Package 'connectivity'

November 26, 2026

Title Multi-Resolution Landscape Connectivity

Version 0.1.0

Description Calculates multi-resolution landscape connectedness, Connected Habitat (a.k.a. connectivity-adjusted condition), the Protected Area Connectedness Index (PARC-connectedness) and the Bioclimatic Ecosystem Resilience Index (BERI) from raster filenames or 'terra' SpatRaster objects. The multi-resolution method is described in Valavi, R. et al. (2026) <doi:10.32942/X2S68V>. Uses 'terra' for raster and vector input and output, input preparation and polygon analysis boundaries, and 'extendr' to call the 'connectivity-core' 'Rust' numerical engine. Calculations load the analysis window into memory, and large rasters can be processed as aligned tiles. Also calculates exact fractional polygon coverage of raster cells.
Bundled example rasters for Tasmania are licensed separately under CC BY-NC-SA 4.0.

License MIT + file LICENSE

Copyright Commonwealth Scientific and Industrial Research Organisation (CSIRO). Vendored Rust crates: see file LICENSE.note.

URL https://github.com/csiro/connectivity

BugReports https://github.com/csiro/connectivity/issues

Imports terra (>= 1.8.0), utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 8.0.0

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.86.0, GNU make, xz

Depends R (>= 4.2)

Author Roozbeh Valavi [aut, cre]

Maintainer Roozbeh Valavi <roozbeh.valavi@csiro.au>

Repository CRAN

Date/Publication 2026-11-26 16:20:14 UTC

Contents

beri 2
connectedness 5
make_tiles 9
pixel_coverage 11
resolution_info 12

Index **13**

`beri` *Calculate the Bioclimatic Ecosystem Resilience Index (BERI)*

Description

The Bioclimatic Ecosystem Resilience Index (BERI) estimates the capacity of each cell to retain its biological diversity under climate change. It measures how much habitat the cell can reach through the landscape that is expected to support ecological communities similar to its present-day communities, under the current climate and under future climate scenarios. As in `connectedness()`, reachable habitat is weighted by condition and discounted by least-cost distance, here with a kernel representing many generations of dispersal over a 50-year horizon.

Compositional similarity comes from generalised dissimilarity modelling (GDM). Each climate raster holds environmental layers transformed by a fitted GDM (transgrids), with the current climate first. Scenario results are combined so that a poor outcome under any one scenario lowers the index. Habitat condition is used directly, without the Connected Habitat transformation.

Usage

```
beri(
  condition_file,
  current_file,
  future_files = NULL,
  polygon_mask = NULL,
  closed_border = FALSE,
  lambdas = c(2, 20, 200),
  max_cost = 2,
  window_size = 5L,
  outer_window = 9L,
  window_mode = "circular",
  levels = c(2, 4, 8, 16, 32),
  scale = NULL,
```

```
  resistance_file = NULL,
  resistance_scale = NULL,
  n_threads = NULL,
  filename = "",
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

condition_file	Single-layer terra::SpatRaster or raster filename. Habitat condition must be in [0, 1] after scaling. Missing values are excluded. All raster inputs must share a CRS, resolution and pixel grid; auxiliary rasters must cover the analysis window. They are not resampled.
current_file	Current-climate terra::SpatRaster or raster filename of GDM-transformed environmental layers (transgrids). One or more layers are supported.
future_files	Optional character vector of raster filenames, list of raster objects/filenames, or a single SpatRaster for one future scenario. Every scenario must have the same number and order of bands as current climate. NULL or an empty list computes the current-climate index only. Missing climate bands give the cell a similarity of zero.
polygon_mask	Optional polygon terra::SpatVector, vector filename or object accepted by terra::vect(). Polygons are reprojected to the raster CRS. Both polygon and raster must have a CRS, or both omit one. The result is cropped to the intersecting polygon bounds and cells outside polygons are NA. Cells touched by a polygon are included.
closed_border	If FALSE, retain surrounding landscape context for the analysis before cropping the output. If TRUE, exclude the landscape outside polygon_mask from both paths and habitat contributions.
lambdas	Positive dispersal bandwidths in kilometres. Scores are averaged across bandwidths.
max_cost	Finite relative traversal cost, at least 1.
window_size	Positive odd window size for non-coarsest levels.
outer_window	Positive odd coarsest window size, at least window_size.
window_mode	Either "circular" (default) or "square".
levels	Aggregation levels. Positive integers are rounded to the nearest power of two, with ties rounded down; level 1 is included. The result must be a continuous power-of-two sequence. Aggregated grids are computed from the base-resolution values.
scale	Optional condition divisor. Climate bands are not scaled.
resistance_file	Optional single-layer resistance raster or filename. After scaling, 1 - resistance is clipped to [0, 1] for path traversal, so resistance of 1 or more, including Inf, gives the highest path cost. Missing resistance at valid condition cells falls back to condition and raises a warning. Resistance does not alter habitat values or the domain.

`resistance_scale`
: Optional divisor applied only to resistance values.

`n_threads`
: Optional positive integer number of worker threads. `NULL` uses Rayon's default thread count.

`filename`
: Optional output filename. An empty string returns an in-memory raster. Files default to single-precision floating-point values.

`overwrite`
: Whether an existing output file may be overwritten.

`wopt`
: Named list of `terra::writeRaster()` options.

Details

For a focal cell i, scenario k ($k = 0$ for the current climate) and dispersal bandwidth λ (km), the accessible, compositionally similar habitat is

$$B_{ik} = \sum_j K(D_{ij}^{adj})\, h_j \tilde{A}_j s_{ijk}, \qquad K(D) = \exp\left(-\frac{(D/\lambda)^2}{283.465}\right),$$

with distances, condition and effective areas as in `connectedness()`. The Gaussian kernel approximates 50 generations of exponential dispersal ($283.465 = 5.785 \times 49$). The similarity of destination j under scenario k to the focal cell under the current climate is $s_{ijk} = \exp(-\sum_b |x_{ib}^{(0)} - x_{jb}^{(k)}|)$ over the transformed layers b. With m future scenarios,

$$\mathrm{BERI}_i = \frac{\frac{1}{2}\left(\frac{1}{m+1}\sum_{k=0}^{m} B_{ik} + \min_k B_{ik}\right)}{\sum_j K(D_{ij}^{int})\, \tilde{A}_j s_{ij0}},$$

the average of the mean and the worst scenario, relative to an intact landscape under the current climate. Scores are averaged over `lambdas`.

Value

A single-layer `terra::SpatRaster`, optionally written to `filename`. Its values are BERI scores; missing and polygon-masked cells are `NA`.

References

Ferrier, S., Harwood, T. D., Ware, C., and Hoskins, A. J. (2020). A globally applicable indicator of the capacity of terrestrial ecosystems to retain biological diversity under climate change: The bioclimatic ecosystem resilience index. Ecological Indicators, 117, 106554. doi:10.1016/j.ecolind.2020.106554. Introduces BERI.

Harwood, T., Love, J., Drielsma, M., Brandon, C., and Ferrier, S. (2022). Staying connected: assessing the capacity of landscapes to retain biodiversity in a changing climate. Landscape Ecology, 37, 3123–3139. doi:10.1007/s10980022015345. Develops the spatial resilience and connectedness approach.

Valavi, R., Mokany, K., Ware, C., Vickers, M., Giljohann, K. M., and Ferrier, S. (2026). A scalable multi-resolution framework for connectivity-based biodiversity indicators. EcoEvoRxiv, preprint. doi:10.32942/X2S68V. Describes the multi-resolution implementation.

See Also

connectedness() for input units, polygon boundaries, memory use and raster alignment requirements; make_tiles() for large rasters.

Examples

```
condition_file <- system.file("extdata", "site_condition.tif", package = "connectivity")
climate_dir <- system.file("extdata", "transgrids", package = "connectivity")
current_file <- file.path(climate_dir, "1990.tif")
future_files <- file.path(climate_dir, c("IPS50_45.tif", "GFD50_85.tif"))


# BERI with the current climate and two future scenarios
beri_map <- beri(
    condition_file = condition_file,
    current_file = current_file,
    future_files = future_files,
    lambdas = c(2, 20, 200),
    max_cost = 2,
    window_size = 5,
    outer_window = 9,
    levels = c(2, 4, 8, 16, 32),
    window_mode = "circular",
    n_threads = 2
)

terra::plot(beri_map, main = "BERI")
```

---

connectedness *Calculate habitat connectedness, Connected Habitat or PARC-connectedness*

Description

Habitat connectedness measures how much of the potentially connected habitat around each cell is retained, relative to a hypothetical intact landscape. Surrounding habitat counts more when it is in better condition and closer along its least-cost path, which lengthens through degraded habitat. Values range from 0 (isolated) to 1 (as connected as an intact landscape).

By default, the function returns Connected Habitat (a.k.a. connectivity-adjusted condition): the geometric mean of a cell's own habitat condition and its connectedness, used as an input to the Biodiversity Habitat Index (BHI). With `pa_file`, it returns the Protected Area Connectedness Index (PARC-connectedness) instead: how well protected cells are connected to other protected areas and to surrounding intact habitat.

Each cell's neighbourhood keeps fine detail nearby and represents more distant landscape with progressively coarser aggregated cells (Valavi et al. 2026). Inputs are raster filenames or `terra` SpatRaster objects, and the result is a georeferenced raster.

Usage

```
connectedness(
  condition_file,
  pa_file = NULL,
  pa_to_pa = FALSE,
  polygon_mask = NULL,
  closed_border = FALSE,
  lambdas = c(2, 20, 200),
  max_cost = 2,
  window_size = 5L,
  outer_window = 9L,
  window_mode = "circular",
  levels = c(2, 4, 8, 16, 32),
  scale = NULL,
  resistance_file = NULL,
  resistance_scale = NULL,
  option = 3L,
  n_threads = NULL,
  filename = "",
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

condition_file	Single-layer terra::SpatRaster or raster filename. Habitat condition must be in [0, 1] after scaling. Missing values are excluded. All raster inputs must share a CRS, resolution and pixel grid; auxiliary rasters must cover the analysis window. They are not resampled.
pa_file	Optional single-layer protected-area proportion raster or filename. Values must be in [0, 1] after scaling. The analysis uses the maximum of condition and protected fraction and evaluates protected cells only. PARC output is raw connectedness, independent of option.
pa_to_pa	Restrict PARC destinations to protected cells, weighted by protected fraction at coarser levels. Ignored if pa_file is NULL.
polygon_mask	Optional polygon terra::SpatVector, vector filename or object accepted by terra::vect(). Polygons are reprojected to the raster CRS. Both polygon and raster must have a CRS, or both omit one. The result is cropped to the intersecting polygon bounds and cells outside polygons are NA. Cells touched by a polygon are included.
closed_border	If FALSE, retain surrounding landscape context for the analysis before cropping the output. If TRUE, exclude the landscape outside polygon_mask from both paths and habitat contributions.
lambdas	Positive dispersal bandwidths in kilometres. Scores are averaged across bandwidths.
max_cost	Finite relative traversal cost, at least 1.

`window_size`	Positive odd window size for non-coarsest levels.
`outer_window`	Positive odd coarsest window size, at least `window_size`.
`window_mode`	Either `"circular"` (default) or `"square"`.
`levels`	Aggregation levels. Positive integers are rounded to the nearest power of two, with ties rounded down; level 1 is included. The result must be a continuous power-of-two sequence. Aggregated grids are computed from the base-resolution values.
`scale`	Optional divisor for condition values (`NULL`, 0 or 1 leaves values unchanged). A numeric pair or list supplies separate condition and protected-area divisors, e.g. `c(100, 100)` or `list(NULL, 100)`.
`resistance_file`	Optional single-layer resistance raster or filename. After scaling, `1 - resistance` is clipped to `[0, 1]` for path traversal, so resistance of 1 or more, including `Inf`, gives the highest path cost. Missing resistance at valid condition cells falls back to condition and raises a warning. Resistance does not alter habitat values or the domain.
`resistance_scale`	Optional divisor applied only to resistance values.
`option`	Habitat output: 1 returns connectedness, 2 multiplies by condition, and 3 (default) takes the square root of that product.
`n_threads`	Optional positive integer number of worker threads. `NULL` uses Rayon's default thread count.
`filename`	Optional output filename. An empty string returns an in-memory raster. Files default to single-precision floating-point values.
`overwrite`	Whether an existing output file may be overwritten.
`wopt`	Named list of `terra::writeRaster()` options.

Details

For a focal cell i and one dispersal bandwidth λ (km), connectedness is

$$C_i = \frac{\sum_j \exp(-D_{ij}^{adj}/\lambda)\, h_j \tilde{A}_j}{\sum_j \exp(-D_{ij}^{int}/\lambda)\, \tilde{A}_j},$$

where j runs over the cells of the multi-resolution neighbourhood, h_j is habitat condition and $\tilde{A}_j$ is effective area. D_{ij}^{adj} is the least-cost distance when crossing a cell of condition h costs its length times $(1-\mu)h+\mu$, with μ = `max_cost`, and D_{ij}^{int} is the distance through an intact landscape. With `resistance_file`, the cost factor is $1+(\mu-1)r$ for resistance r, while h_j remains the destination weight. Scores are averaged over `lambdas`. `option` = 2 returns $h_i C_i$ and `option` = 3 returns Connected Habitat, $\sqrt{h_i C_i}$.

PARC-connectedness replaces h with $\varphi = \max(h, p)$, for protected fraction p, in both path costs and destination weights, and scores only cells with $p > 0$. Paths still cross unprotected land. With `pa_to_pa` = `TRUE`, each destination in the numerator is also weighted by its protected share, so only protected destinations count.

Projected CRS units are converted to metres for the Rust engine. Geographic coordinates use degrees. Dispersal bandwidths are kilometres. An `outer_window` smaller than `window_size` is increased to match it. A missing raster CRS triggers a warning. Coordinates are then assumed to be degrees for cells smaller than one unit and metres for larger cells; mixed cases are judged from whether the extent fits longitude/latitude bounds.

Each call reads the complete analysis window into memory. For large areas, call the function for each tile from `make_tiles()` with open borders and merge the results with `terra::merge()`. The read window is padded by `(outer_window + 3) * max(levels)` pixels and aligned to the full raster's coarsest aggregation grid, so merged tiles equal a whole-raster result.

Value

A single-layer `terra::SpatRaster`, optionally written to `filename`. Missing and masked cells are `NA`. Computation and habitat post-processing use single precision.

References

Valavi, R., Mokany, K., Ware, C., Vickers, M., Giljohann, K. M., and Ferrier, S. (2026). A scalable multi-resolution framework for connectivity-based biodiversity indicators. EcoEvoRxiv, preprint. doi:10.32942/X2S68V. Describes the multi-resolution implementation.

Drielsma, M., Ferrier, S., and Manion, G. (2007). A raster-based technique for analysing habitat configuration: The cost–benefit approach. Ecological Modelling, 202(3–4), 324–332. doi:10.1016/j.ecolmodel.2006.10.016. Introduces the raster cost-benefit approach to habitat connectedness and distance-dependent aggregation.

Drielsma, M. J., Love, J., Taylor, S., Thapa, R., and Williams, K. J. (2022). General Landscape Connectivity Model (GLCM): a new way to map whole of landscape biodiversity functional connectivity for operational planning and reporting. Ecological Modelling, 465, 109858. doi:10.1016/j.ecolmodel.2021.109858. Extends the approach to whole-landscape connectivity mapping across large regions.

See Also

`make_tiles()` to split large rasters into tiles.

Examples

```
condition_file <- system.file("extdata", "site_condition.tif", package = "connectivity")
pa_file <- system.file("extdata", "pa_proportion.tif", package = "connectivity")


# Connectedness
conn <- connectedness(
    condition_file = condition_file,
    lambdas = c(2, 20, 200),
    max_cost = 2,
    window_size = 5,
    outer_window = 9,
    levels = c(2, 4, 8, 16, 32),
    window_mode = "circular",
    option = 1,
```

```
    n_threads = 2
)

# Connected Habitat
habitat <- connectedness(
    condition_file = condition_file,
    lambdas = c(2, 20, 200),
    max_cost = 2,
    window_size = 5,
    outer_window = 9,
    levels = c(2, 4, 8, 16, 32),
    window_mode = "circular",
    option = 3,
    n_threads = 2
)

# PARC-connectedness with default settings
parc <- connectedness(condition_file, pa_file = pa_file, n_threads = 2)

terra::plot(c(conn, habitat, parc),
            main = c("Connectedness", "Connected habitat", "PARC-connectedness"))
```

`make_tiles` *Create aggregation-aligned raster tiles*

Description

Splits a raster grid into non-overlapping rectangular tile polygons for processing large rasters piecewise. Pass each tile as `polygon_mask` to `connectedness()` or `beri()` with open borders (the default), then merge the results with `terra::merge()`. Open-border tiles read their surrounding landscape, so the merged result matches a single whole-raster calculation.

Usage

```
make_tiles(
  raster_file,
  nrows,
  ncols,
  align_to = 1L,
  balanced = FALSE,
  io_weight = 0.1
)

make_tile(
  raster_file,
  nrows,
  ncols,
  tile_id,
```

```
  align_to = 1L,
  balanced = FALSE,
  io_weight = 0.1
)
```

Arguments

raster_file	Reference terra::SpatRaster or raster filename. Only its grid is used, unless balanced = TRUE, which also reads the first layer's missing values.
nrows, ncols	Positive integer numbers of tile rows and columns.
align_to	Positive integer. Internal tile edges are placed at multiples of align_to pixels from the raster origin; the outer raster edge is always kept. Use the largest aggregation level, e.g. 32.
balanced	If FALSE, tiles have near-equal sizes. If TRUE, row breaks, then the column breaks of each tile row, balance an estimated cost. Each align_to block costs its cell count if any of its cells in the first layer has a value, and nothing if all are missing, plus io_weight times its cell count.
io_weight	Non-negative cost per cell added in balanced mode, so that mostly empty areas still carry some reading cost.
tile_id	One-based tile number, in row-major order.

Value

A terra::SpatVector of nrows * ncols polygons (one polygon for make_tile()) in the raster's CRS, ordered by row and then column, with attributes id, row, col, row_start, row_stop, col_start, col_stop, align_to and, in balanced mode, cost. Tile numbers and cell ranges are one-based and inclusive.

See Also

connectedness() and beri() for open-border polygon analyses.

Examples

```
habitat <- terra::rast(nrows = 64, ncols = 96, xmin = 0, xmax = 9600,
                       ymin = 0, ymax = 6400, crs = "EPSG:3857")
terra::values(habitat) <- seq(0.1, 1, length.out = terra::ncell(habitat))
tiles <- make_tiles(habitat, nrows = 2, ncols = 2, align_to = 4)
parts <- lapply(seq_len(nrow(tiles)), function(i) {
    return(connectedness(habitat, polygon_mask = tiles[i], levels = c(2, 4),
                         outer_window = 5, n_threads = 1))
})
mosaic <- terra::merge(terra::sprc(parts))
```

pixel_coverage *Calculate fractional polygon coverage of raster cells*

Description

Unions polygon geometries using `terra`, then calls the existing Rust core's exact per-cell polygon clipping algorithm. Holes and multipart polygons are preserved. This does not use sub-cell sampling or rasterize's approximate coverage option. Use it to prepare the protected-area proportions (`pa_file`) for PARC-connectedness in `connectedness()`.

Usage

```
pixel_coverage(
  polygon,
  raster_file,
  n_threads = NULL,
  filename = "",
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`polygon`	Polygon `terra::SpatVector`, vector filename, or object accepted by `terra::vect()`. Geometries are reprojected to the raster CRS where necessary and unioned so overlapping polygons are not counted twice.
`raster_file`	Reference `terra::SpatRaster` or raster filename. Only its grid and CRS are used; values and nodata do not mask the coverage result.
`n_threads`	Optional positive integer number of worker threads. `NULL` uses Rayon's default thread count.
`filename`	Optional output filename. An empty string returns an in-memory raster. Files default to single-precision floating-point values.
`overwrite`	Whether an existing output file may be overwritten.
`wopt`	Named list of `terra::writeRaster()` options.

Value

A single-layer `terra::SpatRaster` of fractions in `[0, 1]` matching the full reference grid. Empty polygon input produces zero coverage. Fractions refer to area in the raster's coordinate plane.

Examples

```
grid <- terra::rast(nrows = 2, ncols = 3, xmin = 0, xmax = 30,
                    ymin = 0, ymax = 20, crs = "EPSG:3857")
polygon <- terra::as.polygons(terra::ext(5, 20, 2, 18), crs = terra::crs(grid))
fraction <- pixel_coverage(polygon, grid, n_threads = 1)
```

`resolution_info` *Preview raster aggregation levels and search reach*

Description

A planning helper that reports raster metadata, candidate grid dimensions, resolutions, cell counts, approximate search reach and small-grid flags without reading raster values.

Usage

```
resolution_info(
  file_path,
  levels = NULL,
  min_dimension = 16L,
  outer_window = 5L
)
```

Arguments

`file_path`	Raster filename or `terra::SpatRaster`. Multiple layers and rasters without values are supported; only grid metadata are used.
`levels`	Positive integer aggregation factors. `NULL` uses powers of two from 1 to 512. Candidates are sorted and deduplicated, without rounding or inserting intermediate levels. Indicator functions normalise levels separately.
`min_dimension`	Integer small-grid threshold, clamped to at least 1.
`outer_window`	Integer search-window size, clamped to at least 1.

Details

Dimensions round up after division by each candidate level. Search reach is `outer_window * level * max(x_resolution, y_resolution)` using base resolutions. It is an approximate planning span, not a dispersal bandwidth. Geographic distances use 111.32 km per degree of latitude and its cosine adjustment at the raster's centre latitude for longitude. Projected units are converted using the CRS conversion factor. Small grids are labelled when either dimension is strictly below `min_dimension`.

Value

Invisibly, a data frame with `level`, `width`, `height`, `cells`, `percent_base`, `x_resolution`, `y_resolution`, `reach`, `reach_km` and `note`. Resolutions and `reach` use native CRS units (also stored in the `unit` attribute). `reach_km` is `NA` when CRS units are unknown.

Examples

```
grid <- terra::rast(nrows = 65, ncols = 129, xmin = 0, xmax = 12900,
                    ymin = 0, ymax = 6500, crs = "EPSG:3857")
plan <- resolution_info(grid, levels = c(1, 2, 4, 8), outer_window = 7)
```

Index

beri, 2
beri(), *9, 10*

connectedness, 5
connectedness(), *2*, *4, 5*, *9–11*

make_tile *(*make_tiles*)*, 9
make_tiles, 9
make_tiles(), *5*, *8*

pixel_coverage, 11

resolution_info, 12

terra::merge(), *8, 9*
terra::vect(), *3*, *6*, *11*
terra::writeRaster(), *4*, *7*, *11*