Package 'nhSDM'

June 16, 2022

Title Tools for Natural Heritage SDMs

Date 2021-06-08

Version 0.1.1

Description Miscellaneous tools for working with Species Distribution Modelling (SDM) input and output data.

Depends R (>= 3.5.0), raster, sf

Imports methods, sp, stringi, RSQLite, fasterize, lwgeom

Suggests dplyr

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

R topics documented:

nh_best *Extract areas with best (highest value) as polygons from an SDM prediction raster, with optional feature mask*

Description

If spf is given, areas intersecting these features (plus a buffer, if min.dist is specified) will not be included in the returned polygons.

Usage

```
nh_best(
  rast,
  spf = NULL,
  top.percent = NULL,
  min.size = NULL,
  min.dist = NULL,
  num.patches = NULL,
  rank.by = "area"
)
```

Arguments

rast	input raster model output with continuous values
spf	input spatial features (sp or sf spatial object); if given, it can be used to modify areas selected from rast
top.percent	numeric; percent (e.g.; 0.01 = 0.01%) of highest cell values in raster to extract
min.size	numeric; optional minimal area of an extracted polygon
min.dist	numeric; optional minimal distance from spf for patches. Default is 0 (can be adjacent to 'spf')
num.patches	numeric; optional number of patches to return, using rank.by criteria
rank.by	character; used for ranking patches when num.patches is not null; either 'area' (default) or 'value'

Details

top.percent and min.size will both be derived from spf if they are null and spf is given. In this case, top.percent will be set to be equal to spf's cell coverage (prevalence relative to non-NA areas in raster). min.size will be set to the area of the smallest feature in spf. If spf are line or point features, this will be zero.

rank.by only subsets the outputs if num.patches is set. rank.by must be set to "value" to return the mean prediction value of the patch.

For better performance, crop the raster prior to running, and/or use a very low top.percent value, to return a smaller proportion of the cells as polygons.

Value

sp or sf object (polygons)

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- sf::st_read("inputs/species/ambymabe/polygon_data/ambymabe.shp")
rast <- raster::raster("outputs/ambymabe/grids/ambymabe_20171018_130837.tif")
rast <- raster::crop(rast, spf)
best_polys <- nh_best(rast, spf)

## End(Not run)
```

---

nh_burn — *Add areas represented by features ('burn-in') to a binary raster SDM output*

Description

Takes spatial features, a continuous raster (values between 0 and 1), and returns a classified binary (0/1) raster. Areas greater than `orig.thresh` are set to 1, all others 0. Areas intersecting `spf` features are assigned a value of 1 in the returned classified raster. See details for default calculation of `orig.thresh` and usage of `buffer`.

Usage

```
nh_burn(spf, rast, orig.thresh = NULL, buffer = 0, return.thresh = FALSE, ...)
```

Arguments

`spf`	input spatial features (sp or sf spatial object)
`rast`	input raster model output (values between 0 and 1)
`orig.thresh`	numeric between 0 and 1; 'global' threshold value to apply to raster
`buffer`	numeric; spatial buffer around spf to include in burn-in
`return.thresh`	logical; whether to return thresholds along with raster in a list
`...`	Other arguments as to `raster::writeRaster`

Details

When `buffer` is used or `orig.thresh` is not provided, a minimum cell value (min.cell) across all `spf` is calculated. The min.cell value will be used as a threshold for the full raster, or just areas within `buffer` distance of `spf`, if `buffer` is greater than 0.

If `buffer` is greater than 0 and `orig.thresh` is not NULL, `orig.thresh` is used as the threshold for the full raster. Additionally, cells within `buffer` distance of `spf` which are greater than the calculated min.cell value are also set to 1. If a given `orig.thresh` is lower than the calculated min.cell value, `buffer` will have no impact on the output.

The default is to return the raster only. If `return.thresh = TRUE`, the function will return a list with 3 named objects: `rast`, the output raster; `orig.thresh`, the global threshold (if used); `min.cell`, the minimum cell value threshold (if used).

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- st_read("ambymabe/polygon_data/ambymabe.shp")
rast <- raster("ambymabe/grids/ambymabe_20171018_130837.tif")
class_burn <- nh_burn(spf, rast, 0.75, 250, filename = "model_classified.tif", datatype = "INT2U")

## End(Not run)
```

---

nh_crop *Crop extra rows/columns from all sides of a raster*

Description

This function will reduce the extent of a raster, by removing rows/columns from all sides if they do not have any non-NA cells with a value greater than zero. One buffer row/column is left on each side.

Usage

```
nh_crop(rast)
```

Arguments

rast	input raster

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
rast <- rast <- raster("ambymabe_16May2018.tif")
rast.crop <- nh_crop(rast)

# should be TRUE
sum(values(rast), na.rm=T) == sum(values(rast.crop), na.rm=T)

## End(Not run)
```

nh_group *Group spatial features using a defined separation distance*

Description

Input features are allocated into new groups depending on whether they lie within the separation distance of another input feature.

Usage

```
nh_group(spf, sep.dist = 0, union = FALSE)
```

Arguments

spf	input spatial features (sp or sf spatial object)
sep.dist	separation distance with which to define groups (see description)
union	whether to union output groups into multi-features

Details

The grouping is done by input ID, so multi-features are considered one input feature (they will not be be ungrouped).

The separation distance `sep.dist` is numeric and in the units of `spf`'s coordinate system, unless the coordinate system uses latitude/longitude as the unit (e.g. WGS 84). In these cases, geodesic distances will be used and `sep.dist` should be specified in meters.

A column 'group' will be added to the output features. Specifying `union = TRUE` will output one (multi)feature per group, meaning original attributes will be discarded - only 'group' and 'count' (the number of original features in the group) will be returned. This feature requires the package `dplyr` to be installed.

Value

sp or sf object (same as input)

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- rgdal::readOGR("ambymabe/polygon_data", "ambymabe")
spg <- nh_group(spf, 1000)

## End(Not run)
```

`nh_linefill` *For line-based SDM predictions, identify areas between predicted suitable lines*

Description

Provided a network of lines (e.g., reaches in a hydrological network), and an attribute that identifies those lines representing suitable habitat, this function identifies fill-in lines between suitable lines, given certain limits (total distance and/or total reaches). It does not alter original line segments.

Usage

```
nh_linefill(spf, field, max.dist = NA, max.line = NA)
```

Arguments

`spf`	input spatial features (model predictions; sp or sf spatial object)
`field`	name of field in `spf` (binary 1/0), where lines will be filled between features with value == 1
`max.dist`	maximum distance between selected lines to fill in
`max.line`	maximum number of lines between selected lines to fill in

Details

Line directionality is used to identify how they are connected in the network, so start/end nodes must align. These nodes will be calculated; alternatively, the names `startNode` and `endNode` can be provided as columns in `spf` and will be used instead.

Values for `max.dist` should be in the units of the CRS, except when using a lat/lon based CRS, in which case the value should be in meters.

Value

sf object including suitable lines and filled-in lines

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- st_read("SDM_results.shp")
fill <- nh_linefill(spf, "thresh", max.dist = 5000, max.line = 5)

## End(Not run)
```

nh_map *Make a suitable/unsuitable binary map, given a continuous SDM output or raster template*

Description

This function performs a reclassification of a continuous map to a binary one, and then performs additional changes to the binary map, depending on inputs. If no continuous map is to be used, then a template raster should be given to `rast`.

Usage

```
nh_map(
  rast,
  thresh = NULL,
  feature.occ = NULL,
  feature.burn = NULL,
  feature.mask = NULL,
  raster.mask = NULL,
  patch.drop = NULL
)
```

Arguments

`rast`	input raster model output (values between 0 and 1); if no model, a template raster must be provided
`thresh`	numeric between 0 and 1; 'global' threshold value to apply to rast
`feature.occ`	input feature occurrences, to burn-in to map (sp or sf spatial object)
`feature.burn`	input feature(s), to burn-in to map (sp or sf spatial object)
`feature.mask`	input feature mask to apply to map (sp or sf spatial object)
`raster.mask`	input raster mask to apply to map (should match `rast` resolution)
`patch.drop`	patch size (in `rast` area units) to remove from map

Details

Reclassification (when `thresh` is given) is always performed first. After this, optional masks are applied where areas within features (`feature.mask`) and not NoData (`raster.mask`) are kept (value = 1). All other cells are given (value = 0).

Patches (contiguous groups of cells = 1) are then be dropped by specifying a minimum patch size to `patch.drop` in area units of the raster's CRS (see nhSDM::nh_patchdrop).

The final step is to 'burn in' (rasterize with value = 1) features from `feature.occ` and `feature.burn`. Since this is the last step, it is not affected by masks or patch dropping.

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- st_read("ambymabe/polygon_data/ambymabe.shp")
rast <- raster("ambymabe/grids/ambymabe_20171018_130837.tif")
map <- nh_map(rast, 0.75, feature.occ = spf)

## End(Not run)
```

---

nh_patchdrop *Remove contiguous patches smaller than a given patch size from binary output*

Description

Takes a binary/thresholded raster (values either NA, 0, or 1), and returns a binary raster. Clumps of contiguous cells with the value (1) that are smaller than the `min.patch` size are given `updatevalue` (default = 0).

Usage

```
nh_patchdrop(
  spf = NULL,
  rast,
  min.patch = NULL,
  directions = 8,
  updatevalue = 0,
  ...
)
```

Arguments

`spf`	input spatial features (sp or sf spatial object)
`rast`	input binary raster output (values either NA/0 or 1)
`min.patch`	area of minimum patch size, in area units used in `rast`
`directions`	Integer. Which cells are considered adjacent? Should be 8 (default; Queen's case) or 4 (Rook's case). From `raster::clump`
`updatevalue`	Integer or NA. Value to apply to cells which do not meet the min.patch size. Default = 0.
`...`	Other arguments as to `raster::writeRaster`

Details

If 'spf' is given, the smallest feature's area will be used to derive a `min.patch` value, and any given 'min.patch' is ignored. If 'spf' is not given, a 'min.patch' value must be given, in area units of the input raster.

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- st_read("ambymabe/polygon_data/ambymabe.shp")
rast <- raster("screen_lpsh/thumb/ambymabe.tif")

# use minimum patch size from presence features
rast_contig_minpres <- nh_patchdrop(spf, rast)

# use a minimum patch size of 10000 (in units from 'rast')
rast_contig_10km <- nh_patchdrop(rast = rast, min.patch = 10000,
 filename = "rast_contig_10km.tif", datatype = "INT2U")

## End(Not run)
```

`nh_rasterize` *Convert vector format SDM prections to raster format*

Description

Requires spatial features, a template raster, and prediction values vector. The buffer is optional; when provided, the buffer distance can be provided as a single numeric value or a vector matching length of `spf`, for a variable buffer by feature.

Usage

```
nh_rasterize(
  spf,
  rast,
  pred.vals,
  buffer = NULL,
  priority = NULL,
  rast.out = NULL,
  ...
)
```

Arguments

`spf`	input spatial features (model predictions; sp or sf spatial object)
`rast`	input raster template
`pred.vals`	column name in spf holding feature prediction values
`buffer`	numeric (single value or vector matching length of spf); spatial buffer around spf to include in burn-in
`priority`	column name in spf holding priority values for feature rasterization (higher values have priority)
`rast.out`	Output raster file name (with file extension)
`...`	Additional arguments to writeRaster (e.g. overwrite)

Details

Buffer units should be given in the units of `rast`. Make sure to keep in mind the resolution of the raster when choosing a buffer. If rasterizing lines, and a one-cell width is desired, do not use a buffer.

A vector can be given to 'priority' for sorting prior to rasterization, where higher values have priority. When `priority = NULL` (default), priority will be defined as the `pred.vals`.

If `rast.out` is not specified, the raster will remain in temp folder.

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
spf <- st_read("acipoxyr/shapefiles/acipoxyr_20180105_133929_results.shp")
rast <- raster("template.tif")

# rasterize
bla <- nh_rasterize(spf, rast, pred.vals = "prbblty", buffer = spf$strord*15)

## End(Not run)
```

`nh_sample` *Create points in features in reference raster cells*

Description

For each spatial feature, a given number (`num_samps`) of points are created in cells that the feature intersects.

Usage

```
nh_sample(spf, rast, num.samps = NULL, replace = FALSE, force.min = FALSE)
```

Arguments

`spf`	input spatial features (sp or sf spatial object)
`rast`	raster dataset with extent overlapping spf
`num.samps`	number of samples to create in each feature (see details)
`replace`	whether to sample with or without replacement
`force.min`	whether to force `num.samps` points in features, even if they are duplicates

Details

num.samps can be a a proportion (a decimal value < 1), single integer, or vector of integers equal to length of spf indicated the number of samples to take from each feature. If left NULL, num.samps will be set to the number of cells [n] intersecting the feature. If a proportion is given (e.g., 0.5), than [n * num.samps] will be returned. If a single integer is given, num.samps points will be sampled in each feature.

When replace = FALSE and force.min = FALSE (defaults), each cell can only contain one point (across each feature, and the entire returned set of points). In this case, when the number of samples points to create exceeds the number of unique cells intersected by a given feature, the number of samples for that feature equals the number of cells. If replace = TRUE, sampling is done with replacement and duplicates may be taken. The special case replace = FALSE and force.min = TRUE will always return num.samps per feature. It only produces duplicates if num.samps exceeds the number of cells intersecting the feature, in which case it will replicate the samples until num.samps is reached.

If CRS do not match, the features will be transformed to the CRS of the raster.

A column 'feat.id' is added to the output point features to indicate the row number of the feature that the point was generated within.

Value

sp or sf object (points)

Author(s)

David Bucklin

Examples

```
## Not run:
r<-raster::raster("AnnMnTemp.tif")
spf <- rgdal::readOGR("ambymabe/polygon_data", "ambymabe")
# can also use sf: spf <- sf::st_read("ambymabe/polygon_data/ambymabe.shp")
spf.samps <- nh_sample(spf, r, num.samps)

## End(Not run)
```

---

nh_stack *Stack multiple binary SDM rasters into one raster layer*

Description

Takes a rastfiles of binary raster's filenames (values 0/1), a template raster covering the extent desired for the stack, and optionally the codes (names) to use for the rasters. Returns a single raster layer (factor type with an attribute table).

Usage

```
nh_stack(rastfiles, rast, codes = NULL, return.table = TRUE, clip.feat = NULL)
```

Arguments

rastfiles	raster file names; character vector
rast	raster template dataset
codes	species codes; character vector. If given, must match length of rastfiles
return.table	Whether to return a table with nh_stack unique values, species codes, and file-names
clip.feat	sf data with masking features for rastfiles. Must have column named 'code', matching codes

Details

The raster attribute table (accessed using levels()) has four columns: 'ID': the unique integer value in the raster; 'VALUE': the internal nh_stack unique value for that 'ID', 'ALLCODES': the identity of species codes, pasted in a character vector seperated by ';', 'ALLCODES_CT': the number of unique codes for that value.

If codes is not given, the raster layer name will be used as the layer's code. If these are not unique, the internal nh_stack unique value will be pasted to the end of the original code, and a message will be printed.

All rasters must have the same projection and resolution, though they can have different extents - the processing extent is defined by rast. To summarize all rasters across their entire extents, rast should essentially be the union (mosaic) of all raster extents.

When return.table is TRUE, a list with two objects (1), the stack raster, and (2) a summary table of included rasters is returned (with internal nh_stack unique values, species codes, and file names). This table is required for resampling the stack (i.e. with nh_stack_resample).

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
rast<-raster::raster("project_mask.tif")
list <- sort(list.files(paste0(rastout, t, "/thumb"), full.names = T,
 pattern = "^[[:lower:]]{5,12}\\.tif$"))

stack <- nh_stack(list, rast)
# view raster attribute table
levels(stack[[1]])

## End(Not run)
```

nh_stack_resample *Resample a raster from nh_stack to a lower (coarser) resolution*

Description

Takes an output raster from nh_stack, and returns a lower-resolution version, with recalculated species assemblages for the larger cells. New values are "aggregated" by `fact`, the number of cells to aggregate in the x/y dimensions (see `?raster::aggregate`).

Usage

```
nh_stack_resample(rast, lookup, fact = 10, spf = NULL)
```

Arguments

`rast`	raster output from nh_stack
`lookup`	lookup table from nh_stack
`fact`	aggregation factor, in number of cells (see `?raster::aggregate`)
`spf`	Optional vector spatial features to use for aggregation (sp or sf-class polygons). If supplied, `fact` will be ignored

Details

You can also provide polygons (sp or sf-class) to `spf`, over which to aggregate species assemblages. The polygons intersecting areas with data in `rast` are returned, with columns identifying species codes and counts. This method will fail with large rasters (see `raster::zonal`), in which case processing subsets of the stack raster is advised. Polygons will be returned in their original projection, but processing internally is done in the raster's projection.

Value

RasterLayer

Author(s)

David Bucklin

Examples

```
## Not run:
stack <- nh_stack(list, rast, return.table = TRUE)

# resample from 30m to 990m (~1km) resolution
stack1km <- nh_stack_resample(stack[[1]], stack[[2]], fact = 33)

# view species count raster
ct <- deratify(stack1km, att = "ALLCODES_CT")
plot(ct)

## End(Not run)
```

Index