

# How do IUCN Range maps of birds compare to SDMs in Indonesian New Guinea? What are the main predictors for this distribution?

Jonathan Gehret, Paul Leister

31.08.2021

## Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Methods</b>	<b>1</b>
2.1. Main script . . . . .	2
2.2. Species data . . . . .	3
2.3. Create presence-absence data . . . . .	4
2.4. Indicators . . . . .	9
2.5. Species distribution models . . . . .	10
2.6. Comparison SDMs and IUCN . . . . .	35
7.2 Code . . . . .	40
<b>1. Introduction</b>	<b>40</b>
<b>2. Methods</b>	<b>41</b>
2.1. Main script . . . . .	41
2.2. Species data . . . . .	41
2.3. Create presence-absence data . . . . .	41
2.4. Indicators . . . . .	42
2.5. Species distribution models . . . . .	42
2.6. Comparison SDMs and IUCN . . . . .	42
7.2 Code . . . . .	42

## 1. Introduction

## 2. Methods

To find the species to be used for our analysis we used the loops in `bird_plot_loops.R` to plot all species from the Gbif and IUCN data sets to find birds with sufficient observation points and different ranges to be used for our project.

## 2.1. Main script

The required libraries and the setup of the IUCN and gbif data sets are done in the MCMMB\_Main.R script. Notably, the HBIF data are created only for species with greater than 100 observations. The GBIF data were downloaded manually beforehand, however the r library “rgbif” could be used instead. Additionally, the island borders are read in and given the name “regio”.

```
source("MCMMB_Main.R")

## Warning: package 'biomod2' was built under R version 4.0.5

## biomod2 3.5.1 loaded.
##
## Type browseVignettes(package='biomod2') to access directly biomod2 vignettes.

## Warning: package 'rgdal' was built under R version 4.0.5

## Loading required package: sp

## rgdal: version: 1.5-23, (SVN revision 1121)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.2.1, released 2020/12/29
## Path to GDAL shared files: D:/Documents/R/win-library/4.0/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 7.2.1, January 1st, 2021, [PJ_VERSION: 721]
## Path to PROJ shared files: D:/Documents/R/win-library/4.0/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-5
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.
## Overwritten PROJ_LIB was D:/Documents/R/win-library/4.0/rgdal/proj

## Warning: package 'raster' was built under R version 4.0.5

## Warning: package 'rgeos' was built under R version 4.0.5

## rgeos version: 0.5-5, (SVN revision 640)
## GEOS runtime version: 3.8.0-CAPI-1.13.1
## Linking to sp version: 1.4-5
## Polygon checking: TRUE

## Warning: package 'sf' was built under R version 4.0.5

## Linking to GEOS 3.9.0, GDAL 3.2.1, PROJ 7.2.1

## Warning: package 'spatialEco' was built under R version 4.0.5

##
## Attaching package: 'spatialEco'
```

```

## The following object is masked from 'package:raster':
##
##      shift

## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:spatialEco':
##
##      combine

## The following objects are masked from 'package:rgeos':
##
##      intersect, setdiff, union

## 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

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## EOF in Zeichenkette

## Warning in sp::proj4string(obj): CRS object has comment, which is lost in output

## although coordinates are longitude/latitude, st_intersection assumes that they are planar

## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries

```

## 2.2. Species data

The script species\_loops.R includes two functions each for creating lists with the input bird species data for gbif (“get\_gbif\_birds(bird\_names,gbif\_crop,regio)”) and IUCN (“get\_iucn\_birds(bird\_names,birdlife,regio)”) and plotting them, respectively.

```

# loading of the described functions
source("species_loops.R")

# cassowaries didn't have enough data
#bird_names = c("Casuarius bennetti", "Casuarius casuarius", "Casuarius unappendiculatus")

```

```

# we chose Pachycephala instead
bird_names = c("Pachycephala lorentzi",
               "Pachycephala meyeri",
               "Pachycephala schlegelii",
               "Pachycephala simplex",
               "Pachycephala soror")

# these names can be changed to any other bird species occurring in the data sets, for fully automated c
# bird_names = c(your_favorite_species_here)

# for plotting gbif data next to iucn data
par(mfrow = c(2,length(bird_names)))

# create list with birds out of the iucn and gbif data
iucn_birds = get_iucn_birds(bird_names,birdlife,regio)
gbif_birds = get_gbif_birds(bird_names,gbif_crop,regio)

```

Pachycephala lorentzi      Pachycephala meyeri      Pachycephala schlegelii      Pachycephala simplex      Pachycephala soror



Pachycephala lorentzi      Pachycephala meyeri      Pachycephala schlegelii      Pachycephala simplex      Pachycephala soror



## 2.3. Create presence-absence data

While biomod2 also brings with it capabilities to create pseudo-absence data, we built a script to do it. First centroids for the area were created, based on that an equal number to the presence data of absence points is created.

"Be sure to take care when considering the use of pseudo-absences versus true absences for species dist

modeling. Similarly, it is extremely important to consider the influence of sampling bias in the data used to train models. Further reading: e.g., Guillera-Arroita et al. 2015, Kramer-Schadt et al. 2013, and Merow et al. 2013."

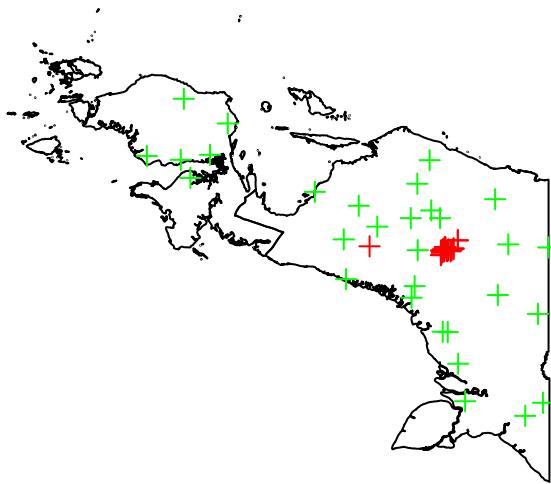
```
# load presence_absence.R script including the function
source("presence_absence.R")
presence_absence_list = create_pseudo_absence(regio,gbif_birds)

## Warning in showSRID(upprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition

## Warning in showSRID(upprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition

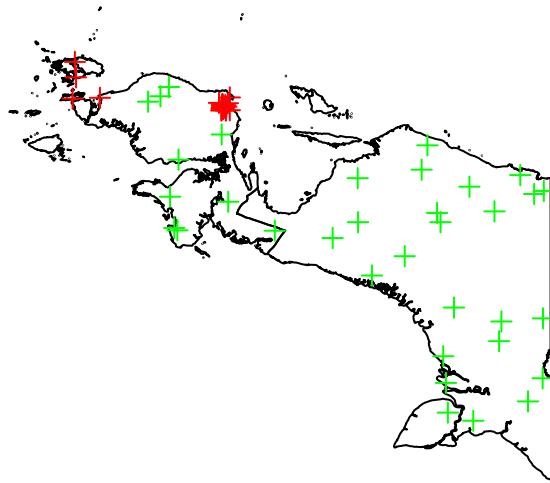
## Warning in showSRID(upprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## Pachycephala lorentzi



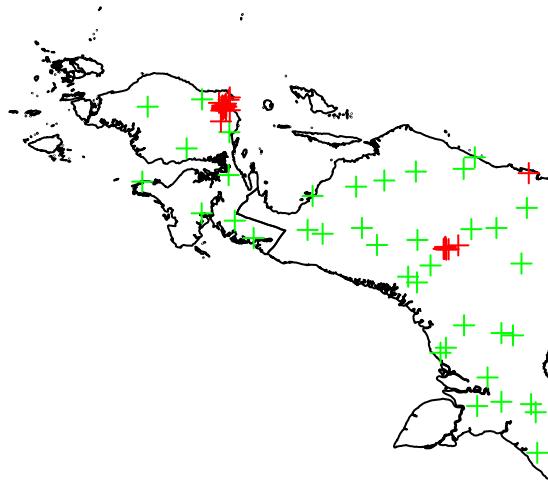
```
## Warning in showSRID(upprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala meyeri**



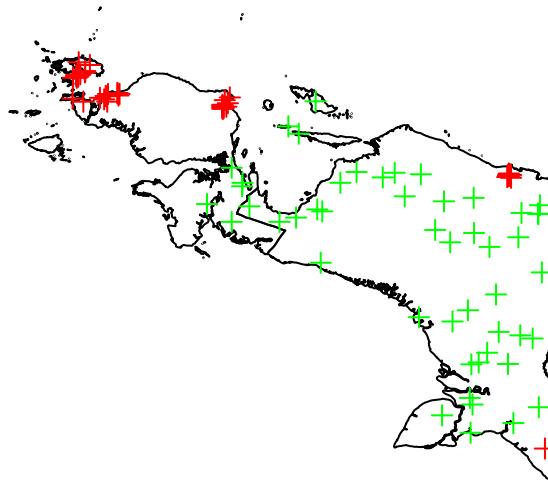
```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala schlegelii**

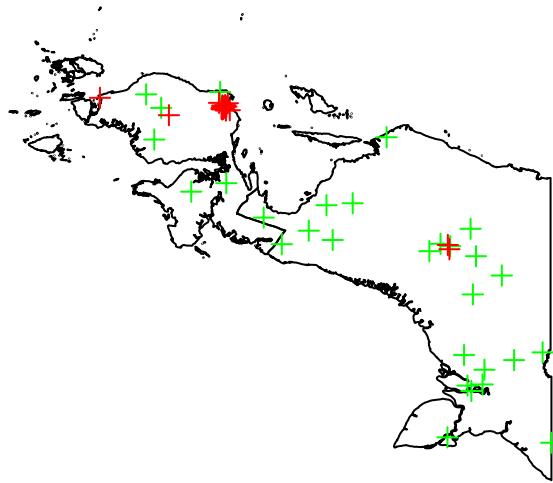


```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala simplex**



## Pachycephala soror



### 2.4. Indicators

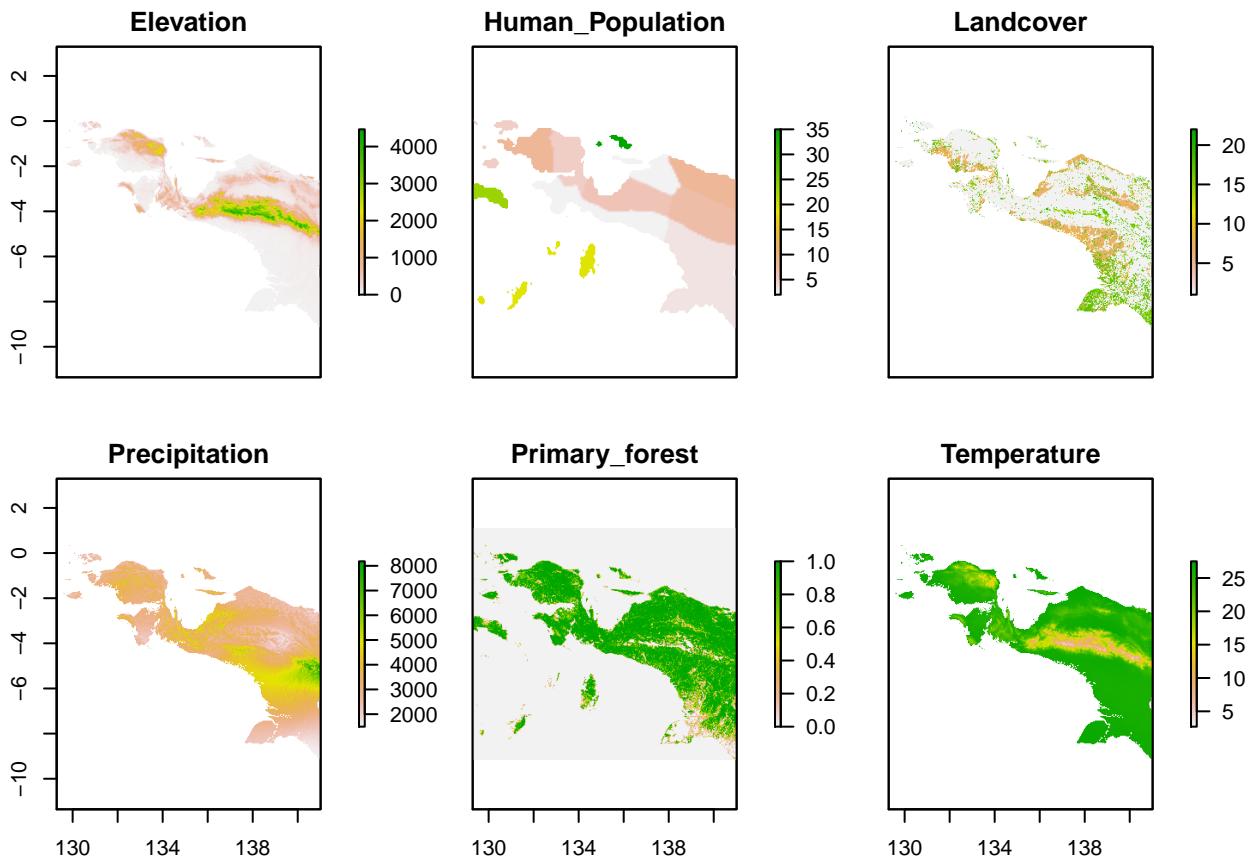
The Indicator.R script is used to read in data for the various indicators (precipitation, elevation, landcover, primary\_forest, temperature, human population) to be used for the SDM. These are then cropped to the target region and saved in .tif format in the folder data/indicator\_stack/ for later use. As the process for some of the bigger files takes a lot of time, ready to use indicator .tif files are to be found in the aforementioned folder. These will then be loaded and stacked later on before the sdm creation.

```
# where all the predictors are loaded in
source("Indicator.R")

# stacking all predictors found in folder indicator stack in .tif format
# predictors: elevation, precipitation, temperature, primary forest, landcover, population
tif_predictors = stack(list.files(path = "data/indicator_stack/",
                                    full.names = TRUE,
                                    pattern = ".tif"))

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition

plot(tif_predictors)
```



## 2.5. Species distribution models

Using biomod2, the indicator stack, and the presence absence data with the gbif birds of interest the species distribution models with the methods GLM, GAM, RF and ANN are produced. There are also alternative methods available, which can be shown using “BIOMOD\_ModelingOptions()”. Each of the individual SDM versions can also be modified further, as can be seen in our script we modified k for the GAM (line 25). The data set was not split in two for evaluation, however this step could be done next time. The calculated models are saved to file (argument SaveObj = TRUE in BIOMOD\_Modeling()).

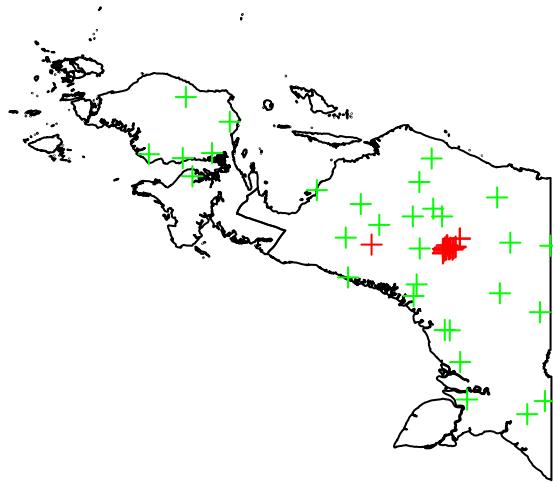
```
# loading biomod script and passing ensemble models to test_sdm
source("sdm_biomod2.R")

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition

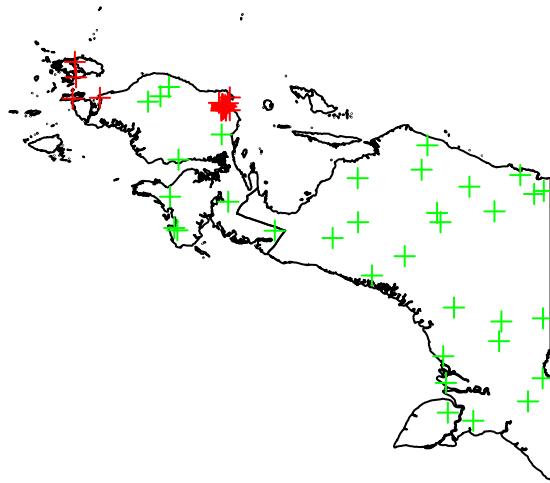
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala lorentzi**



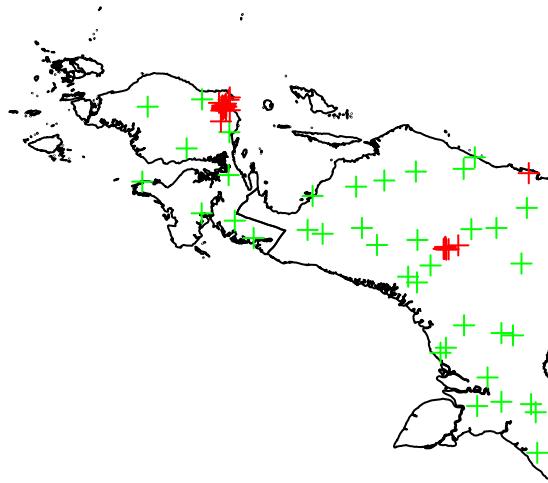
```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala meyeri**



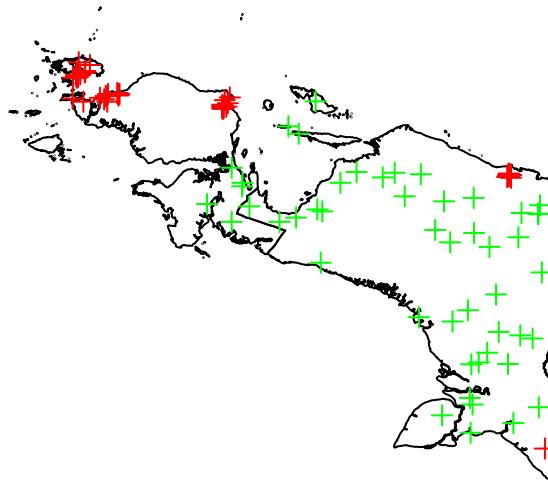
```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

## **Pachycephala schlegelii**

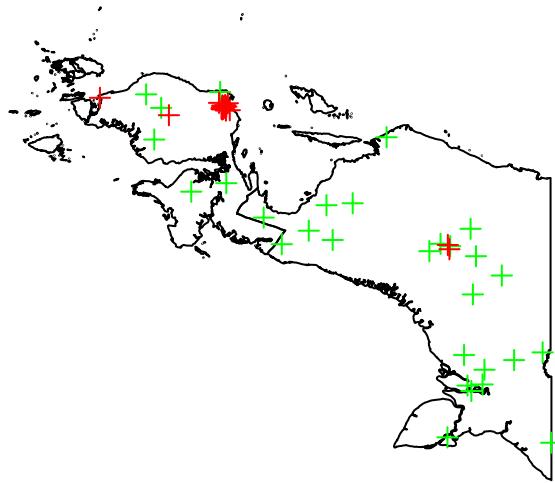


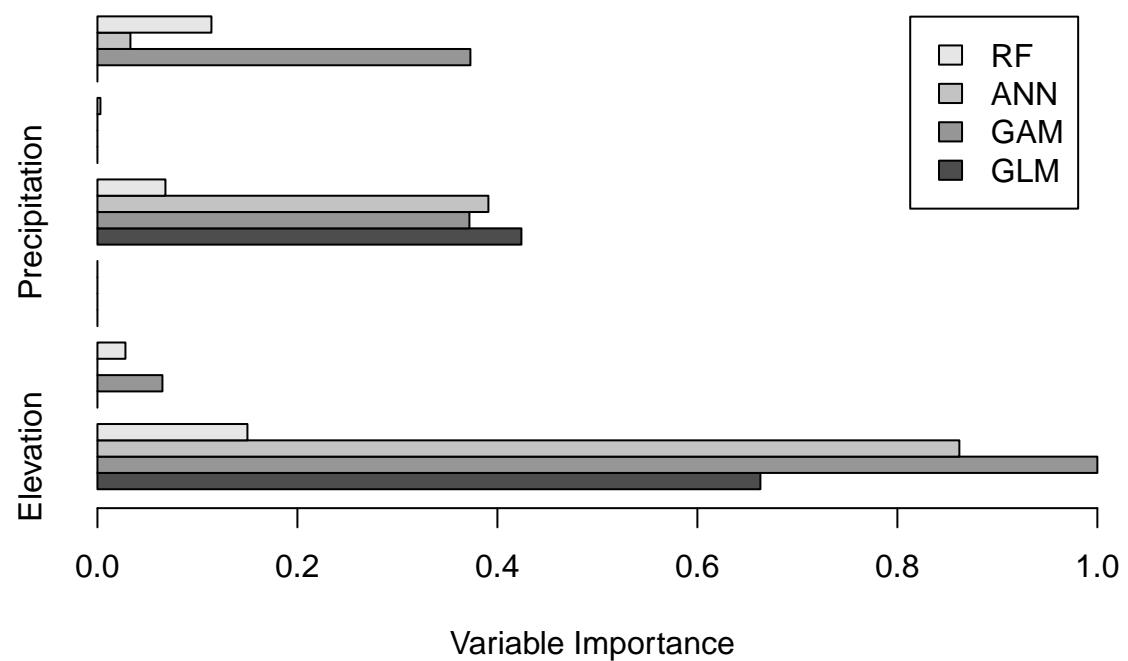
```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on WGS84 ellipsoid in Proj4
## definition
```

**Pachycephala simplex**

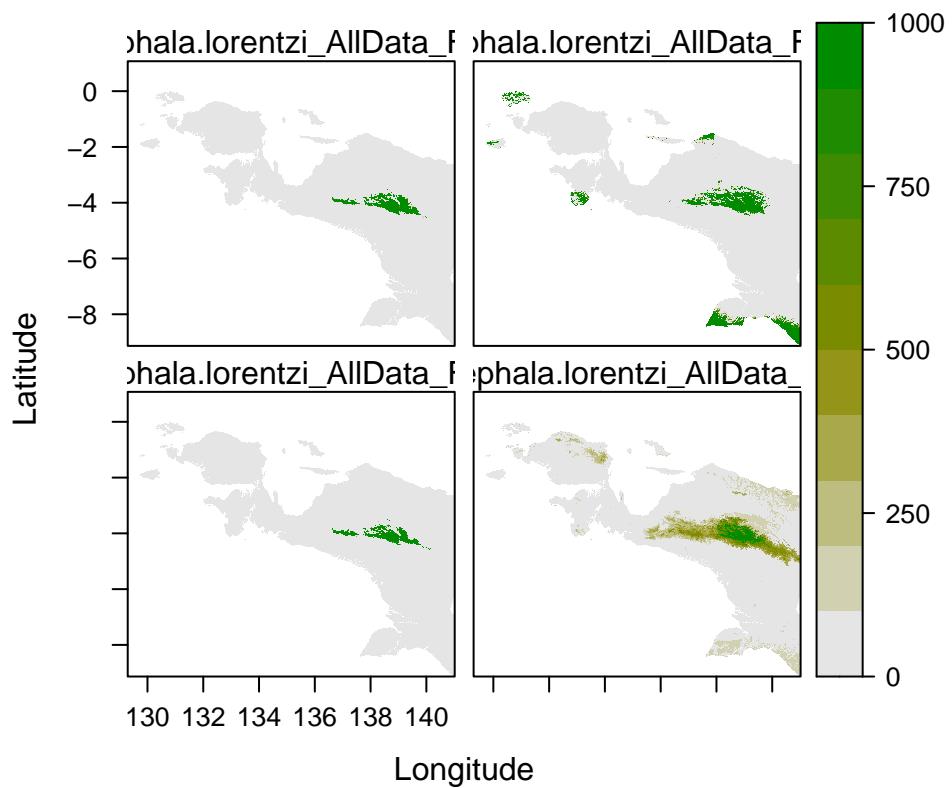


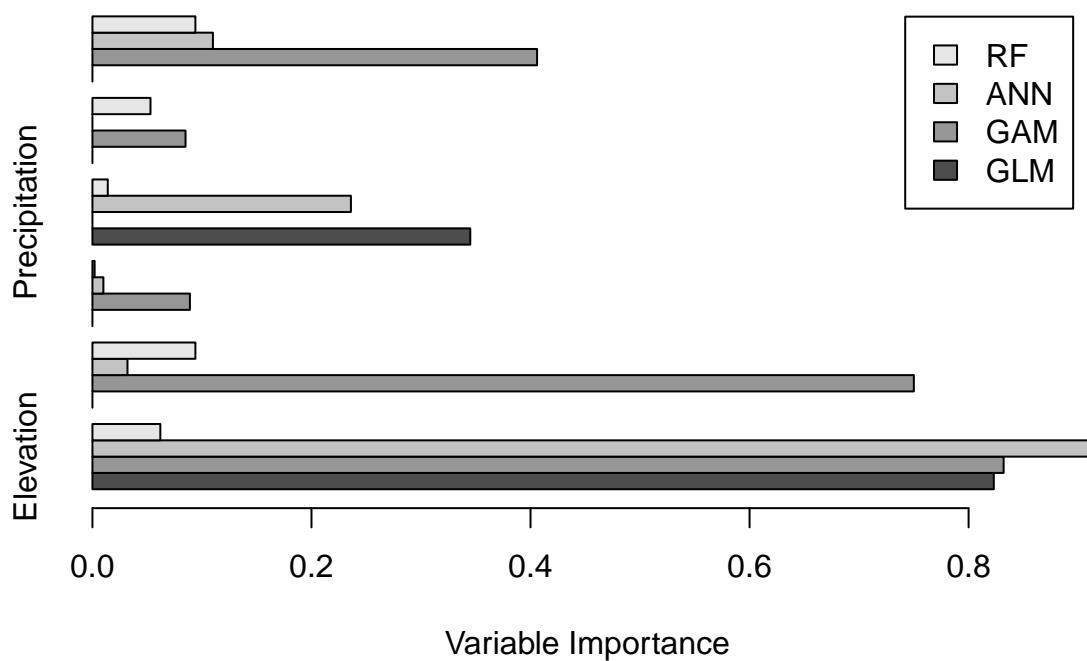
**Pachycephala soror**



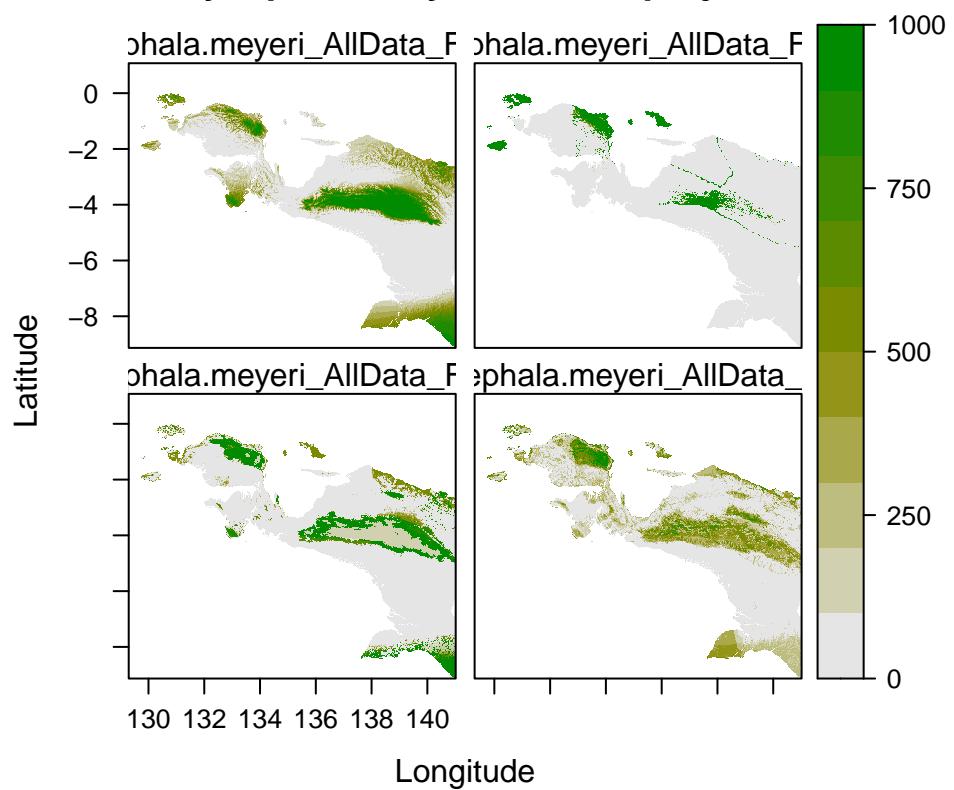


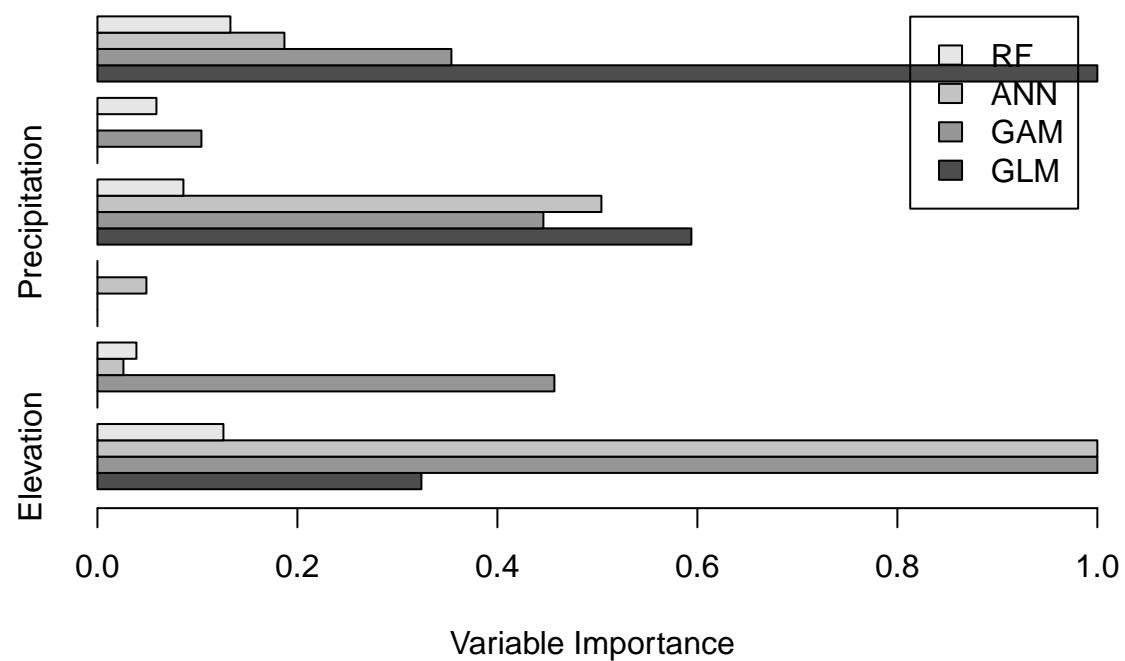
### **Pachycephala.lorentzi current projections**



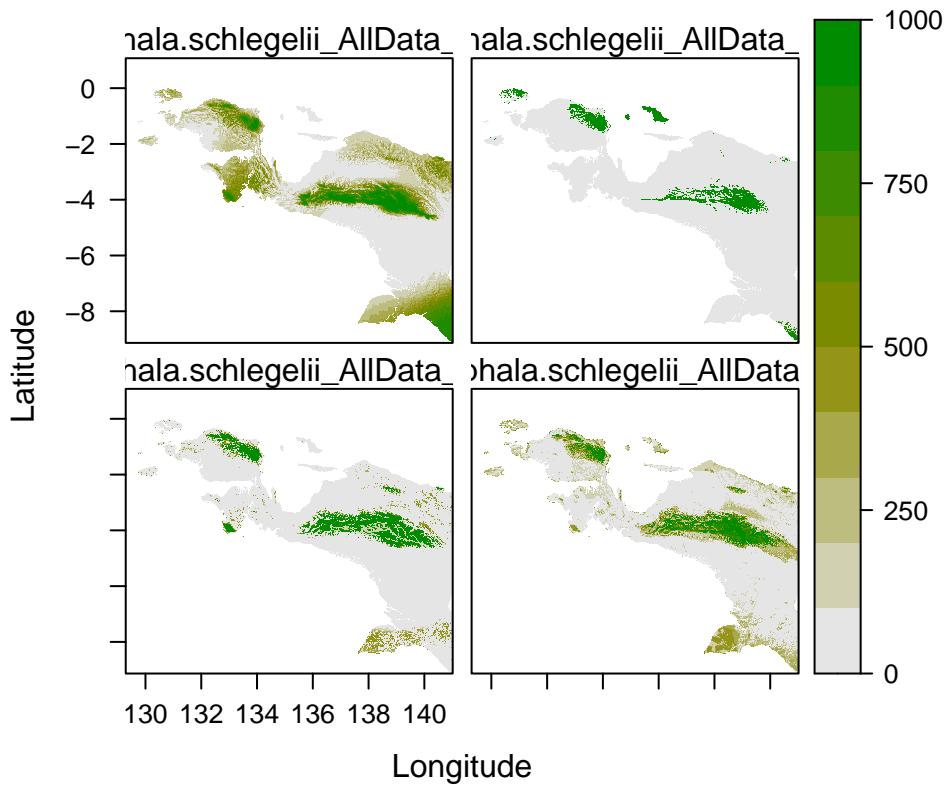


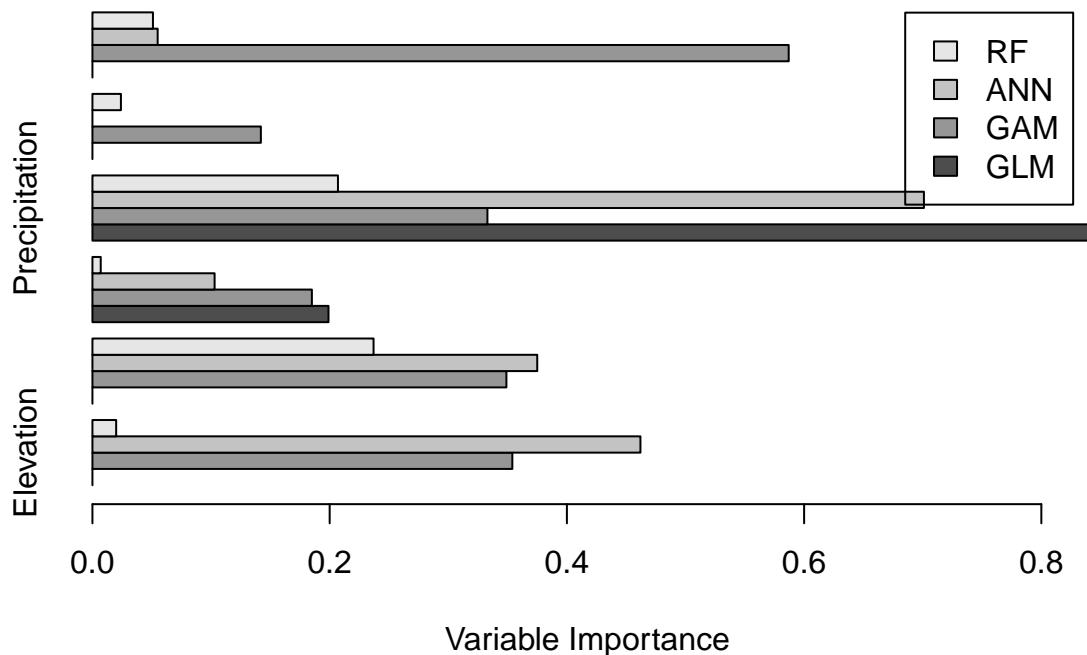
### **Pachycephala.meyeri current projections**





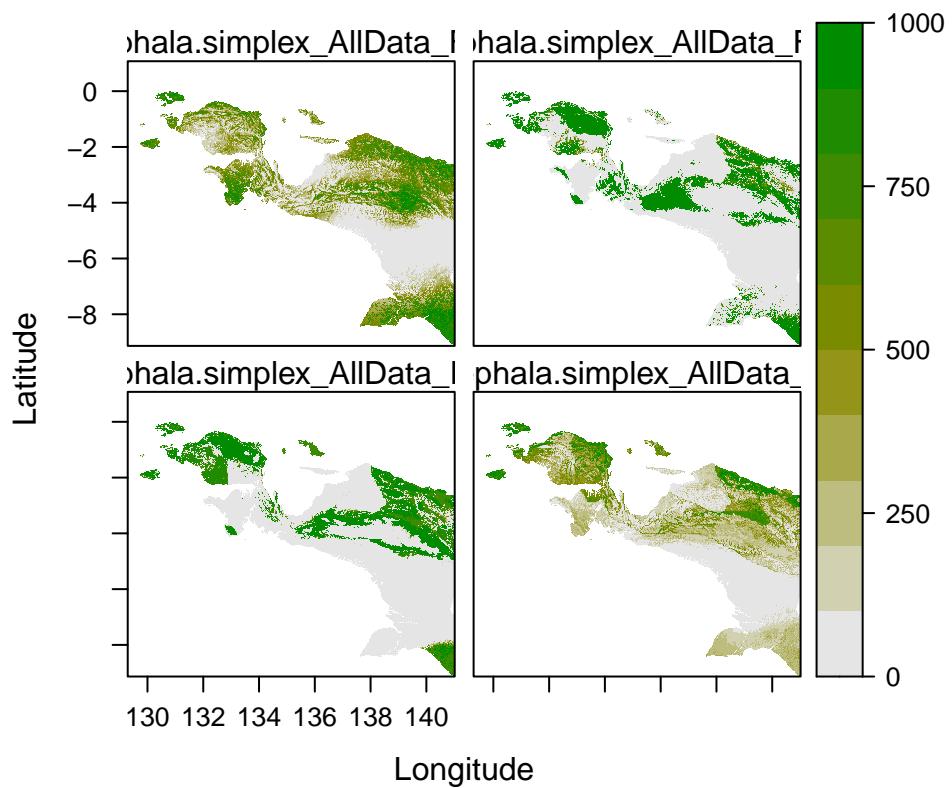
### Pachycephala.schlegelii current projections

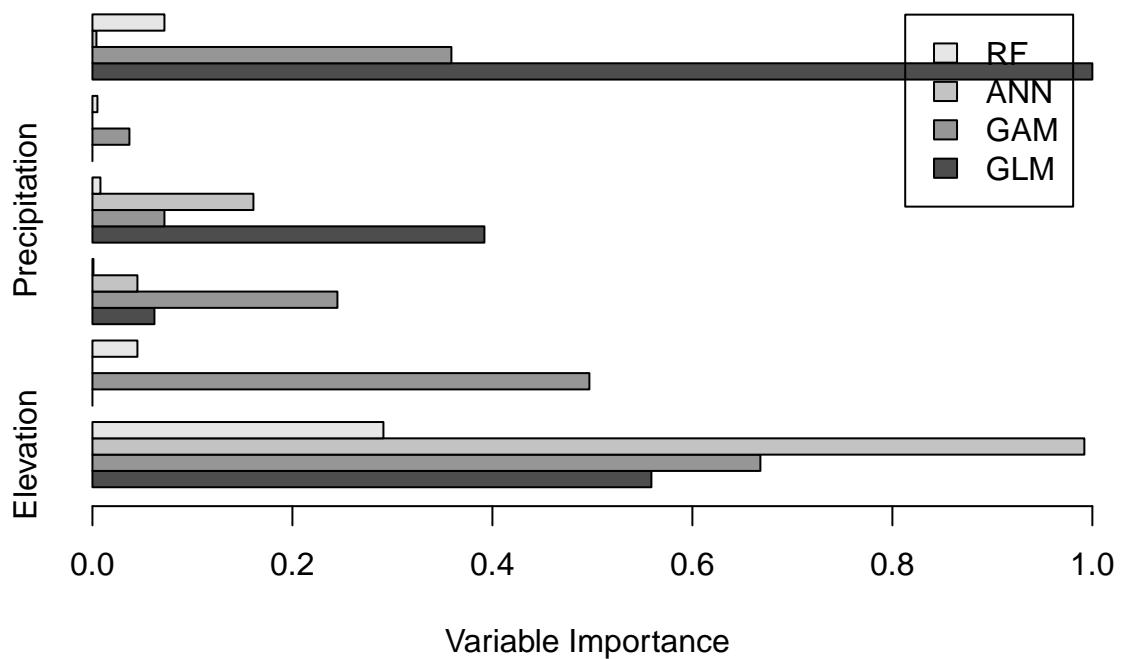




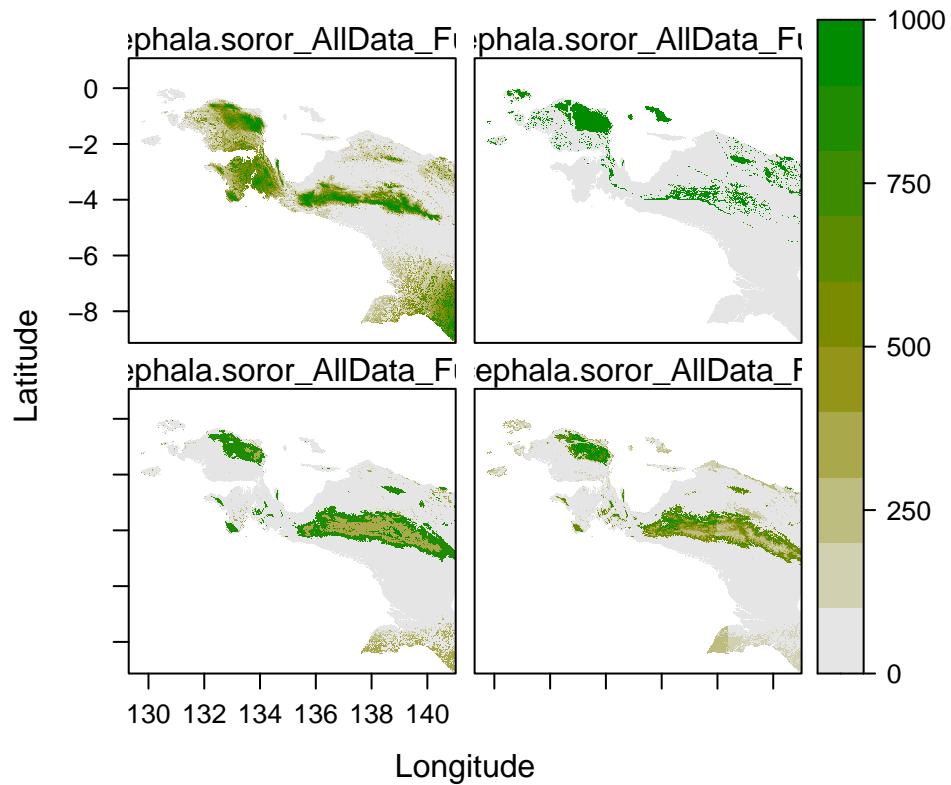
```
## Warning in newton(lsp = lsp, X = G$X, y = G$y, Eb = G$Eb, UrS = G$UrS, L =
## G$L, : Fitting terminated with step failure - check results carefully
```

### Pachycephala.simplex current projections

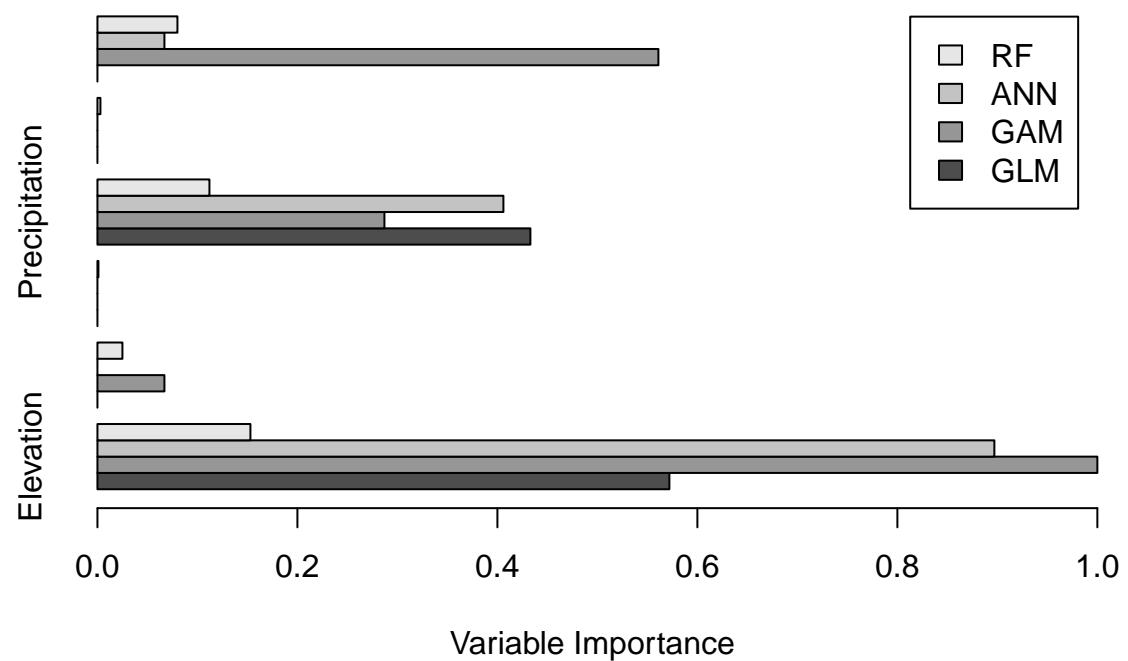




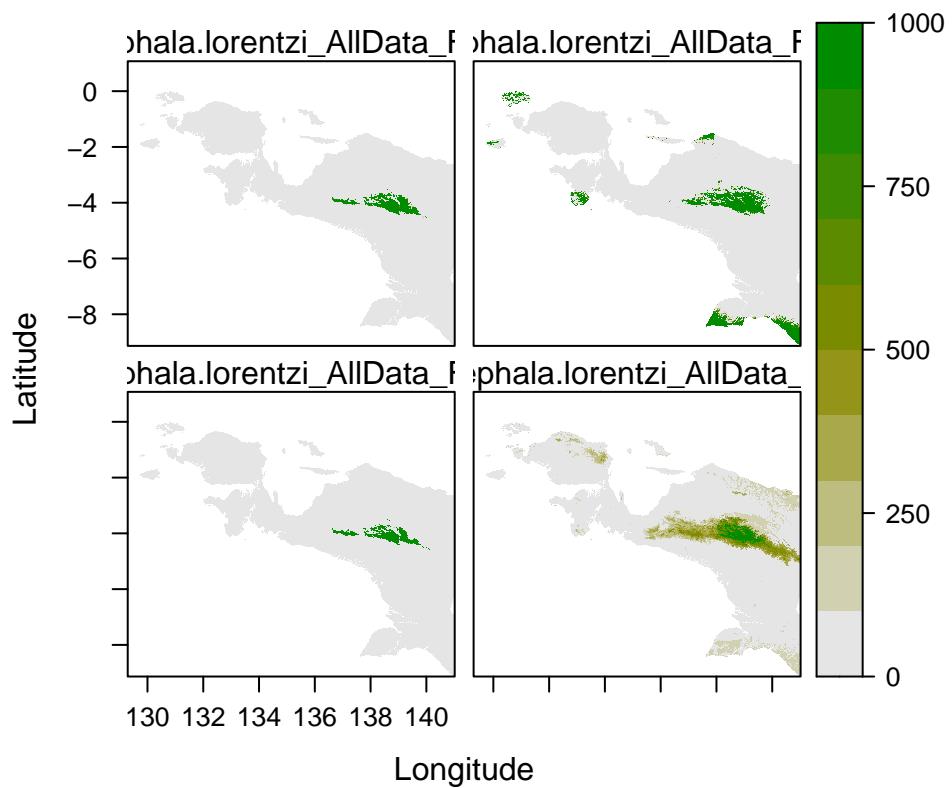
## Pachycephala.soror current projections

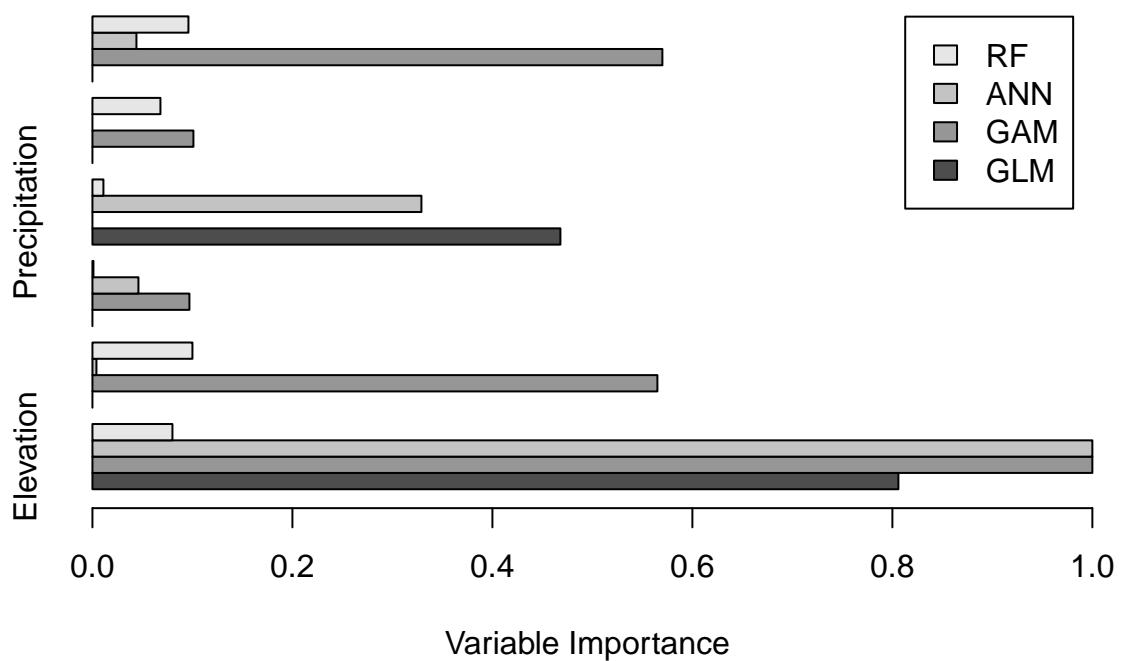


```
test_sdms = calculate_sdm(presence_absence_list, tif_predictors)
```

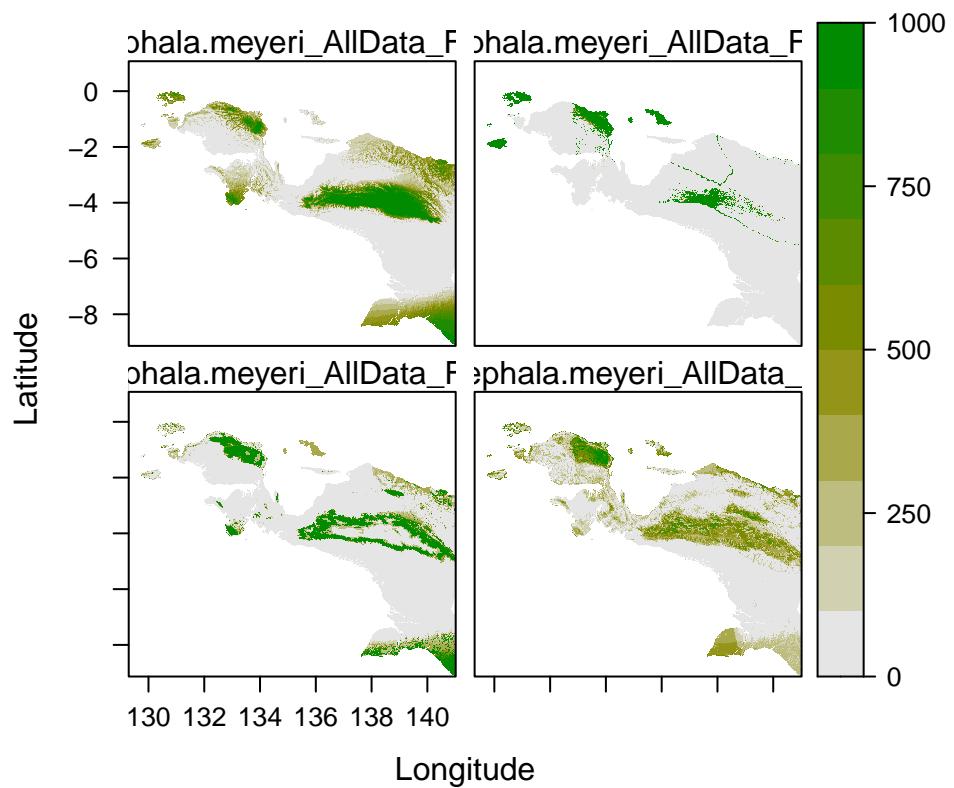


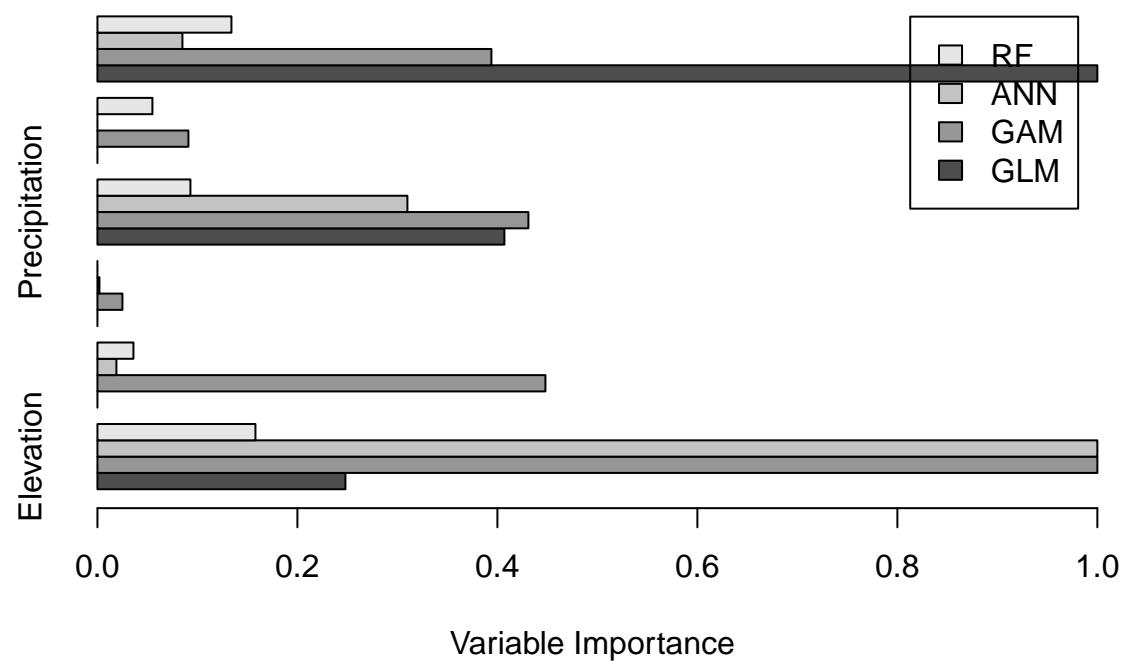
### **Pachycephala.lorentzi current projections**



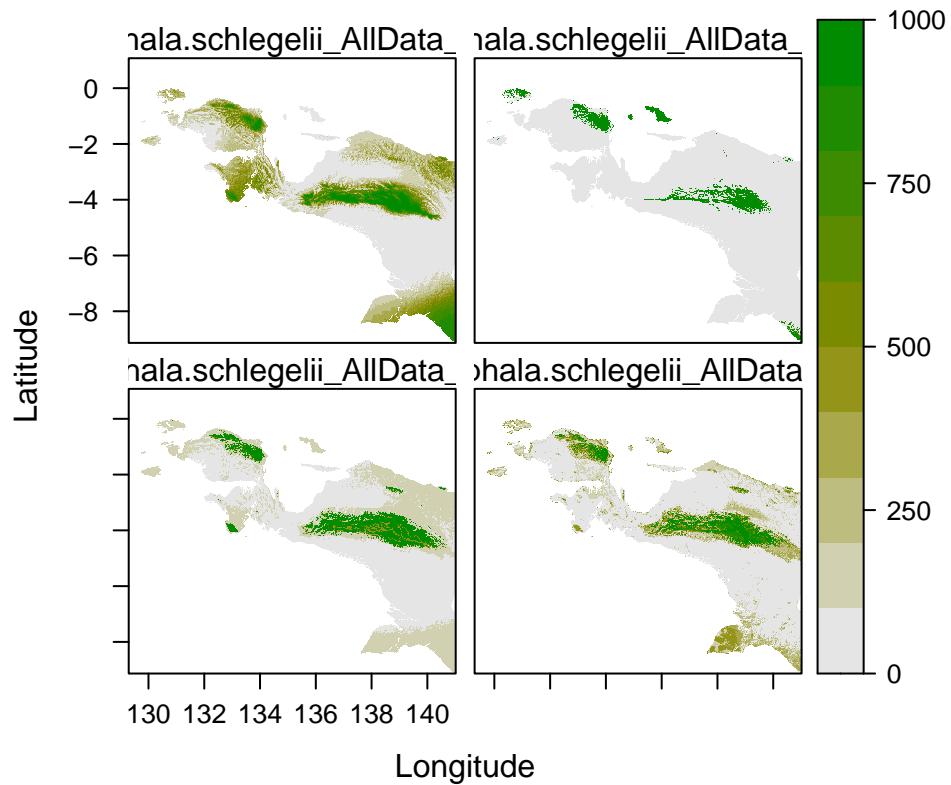


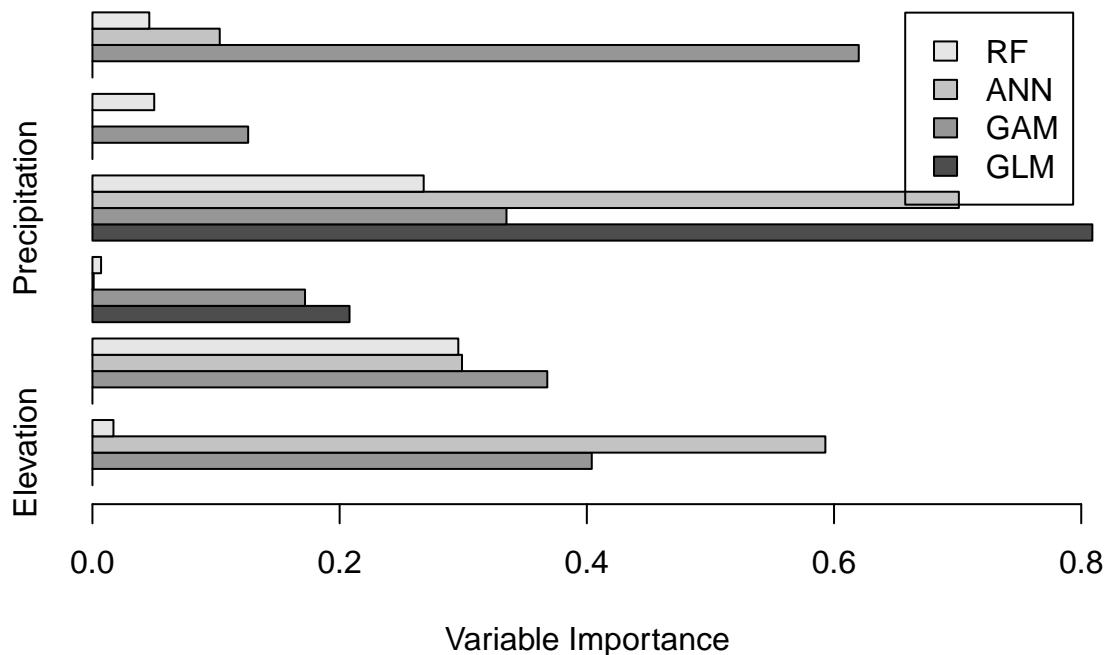
### **Pachycephala.meyeri current projections**





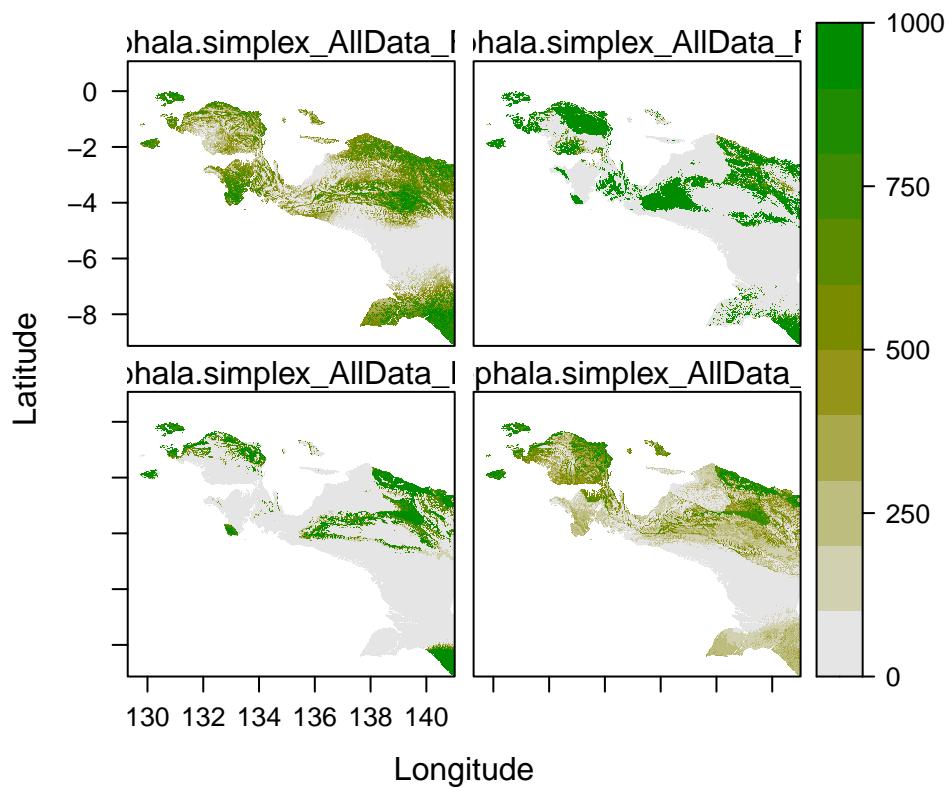
## Pachycephala.schlegelii current projections

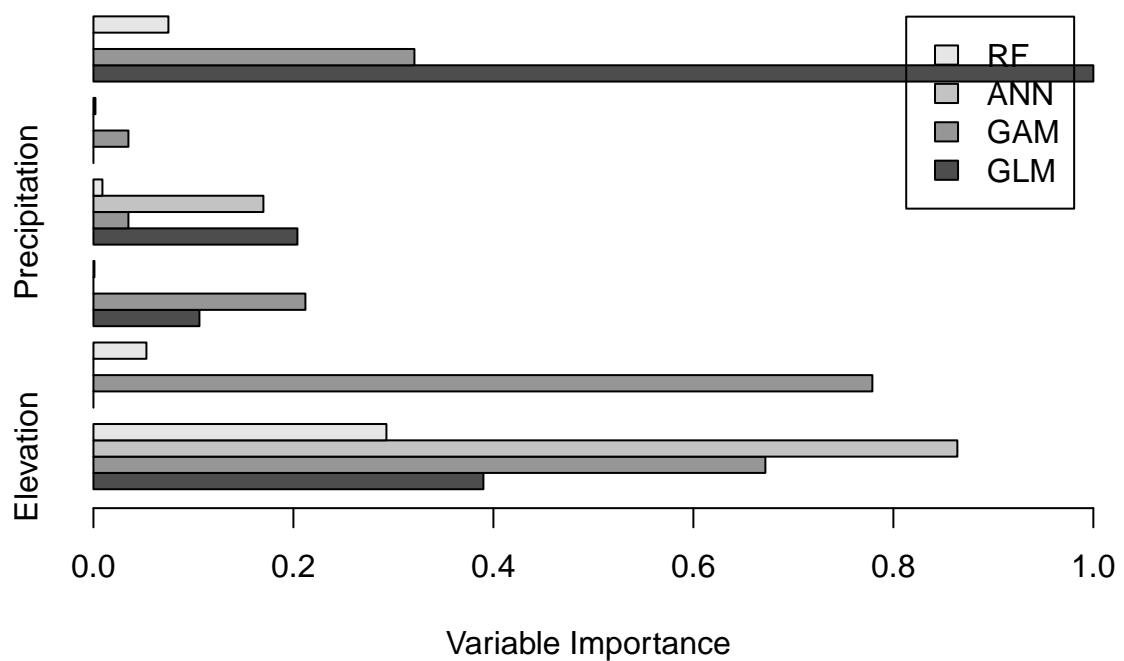


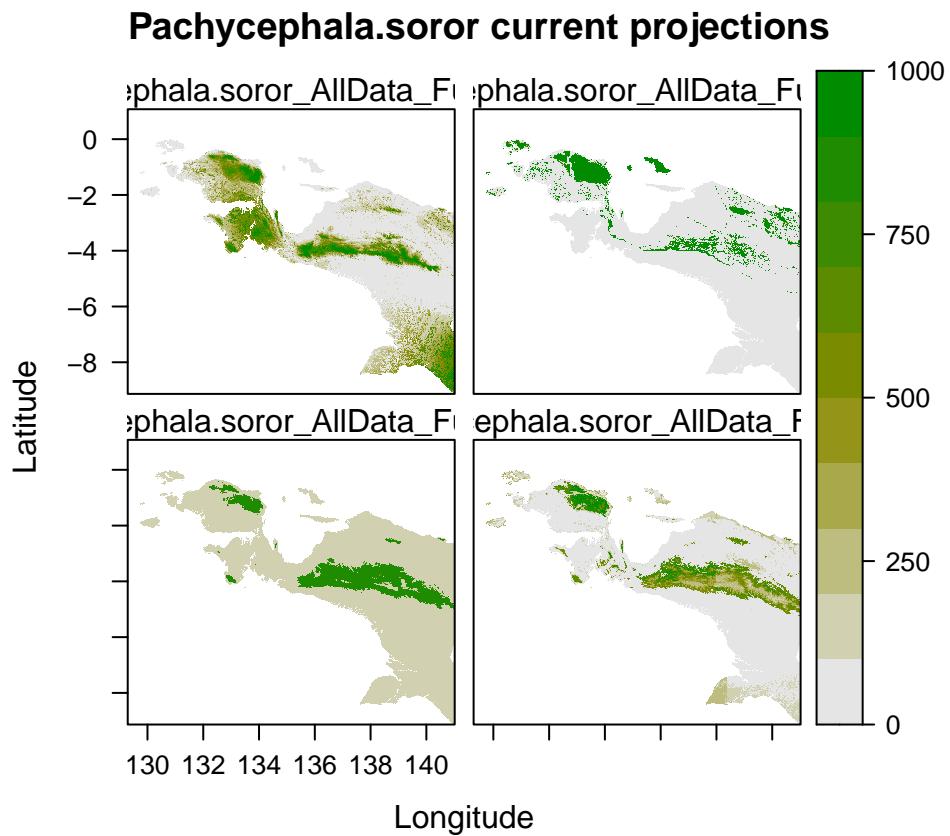


```
## Warning in newton(lsp = lsp, X = G$X, y = G$y, Eb = G$Eb, UrS = G$UrS, L =
## G$L, : Fitting terminated with step failure - check results carefully
```

### Pachycephala.simplex current projections





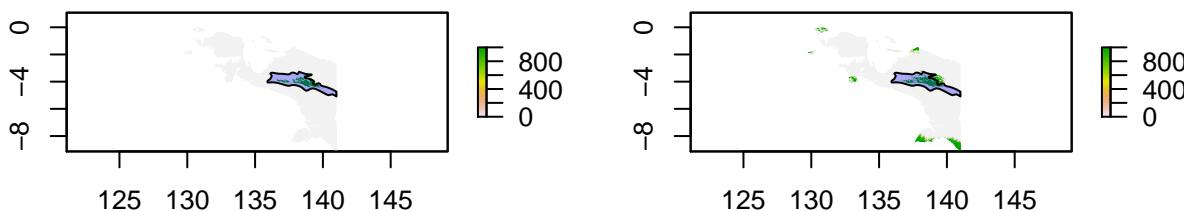


## 2.6. Comparison SDMs and IUCN

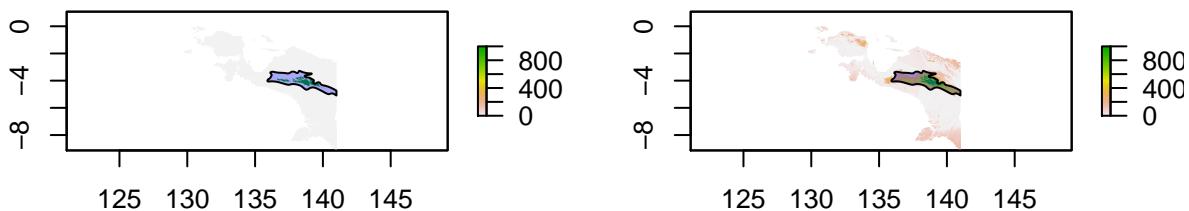
Plotting and comparing sdms with birdlife data.

```
source("plotting_sdms.R")
```

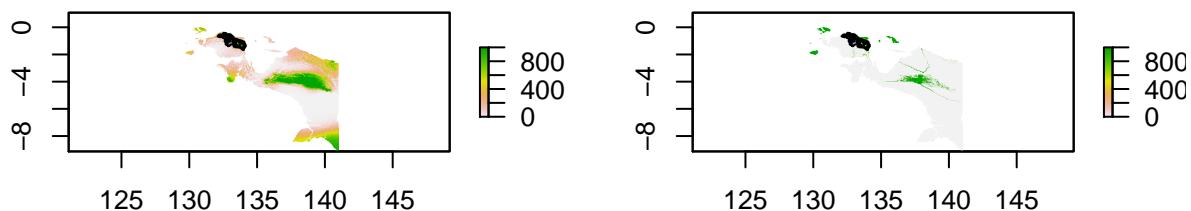
**Pachycephala.lorentzi\_AllData\_Full\_GLN** **Pachycephala.lorentzi\_AllData\_Full\_GAM**



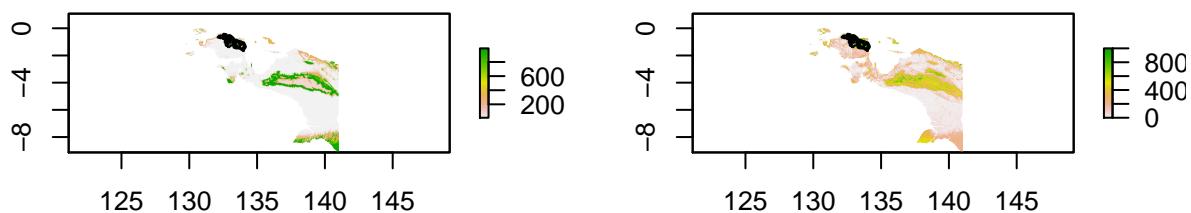
**Pachycephala.lorentzi\_AllData\_Full\_ANN** **Pachycephala.lorentzi\_AllData\_Full\_RF**



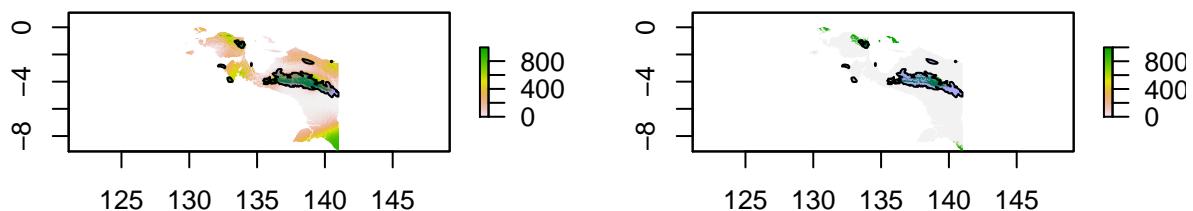
**Pachycephala.meyeri\_AllData\_Full\_GLM** **Pachycephala.meyeri\_AllData\_Full\_GAM**



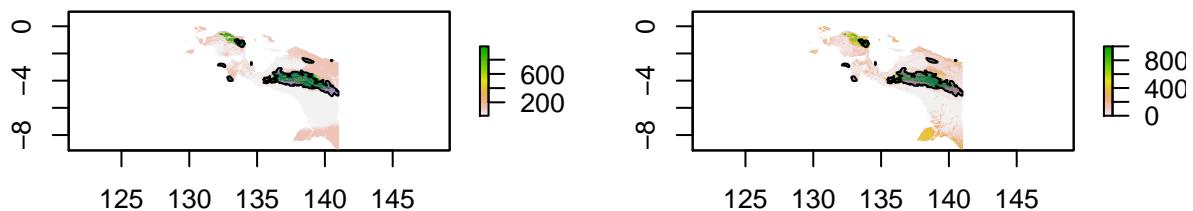
**Pachycephala.meyeri\_AllData\_Full\_ANN** **Pachycephala.meyeri\_AllData\_Full\_RF**



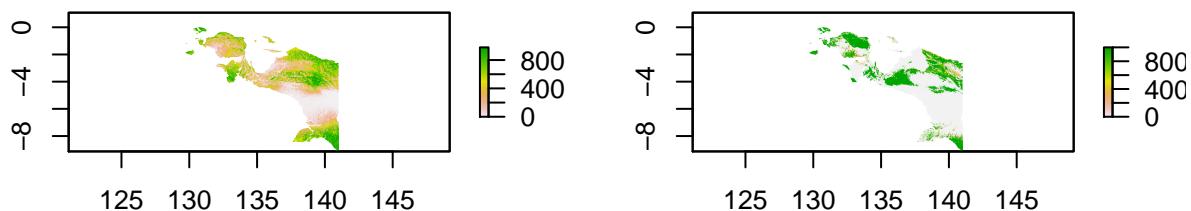
**Pachycephala.schlegelii\_AllData\_Full\_GL** **Pachycephala.schlegelii\_AllData\_Full\_GA**



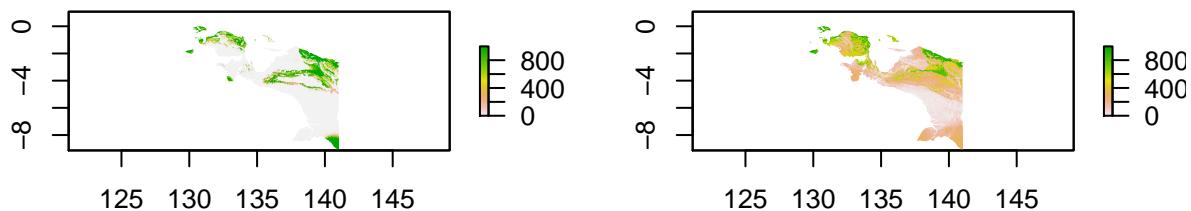
**Pachycephala.schlegelii\_AllData\_Full\_AN** **Pachycephala.schlegelii\_AllData\_Full\_RI**



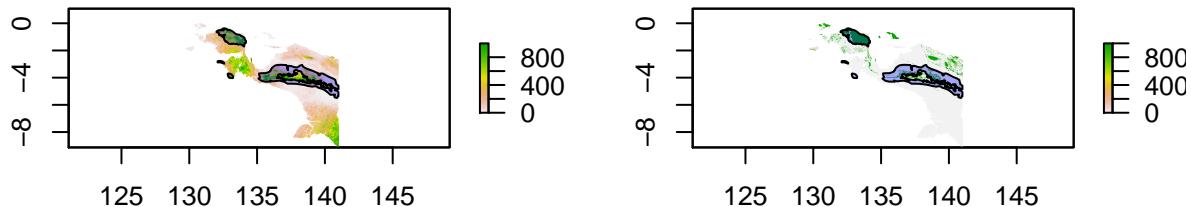
**Pachycephala.simplex\_AIIData\_Full\_GLM** **Pachycephala.simplex\_AIIData\_Full\_GAM**



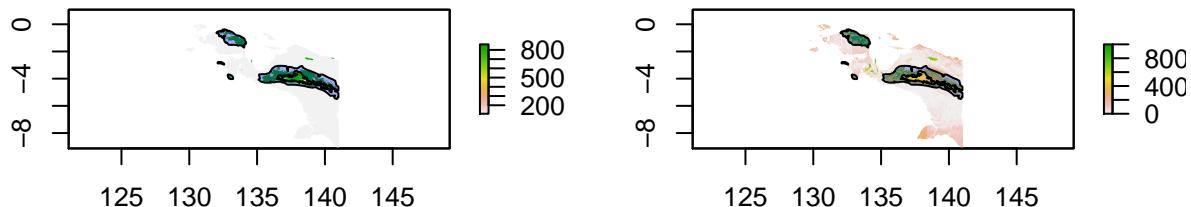
**Pachycephala.simplex\_AIIData\_Full\_ANI** **Pachycephala.simplex\_AIIData\_Full\_RF**



**Pachycephala.soror\_AllData\_Full\_GLM**   **Pachycephala.soror\_AllData\_Full\_GAM**



**Pachycephala.soror\_AllData\_Full\_ANN**   **Pachycephala.soror\_AllData\_Full\_RF**



```
#plot some nice plots here!
```

## 7.2 Code

MCMMB.Rmd

```
---
title: How do IUCN Range maps of birds compare to SDMs in Indonesian New Guinea? What
  are the main predictors for this distribution?
author: "Jonathan Gehret, Paul Leister"
date: "31.08.2021"
output:
  pdf_document:
    toc: yes
  html_notebook:
    toc: yes
    code_folding: show
---
````{r setup, include=FALSE, message = FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## 1. Introduction

## 2. Methods

To find the species to be used for our analysis we used the loops in bird\_plot\_loops.R to plot all species from the Gbif and IUCN data sets to find birds with sufficient observation points and different ranges to be used for our project.

### 2.1. Main script

The required libraries and the setup of the IUCN and gbif data sets are done in the MCMMB\_Main.R script. Notably, the HBIF data are created only for species with greater than 100 observations. The GBIF data were downloaded manually beforehand, however the r library “rgbif” could be used instead. Additionally, the island borders are read in and given the name “regio”.

```
{r, results='hide'} source("MCMMB_Main.R")
```

### 2.2. Species data

The script species\_loops.R includes two functions each for creating lists with the input bird species data for gbif (“get\_gbif\_birds(bird\_names,gbif\_crop,regio)”) and IUCN (“get\_iucn\_birds(bird\_names,birdlife,regio)”) and plotting them, respectively.

```
# loading of the described functions
source("species_loops.R")

# cassowaries didn't have enough data
#bird_names = c("Casuarius bennetti", "Casuarius casuarius", "Casuarius unappendiculatus")

# we chose Pachycephala instead
bird_names = c("Pachycephala lorentzi",
              "Pachycephala meyeri",
              "Pachycephala schlegelii",
              "Pachycephala simplex",
              "Pachycephala soror")

# these names can be changed to any other bird species occurring in the data sets, for fully automated c
# bird_names = c(your_favorite_species_here)

# for plotting gbif data next to iucn data
par(mfrow = c(2,length(bird_names)))

# create list with birds out of the iucn and gbif data
iucn_birds = get_iucn_birds(bird_names,birdlife,regio)
gbif_birds = get_gbif_birds(bird_names,gbif_crop,regio)
```

### 2.3. Create presence-absence data

While biomod2 also brings with it capabilities to create pseudo-absence data, we built a script to do it. First centroids for the area were created, based on that an equal number to the presence data of absence points is created.

"Be sure to take care when considering the use of pseudo-absences versus true absences for species disti

modeling. Similarly, it is extremely important to consider the influence of sampling bias in the data used to train models. Further reading: e.g., Guillera-Arroita et al. 2015, Kramer-Schadt et al. 2013, and Merow et al. 2013."

```
# load presence_absence.R script including the function
source("presence_absence.R")
presence_absence_list = create_pseudo_absence(regio,gbif_birds)
```

## 2.4. Indicators

The Indicator.R script is used to read in data for the various indicators (precipitation, elevation, landcover, primary\_forest, temperature, human population) to be used for the SDM. These are then cropped to the target region and saved in .tif format in the folder data/indicator\_stack/ for later use. As the process for some of the bigger files takes a lot of time, ready to use indicator .tif files are to be found in the aforementioned folder. These will then be loaded and stacked later on before the sdm creation.

```
{r, eval = FALSE} # where all the predictors are loaded in source("Indicator.R")

# stacking all predictors found in folder indicator stack in .tif format
# predictors: elevation, precipitation, temperature, primary forest, landcover, population
tif_predictors = stack(list.files(path = "data/indicator_stack/",
                                    full.names = TRUE,
                                    pattern = ".tif"))
plot(tif_predictors)
```

## 2.5. Species distribution models

Using biomod2, the indicator stack, and the presence absence data with the gbif birds of interest the species distribution models with the methods GLM, GAM, RF and ANN are produced. There are also alternative methods available, which can be shown using "BIOMOD\_ModelingOptions()". Each of the individual SDM versions can also be modified further, as can be seen in our script we modified k for the GAM (line 25). The data set was not split in two for evaluation, however this step could be done next time. The calculated models are saved to file (argument SaveObj = TRUE in BIOMOD\_Modeling()).

```
# loading biomod script and passing ensemble models to test_sdms
source("sdm_biomod2.R")
test_sdms = calculate_sdm(presence_absence_list, tif_predictors)
```

## 2.6. Comparison SDMs and IUCN

Plotting and comparing sdms with birdlife data.

```
source("plotting_sdms.R")
#plot some nice plots here!
```

## 7.2 Code

MCMMB.Rmd

```
{r, eval = FALSE, code = xfun::read_utf8("MCMMB.Rmd"), class.source = "fold-hide"}
```

```

MCMMB_Main.R
{r, eval = FALSE, code = xfun::read_utf8("MCMMB_Main.R"), class.source = "fold-hide"}
species_loops.R
{r, eval = FALSE, code = xfun::read_utf8("species_loops.R"), class.source = "fold-hide"}
presence_absence.R
{r, eval = FALSE, code = xfun::read_utf8("presence_absence.R"), class.source = "fold-hide"}
Indicator.R
{r, eval = FALSE, code = xfun::read_utf8("Indicator.R"), class.source = "fold-hide"}
sdm_biomod2.R
{r, eval = FALSE, code = xfun::read_utf8("sdm_biomod2.R"), class.source = "fold-hide"}
plotting_sdms.R
{r, eval = FALSE, code = xfun::read_utf8("plotting_sdms.R"), class.source = "fold-hide"}
bird_plot_loops.R
{r, eval = FALSE, code = xfun::read_utf8("bird_plot_loops.R"), class.source = "fold-hide"}

MCMMB_Main.R

```{.r .fold-hide}
#package libraries

library(biomod2) # for calculation of sdms
library(rgdal)
library(raster)
library(rgeos)
library(sf)
library(spatialEco) # for intersecting with point.in.poly
library(dplyr) # for removing species with n < threshold

# loading of IUCN birdlife data
birdlife <- readOGR("data/Papua_Birdlife_project/Birdlife_Papua.shp", integer64="allow.loss")

# iucn birdlife species names dataframe:
bl_species <- birdlife$SCINAME
bl_species <- data.frame(bl_species)

# renaming Amaurornis moluccanus/Amaurornis moluccana (#33)
# and Threskiornis molucca/Threskiornis moluccus (#527) as their names on gbif are different than on iucn
bl_species[33,] = "Amaurornis moluccanus"
bl_species[527,] = "Threskiornis molucca"
bl_species = data.frame(bl_species)

# reading in island borders
regio <- readOGR("data/Papua_Birdlife_project/Papua_region.shp", integer64="allow.loss")

# loading of all birds Isndonesia gbif
# remove all data without sufficient geometry

```

```

gbif_birds = read.csv("data/gbif/aves_indonesia/0303155-200613084148143.csv", header = TRUE, sep = "\t")
gbif_birds = select(gbif_birds, gbifID, species, decimalLatitude, decimalLongitude)
gbif_birds_corrected = gbif_birds[!(is.na(gbif_birds$decimalLatitude) | gbif_birds$decimalLatitude=="")]

# create points
gbif_points = st_as_sf(gbif_birds_corrected, coords = c("decimalLongitude", "decimalLatitude"), crs = 4326)

#crop to indonesian papua
gbif_crop_all = st_crop(gbif_points, regio)

#remove birds with occurance < x
# using 100 as per Van-Proosdij et al. (2016)
gbif_crop = gbif_crop_all %>% group_by(species) %>% filter(n() >= 100) %>% ungroup()

# species list of all bird species with occurrence > 100
species = sort(unique(gbif_crop$species))

# for plotting:
#plot(regio)
#plot(st_geometry(gbif_crop[gbif_crop$species == ""])), add = TRUE)

species_loops.R

# Loops to access iucn birdlife and gbif data by bird_names

# species vector with all scientific species names, example species
#bird_names = c(cas_ben_species, cas_cas_species, cas_una_species)
#bird_names = c("Casuarius bennetti", "Casuarius casuarius", "Casuarius unappendiculatus")
#bird_names = c("Pachycephala lorentzi",
#              "Pachycephala meyeri",
#              "Pachycephala schlegelii",
#              "Pachycephala simplex",
#              "Pachycephala soror")

#1. IUCN
# getting and plotting iucn birdlife birds in loop using scientific names vector:
# Function to create list of with IUCN Birdlife data for spedcified bird_names
get_iucn_birds = function(bird_names, birdlife, regio) {
  iucn_birds = list()
  for (i in bird_names) {
    iucn_bird = birdlife[birdlife$SCINAME == i,]
    iucn_birds[[i]] = iucn_bird
    #plotting optional
    plot(regio, main = i, cex.main = 0.8)
    plot(iucn_bird, add = T, col = "red")
  }
  return(iucn_birds)
}

#2. GBIF
# loop for getting gibf data for any scientific bird names ("Genus species") in vector
get_gbif_birds <- function(bird_names, gbif_crop, regio) {
  gbif_birds = list() # creaty empty list
  for (i in bird_names) {

```

```

gbif_bird = gbif_crop[gbif_crop$species == i,]
gbif_birds[[i]] = gbif_bird
#plotting optional
plot(regio, main = i, cex.main = 0.8)
  plot(st_geometry(gbif_bird), pch=16, col="green", add = TRUE)
}
return(gbif_birds)
}

# legend("bottomleft", legend = c("Casuarius unappendiculatus", "Casuarius bennetti", "Casuarius casuarius",
#       fill = c("black", "red", "yellow"))

```

presence\_absence.R

```

#function to add absence data to bird_list with spatialpointdataframe of bird presence data,
# using elevation raster for grid coordinates and regio raster for extents
create_pseudo_absence = function(regio, bird_list) {

  #1.: get centroids for all grid cells from established raster (here: elevation)
  elevation = raster("data/indicator_stack/Elevation.tif") # from extraction.R

  elevation_mask = mask(elevation,regio) # masking elevation over regio

  # create centroids for grid with column "present" filled with 0 and transformed to sp
  centroids_all = xyFromCell(elevation_mask, which(elevation_mask[]>=0))
  centroids_all = cbind(centroids_all,present = rep(0,nrow(centroids_all)))
  centroids_all = SpatialPointsDataFrame(coords = data.frame(centroids_all), data = data.frame(centroids_all))
  centroids_all@proj4string = CRS("+proj=longlat +ellps=WGS84 +no_defs")

  # empty output list to be filled aith all pres/abs data
  presence_absence_list = list()

  set.seed(100)
  for (bird in bird_list) {

    #2.: get centroids of all grid cells for specific species by use of mask
    # 2nd raster
    # cas_una_gbif is st -> sp with as_Spatial
    species_raster = rasterize(as_Spatial(bird),elevation,1)
    raster_mask = mask(elevation,species_raster)

    # create centroids for occurrence data
    centroids_occ = xyFromCell(raster_mask, which(raster_mask[] > 0))

    # add columns with presence data 1
    centroids_occ = cbind(centroids_occ,present = c(rep(1,length(centroids_occ[,1]))))

    # transform to SP:
    centroids_occ = SpatialPointsDataFrame(coords = data.frame(centroids_occ), data = data.frame(centroids_occ))
    centroids_occ@proj4string = CRS("+proj=longlat +ellps=WGS84 +no_defs")

    # create absence data added to presence data
    presence_absence <- rbind(centroids_all[sample(nrow(centroids_all),nrow(centroids_occ)), replace = T],

```

```

            centroids_occ)

# add to list
bird_name = unique(bird$species)
presence_absence_list[[bird_name]] = presence_absence

# plotting the presence absence data
plot(regio, main = bird_name)
plot(presence_absence, add = T, col = ifelse(presence_absence$present == 1, "red", "green"))
}

return(presence_absence_list)
}

```

Indicator.R

```

#package libraries

#library(rgdal)
#library(raster)
#library(rgeos)

#Read in Data

#regio <- readOGR("data/Papua_Birdlife_project/Papua_region.shp", integer64="allow.loss")
#plot(regio)

#elevation map

elev <- raster("data/Indicator/elevation/mn30_grd")
plot(elev)

e <- mask(elev,regio)
Elevation <- crop(x = e, y = extent(regio))
plot(Elevation)

#writeRaster(Elevation, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods"))
#writeRaster(Elevation, filename=file.path("data/indicator_stack/Elevation"), format="GTiff", overwrite=TRUE)

#Temperature

temp <- raster("data/Indicator/CHELSA_bio10_01.tif")

regio_temp <- mask(temp,regio)
r_temp <- crop(x = regio_temp, y = extent(regio))
plot(r_temp)

#writeRaster(r_temp, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods"))

```

```

#writeRaster(r_temp, filename=file.path("data/Indicator/Temperatur/r_temp"), format="GTiff", overwrite=TRUE)

Temperature = raster("data/Indicator/Temperatur/r_temp.tif") / 10 # temperature*10
Temperature = r_temp / 10 # temperature*10
plot(Temperature)
#writeRaster(Temperature, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods/"))
#writeRaster(Temperature, filename=file.path("data/indicator_stack/Temperature"), format="GTiff", overwrite=TRUE)

#Precipitation

precip <- raster("data/Indicator/CHELSA_bio10_12.tif") # precipitation

Precip_mask <- mask(precip, regio)
Precipitation <- crop(x = Precip_mask, y = extent(regio))
plot(Precipitation)

#writeRaster(Precipitation, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods/"))
#writeRaster(Precipitation, filename=file.path("data/indicator_stack/Precipitation"), format="GTiff", overwrite=TRUE)

#Population-density

regio_popul <- raster("data/Indicator/population/idn_msk_pop.grd")

pop_mask <- mask(regio_popul, regio)

Human_Population <- crop(x = pop_mask, y = extent(regio))
plot(Human_Population)

#writeRaster(Human_Population, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods/"))
#writeRaster(Human_Population, filename=file.path("data/indicator_stack/Human_Population"), format="GTiff", overwrite=TRUE)

#Primary-forest

primforest <- raster("data/Indicator/Primary_Forest/Asia_2001_primary.tif")
prim_mask <- mask(primforest, regio)

Primary_forest <- crop(x = prim_mask, y = extent(regio))
plot(Primary_forest)

#writeRaster(Primary_forest, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods/"))
#writeRaster(Primary_forest, filename=file.path("data/indicator_stack/Primary_forest"), format="GTiff", overwrite=TRUE)

#Landcover

landcover <- raster("data/Indicator/landcover/IDN_msk_cov.grd")
plot(landcover)

land_mask <- mask(landcover, regio)
Landcover <- crop(x = land_mask, y = extent(regio))
plot(Landcover)

```

```

#writeRaster(Landcover, filename=file.path("C:/Users/Paul/Desktop/3_Semester/modern concepts and methods/SDM/stacked_rasters/Landcover"), format="GTiff", overwrite=TRUE)

# plotting
plot(regio)

plot(Elevation)
plot(regio, add=T)

plot(Precipitation)
plot(regio, add=T)

plot(Temperature)
plot(regio, add=T)

plot(Human_Population)
plot(regio, add=T)

plot(Primary_forest)
plot(regio, add=T)

```

sdm\_biomod2.R

```

# following https://griffithdan.github.io/pages/outreach/SDM-Workshop-OSU-FALL2017.pdf

# shows biomod modelling possibilities:
#BIOMOD_ModelingOptions()

# create pres/abs data with script presence_absence.R (move to main script?)
source("presence_absence.R")
presence_absence_list = create_pseudo_absence(regio,gbif_birds)

# function to calculate sdm with biomod2
calculate_sdm = function(presence_absence_list, tif_predictors) {

  # for testing purposes
  #species = presence_absence_list[1]

  # list to be filled with sdm for every species put int
  sdm_list = list()

  # modify model settings (i.e. k value (degrees of freedom) for GAM)
  # big but not too big for k
  myBiomodOptions = BIOMOD_ModelingOptions(
    GAM = list(algo = 'GAM_mgcv',
               type = 's_smoothen',
               #k = 0,
               k = 4,
               interaction.level = 0,
               myFormula = NULL,
               family = binomial(link = 'logit'),

```

```

    method = 'GCV.Cp',
    optimizer = c('outer','newton'),
    select = FALSE,
    knots = NULL,
    paraPen = NULL,
    control = list(nthreads = 1, irls.reg = 0, epsilon = 1e-07, maxit = 200, trace = FALSE,
                   , mgcv.half = 15, rank.tol = 1.49011611938477e-08
                   , nlm = list(ndigit=7, gradtol=1e-06, stepmax=2, steptol=1e-04, iterlim=1000),
                   , optim = list(factr=1e+07), newton = list(conv.tol=1e-06, maxNstep=5, maxIter=1000),
                   , outerPIsteps = 0, idLinksBases = TRUE, scalePenalty = TRUE, efs.lspmax = 1000),
                   , scale.est = "fletcher", edge.correct = FALSE) ),
  # standard settings
  RF = list( do.classif = TRUE,
             ntree = 500,
             mtry = 'default',
             nodesize = 5,
             #nodesize = 50,
             maxnodes = NULL)
  )

for (i in 1:length(presence_absence_list)) {

  # get bird_names
  help_species = presence_absence_list[i]
  bird_name = names(help_species)

  # species
  species = help_species[[1]]

  # get coordinates from prese/abs data
  species_xy = data.frame(cbind(species$x, species$y))

  # potentially: split dataset into two for evaluation data
  # putting the data into right format
  format_bm <- BIOMOD_FormattingData(resp.var = species$present,
                                         expl.var = stack(tif_predictors),
                                         resp.xy = species_xy,
                                         resp.name = bird_name)

  # modeling the sdms
  biomodels_1 <- BIOMOD_Modeling(data = format_bm,
                                   models = c('GLM','GAM','ANN','RF'),
                                   SaveObj = TRUE,
                                   models.options = myBiomodOptions,
                                   # DataSplit = 80, # common practice to validate!
                                   VarImport = 1)

  "We will focus TSS (see Allouche et al. 2006 for a comparison of all three). TSS is the sum of the absolute values of the differences between the observed and predicted presence/absence values. It ranges from -1 to 1, where 1 means we correctly classified presences and absences, minus 1. Higher is better (in the range -1 to 1), as it is a balance between model maximizing sensitivity and specificity."
}

# add to markdown?? use other indicators??
#biomod_eval = get_evaluations(biomodels_1)

```

```

#biomod_eval["TSS", "Testing.data", , , ]
#biomod_eval["KAPPA", "Testing.data", , , ]
#biomod_eval["ROC", "Testing.data", , , ]
#evaluate(biomodels_1, data, stat, as.array=FALSE)

" GLM    GAM    ANN     RF
0.725    NA 0.757 0.938 "
#>   biomod_eval["TSS", "Testing.data", , , ]
#GLM    GAM    ANN     RF
#1.000 1.000 1.000 0.967
#>   biomod_eval["KAPPA", "Testing.data", , , ]
#GLM    GAM    ANN     RF
#1.000 1.000 1.000 0.967
#>   biomod_eval["ROC", "Testing.data", , , ]
#GLM    GAM    ANN     RF
#1.000 1.000 1.000 0.999

"We can also calculate variable importances to compare influences of individual predictor variables among models. For a publication, you might also create partial dependence plots. Do the different models agree on the importance of the variables?""

biomod_var_importance = drop(get_variables_importance(biomodels_1))
barplot(height = t(biomod_var_importance),
        beside = T,
        horiz = T,
        #ylab = c("Precipitation", ...),
        xlab = "Variable Importance",
        legend = c("GLM", "GAM", "ANN", "RF"))

"
One approach for using the information in these various models is to combine them into an ensemble, collection of models merged together (Thuiller et al. 2009). We can take all models above a given "threshold and combine them.""

biomod_ensemble = BIOMOD_EensemleModeling(modeling.output = biomodels_1,
                                             chosen.models = 'all',
                                             em.by = 'all',
                                             eval.metric = c('TSS'),
                                             eval.metric.quality.threshold = c(0.3),
                                             prob.mean = TRUE,
                                             prob.mean.weight = FALSE,
                                             prob.cv = FALSE,
                                             prob.ci = FALSE,
                                             prob.median = FALSE)

#get_evaluations(biomod_ensemble)
"$Test.Spec_EMmeanByTSS_mergedAlgo_mergedRun_mergedData
  Testing.data Cutoff Sensitivity Specificity
KAPPA        0.846    564      93.75     90.909
TSS          0.847    564      93.75     90.909
ROC          0.979    562      93.75     90.909
"

```

"In order to visualize the model outputs, we should now use the SDMs to predict species occurrence at space using our modern environmental data. The models predict probabilities of occurrence, and it is important to remember the interpretation of these outputs depend on the specific model, our data, and assumptions (e.g., see discussion of Maxent outputs in Merow et al 2013)."

```

biomod_proj = BIOMOD_Projection(modeling.output = biomodels_1,
                                  new.env = stack(tif_predictors), # modern environment
                                  proj.name = 'current' ,
                                  selected.models = 'all' ,
                                  binary.meth = 'TSS' ,
                                  compress = 'xz' ,
                                  clamping.mask = F,
                                  output.format = '.grd' )

# to-do: make plots beautiful
#plot(biomod_proj,xlab="Longitude")
plot(biomod_proj)

# write sd models to list for return
sdm_list[[bird_name]] = biomod_proj
"Fehler in sdm_list[[bird_name]] <- biomod_proj :
attempt to select less than one element in OneIndex"
}
return(sdm_list)
}

test_sdms = calculate_sdm(presence_absence_list, tif_predictors)
#sdm_list[[bird_name]] = biomod

```

plotting\_sdms.R

```

# Evaluation
# compare gbif with iucn for all birds j and sdms i
par(mfrow=c(2,2))
#pdf("plots.pdf")
for (j in 1:length(test_sdms)) {
  bird = test_sdms[[j]]
  sdms = bird@proj@val@layers
  #bird_names = names(test_sdms[j])
  for (i in 1:length(sdms)) {
    sdm = sdms[[i]]
    plot(sdm, main = names(sdm@data))
    plot(iucn_birds[[j]], add = T, col =  rgb(red = 0, green = 0, blue = 1, alpha = 0.3))
  }
  #plot_names = paste("images/sdm_iucn/",names(i),".png")

  #savePlot(filename = plot_names, type = "png")
  #tiff("images/sdm_iucn/.tiff")
}

```

bird\_plot\_loops.R

```

# These two loops were used to plot all species distributions from the two data sets (GBIF and IUCN)
#to find species that had sufficient data for distribution modeling and
#to find related species (i.e. of the same genus) with different ranges.

#3. plot all iucn species (for finding the desired one)

# plotting a map for every single bird species in indonesia!
# adjusting for cropped 100 species!

par(mfrow=c(4,4))
for (i in 1:length(bl_species$species)) {
  plot(regio, main = paste(birdlife[i,]$SCINAME, i))
  plot(birdlife[i,], add=T, col = "green")
}

#4. plot all gbif species
par(mfrow=c(3,3))

#adjusting species list for gbif_crop
bird_names_2 = sort(unique(gbif_crop$species))
gbif_birds = list() # creaty emtpy list
for (i in bird_names) {
  bird = gbif_crop[gbif_crop$species == i,]
  gbif_birds[[i]] = bird
  plot(regio, main = paste(i,nrow(bird)))
  plot(st_geometry(bird), pch=16, col="green", add = TRUE)
}

# accessing all plots of this r session:
# accessing save path
plots.dir.path <- list.files(tempdir(), pattern="rs-graphics", full.names = TRUE);
plots.png.paths <- list.files(plots.dir.path, pattern=".png", full.names = TRUE)

# copy these files to specified directory:

file.copy(from=plots.png.paths, to="images/gbif_maps_2/")

```