Biodiversity Metrics

P Galante, A Paz

2022-11-29

To aid conservation planning and decision-making, for example by understanding species' representativeness and complementarity across conservation areas, we can combine species' range estimates across multiple species groups to calculate community-level metrics of conservation interest that reflect different dimensions of biological diversity such as species richness, spatial and temporal turnover, endemism, and phylogenetic diversity. We provide an example below to calculate these metrics for a dataset of Colombian primates (Henao-Díaz et al. 2020) using the 'changeRangeR' package. For information on single species range change metrics, see the single-species vignette .

Biodiversity metrics

Using multiple species for landscape-level species metrics

Load packages that will be useful for these analyses

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## Please note that rgdal will be retired during 2023,
## plan transition to sf/stars/terra functions using GDAL and PROJ
## at your earliest convenience.
## See https://r-spatial.org/r/2022/04/12/evolution.html and https://github.com/r-spatial/evolution
## rgdal: version: 1.6-2, (SVN revision 1183)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.5.3, released 2022/10/21
## Path to GDAL shared files: /Users/kass1/Library/R/arm64/4.2/library/rgdal/gdal
##  GDAL does not use iconv for recoding strings.
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 9.1.0, September 1st, 2022, [PJ_VERSION: 910]
## Path to PROJ shared files: /Users/kass1/Library/R/arm64/4.2/library/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.5-1
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
```

```
library(raster)
library(picante)
```

```
## Loading required package: ape
```

```
##
## Attaching package: 'ape'

## The following objects are masked from 'package:raster':
##
##     rotate, zoom

## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:raster':
##
##     getData
```

```
library(phylobase)
```

```
##
## Attaching package: 'phylobase'

## The following object is masked from 'package:ape':
##
##     edges
```

```
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:nlme':
##
##     collapse

## The following objects are masked from 'package:raster':
##
##     intersect, select, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(changeRangeR)
```

```
## To get started with multiple species, see the demo with
## vignette(package='changeRangeR')
```

```
library(ape)
```

Species Richness

Here we measure species richness as the number species predicted to be present in each pixel.

Load and manage spatial data

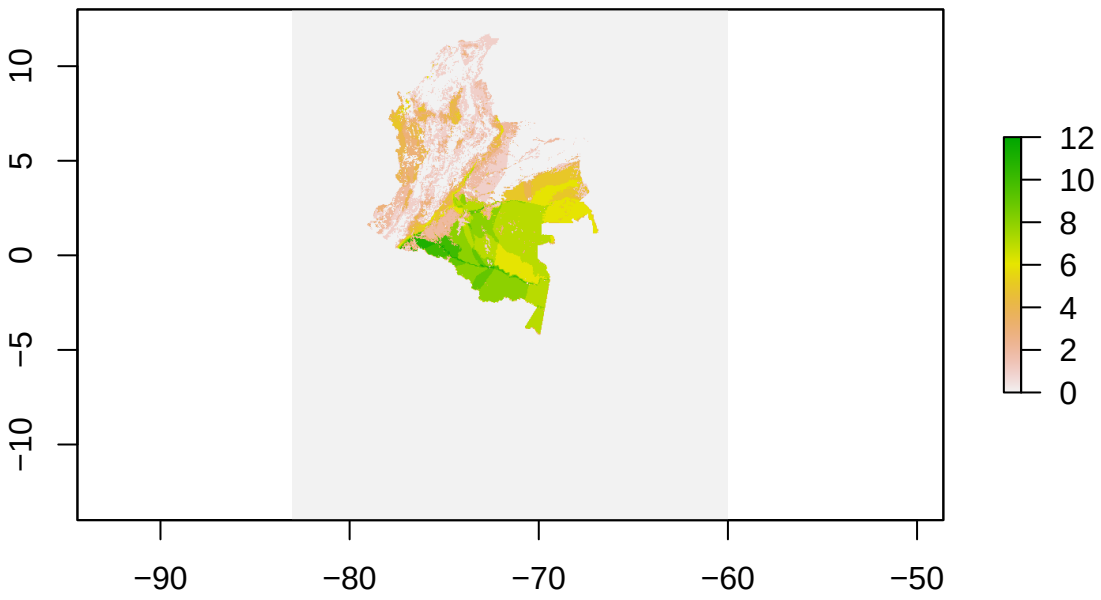
Loading the SDM data and ensuring the names match the phylogeny, and removing some resolved species. NOTE: For calculating species richness, named rasters are not strictly necessary.

Data are from: Henao-Díaz, F. et al. (2020). Atlas de la biodiversidad de Colombia. Primates. Instituto de Investigación de Recursos Biológicos Alexander von Humboldt. Bogotá D. C., Colombia. 51 pp.

```
# Load binary maps/SDMs
primRas <- raster::stack(list.files(path = paste0(system.file(package="changeRangeR"), "/extdata/DemoDa
# Because of species' name resolutions in the phylogeny we use here, we need to rename some of the spec
names(primRas) <- gsub("*_veg_coarse", "", names(primRas))
# Drop models that are resolved into one group; Cebus_albifrons
primRas <- dropLayer(primRas, c("Cebus_versicolor", "Cebus_leucocephalus", "Cebus_malitiosus", "Cebus_a
oldnames <- names(primRas)
namesToChange <- c("Cebus_all_albifrons", "Cheracebus_lugens", "Cheracebus_medemi", "Lagothrix_lagothri
newNames <- c("Cebus_albifrons", "Callicebus_lugens", "Callicebus_medemi", "Lagothrix_lagotricha", "Leo
oldnames[oldnames %in% namesToChange] <- newNames
names(primRas) <- oldnames
```

Calculate species richness by summing the raster values

```
SR <- sum(primRas, na.rm = T)
plot(SR)
```



Phylogenetic diversity

Phylogenetic diversity can be calculated from binary range-maps, or as here, thresholded SDMs. We combine these maps with phylogenetic trees to calculate this measure of diversity. Here, we calculate phylogenetic diversity as the sum of branch lengths using Faith's PD (Faith, 1992).

It is important to note that species-experts can edit binary SDMs, such as is done in the R package *maskRangeR* to add or remove areas.

Load and manage spatial data

Loading the SDM data and ensuring the names match the phylogeny

```
# Load binary maps/SDMs
primRas <- stack(list.files(path = paste0(system.file(package="changeRangeR"), "/extdata/DemoData/SDM/c
# Because of species' name resolutions in the phylogeny we use here, we need to rename some of the spec
names(primRas) <- gsub("*_veg_coarse", "", names(primRas))
# Drop models that are resolved into one group; Cebus_albifrons
primRas <- dropLayer(primRas, c("Cebus_versicolor", "Cebus_leucocephalus", "Cebus_malitiosus", "Cebus_a
oldnames <- names(primRas)
namesToChange <- c("Cebus_all_albifrons", "Cheracebus_lugens", "Cheracebus_medemi", "Lagothrix_lagothri
newNames <- c("Cebus_albifrons", "Callicebus_lugens", "Callicebus_medemi", "Lagothrix_lagotricha", "Leo
oldnames[oldnames %in% namesToChange] <- newNames
names(primRas) <- oldnames
```

To save time in this vignette, let's crop down the primate rasters to a much smaller extent

```
e <- extent(c(-75, -70, 0, 2.5))
primRas <- crop(primRas, e)
```

Load phylogenetic tree

THIS NEEDS A CITATION (ANDREA?)

```
primTree <- read.nexus(paste0(system.file(package="changeRangeR"), "/extdata/DemoData/phyloTree/output.
```

Convert NA values to 0

```
# set all cells that have NA values to 0
for (i in 1:nlayers(primRas)){
  primRas[[i]][is.na(primRas[[i]])] <- 0
}
```

Convert the rasterStack to a data.frame

```
# convert raster to dataframe
commDat <- as.data.frame(primRas)
# remove rows that are NA
commDat <- na.omit(commDat)
row.names(commDat) <- 1:nrow(commDat)
```

Calculate phylogenetic diversity, here using the 500th tree as an example

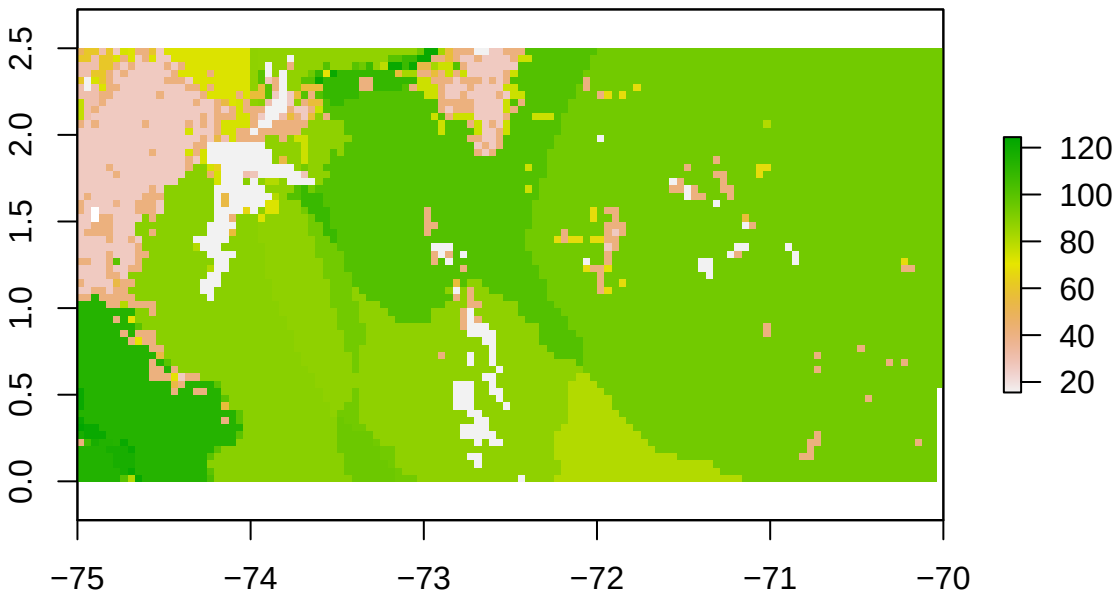
```
user_phylogeny <- primTree[1]$tree_6532
phydiv <- pd(samp = commDat, tree = user_phylogeny, include.root = TRUE)
```

Add the phylogenetic diversity pixel values to the community data data.frame

```
commDat$pd <- phydiv$PD
```

To create a raster of phylogenetic diversity, start with an empty raster. Add the primate raster values from the primRas rasterstack to get a single raster showing every pixel for which we have a PD value.

```
pd_ras <- sum(primRas, na.rm=T)
pd_ras[!is.na(pd_ras)] <- commDat$pd
pd_ras[pd_ras == 0] <- NA
plot(pd_ras)
```



Phylogenetic endemism Phylogenetic endemism combines phylogenetic trees with spatial data. PE is calculated by dividing each branch length over the total number of locations where the species that descend from that branch are found (Rosauer et al., 2009).

Here, we can calculate PE from a function written by, and hosted by Dan Rosauer on Github

To begin, load and manage the spatial data to make sure the names match those in the phylogeny

```
# Load binary maps/SDMs
primRas <- stack(list.files(path = paste0(system.file(package="changeRangeR"), "/extdata/DemoData/SDM/c
# Because of species' name resolutions in the phylogeny we use here, we need to rename some of the spec
names(primRas) <- gsub("*_veg_coarse", "", names(primRas))
# Drop models that are resolved into one group; Cebus_albifrons
primRas <- dropLayer(primRas, c("Cebus_versicolor", "Cebus_leucocephalus", "Cebus_malitiosus", "Cebus_a
oldnames <- names(primRas)
namesToChange <- c("Cebus_all_albifrons", "Cheracebus_lugens", "Cheracebus_medemi", "Lagothrix_lagothri
newNames <- c("Cebus_albifrons", "Callicebus_lugens", "Callicebus_medemi", "Lagothrix_lagotricha", "Leo
oldnames[oldnames %in% namesToChange] <- newNames
names(primRas) <- oldnames
```

As before, let's crop the rasters for processing speed.

```
e <- extent(c(-75, -70, 0, 2.5))
primRas <- crop(primRas, e)
```

Load in the phylogenetic trees

```
primTree <- read.nexus(paste0(system.file(package="changeRangeR"), "/extdata/DemoData/phyloTree/output.n
```

The function, provided, on github, by Dan Rosauer (https://github.com/DanRosauer/phylospatial) as defined in:

Rosauer, D.F., Blom, M.P.K., Bourke, G., Catalano, S., Donnellan, S., Gillespie, G., Mulder, E., Oliver, P.M., Potter, S., Pratt, R.C. and Rabosky, D.L., 2016. Phylogeography, hotspots and conservation priorities: an example from the Top End of Australia. Biological Conservation, 204, pp.83-93.

Prepare the raster data

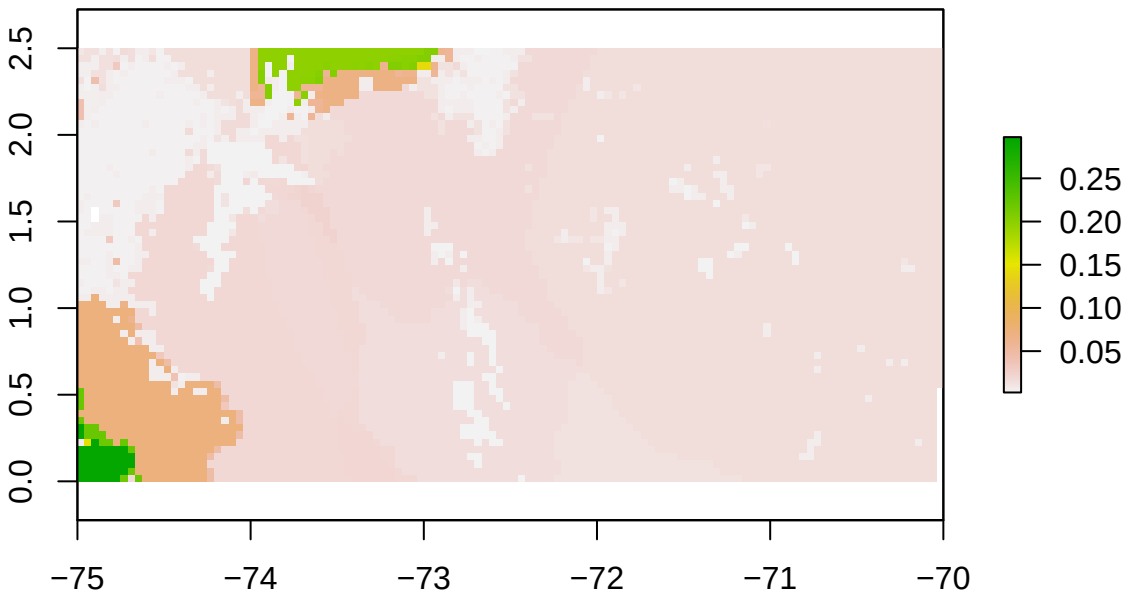
```
## Convert the binary primate SDMs to a point data.frame, removing the X and Y data.
Allxy <- rasterToPoints(primRas)
sites <- as.data.frame(Allxy[,1:ncol(Allxy)])
## Change all NA values to 0
sites[is.na(sites)] <- 0
```

Calculate PE

```
pEprimates <- calc_PE(phylo.tree = primTree[[1]], sites_x_tips = sites, presence = "presence")
```

Next, transform the pixel-wise PE values back into a raster

```
## cbind the pixel centroids with PE values and convert to a raster
PExyz <- cbind(Allxy[,1:2], pEprimates$PE)
PE.ras <- rasterFromXYZ(PExyz)
PE.ras[PE.ras == 0] <- NA
plot(PE.ras)
```



Species Endemism

Here, species endemism is defined as a value for each cell equal to the number of species found in that cell divided by the total number of cells in which they are found

Load and manage spatial data

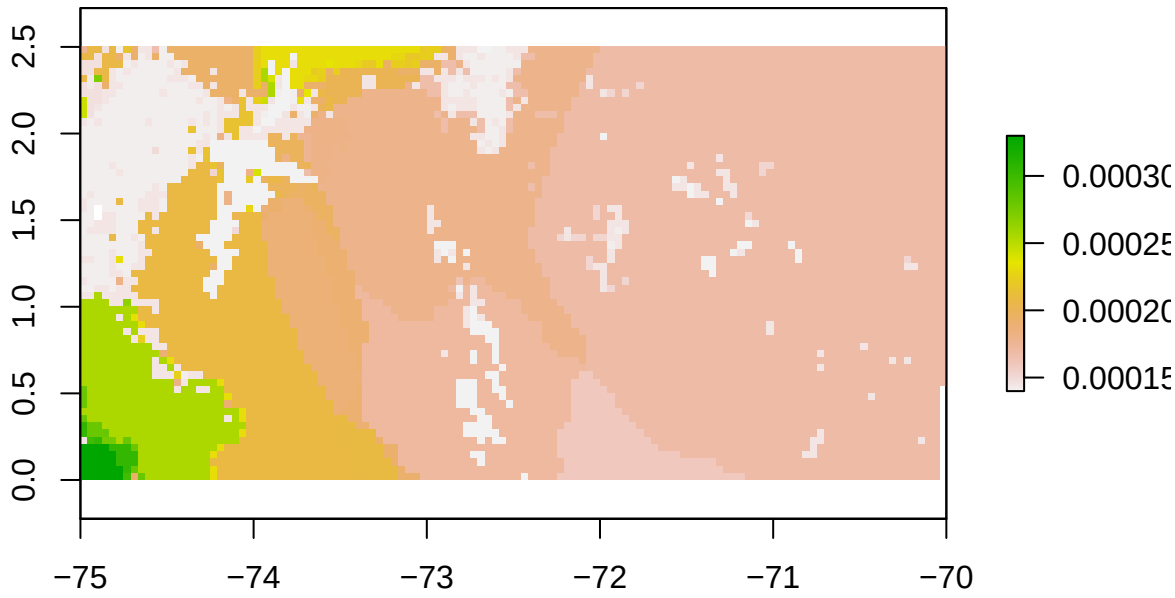
```
# Load binary maps/SDMs
primRas <- stack(list.files(path = paste0(system.file(package="changeRangeR"), "/extdata/DemoData/SDM/c
# Because of species' name resolutions in the phylogeny we use here, we need to rename some of the spec
names(primRas) <- gsub("*_veg_coarse", "", names(primRas))
# Drop models that are resolved into one group; Cebus_albifrons
primRas <- dropLayer(primRas, c("Cebus_versicolor", "Cebus_leucocephalus", "Cebus_malitiosus", "Cebus_a
oldnames <- names(primRas)
namesToChange <- c("Cebus_all_albifrons", "Cheracebus_lugens", "Cheracebus_medemi", "Lagothrix_lagothri
newNames <- c("Cebus_albifrons", "Callicebus_lugens", "Callicebus_medemi", "Lagothrix_lagotricha", "Leo
oldnames[oldnames %in% namesToChange] <- newNames
names(primRas) <- oldnames
```

Again, let's crop the rasters for processing speed.

```
e <- extent(c(-75, -70, 0, 2.5))
primRas <- crop(primRas, e)
```

Calculate species endemism

```
sp.End <- SpiesEndemism(primRas)
plot(sp.End)
```



Complementarity

It is often useful to spatially quantify biodiversity data to assess geographically where patterns exist. For example, to find or compare areas that contain more than 50% of species richness, thresholding a raster by a statistical value can be important.

Loading the SDM data and ensuring the names match the phylogeny

```
# Load binary maps/SDMs
primRas <- stack(list.files(path = paste0(system.file(package="changeRangeR"), "/extdata/DemoData/SDM/c
# Because of species' name resolutions in the phylogeny we use here, we need to rename some of the spec
names(primRas) <- gsub("*_veg_coarse", "", names(primRas))
# Drop models that are resolved into one group; Cebus_albifrons
primRas <- dropLayer(primRas, c("Cebus_versicolor", "Cebus_leucocephalus", "Cebus_malitiosus", "Cebus_a
oldnames <- names(primRas)
namesToChange <- c("Cebus_all_albifrons", "Cheracebus_lugens", "Cheracebus_medemi", "Lagothrix_lagothri
newNames <- c("Cebus_albifrons", "Callicebus_lugens", "Callicebus_medemi", "Lagothrix_lagotricha", "Leo
oldnames[oldnames %in% namesToChange] <- newNames
names(primRas) <- oldnames
```

Calculate species richness again and convert zeros to NA

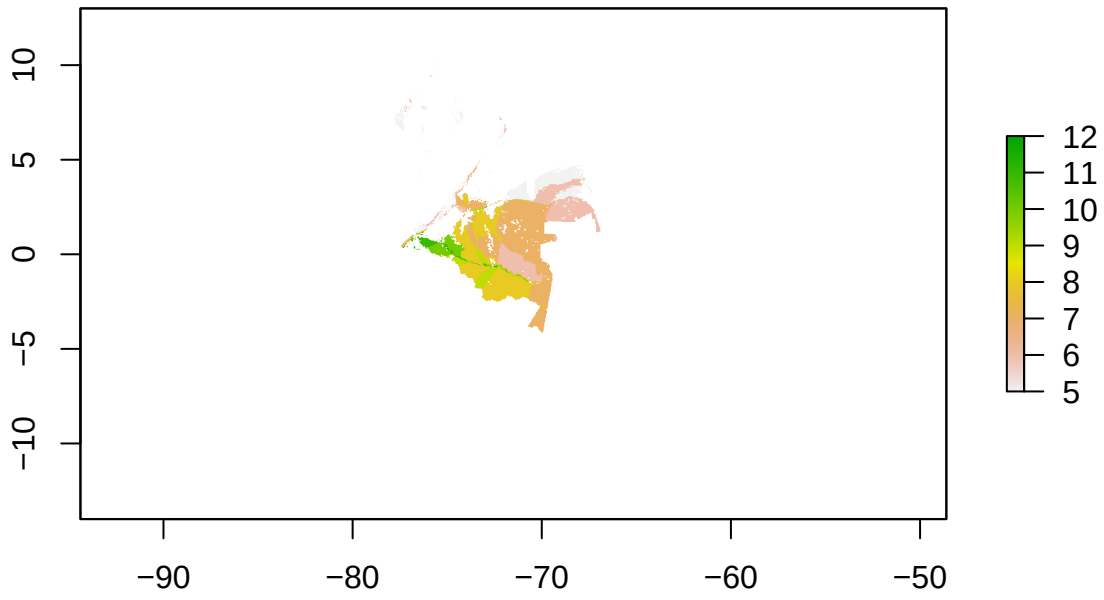
```
SR <- sum(primRas, na.rm = T)
SR[SR == 0] <- NA
```

Calculate the raster value at the 50th percentile

```
thresholdValue <- quantile(SR, probs = 0.5)
```

Threshold the raster by the chosen value and plot

```
SR[SR < thresholdValue] <- NA
plot(SR)
```



Users can then calculate the overlap of this new raster with a shapefile. For example, a user can find the proportion of 50% of species richness that is protected.

Load protected areas in Colombia and calculate the ratio of overlap

```
PAs <- readRDS(file.path(system.file(package="changeRangeR"), "extdata/DemoData/shapefiles", "WDPA_COL_
# View the fields
colnames(PAs)
```

```
## NULL
```

```
# Pick the field you are interested in
field <- "DESIG_ENG"
category <- "All"
ratio.Overlap <- ratioOverlap(r = SR, shp = PAs, field = field, category = category, subfield = FALSE)
print(ratio.Overlap)
```

```
## $maskedRange
## $maskedRange[[1]]
## class      : RasterLayer
## dimensions : 657, 562, 369234  (nrow, ncol, ncell)
## resolution : 0.0417, 0.0417  (x, y)
## extent     : -83.2085, -59.7731, -14.1884, 13.2085  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : memory
## names      : layer
## values     : 5, 10  (min, max)
##
##
##
## $ratio
## [1] "Percentage of range within All is 0.844%"
##
## $correlation
## NULL
```

Users can also find the amount of species richness is currently protected.

Calculate species richness again and convert zeros to NA

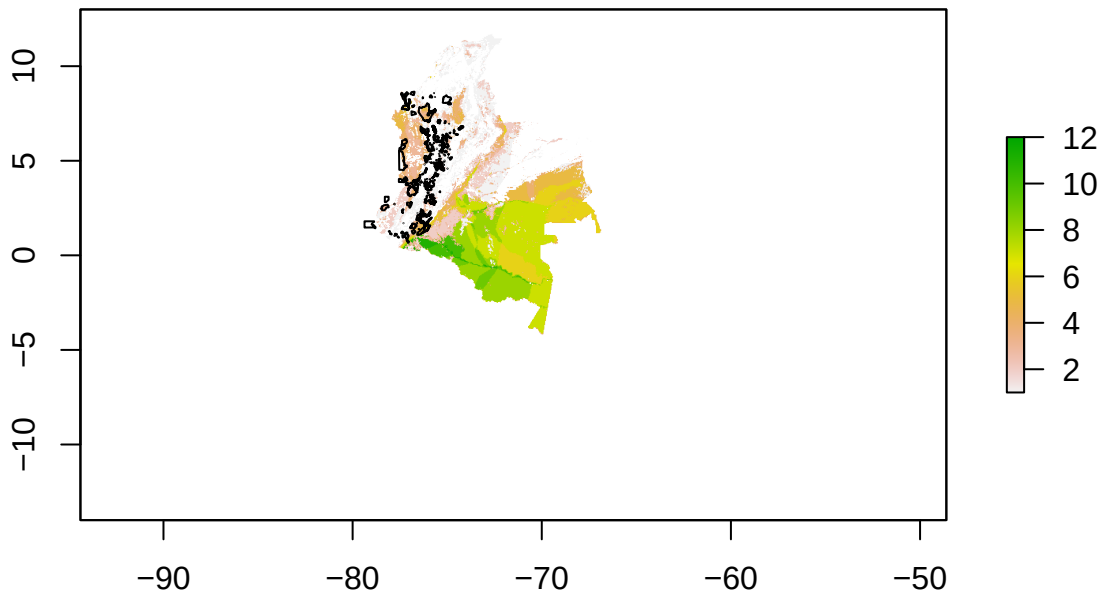
```
SR <- sum(primRas, na.rm = T)
SR[SR == 0] <- NA
```

Load protected areas in Colombia and calculate what percentage of species richness is currently protected.

```
PAs <- readRDS(file.path(system.file(package="changeRangeR"), "extdata/DemoData/shapefiles", "WDPA_COL_
# View the fields
colnames(PAs)
```

```
## NULL
```

```
# Pick the field you are interested in
field <- "DESIG_ENG"
category <- "All"
# Mask species richness by protected areas to find richness within protected areas
rmask <- raster::mask(SR, PAs)
# Take a look
plot(SR)
plot(PAs, add = T)
```




```
comp <- complementarity(ras1 = SR, ras1mask = rmask)
print(comp)
```

```
## $Percent_of_Total
## [1] 2.14925
##
## $Percent_unique_values
##    Var1 Freq.x Freq.y     percent
## 1     1   9314    300  3.22095770
## 2     2   4268    216  5.06091846
## 3     3   3054    273  8.93909627
## 4     4   2822    344 12.18993622
## 5     5   3144    136  4.32569975
## 6     6   4012     21  0.52342971
## 7     7   6853     13  0.18969794
## 8     8   4402      3  0.06815084
## 9     9    814      0  0.00000000
## 10   10    834      3  0.35971223
## 11   11    302      0  0.00000000
## 12   12     14      0  0.00000000
```