

# Package ‘nhSDM’

February 28, 2018

**Title** Tools for Natural Heritage SDMs

**Date** 2018-02-08

**Version** 0.0.1

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

**Depends** R (>= 3.4.1)

**Imports** methods, sf, raster, sp

**Suggests** dplyr, lwgeom

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

## R topics documented:

nh_burn . . . . .	1
nh_group . . . . .	2
nh_sample . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

nh_burn	<i>Add areas represented by features ('burn-in') to a binary raster SDM output</i>
---------	--

---

## 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.

## Usage

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

**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; threshold value to apply to raster
buffer	numeric; spatial buffer around spf to include in burn-in
...	additional paramaters to raster::writeRaster

**Details**

If buffer is not 0, areas within that buffer around spf that are greater than the minimum cell value (min.cell) within spf are set to 1 in the classified raster.

If orig.thresh is NULL, the min.cell value will be used as a global threshold. When a given orig.thresh is less than the min.cell value, the result will be the same as a thresholded raster using orig.thresh.

**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.

## Author(s)

David Bucklin

## Examples

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

## End(Not run)
```

---

nh\_sample

*Create points in polygons in reference raster cells*

---

## Description

For each spatial polygon, a given number (`num.samps`) of points are created in cells that the polygon 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 polygon (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 polygons, even if they are duplicates

## Details

num.samps can be 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 polygon. If left NULL, num.samps will be set to the number of cells [n] intersecting the polygon. If a proportion is given (e.g., 0.5), then  $[n * \text{num.samps}]$  will be returned. If a single integer is given, num.samps points will be sampled in each polygon.

When replace = FALSE and force.min = FALSE (defaults), each cell can only contain one point (across each polygon, 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 polygon, the number of samples for that polygon 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 polygon. It only produces duplicates if num.samps exceeds the number of cells intersecting the polygon, in which case it will replicate the samples until num.samps is reached.

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

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

## 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)
```

# Index

nh\_burn, [1](#)  
nh\_group, [2](#)  
nh\_sample, [3](#)