

# Package ‘nhSDM’

July 23, 2018

**Title** Tools for Natural Heritage SDMs

**Date** 2018-07-23

**Version** 0.0.3

**Description** Miscellaneous tools for working with Natural Heritage data in Species Distribution Modelling (SDM) workflows.

**Depends** R (>= 3.5.0)

**Imports** methods, sf, raster, sp, gdalUtils, stringi, RSQLite

**Suggests** dplyr, lwgeom

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

## R topics documented:

nh_best	1
nh_burn	2
nh_group	4
nh_patchdrop	5
nh_sample	6
nh_stack	7
<b>Index</b>	<b>9</b>

---

nh_best	<i>Extract areas with best (highest value) as polygons from an SDM prediction raster, with optional feature mask</i>
---------	--

---

## Description

If spf is given, areas of cells intersecting these features will not be included in the returned polygons.

## Usage

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

## Arguments

spf	input spatial features (sp or sf spatial object); will be erased from output
rast	input raster model output with continuous values
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
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 <- st_read("ambymabe/polygon_data/ambymabe.shp")
rast <- raster("ambymabe/grids/ambymabe_20171018_130837.tif")
best_polys <- nh_burn(spf, rast)

## 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\_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_patchdrop	<i>Remove contiguous patches smaller than a given patch size from binary output</i>
--------------	---

---

## 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 a value of 0 (if 0 values are present in the input 'rast'); otherwise, they are given an NA value.

## Usage

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

## Arguments

<code>spf</code>	input spatial features (sp or sf spatial object)
<code>rast</code>	input binary raster output (values either NA/0 or 1)
<code>min.patch</code>	area of minimum patch size, in area units used in <code>rast</code>
<code>directions</code>	Integer. Which cells are considered adjacent? Should be 8 (default; Queen's case) or 4 (Rook's case). From <code>raster::clump</code>
<code>...</code>	Other arguments as to <code>raster::writeRaster</code>

## 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\_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**

<code>spf</code>	input spatial features (sp or sf spatial object)
<code>rast</code>	raster dataset with extent overlapping <code>spf</code>
<code>num.samps</code>	number of samples to create in each feature (see details)
<code>replace</code>	whether to sample with or without replacement
<code>force.min</code>	whether to force <code>num.samps</code> 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), then `[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 = FALSE)
```

## 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

## 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, template 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).

## 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)
```



# Index

nh\_best, [1](#)  
nh\_burn, [2](#)  
nh\_group, [4](#)  
nh\_patchdrop, [5](#)  
nh\_sample, [6](#)  
nh\_stack, [7](#)