

# How to displaying the IUCN range of your species on top of an assignment map in IsoriX?

*The IsoriX core team*

*2017-06-20*

Welcome to **IsoriX**, this vignette will explain to you how to add a second layer to your assignment map. This additional layer could for example display the IUCN range of your species.

In this vignette we are using a dataset that contains distribution information on *Myotis bechsteinii* assessed for *The IUCN Red List of Threatened Species<sup>TM</sup>* (IUCN 2016). We downloaded the data from the IUCN Red List website (<http://www.iucnredlist.org/>) for free (after filling a short questionnaire).

## Before starting

### Performing the IsoriX workflow

Because we want to illustrate the IUCN range of a species on top of an assignment map, we need to have proceeded thorough an entire IsoriX workflow including the assignment itself. To make this quick, we will simply replicate the minimal example given for the function `isofind()` to run an entire workflow:

```
## We load the necessary data:
data(calibdata)
data(assigndata)
data(elevraster)
data(countries)
data(oceanmask)
data(Europefit)

## We prepare the elevation raster
elevationraster <- relevate(elevation.raster = elevraster, isofit = Europefit)

## We build the isoscape:
isoscape <- isoscape(elevation.raster = elevationraster, isofit = Europefit)

## Loading required package: raster
## Loading required package: sp

## We fit the calibration model:
calib <- calibfit(calib.data = calibdata, isofit = Europefit)

## We perform the assignment:
assignment <- isofind(assign.data = subset(assigndata, species == "Myotis_bechsteinii"),
                      isoscape = isoscape,
                      calibfit = calib)
```

### Getting the files containing the IUCN range for your species ready

Download the .ZIP file containing the information about the geographical range of your species from the IUCN Red List website (<http://www.iucnredlist.org/>). The .ZIP file stores the species range as an ESRI

shapefile. We assume below that the zip file is present in your working directory. If this is not the case, either change your directory (see `?setwd()`) or identify your working directory using `getwd()` and then move your file .ZIP there.

We first need to unzip the .ZIP file. You can do that outside R, but you can also do that directly within R:

```
unzip("species_14123.zip")
```

If you have unzipped the compressed file successfully, a new folder with a code name corresponding to the species should be present in your working directory and this folder should contain your ESRI shapefile (which consists of a collection of different files):

```
dir("./species_14123")
```

```
## [1] "METADATA for Digital Distribution Maps of The IUCN Red List of Threatened Species.pdf"
## [2] "ReadMe.txt"
## [3] "Red List Terms & Conditions of Use.pdf"
## [4] "species_14123.cpg"
## [5] "species_14123.dbf"
## [6] "species_14123.lyr"
## [7] "species_14123.prj"
## [8] "species_14123.sbn"
## [9] "species_14123.sbx"
## [10] "species_14123.shp"
## [11] "species_14123.shp.xml"
## [12] "species_14123.shx"
```

## Adding the additional layer

First, we need to transform the shapefile into a `SpatialPolygonsDataFrame`. This can be done with the package **rgdal** which provides the function `readOGR()` to read in shapefiles in R. We must provide the path to the data as a first argument and the name of the layer as a second one.

```
library(rgdal)
myotis_bechsteinii <- readOGR("./species_14123/", "species_14123")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "./species_14123/", layer: "species_14123"
## with 5 features
## It has 18 fields
```

The `readOGR()` function will also read the .prj file. If it exists, you can return the projection as follows:

```
proj4string(myotis_bechsteinii)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
```

If the shapefile has a different projection than your assignment raster you must project the layer according to the projection of your assignment. You can extract such projection from any raster contained in the assignment object. We will use here the one storing p-values for individuals. Here, the projections do not differ between our assignment rasters and the IUCN data, but we can nonetheless perform such a projection to illustrate the code (it just has no effect). This step requires the package **raster**.

```
library(raster)
myotis_bechsteinii <- spTransform(myotis_bechsteinii, crs(assignment$indiv$pv))
```

Now we have created a polygon that shows the IUCN range of our species in the same projection as our assignment. This could be added to an assignment map to display the boundaries of the species range. In

order to mask area outside the range of the species, we need to get the inverse polygon. This step requires the package **rgeos**.

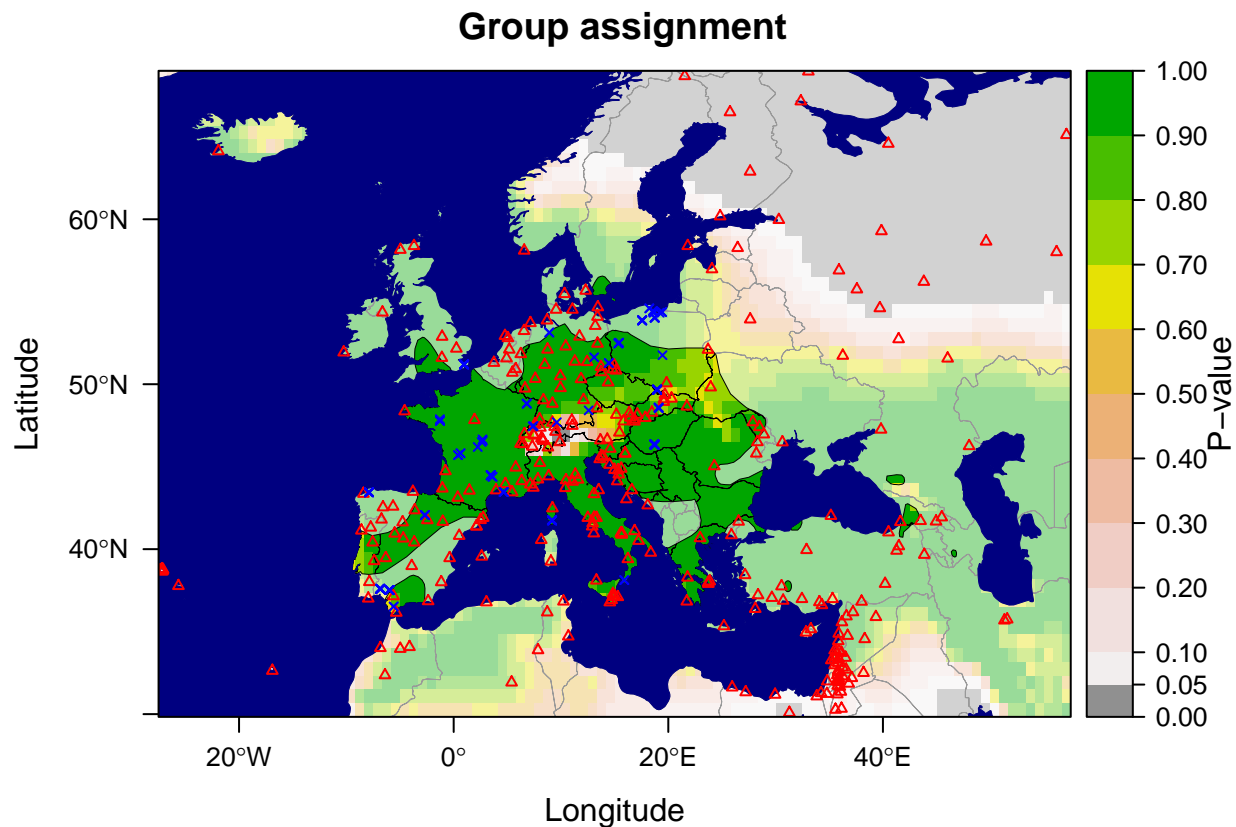
```
library(rgeos)

## rgeos version: 0.3-23, (SVN revision 546)
## GEOS runtime version: 3.4.2-CAPI-1.8.2 r3921
## Linking to sp version: 1.2-4
## Polygon checking: TRUE

limits <- as(extent(assignment$indiv$pv), "SpatialPolygons")
proj4string(limits) <- crs(assignment$indiv$pv)
iucnmask <- gDifference(limits, myotis_bechsteinii)
```

We need to pass our **SpatialPolygons** in the first mask argument of the **plot()** function and use the second mask argument to display the oceans (so that the ocean masks the species range). The fill argument is specifying the displaying colour of your polygon. We can use the function **rgb()** to specify a color with an alpha transparency.

```
plot(assignment, who = "group",
     borders = list(borders = countries),
     mask = list(mask = iucnmask, fill = rgb(1, 1, 1, 0.6)),
     mask2 = list(mask = oceanmask, fill = "navyblue", col = "navyblue"))
```



## References

IUCN (International Union for Conservation of Nature). 2016. *Myotis bechsteinii*. *The IUCN Red List of Threatened Species*. Version 2017-1. Available at: <http://www.iucnredlist.org>.