Package: ibis.insights (via r-universe)

June 6, 2026

Title An R Implementation of the InSiGHTS Framework

Version 0.8

Description Implements the InSiGHTS (Index of Habitat Availability) modelling framework for assessing how climate change and land-use change affect the availability of suitable habitat for species over time. The package refines binary range maps or probabilistic species distribution models (SDMs) with fractional land-use layers to produce area-of-habitat (AOH) estimates across present and future time steps. An optional temporal discount function accounts for habitat maturity, allowing newly established land cover to be weighted according to its effective ecological value. Output summaries express habitat availability as absolute area or as a relative index with respect to a user-defined reference year. The package is designed to work seamlessly with the ibis.iSDM package but can also accept any SpatRaster or stars object as range input.

License CC BY 4.0

URL https://github.com/iiasa/ibis.insights, https://iiasa.github.io/ibis.insights/

BugReports https://github.com/iiasa/ibis.insights/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports FNN, ibis.iSDM, sf (>= 1.0-17), terra (>= 1.7-80), stars (>= 0.6-7), assertthat, lubridate, methods, units

Remotes iiasa/ibis.iSDM

Depends R (>= 4.2.0)

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

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

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

Date/Publication 2026-06-06 21:51:14 UTC

RemoteUrl https://github.com/iiasa/ibis.insights

RemoteRef HEAD

RemoteSha 328f8ab5abee6663743cb9c75330185acd058b01

Contents

`align_temporal` *Harmonize and align temporal raster layers*

Description

Align the temporal dimension of a source raster-like object to the time steps of a target object. For each target time step, `align_temporal()` selects the most recent source time step that is less than or equal to it. Target time steps earlier than the first source time step use the first source layer.

Before alignment, time coordinates are harmonized to integer calendar years. This supports common `stars` time dimensions stored as `Date`, `POSIXct`, numeric day offsets from 1970-01-01, or `units` values such as `"days since 1970-01-01"`. For `terra::SpatRaster` objects, the output time is written with `tstep = "years"`. For `stars` objects, the `time` dimension values are returned as `units` in `"years"`.

Usage

```
align_temporal(source, target, unit = "years")

## S4 method for signature 'SpatRaster,SpatRaster'
align_temporal(source,target,unit)

## S4 method for signature 'stars,stars'
align_temporal(source,target,unit)
```

```
## S4 method for signature 'ANY,ANY'
align_temporal(source,target,unit)
```

Arguments

source	A terra::SpatRaster or stars object providing the values to align.
target	A terra::SpatRaster or stars object providing the target time steps. It must be the same object type as source.
unit	A single character value describing the harmonized time unit. Currently only "years" is supported.

Details

align_temporal() is implemented as an S4 generic with methods for SpatRaster,SpatRaster and stars,stars inputs. It is useful when a range or environmental raster is available at coarser time steps than a land-use or scenario raster. The function implements a "previous value carried forward" alignment: a target year of 2035 will use a source layer from 2030 when the next source layer is 2040.

For stars inputs, a time dimension named Time is normalized to time. The function does not alter spatial dimensions, attributes, coordinate reference systems, or cell values beyond selecting/repeating temporal slices.

Value

An object of the same class as source, with temporal layers selected from source and time coordinates matching the harmonized target time steps.

Author(s)

Martin Jung

Examples

```
require(terra)

source <- terra::rast(nrow = 1, ncol = 1, nlyr = 3, vals = c(10, 20, 30))
terra::time(source) <- as.Date(c("2000-01-01", "2010-01-01", "2020-01-01"))

target <- terra::rast(nrow = 1, ncol = 1, nlyr = 4, vals = 1)
terra::time(target) <- as.Date(c("1995-01-01", "2005-01-01",
                                 "2015-01-01", "2025-01-01"))

align_temporal(source, target)
```

`create_derivate_range` *Recreate a derivative variable based on their range*

Description

The purpose of this function is to create the range from several derivative variables. The information to do so is taken from the variable name and it is assumed that those have been created by `ibis.iSDM::predictor_derivate()` function.

This function return the range of values from the original data that fall within the set of coefficients. Currently only positive coefficients are taken by default.

Usage

```
create_derivate_range(env, varname, co, to_binary = FALSE)
```

Arguments

`env`	The original variable stacks as `SpatRaster` or `stars`.
`varname`	A `character` of the variable name. Needs to be present in `"env"`.
`co`	A set of coefficients obtained via `stats::coef()` and a `ibis.iSDM::BiodiversityDistribution` object.
`to_binary`	A `logical` flag if the output should be converted to binary format or left in the original units (Default: `FALSE`).

Details

- This function really only makes sense for `'bin'`, `'thresh'` and `'hinge'` transformations.
- For `'hinge'` the combined `min` is returned.

Value

A `SpatRaster` object containing the predictor range.

Note

This is rather an internal function created for a specific use and project. It might be properly described in an example later.

Author(s)

Martin Jung

Examples

```
## Not run:
 # Assuming derivates of temperature have been created for a model, this
 # recreates the range over which they apply.
 deriv <- create_derivate_range(env, varname = "Temperature",
  co = coef(fit), to_binary = TRUE)

## End(Not run)
```

`insights_area` *Apply InSiGHTS with continuous areal land-use data*

Description

Apply an area-of-habitat (AOH) refinement to a binary or fractional range estimate using land-use or habitat layers already expressed as area per cell. Use this method when `lu` values are continuous areal units such as `km2`. For cell shares or suitability weights in `[0, 1]`, use `insights_fraction()` instead.

If `lu` has multiple layers or attributes, they are summed before being applied to the range. Optional `other` layers can be supplied as additional suitability or condition masks, but they must already be scaled to `[0, 1]`; raw environmental layers such as elevation should be converted to a suitability mask before calling this function.

Raster inputs are reprojected, cropped, and resampled to the range geometry when needed. Temporal `stars` inputs may use a `time` or `Time` dimension name. If `outfile` is supplied, the extension is adjusted to `.tif` for raster output or `.nc` for `stars` output.

Usage

```
insights_area(range, lu, other, outfile = NULL)

## S4 method for signature 'SpatRaster,SpatRaster'
insights_area(range,lu,other,outfile)

## S4 method for signature 'SpatRaster,stars'
insights_area(range,lu,other,outfile)

## S4 method for signature 'stars,stars'
insights_area(range,lu,other,outfile)

## S4 method for signature 'stars,SpatRaster'
insights_area(range,lu,other,outfile)

## S4 method for signature 'ANY,ANY'
insights_area(range,lu,other,outfile)
```

Arguments

range	A SpatRaster or temporal stars object describing the estimated distribution of a biodiversity feature (e.g. species). Values must be binary or fractional in [0, 1]. Alternatively a DistributionModel fitted with ibis.iSDM package can be supplied.
lu	A SpatRaster or temporal stars object of the future land-use areas to be applied to the range. **Each layer has to be in area units and greater than or equal to 0.** Multi-layer inputs are summed.
other	Optional SpatRaster or temporal stars object describing additional suitable conditions for the species. Values must already be suitability weights in [0, 1].
outfile	A writeable character of where the output should be written to. If missing, the function will return a SpatRaster or stars object respectively. Missing .tif or .nc extensions are added as needed.

Value

Either a SpatRaster or temporal stars object or nothing if outputs are written directly to drive.

Note

This function does not infer species-habitat relationships from raw land-use classes. Select, weight, or transform land-use and condition layers before calling insights_area(). *No checks are conducted on whether supplied area layers add up to the area of a grid cell.*

Author(s)

Martin Jung

References

- Rondinini, Carlo, and Piero Visconti. "Scenarios of large mammal loss in Europe for the 21st century." Conservation Biology 29, no. 4 (2015): 1028-1036.
- Visconti, Piero, Michel Bakkenes, Daniele Baisero, Thomas Brooks, Stuart HM Butchart, Lucas Joppa, Rob Alkemade et al. "Projecting global biodiversity indicators under future development scenarios." Conservation Letters 9, no. 1 (2016): 5-13.

Examples

```
require(terra)
range <- terra::rast(system.file(
  "extdata/example_range.tif", package = "ibis.insights", mustWork = TRUE
))
lu_fraction <- terra::rast(system.file(
  "extdata/Grassland.tif", package = "ibis.insights", mustWork = TRUE
)) / 10000

# Convert a fractional land-use layer to area per cell in km2.
lu_km2 <- lu_fraction * terra::cellSize(lu_fraction, unit = "km")
out <- insights_area(range = range, lu = lu_km2)
```

```
# The output is already in area units, so do not multiply by cell area again.
insights_summary(out, toArea = FALSE)
```

`insights_discount` *Apply temporal discount to land-use layers based on an age variable*

Description

This function applies a temporal discount to land-use layers based on a corresponding age or maturity variable. The age variable is always connected to a specific land-use class (e.g. forest age linked to forest fraction) and represents increasing value over time.

Newly established habitat (low age) does not provide full habitat value. The `target_age` parameter controls how quickly the age translates to effective habitat value: it is the age at which habitat reaches `target` of its full value. The function produces a discounted version of `lu` by applying a maturity factor derived from the age:

$$H_{\text{eff}} = H \times [1 - (1 - p)^{a/a_p}]$$

where H is the land-use value, a is the cell age, a_p is `target_age`, and p is `target`. Internally, this is equivalent to deriving the per-age discount rate:

$$d = 1 - (1 - p)^{1/a_p}$$

and applying:

$$H_{\text{eff}} = H \times [1 - (1 - d)^{a}]$$

- At `age = 0`: the factor is `0` – no habitat value for brand-new land-use.
- At `age = target_age`: the factor is `target`, e.g. `0.95` by default.
- As `age` increases: the factor approaches `1` – mature habitat reaches full value.

Usage

```
insights_discount(lu, age, target_age = 20, target = 0.95)

## S4 method for signature 'SpatRaster,SpatRaster'
insights_discount(lu,age,target_age,target)

## S4 method for signature 'SpatRaster,stars'
insights_discount(lu,age,target_age,target)

## S4 method for signature 'stars,SpatRaster'
insights_discount(lu,age,target_age,target)

## S4 method for signature 'stars,stars'
insights_discount(lu,age,target_age,target)
```

Arguments

lu	A SpatRaster or temporal stars object of the land-use variable (e.g. forest fraction or area). Can be single or multi-layer.
age	A SpatRaster or temporal stars object of the corresponding age or maturity variable (values >= 0). Must match lu in number of layers / time steps.
target_age	A single positive numeric age at which habitat reaches target of full value. Default: 20.
target	A single numeric target maturity value strictly between 0 and 1. Default: 0.95.

Value

A discounted version of lu in the same format as the input.

Author(s)

Martin Jung

Examples

```
require(terra)
# Load package example rasters
range <- terra::rast(system.file(
  "extdata/example_range.tif", package = "ibis.insights", mustWork = TRUE
))
lu <- terra::rast(system.file(
  "extdata/Grassland.tif", package = "ibis.insights", mustWork = TRUE
))
lu <- lu / 10000

# Use sparse vegetation as a simple proxy for habitat age/maturity.
# In real applications, use an age or maturity layer for the same land-use class.
age <- terra::rast(system.file(
  "extdata/Grassland.tif", package = "ibis.insights", mustWork = TRUE
))
age <- age / 10000
age <- age * 20

# Specify that habitat reaches 95% of full value at age 20.
lu_discounted <- insights_discount(lu, age, target_age = 20, target = 0.95)
out1 <- insights_fraction(range = range, lu = lu)
out2 <- insights_fraction(range = range, lu = lu_discounted)
op <- graphics::par(mfrow = c(1, 2))
terra::plot(out1, main = "Original grassland")
terra::plot(out2, main = "Discounted grassland")
graphics::par(op)
```

`insights_fraction` *Apply InSiGHTS with fractional land-use data*

Description

Apply an area-of-habitat (AOH) refinement to a binary or fractional range estimate using land-use or habitat layers expressed as fractions. Use this method when the land-use values represent cell shares or suitability weights in the interval `[0, 1]`. For land-use inputs already expressed as area per cell, use `insights_area()` instead.

If `lu` has multiple layers or attributes, they are summed before being applied to the range. This is useful when several land-use classes jointly represent suitable habitat. Optional `other` layers can be supplied as additional suitability or condition masks, but they must already be scaled to `[0, 1]`; raw environmental layers such as elevation should be converted to a suitability mask before calling this function.

Raster inputs are reprojected, cropped, and resampled to the range geometry when needed. Temporal `stars` inputs may use a `time` or `Time` dimension name. If `outfile` is supplied, the extension is adjusted to `.tif` for raster output or `.nc` for `stars` output.

Usage

```
insights_fraction(range, lu, other, outfile = NULL, clamp = FALSE)

## S4 method for signature 'SpatRaster,SpatRaster'
insights_fraction(range,lu,other,outfile,clamp)

## S4 method for signature 'SpatRaster,stars'
insights_fraction(range,lu,other,outfile,clamp)

## S4 method for signature 'stars,stars'
insights_fraction(range,lu,other,outfile,clamp)

## S4 method for signature 'stars,SpatRaster'
insights_fraction(range,lu,other,outfile,clamp)

## S4 method for signature 'ANY,ANY'
insights_fraction(range,lu,other,outfile,clamp)
```

Arguments

`range`	A `SpatRaster` or temporal `stars` object describing the estimated distribution of a biodiversity feature (e.g. species). Values must be binary or fractional in `[0, 1]`. Alternatively a `DistributionModel` fitted with `ibis.iSDM` package can be supplied.
`lu`	A `SpatRaster` or temporal `stars` object of the future land-use fractions to be applied to the range. **Each layer has to be in fractional units between 0 and 1.** Multi-layer inputs are summed.

other	Optional SpatRaster or temporal stars object describing additional suitable conditions for the species. Values must already be suitability weights in [0, 1].
outfile	A writeable character of where the output should be written to. If missing, the function will return a SpatRaster or stars object respectively. Missing .tif or .nc extensions are added as needed.
clamp	A logical on whether lu and summed fractional suitability should be clamped to 0 and 1 beforehand (Default: FALSE).

Value

Either a SpatRaster or temporal stars object or nothing if outputs are written directly to drive.

Note

This function does not infer species-habitat relationships from raw land-use classes. Select, weight, or transform land-use and condition layers before calling insights_fraction().

Author(s)

Martin Jung

References

- Rondinini, Carlo, and Piero Visconti. "Scenarios of large mammal loss in Europe for the 21st century." Conservation Biology 29, no. 4 (2015): 1028-1036.
- Visconti, Piero, Michel Bakkenes, Daniele Baisero, Thomas Brooks, Stuart HM Butchart, Lucas Joppa, Rob Alkemade et al. "Projecting global biodiversity indicators under future development scenarios." Conservation Letters 9, no. 1 (2016): 5-13.

Examples

```
require(terra)
# Load package example rasters
range <- terra::rast(system.file(
  "extdata/example_range.tif", package = "ibis.insights", mustWork = TRUE
))
lu <- c(
  terra::rast(system.file(
    "extdata/Grassland.tif", package = "ibis.insights", mustWork = TRUE
  )),
  terra::rast(system.file(
    "extdata/Sparsely.vegetated.areas.tif", package = "ibis.insights", mustWork = TRUE
  ))
)

# Convert example land-use layers to fractions between 0 and 1.
lu <- lu / 10000

out <- insights_fraction(range = range, lu = lu, clamp = TRUE)
head(insights_summary(out))
```

insights_summary *Summarize inSiGHTS into an index*

Description

This function handily summarizes the suitable habitat for a given species or biodiversity feature into an index. If a single timestep (or object with a single layer) is provided, this function simply summarizes the suitable area.

Usage

```
insights_summary(
  obj,
  toArea = TRUE,
  fun = "sum",
  relative = TRUE,
  symmetric = FALSE
)

## S4 method for signature 'SpatRaster'
insights_summary(obj,toArea,fun,relative,symmetric)

## S4 method for signature 'stars'
insights_summary(obj,toArea,fun,relative,symmetric)
```

Arguments

obj	A SpatRaster or temporal stars object with the applied InSiGHTS outputs from insights_fraction or insights_area. If the number of layers is greater than 1, the parameter "relative" might be applied.
toArea	A logical flag whether fractional suitable habitat should be multiplied by cell area before summarizing (Default: TRUE). Use FALSE for outputs from insights_area(), which are already in area units.
fun	A character indicating the summary function to be applied (Default: 'sum'). Currently supported are 'sum', 'min', 'max', 'median' and 'mean'.
relative	A logical flag whether a relative index is to be constructed (Default: TRUE).
symmetric	A logical flag whether to additionally compute the symmetric relative difference (Default: FALSE). Requires relative = TRUE.

Details

When relative = TRUE, the standard relative change (in percent) is computed as $D(t) = (x_t - x_0)/x_0 \times 100$.

When symmetric = TRUE, the symmetric relative difference is also reported as an additional column relative_change_sym:

$$D_{sym}(t) = \frac{x_t - x_0}{x_t + x_0}$$

This metric is bounded in $[-1, 1]$ and is preferred over the standard relative change when the baseline habitat area x_0 is small (causing the standard metric to become arbitrarily large), or when a bounded, symmetric index is needed for cross-species comparisons. Requires the baseline suitability (x_0) to be positive.

Value

A `data.frame` with area estimates or the respective indicator.

Author(s)

Martin Jung

References

- Baisero, Daniele, Piero Visconti, Michela Pacifici, Marta Cimatti, and Carlo Rondinini. "Projected global loss of mammal habitat due to land-use and climate change." One Earth 2, no. 6 (2020): 578-585.
- Powers, Ryan P., and Walter Jetz. "Global habitat loss and extinction risk of terrestrial vertebrates under future land-use-change scenarios." Nature Climate Change 9, no. 4 (2019): 323-329.

Examples

```
require(terra)
range <- terra::rast(system.file(
  "extdata/example_range.tif", package = "ibis.insights", mustWork = TRUE
))
lu <- terra::rast(system.file(
  "extdata/Grassland.tif", package = "ibis.insights", mustWork = TRUE
)) / 10000

out <- insights_fraction(range = range, lu = lu)
insights_summary(out, relative = FALSE)

ts <- c(out, out * 0.8, out * 0.6)
terra::time(ts, tstep = "years") <- c(2020, 2040, 2060)
insights_summary(ts, relative = TRUE, symmetric = TRUE)
```

relChange *Relative change function*

Description

This function calculates the relative change of a vector, taking the first value as a reference value.

Usage

```
relChange(v, fac = 100)
```

Arguments

v	A numeric vector with length greater than 1.
fac	A numeric constant multiplier on the resulting metric.

Value

A numeric vector.

Author(s)

Martin Jung

Examples

```
# Example vector
x <- c(20,6,2,1,15,25)
relChange(x)
```

relChangeSym *Symmetric relative difference function*

Description

Computes the symmetric relative difference of a numeric vector with respect to its first element. Unlike the standard relative change (relChange), this metric is bounded in $[-1, 1]$ and remains well-defined when the baseline value is small.

Usage

```
relChangeSym(v)
```

Arguments

v	A numeric vector with length greater than 1 and first element > 0.

Details

The symmetric relative difference is defined as:

$$D_{sym}(t) = \frac{x_t - x_0}{x_t + x_0}$$

This formulation is preferred over the standard relative change when:

- The baseline value x_0 is small, causing the standard formula to produce arbitrarily large values (common for rare species or freshly colonised habitat).
- Symmetric treatment of gains and losses is required: a change from a to b has the same magnitude (opposite sign) as from b to a.
- A bounded, directly comparable index across species or regions with very different baseline areas is needed.

Value

A `numeric` vector of symmetric relative differences, bounded in $[-1, 1]$. Returns `NA` where the denominator $x_t + x_0 = 0$.

Author(s)

Martin Jung

Examples

```
x <- c(20, 6, 2, 1, 15, 25)
relChangeSym(x)
```

`st_clamp` *Clamp raster values to specific bounds*

Description

Clamp all cell values in a `terra::SpatRaster` or `stars` object to a provided numeric interval. Values below the lower bound are set to the lower bound, and values above the upper bound are set to the upper bound. Missing values are preserved.

This is useful for keeping suitability, fractional land-use, or habitat condition layers inside valid ranges before they are passed to functions such as `insights_fraction()`.

Usage

```
st_clamp(env, lb = -Inf, ub = Inf, lower = lb, upper = ub)
```

Arguments

env	A terra::SpatRaster or stars object.
lb	A single numeric lower bound for clamping. Defaults to -Inf, which leaves the lower tail unchanged.
ub	A single numeric upper bound for clamping. Defaults to Inf, which leaves the upper tail unchanged.
lower	Alias for lb.
upper	Alias for ub.

Details

For terra::SpatRaster inputs, st_clamp() delegates to terra::clamp(). For stars inputs, each attribute array is clamped directly with pmax() and pmin(), preserving the original dimensions and attribute names.

Value

An object of the same class as env, with values clamped to [lb, ub].

Author(s)

Martin Jung

Examples

```
require(terra)
r <- terra::rast(nrow = 2, ncol = 2, vals = c(-0.2, 0.4, 1.2, NA))
st_clamp(r, lower = 0, upper = 1)

require(stars)
s <- stars::st_as_stars(r)
st_clamp(s, lower = 0, upper = 1)
```

Index