

# Package ‘RDH’

September 20, 2016

**Type** Package

**Title** Range Data Helper

**Version** 1.01

**Date** 2014-24-01

**Author** Nicholas M A Crouch

**Maintainer** Nicholas Crouch <ncrouc2@uic.edu>

**Description** A collection of functions for help in analyzing spatial data available from Birdlife International and NatureServe

**Depends** sp, maptools, ape, rgeos, raster

**License** GPL-3

## R topics documented:

RDH-package . . . . .	2
example.phylogeny . . . . .	2
example.shapefiles . . . . .	3
migration.distance . . . . .	3
nameFix . . . . .	4
num.polygons . . . . .	5
pairwise.range.overlap . . . . .	6
polygon.cleaning . . . . .	7
rangeOverlaps.phyloDist . . . . .	8
raster.extract . . . . .	9
seasonal.distribution.ranges . . . . .	10
seasonal.values . . . . .	12
<b>Index</b>	<b>13</b>

---

RDH-package

*Range Data Helper*

---

### Description

A collection of functions for help in analyzing spatial data available from Birdlife International and NatureServe

### Details

Package: RDH  
Type: Package  
Version: 1.0  
Date: 2014-24-01  
License: GPL-3

### Author(s)

Nicholas M A Crouch

Maintainer: Nicholas M A Crouch <ncrouc2@uic.edu>

### See Also

rgeos, raster

---

example.phylogeny

*Example phylogenetic data for nine species*

---

### Description

Example phylogenetic data for nine species

### Usage

```
data(example.phylogeny)
```

### Format

A phylogeny of class phylo

---

example.shapefiles	<i>Example species range information</i>
--------------------	--

---

**Description**

Example range information for nine species

**Usage**

```
data(example.shapefiles)
```

**Format**

A list with each component being an object of class SpatialPolygonsDataFrame.

**Source**

BirdLife International and Natureserve, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

---

migration.distance	<i>Migration distance</i>
--------------------	---------------------------

---

**Description**

Calculates the distance between the breeding and wintering ranges of migratory species.

**Usage**

```
migration.distance(shp)
```

**Arguments**

shp	A shapefile
-----	-------------

**Details**

If you are unsure whether your species of interest has breeding and wintering components to its range you can check using the function `seasonal.values`

**Value**

Returns a `data.frame` of shapefiles with species broken down into wintering and breeding ranges

**Author(s)**

Nicholas M A Crouch

## References

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

---

nameFix

*Birdlife & NatureServe Name Fix*

---

## Description

Cleans the names of shapefiles downloaded from Birdlife International and NatureServe, leaving only Genus and species names

## Usage

```
nameFix(x, as.list=FALSE)
```

## Arguments

x	A character string. Can either be a single value or a list of names.
as.list	Logical. Should the output be returned in the format of a list?

## Details

The format of the input values needs to be unchanged from the format used by Birdlife International and NatureServe.

## Value

Returns a character string of the same length as the input

## Author(s)

Nicholas M A Crouch

## Examples

```
# Using the filename for Accipiter collaris

shp.name <- "Accipiter_collaris_3443_NS.shp"
require(RDH)

nameFix(shp.name)

#Accipiter_collaris
```

---

num.polygons	<i>Number of Polygons</i>
--------------	---------------------------

---

**Description**

Counts the number of unique polygons that make up a shapefile

**Usage**

```
num.polygons(x, return.sizes=FALSE)
```

**Arguments**

x	A vector containing one or more shapefiles of class "SpatialPolygonsDataFrame"
return.sizes	Logical. If TRUE, output returns the sizes of the polygons contained within a shapefile

**Details**

Counts the number of unique polygons that make up a shapefile. Using `return.sizes=TRUE` means that the sizes of each polygon is returned, rather than the number of files.

**Value**

If `return.sizes=FALSE`, the default, then function returns numeric string of the same length as the number of shapefiles passed to the function.

If `return.sizes=TRUE`, then a `data.frame` will be returned containing the size of each polygon within the shapefile will be returned.

Use `lapply` when passing a list of shapefiles to the function, see examples. When using `lapply`, a list will be returned where each element is either a numeric value of polygons with the shapefile, or a `data.frame` containing the size of each polygon within the shapefile.

**Author(s)**

Nicholas M A Crouch

**Examples**

```
library(RDH)
require(maptools)
require(rgeos)
require(sp)

data(example.shapefiles)

shp <- example.shapefiles[[1]]
```

```
#Number of polygons within the shapefile
num.polygons(shp)

#Sizes of the polygons within a shapefile
num.polygons(shp, return.sizes=TRUE)
```

---

```
pairwise.range.overlap
```

*Pairwise percentage range overlap*

---

### Description

Calculate what percentage of each species' range is shared with every other species.

### Usage

```
pairwise.range.overlap(x)
```

### Arguments

x	A list of shapefiles
---	----------------------

### Details

A value of 0 signifies that the species in that row, shares none of its range with the species in the corresponding column. A value of 100 would signify it shares all of its range.

Important! Some species spatial data provided by Birdlife International and NatureServe have some slight anomalies which might need to be fixed before using this function. See `polygon.cleaning`.

### Value

Returns a `data.frame` where the rows correspond to the species being analyzed, and columns to each other species in the set.

### Author(s)

Nicholas M A Crouch

### References

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

**Examples**

```
library(RDH)

data(example.shapefiles)

# An initial run reveals two shapefiles have some problems
## Not run:
output <- pairwise.range.overlap(example.shapefiles)

## End(Not run)

example.shapefiles[[5]] <- polygon.cleaning(example.shapefiles[[5]])
example.shapefiles[[8]] <- polygon.cleaning(example.shapefiles[[8]])

output <- pairwise.range.overlap(example.shapefiles)
```

---

polygon.cleaning	<i>Polygon Cleaner</i>
------------------	------------------------

---

**Description**

Cleans polygon files to remove any errors caused by missing data/intersecting nodes

**Usage**

```
polygon.cleaning(x, resolution=0.5)
```

**Arguments**

x	A shapefile
resolution	Amount of detail of original polygon retained during process, see Details

**Details**

Due to some problems encountered with functions in other packages, this function will clean a polygon through a rasterization process. The `resolution` argument describes how much of the detail of the original shapefile will be retained. Note: increasing the resolution (a lower value passed to the function) can result in huge amounts of memory required to perform the task, due to the large shapefiles used for avian ranges. Set at 0.5, a satisfactory amount of resolution is retained for large ranges. For smaller, particularly island ranges, I recommend lowering this value to 0.1, see Examples for comparisons.

**Author(s)**

Nicholas M A Crouch

## References

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

## Examples

```
library(RDH)

data(example.shapefiles)

shp <- example.shapefiles[[1]]

shp.clean <- polygon.cleaning(shp)

par(mfrow=c(1,2))

plot(shp, col=rgb(1,0,0,1/2))
title("Original Shapefile")

plot(shp.clean, col=rgb(0,0,1,1/2))
title("After Cleaning Process")
```

---

rangeOverlaps.phyloDist

*Range overlaps and species phylogentic distances*

---

## Description

For a group of species, counts the number of times each species range overlaps with every other species. Also calculates the total phylogenetic distance between species with overlapping ranges.

## Usage

```
rangeOverlaps.phyloDist(shapefiles, phylo, weight.relative=TRUE)
```

## Arguments

shapefiles	A list containing species shapefiles.
phylo	A phylogeny of class phylo where the tips correspond to the species shapefiles contained within shapefiles. Can be NULL, see details.
weight.relative	Logical. Should more closely related species be weighted more than more distantly related species? Defaults to TRUE. When TRUE, performs $1/\text{dist}$ , where $\text{dist}$ is the patristic distance between two species.



**Details**

Progress will be returned whilst the function is running. If `phylo = NULL` then only number of overlapping ranges will be returned. When running with phylogenetic information, function will check whether all species with range data appear in the phylogeny. If species are missing from the phylogeny, an error message is returned. When calculating phylogenetic distance, running time for the function is increased.

**Value**

If `phylo = NULL`, returns a `data.frame` with two columns: species ID and the number of overlapping ranges. If phylogenetic data included, returns a `data.frame` with three columns: species ID, total phylogenetic distance and the number of overlapping ranges.

**Author(s)**

Nicholas M A Crouch

**References**

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

**Examples**

```
library(RDH)

data(example.shapefiles)
data(example.phylogeny)

# With phylogeny
output <- rangeOverlaps.phyloDist(example.shapefiles, example.phylogeny)

output

# Without phylogeny

output1 <- rangeOverlaps.phyloDist(example.shapefiles, phylo=NULL)

output1
```

---

raster.extract

*Raster extraction*


---

**Description**

Extracts data for a given species range from a raster file and returns statistical test based on that extracted values.

**Usage**

```
raster.extract(x, raster.layer, stat.test="mean", na.val=NULL)
```

**Arguments**

<code>x</code>	A shapefile
<code>raster.layer</code>	An object of class <code>rasterLayer</code>
<code>stat.test</code>	Character string specifying the test to perform on the extracted values. Defaults to mean
<code>na.val</code>	Raster files can contain specific values which correspond to missing data. Passing a numeric value to <code>na.val</code> will remove all instances corresponding to missing data before performing the specified statistical test

**Details**

Can pass to a list of shapefiles using `lapply`, will return a list of numeric values.

**Value**

Returns a numeric value based on the statistical test used.

**Author(s)**

Nicholas M A Crouch

**References**

Hijmans, R. J., S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis, 2005. Very high resolution interpolated climate surfaces for global land areas. *International journal of Climatology* 25:1965–1978. URL [www.worldclim.org](http://www.worldclim.org)

---

```
seasonal.distribution.ranges
```

*Seasonal range distribution*

---

**Description**

Breaks down a species range into new shapefiles which represent the breeding and wintering ranges respectively

**Usage**

```
seasonal.distribution.ranges(x)
```

**Arguments**

<code>x</code>	A shapefile
----------------	-------------

**Details**

If you are unsure about which seasonal values are present for a species, you can use the function [seasonal.values](#). When x is a list, use lapply followed by unlist, this is to put all sub-lists into one single list, see examples.

**Value**

Returns a list of shapefiles with species broken down into wintering and breeding ranges

**Author(s)**

Nicholas M A Crouch

**References**

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

**Examples**

```
require(RDH)

data(example.shapefiles)

# Using a single file
s <- example.shapefiles[[1]]

output <- seasonal.distribution.ranges(s)

# Using a list
output.from.list <- lapply(example.shapefiles, seasonal.distribution.ranges)

# Without using unlist

length(output.from.list)

#Correcting

corrected.output.from.list <- unlist(output.from.list)

length(corrected.output.from.list)

# All new shapefiles are in a single list
```

---

seasonal.values	<i>Seasonal values</i>
-----------------	------------------------

---

**Description**

Returns description of which discrete seasonal occurrences are recorded for a species, given its range information

**Usage**

```
seasonal.values(x)
```

**Arguments**

x	A shapefile
---	-------------

**Value**

Returns a character string with specified seasonal occurrences of each species

**Author(s)**

Nicholas M A Crouch

**References**

BirdLife International and NatureServe, 2012, Bird species distribution maps of the world, BirdLife International, Cambridge, UK and NatureServe, Arlington, USA

**Examples**

```
library(RDH)

data(example.shapefiles)

shp <- example.shapefiles[[1]]

seasonal.values(shp)
```

# Index

\*Topic **package**

RDH-package, [2](#)

example.phylogeny, [2](#)

example.shapefiles, [3](#)

migration.distance, [3](#)

nameFix, [4](#)

num.polygons, [5](#)

pairwise.range.overlap, [6](#)

polygon.cleaning, [7](#)

rangeOverlaps.phyloDist, [8](#)

raster.extract, [9](#)

RDH (RDH-package), [2](#)

RDH-package, [2](#)

seasonal.distribution.ranges, [10](#)

seasonal.values, [11](#), [12](#)