between 0 and 1 if comparing two rasters, or a matrix of pairwise coefficients if using rstack.

See Also

Other postprocessing: cost_summary(), coverage_distribution(), feature_curves(), feature_representation(), priority_map(), summary_curves()

Examples

```
# Create two example priority rank maps
r1 <- terra::rast(nrows = 5, ncols = 5)
terra::values(r1) <- runif(terra::ncell(r1))

r2 <- terra::rast(r1)
terra::values(r2) <- runif(terra::ncell(r2))

# Schoener's D
rank_similarity(r1, r2, method = "schoener")

# Jaccard index with threshold
```

```
rank_similarity(r1, r2, method = "jaccard", threshold = 0.7)
```

`run_command_file` *Run a Zonation command file*

Description

This function runs a Zonation 5 analysis directly from R by executing a command file located in the specified folder. On Windows, the function looks for a single `.cmd` file; on Linux, it looks for a single `.sh` file.

Usage

```
run_command_file(folder)
```

Arguments

`folder`	A character string specifying the path to the folder containing the Zonation command file.

Value

Nothing is returned. The function is used to run the Zonation analysis.

Examples

```
## Not run:
run_command_file("C:/path/to/folder")

## End(Not run)
```

`settings_file` *Create a settings file for a Zonation analysis*

Description

This function generates a settings file with various parameters related to the input data and analysis configuration. The resulting settings file is saved with a `.z5` extension and can be used directly in the Zonation software.

Usage

```
settings_file(
  feature_list_file,
  external_solution_file = NULL,
  analysis_area_mask_layer = NULL,
  hierarchic_mask_layer = NULL,
  cost_layer = NULL
)
```

Arguments

feature_list_file	A character string specifying the feature list file. This is a compulsory parameter.
external_solution_file	A character string specifying the full path and/or name of the external solution file.
analysis_area_mask_layer	A character string specifying the full path and/or name of the analysis area mask layer file.
hierarchic_mask_layer	A character string specifying the full path and/or name of the hierarchic mask layer file.
cost_layer	A character string specifying the full path and/or name of the cost layer file.

Value

A `.z5` file containing the specified settings.

See Also

Other preprocessing: `command_file()`, `feature_list()`

Examples

```
withr::with_tempdir({

  tmp <- tempfile(fileext = ".txt")
  writeLines("feature list file = feature_list.txt", tmp)

  settings_file(feature_list_file = "feature_list.txt")
})
```

summary_curves *Plot summary performance curves*

Description

This function reads a Zonation summary curves file and plots one or more summary metrics against the priority rank. The output is a ggplot object, which can be further customized by the user. Optionally, the plot can be saved to disk as a high-quality figure.

Usage

```
summary_curves(
  dir,
  output_folder_name = "output",
  metrics,
  facet = FALSE,
  save_path = NULL,
  dpi = 300,
  width = 8,
  height = 6
)
```

Arguments

dir	Character. Path to the variant folder containing the output folder.
output_folder_name	Character. Name of the output folder inside dir. Default is "output".
metrics	Character vector. Names of the summary metrics to plot. Metrics can be overlaid only if they share the same units and value range. Fraction-based metrics can be overlaid together, while remaining_area and remaining_cost cannot be overlaid with other metrics.
facet	Logical. If TRUE, metrics are plotted in separate panels. This should be used when plotting metrics with different units or value ranges. Default is FALSE.
save_path	Character. Optional file path to save the plot. The file format is inferred from the file extension (e.g. ".tiff").
dpi	Numeric. Resolution (dots per inch) for saved figures. Default is 300.
width	Numeric. Width of the saved figure in inches. Default is 8.
height	Numeric. Height of the saved figure in inches. Default is 6.

Value

A ggplot object visualizing one or more Zonation summary metrics plotted against priority rank.

See Also

Other postprocessing: cost_summary(), coverage_distribution(), feature_curves(), feature_representation(), priority_map(), rank_similarity()

Examples

```
withr::with_tempdir({

  data_path <- system.file(
    "extdata",
    package = "ZonationR"
  )

  dir.create("output")

  file.copy(
    file.path(data_path, "summary_curves.csv"),
    "output/summary_curves.csv",
    overwrite = TRUE
  )

  p1 <- summary_curves(
    dir = ".",
    output_folder_name = "output",
    metrics = c("mean", "max")
  )

  print(p1)

})
```

Index