

Package: BNRTools (via r-universe)

October 7, 2026

Title A suite of convenience functions created by IIASA BNR
Researchers

Version 0.1

Description What the package does (one paragraph).

License MIT + file LICENSE

URL <https://github.com/iiasa/BNRTools>,
<https://iiasa.github.io/BNRTools/>

BugReports <https://github.com/iiasa/BNRTools/issues>

Depends R (>= 3.6)

Imports cli, janitor, dplyr, tibble, assertthat, sf, terra, stars,
abind, RNetCDF, ncd4, yaml

Suggests gdalUtilities, cubelyr, purrr, progress, testthat (>= 3.0.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libnetcdf-
dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://iiasa.r-universe.dev>

Date/Publication 2026-10-06 21:39:32 UTC

RemoteUrl <https://github.com/iiasa/BNRTools>

RemoteRef HEAD

RemoteSha 9cd1ae00c5c2a6af237e0a11e744321ea40754ba

Contents

%notin%	2
conv_downscalr2ibis	3
misc_emptyfolder	4
misc_objectSize	5
misc_sanitizeNames	6
spl_clipMask	6
spl_exportNetCDF	7
spl_growGrid	9
spl_mosaicTiffs	10
spl_replaceGriddedNA	11
spl_resampleRas	12
spl_rwr	13
spl_summarizeGPKGlayers	14
Index	15

%notin%	<i>Inverse of 'in' call</i>
---------	-----------------------------

Description

Calculates the set of entries not present in the second vector. Added for convenience since this is not supported by default in R.

Usage

a %notin% b

Arguments

- a First [vector](#) object.
- b Second [vector](#) object.

Value

A [vector](#) of [logical](#) entries of 'a' not present in 'b'.

Author(s)

Martin Jung

Examples

example code

conv_downscalr2ibis	<i>Function to format a prepared GLOBIOM netCDF file for use in ibis.iSDM</i>
---------------------	---

Description

This function expects a downscaled GLOBIOM output as created in the BIOCLIMA project. It converts the input to a stars object to be fed to the `ibis.iSDM` R-package.

Usage

```
conv_downscalr2ibis(  
  fname,  
  ignore = NULL,  
  period = "all",  
  template = NULL,  
  shares_to_area = FALSE,  
  use_gdalutils = FALSE,  
  verbose = TRUE  
)
```

Arguments

fname	A filename in character pointing to a GLOBIOM output in netCDF format.
ignore	A vector of variables to be ignored (Default: NULL).
period	A character limiting the period to be returned from the formatted data. Options include "reference" for the first entry, "projection" for all entries but the first, and "all" for all entries (Default: "reference").
template	An optional SpatRaster object towards which projects should be transformed.
shares_to_area	A logical on whether shares should be corrected to areas (if identified).
use_gdalutils	(Deprecated) logical on to use gdalutils hack-around.
verbose	logical on whether to be chatty.

Value

A [SpatRaster](#) stack with the formatted GLOBIOM predictors.

Author(s)

Martin Jung

Examples

```
## Not run:  
## Does not work unless downscalr file is provided.  
# Expects a filename pointing to a netCDF file.  
covariates <- conv_downscalr2ibis(fname)  
  
## End(Not run)
```

misc_emptyfolder	<i>Check if a directory is empty (and create it if it doesn't exist)</i>
------------------	--

Description

This function checks whether a specified directory is empty. If the directory does not exist, it will be created.

Usage

```
misc_emptyfolder(dir_path)
```

Arguments

dir_path	A character string specifying the path to the directory.
----------	--

Details

It also checks whether the directory is newly created or already existed.

Value

A logical value: TRUE if the directory is empty (or newly created), FALSE otherwise.

Author(s)

Martin Jung

Contributors: ChatGPT

Examples

```
# Check and create a directory  
dir_path <- tempfile()  
is_empty <- misc_emptyfolder(dir_path)  
print(is_empty)  
unlink(dir_path, recursive = TRUE)
```

misc_objectSize	<i>Shows size of objects in the R environment</i>
-----------------	---

Description

Shows the size of the objects currently in the R environment. Helps to locate large objects cluttering the R environment and/or causing memory problems during the execution of large workflows.

Usage

```
misc_objectSize(n = 10)
```

Arguments

n	Number of objects to show, Default: 10
---	--

Value

A data frame with the row names indicating the object name, the field 'Type' indicating the object type, 'Size' indicating the object size, and the columns 'Length/Rows' and 'Columns' indicating the object dimensions if applicable.

Author(s)

Bias Benito

Examples

```
if(interactive()){  
  
  #creating dummy objects  
  x <- matrix(runif(100), 10, 10)  
  y <- matrix(runif(10000), 100, 100)  
  
  #reading their in-memory size  
  misc_objectSize()  
  
}
```

misc_sanitizeNames	<i>Sanitize variable names</i>
--------------------	--------------------------------

Description

Prepared covariates often have special characters in their variable names which can or can not be used in formulas or cause errors for certain procedures. This function converts special characters (points, underscores or similar) of variable names into a species format.

Usage

```
misc_sanitizeNames(names)
```

Arguments

names A [vector](#) of [character](#) vectors to be sanitized.

Value

A [vector](#) of sanitized [character](#).

Examples

```
# Correct variable names
vars <- c("Climate-temperature2015", "Elevation__sealevel", "Landuse.forest..meanshare")
misc_sanitizeNames(vars)
```

spl_clipMask	<i>Clip and mask a raster using a raster or vector template</i>
--------------	---

Description

Aligns a [SpatRaster](#) or [stars](#) raster to a raster template, or clips it to a vector template. Vector geometries are rasterized on the target grid and used as a mask.

Usage

```
spl_clipMask(x, template, method = c("bilinear", "near"), touches = FALSE)
```

Arguments

x	A SpatRaster or stars object to clip and mask.
template	A SpatRaster or stars raster template, an sf or sfc object, or a path to a raster or vector file.
method	Resampling method: "bilinear" for continuous data or "near" for categorical data (Default: "bilinear").
touches	If TRUE, rasterize all cells touched by vector geometries; otherwise rasterize cells whose centers fall within them (Default: FALSE).

Value

An object of the same class as x, with its layer count and names preserved. Raster templates set the output grid and, when they contain values, mask cells where the first layer is NA. Vector templates crop the output to their extent and mask it to the rasterized geometry.

Author(s)

Martin Jung

See Also

[terra::crop\(\)](#), [terra::extend\(\)](#), [terra::mask\(\)](#), [terra::project\(\)](#), [terra::rasterize\(\)](#), [terra::resample\(\)](#)

Examples

```
target <- terra::rast(nrows = 10, ncols = 10, xmin = 0, xmax = 10,
  ymin = 0, ymax = 10, crs = "EPSG:4326", vals = 1:100)
boundary <- sf::st_sf(
  id = 1,
  geometry = sf::st_sfc(sf::st_polygon(list(rbind(
    c(2, 2), c(8, 2), c(8, 8), c(2, 8), c(2, 2)
  ))), crs = 4326)
)
clipped <- spl_clipMask(target, boundary)
```

spl_exportNetCDF

RExport a gridded raster to a NetCDF format

Description

This function serves as a general wrapper function to export a provided spatial gridded layer as multi-dimensional NetCDF file. It furthermore requires the specification of a list containing meta-data information.

Usage

```
spl_exportNetCDF(
  obj,
  filename,
  global_meta = system.file("iiasa_meta.yaml", package = "BNRTools"),
  separate_meta = FALSE,
  ...
)
```

Arguments

obj	A SpatRaster object to be exported.
filename	A character with the output filename.
global_meta	A global metadata descriptor by default using IIASA standard metadata (Default: "iiasa_meta data").
separate_meta	A logical flag on whether the metadata should be written separately in "yaml" format (Default: FALSE).
...	Any other metadata that should be overwritten or added to "global_meta".

Details

The default metadata is contained in "inst/iiasa_meta". See examples.

Note

A support for 'stars' could be added.

Author(s)

Martin Jung

See Also

[writeCDF](#), [writeRaster](#)

Examples

```
## Not run:
# Load default metadata (loaded by default by function too)
meta <- yaml::read_yaml(system.file("iiasa_meta.yaml", package = "BNRTools"))

# Dummy raster
obj <- terra::rast(ncol = 100, nrow = 100,
  xmin = 0, xmax = 100,
  ymin = 0, ymax = 100,
  resolution = 5, crs = terra::crs("WGS84"),
  val = runif(400)
)
```

```
# Export
spl_exportNetCDF(obj, filename = "test.nc",
                  global_meta = meta, title = "Super cool analysis")

## End(Not run)
```

spl_growGrid	<i>Grow a categorical SpatRaster by certain amount of pixels or distance.</i>
--------------	---

Description

A common issue in aggregating categorical rasters is that coastal and boundary gridcells tend to decrease owing to aggregation effects. This can cause issues of non-matching grid cells later on. This simple function takes a categorical [SpatRaster](#) object and grows it into no-data areas (those with NA values) within an optionally provided distance.

Usage

```
spl_growGrid(x, iter = 2)
```

Arguments

x	A categorical SpatRaster to be grown.
iter	A numeric values of the maximum number of iterations from the grid cell border of which x should be grown (Default: 2).

Value

[SpatRaster](#)

Note

Only NA grid cells will be filled!

See Also

[buffer](#), [focal](#)

Examples

```
set.seed(42)
ras <- terra::rast(ncol = 100, nrow = 100, xmin = 0, xmax = 100,
                  ymin = 0, ymax = 100, resolution = 10, crs = NA)

# Fill with dummy values but keep half as NA
terra::values(ras) <- c(rep(NA,50),rbinom(terra::ncell(ras)/2, 10, 0.5))

# Convert to factor
ras <- terra::as.factor(ras)
```

```
assertthat::assert_that(terra::is.factor(ras))

ras_nona <- spl_growGrid(x = ras, iter = 10)
terra::plot(ras_nona)
```

spl_mosaicTiffs	<i>Mosaic a list of spatial layers together.</i>
-----------------	--

Description

Particular for large spatial files, it is often impractical to process the entire layer as single file. Instead such a file could be split in smaller chunks and processed as such.

A common issue then is to reconstruct a spatial file of the original extent. This function takes a list of filenames as input and mosaics them together using a 'vrt' file format.

Usage

```
spl_mosaicTiffs(files, ofname = NULL, tempdir = NULL, dt = "INT2S", ...)
```

Arguments

files	A character vector with filenames of spatial files.
ofname	A character where the output should be written (Default: NULL).
tempdir	A character with a temporary folder that must exist (Default: NULL).
dt	A character with the output datatype of the spatial file (Default: "INT2S").
...	Any other parameters passed to writeRaster .

Details

This function by default uses the 'gdalUtilities' R-package and tools for most of the projections.

Value

A [SpatRaster](#) or just a file.

Author(s)

Martin Jung

See Also

[mosaic](#), [gdalbuildvrt](#)

Examples

```
## Not run:
# Get list of files
ll <- list.files(path_to_folder)

# Mosaic
spl_mosaicTiffs(ll, "full_file.tif")

## End(Not run)
```

spl_replaceGriddedNA *Replace NA values in gridded layers with a fixed value.*

Description

This function replaces all NA values in a spatial gridded layer with a fixed value such as for example 0. Accepted input layers are for [SpatRaster](#) from the "terra" R-package and [stars](#) from the "stars" R-package.

Usage

```
spl_replaceGriddedNA(obj, value = 0, mask, verbose = FALSE)
```

Arguments

obj	A SpatRaster , SpatRasterDataset or stars object.
value	A fixed numeric value of which all NA values are to be replaced with (Default: 0).
mask	An optional SpatRaster object used instead of the value.
verbose	Be chatty about what is processed (Default: FALSE).

Details

Required inputs are a single "obj" gridded data object and a numeric value. In addition an optional mask layer can be provided that to use a mask. In this case all no-data values are replaced with the value in this mask.

Value

A object of the same type as the input but with no-data values replaced with 'value'.

Author(s)

Martin Jung

Examples

```
# Example
s <- terra::rast(system.file("ex/logo.tif", package="terra"))
s[sample(1:terra::ncell(s), 100)] <- NA
sfill <- spl_replaceGriddedNA(s, value = 100)
terra::plot(sfill)
```

spl_resampleRas	<i>Resample raster</i>
-----------------	------------------------

Description

Resample two raster to the spatial resolution using aggregation or disaggregation.

Usage

```
spl_resampleRas(x, y, discrete = FALSE)
```

Arguments

x	A SpatRaster to be resampled.
y	A SpatRaster to which x will be resampled
discrete	logical to specify if input raster has continuous or discrete values

Value

[SpatRaster](#)

See Also

[aggregate](#), [disagg](#)

Examples

```
set.seed(42)
ras_a <- terra::rast(ncol = 100, nrow = 100, xmin = 0, xmax = 100,
  ymin = 0, ymax = 100, resolution = 20, crs = NA)

ras_b <- terra::rast(ncol = 100, nrow = 100, xmin = 0, xmax = 100,
  ymin = 0, ymax = 100, resolution = 5, crs = NA)

terra::values(ras_a) <- runif(n = terra::ncell(ras_a))
terra::values(ras_b) <- runif(n = terra::ncell(ras_b))

spl_resampleRas(x = ras_a, y = ras_b)
```

spl_rwr	<i>Calculates a rarity-weighted richness estimate from modelled species distributions</i>
---------	---

Description

This function calculates a rarity-weighted richness estimate from modelled species distributions, which can for example be obtained from the ``ibis.iSDM`` R-package. The input maps should ideally be binary presence-absence maps, but the function can also handle continuous predictions.

Usage

```
spl_rwr(x, normalize = TRUE, column = NULL)
```

Arguments

x	A SpatRaster or alternatively data.frame object containing the modelled species distributions.
normalize	A logical flag on whether to normalize the rarity-weighted richness ranks.
column	An optional character value on whether the (Default: NULL).

Details

The function

Value

A logical value: TRUE if the directory is empty (or newly created), FALSE otherwise.

Author(s)

Martin Jung

References

- Albuquerque, F., & Beier, P. (2016). Predicted rarity-weighted richness, a new tool to prioritize sites for species representation. *Ecology and Evolution*, 6(22), 8107-8114.
- Albuquerque, F., & Beier, P. (2015). Rarity-weighted richness: a simple and reliable alternative to integer programming and heuristic algorithms for minimum set and maximum coverage problems in conservation planning. *PloS one*, 10(3), e0119905.

Examples

```
## Not run:
# Calculate rarity-weighted richness from a [`SpatRaster`].

## End(Not run)
```

spl_summarizeGPKGlayers

Summarize a set of layers from geopackages

Description

Geopackages are a common format for storing spatial data, allowing multiple layers within a single file. These can be for example point or polygon data.

Usage

```
spl_summarizeGPKGlayers(folder, verbose = TRUE)
```

Arguments

folder	A character string specifying the path where the geopackage files are stored. This looks specifically for files with the '.gpkg' extension, skipping others.
verbose	A logical value indicating whether to print additional information during processing.

Value

A [data.frame](#) summarizing the layers in the geopackage, including their names, geometry types, and feature counts.

Author(s)

Martin Jung

Examples

```
## Not run:  
# Folder  
spl_summarizeGPKGlayers(folder = "path/to/your/geopackages")  
  
## End(Not run)
```

Index

- * **Inspired from** [`'inlabru'`]
 - `misc_sanitizNames`, 6
- * **conversion**
 - `conv_downscalr2ibis`, 3
- * **function.**
 - `misc_sanitizNames`, 6
- * **internal,**
 - `spl_replaceGriddedNA`, 11
- * **`list(\bru_standardise_names\')`**
 - `misc_sanitizNames`, 6
- * **spatial**
 - `spl_clipMask`, 6
 - `spl_exportNetCDF`, 7
 - `spl_growGrid`, 9
 - `spl_mosaicTiffs`, 10
 - `spl_resampleRas`, 12
- * **utils**
 - `%notin%`, 2
 - `misc_sanitizNames`, 6
 - `spl_replaceGriddedNA`, 11
- `%notin%`, 2
- `aggregate`, 12
- `buffer`, 9
- `character`, 3, 6, 8, 10, 13, 14
- `conv_downscalr2ibis`, 3
- `data.frame`, 13, 14
- `disagg`, 12
- `focal`, 9
- `gdalbuildvrt`, 10
- `logical`, 2, 3, 8, 12–14
- `misc_emptyfolder`, 4
- `misc_objectSize`, 5
- `misc_sanitizNames`, 6
- `mosaic`, 10
- `numeric`, 9
- `object_size (misc_objectSize)`, 5
- `sf`, 7
- `SpatRaster`, 3, 6–13
- `SpatRasterDataset`, 11
- `spl_clipMask`, 6
- `spl_exportNetCDF`, 7
- `spl_growGrid`, 9
- `spl_mosaicTiffs`, 10
- `spl_replaceGriddedNA`, 11
- `spl_resampleRas`, 12
- `spl_rwr`, 13
- `spl_summarizeGPKGlayers`, 14
- `stars`, 6, 7, 11
- `terra::crop()`, 7
- `terra::extend()`, 7
- `terra::mask()`, 7
- `terra::project()`, 7
- `terra::rasterize()`, 7
- `terra::resample()`, 7
- `vector`, 2, 3, 6
- `writeCDF`, 8
- `writeRaster`, 8, 10