Mapping species richness and endemism

Ian Ondo

2023-05-28

Contents

Package notes:
We need to install the following set of R packages to successfully run the codes in this vignette:

- **data.table** (Dowle and Srinivasan 2021)
- **doParallel** (Corporation and Weston 2020)
- **foreach** (Microsoft and Weston 2020)
- **rWCVP** (Brown et al. 2023)
- **ggplot2** (Wickham 2016)
- **ggpubr** (Kassambara 2023)

Introduction

In this vignette, we will demonstrate how to build a species richness and an endemism map based on species probability of occurrence (Crisp et al. 2001; Calabrese et al. 2014).
The grid-level species richness S_j prediction at the j^{th} grid cell location follows a Poisson binomial distribution (Calabrese et al. 2014), meaning that the expected value (mean) and variance of stacking the probability of occurrence of K species is given by:

$$E(S_j) = \sum_{k=1}^{K} p_{j,k} \tag{1}$$

$$Var(S_j) = \sum_{k=1}^{K} (1 - p_{j,k}) p_{j,k} \tag{2}$$

Where, $p_{j,k}$ is the probability of occurrence of species k in grid cell j.

The grid-level endemism (hereafter called *weighted endemism*) WE_j prediction at j is computed as the species richness S_j weighted by the inverse of the range size of species (Crisp et al. 2001):

$$WE_j = \sum_{k=1}^{K} (p_{j,k} \times \frac{1}{RangeSize_k}) \tag{3}$$

$$RangeSize_k = \sum_{j=1}^{J_k} p_{j,k} \tag{4}$$

Where, J_k is the number of grid cells species k is predicted to occur in.

Step 1: Obtaining species probability of occurrence maps

If you do not have probability maps of species occurrences yet, please have a look at the vignette **Modelling species distribution** to obtain such maps using **UsefulPlants**.

Step 2: Aggregating probability maps

We must aggregate a collection of raster maps usually derived with different geographic extents because of the differences in the spatial coverage of species distribution. One solution is use a reference raster $rast_{ref}$ encompassing the total geographic extent of all species distribution maps, e.g. a worldwide raster map, extract the grid cells in $rast_{ref}$ that overlap with each species probability map and retrieve the corresponding species occurrence probabilities.

Let's assume that following **Step 1**, we stored our species probability maps predictions in a folder named "usefulplants_rast_maps", then:

```
# get the list of raster maps files
rast_list_files =list.files("~/usefulplants_rast_maps",
                              # replace .tif by the correct file extension
                              pattern="\\.tif$",
                              full.names=T)
## make sure that raster's names correspond to species names !!

# create a function that will extract cell numbers from the reference raster
# and the individual maps predicted probabilities
get_grid_cell_prob <- function(k,ref){
  # read the probability maps in memory
  r <- raster::raster(k)
  # extract cells with values
  # (we could also select cells with values>0 to speed up the computations)
  vals <- raster::values(r)
  cells <- which(!is.na(vals))
  # get the position of the species in the list of species maps files
  # using the raster's name and a out-of-scope vector of species names
  # defined later
  species_idx = match(gsub("_"," ", gsub("\\.","-",names(r))), sp.names)
  # get cells in reference raster ref
  cells_ref <- raster::cellFromXY(ref, raster::xyFromCell(r,cells))
  # returns a data.table
  data.table::data.table(species=species_idx,
                                cell=cells_ref,
                                proba=r[cells])
}
```

```r
# path to your reference raster. Make sure this raster has the same geographic
# projection and resolution as your probability maps. It can be, for example,
# one of the environmental raster layer used to predict your species probability
# of occurrence.
rast_ref_file <- "<path/to/reference/raster.tif>"

# split the vector of files paths into 100 chunks for parallel processing
raster_split_list <- chunk2(rast_list_files, 100)

# create a vector of species names
sp.names <- gsub("_"," ",strip_extension(basename(rast_list_files)))

# run function in parallel
# foreach will copy all R objects in Global Environment within child processes,
# so be careful to not store big objects in your environment !
doParallel::registerDoParallel(parallel::detectCores()-1)
started.at=proc.time()
species_proba_per_cell <- foreach::foreach(k=raster_split_list,
                                           .packages=c("data.table","raster"),
                                           .combine='rbind') %dopar% {
  gc()
  # apply the function "get_grid_cell_prob" to the set of raster maps k
  l <- lapply(k, get_grid_cell_prob, ref=raster::raster(rast_ref_file))
  # bind outputs and return
  rbindlist(l,fill=TRUE)
}
doParallel::stopImplicitCluster()
finished.at=proc.time()
time_taken <- finished.at - started.at # calculate amount of time elapsed
cat(
  sprintf("\n > End of computations ( Time elapsed : %g hr %g min %g sec. )\n\n",
          as.vector(time_taken['elapsed'])/3600, as.vector(time_taken['elapsed'])/60,
          as.vector(time_taken['elapsed'])))
unregister()

# make sure that probabilities lie between 0 and 1
species_proba_per_cell[,proba:=pmin(pmax(proba, 0, na.rm=T), 1, na.rm=T)]

# save the data.table in a rds file for later processing
# (this make take a while !)
saveRDS(species_proba_per_cell, file.path(getwd(),"species_proba_per_cell.rds"))
```

Important note:
The method that we used to retrieve cells that overlap between the reference and a probability raster only works if the two raster grids are aligned.
If they don't, this can be done by snapping the lower left corner and upper right corner of the raster we want to aligned to the lower left corner and upper right corner respectively of the nearest cell in the snap raster (i.e. reference raster).
The **snapRaster** function included in the *Spatial Analyst* extension of ArcGIS can be used to generate probability maps *snapped* to the reference map.

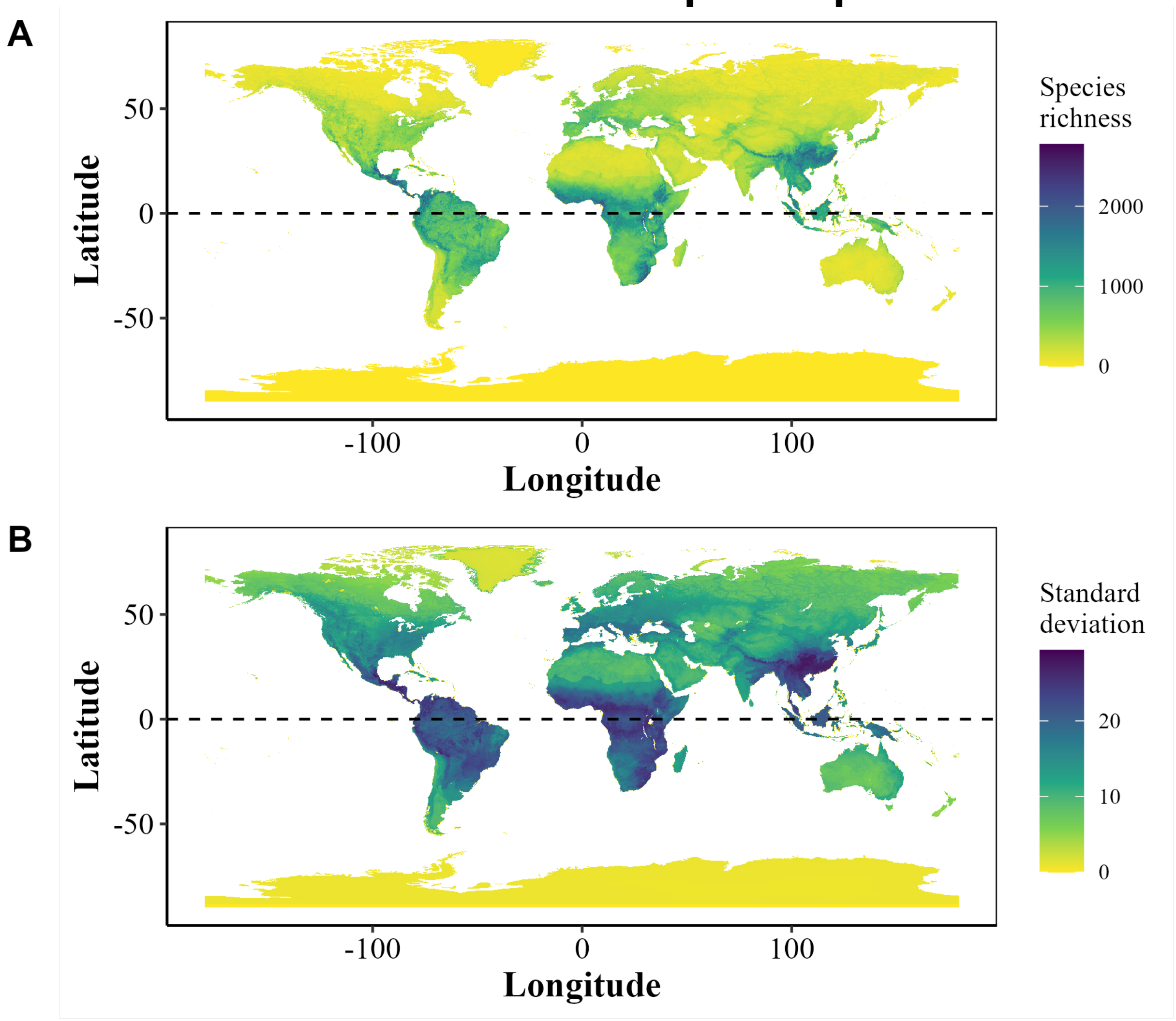
Step 3: Computing the species richness *SR* and its standard deviation *SD*

We just need to apply equations (1) and (2) on the `data.table` object `species_proba_per_cell`, but instead of computing the variance at each grid cell j, we are going to compute the standard deviation SD as follows: $SD(S_j) = \sqrt{\frac{Var(S_j)}{N}} = \sqrt{\frac{\sum_{k=1}^{K}(1-p_{j,k})p_{j,k}}{N}}$, where N is the number of predicted values in j.

```r
# load the species probabilities per grid cell data.table (if not done yet)
species_proba_per_cell <- readRDS(file.path(getwd(),"species_proba_per_cell.rds"))

# compute species richness SR and its standard deviation SD using equation 1 and 2
species_richness<-species_proba_per_cell[,list(SR=sum(proba,na.rm=T),
                                               SD=sqrt(sum((1-proba)*proba,na.rm=T)/sqrt(.N)),
                                               by=cell]

# get valid cell numbers (i.e. cells with values)
valid.cells <- !is.na(species_richness[['cell']])

# create an empty raster with 2 layers to accomodate the species richness and
# standard deviation estimates
rast_ref <- raster::raster(rast_ref_file)
predicted_species_richness <- raster::stack(rast_ref,rast_ref)

# make sure to mask out the original values
raster::values(predicted_species_richness) <- NA

# fill up raster values with corresponding estimates
predicted_species_richness[[1]][species_richness[['cell']][valid.cells]] <-
  species_richness[['SR']][valid.cells]
predicted_species_richness[[2]][species_richness[['cell']][valid.cells]] <-
  species_richness[['SD']][valid.cells]
```

Now, let's visualise the maps using the R packages **ggplot2** and **ggpubr**.

```r
# set up graphical parameters
my_font_family = "serif"
win_family_font = setNames(my_font_family,my_font_family)
invisible(windowsFonts(win_family_font))

# create a data.frame with x,y coordinates and predicted values
m <- raster::rasterToPoints(predicted_species_richness) %>%
  as.data.frame()

# create SR plot
srplt <- m %>%
  ggplot2::ggplot() +
  ggplot2::geom_raster(ggplot2::aes(x=x,y=y,fill=GlobalSRmv4)) +
  ggplot2::theme_classic() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  # ggplot2::labs(title="Global distribution of useful plant species richness",
  #               subtitle = "based on species occurrence probability") +
  ggplot2::theme(
    text=ggplot2::element_text(family=my_font_family),
    plot.title = ggplot2::element_text(face="bold", size=18, hjust = 0),
    axis.text=ggplot2::element_text(colour="black", size=12),
    axis.title=ggplot2::element_text(colour="black", size=14,face="bold"),
    panel.background = ggplot2::element_rect(fill=NA, color="black", size=12)
  )+
  ggplot2::geom_hline(yintercept=0,linetype="dashed") +
  ggplot2::scale_fill_viridis_c(name='Species\nrichness', direction=-1) +
  ggplot2::coord_equal()

# create SR std dev. plot
sdplt <- m %>%
  ggplot2::ggplot() +
  ggplot2::geom_raster(ggplot2::aes(x=x,y=y,fill=GlobalSRsdv4)) +
  ggplot2::theme_classic() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  ggplot2::theme(
    text=ggplot2::element_text(family=my_font_family),
    plot.title = ggplot2::element_text(face="bold", size=18, hjust = 0),
    axis.text=ggplot2::element_text(colour="black", size=12),
    axis.title=ggplot2::element_text(colour="black", size=14,face="bold"),
    panel.background = ggplot2::element_rect(fill=NA, color="black", size=12)
  )+
  ggplot2::geom_hline(yintercept=0,linetype="dashed") +
  ggplot2::scale_fill_viridis_c(name='Standard\ndeviation', direction=-1) +
  ggplot2::coord_equal()

# create co-plot
fig1 <- ggpubr::ggarrange(srplt,
                          sdplt,
                          ncol=1,
                          align="v",
                          labels = c("A","B"))
fig1_ann <- ggpubr::annotate_figure(fig1,
               top = ggpubr::text_grob("Global distribution of useful plant species richness",
                                     face = "bold", size = 18),
               bottom = ggpubr::text_grob("Based on species occurrence probability",
                                  hjust = 1, x = 1, face = "italic", size = 10)
)

ggplot2::ggsave("H:/Useful_Plants_2/SR_andSD.png", fig1_ann)
```


Global distribution of useful plant species richness

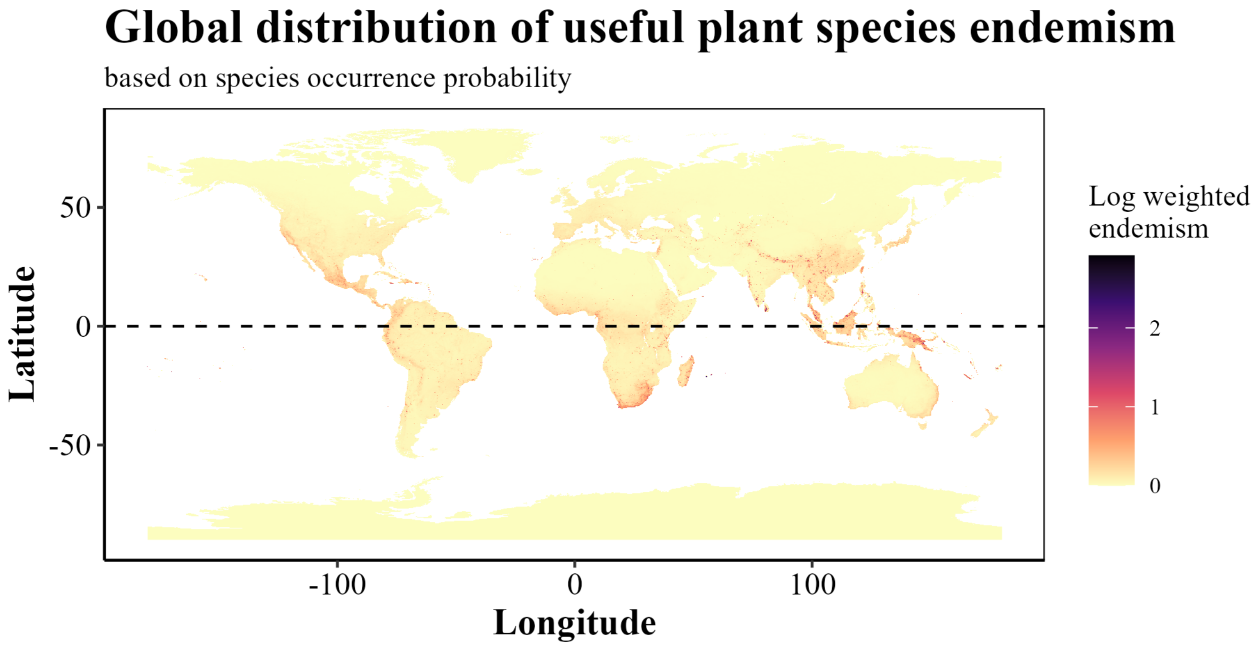
Based on species occurrence probability

Step 4: Computing the weighted endemism *WE*

We apply equations (3) and (4) on `species_proba_per_cell`.

```
# compute the range size weights
weights <- species_proba_per_cell[, list(weight=1/sum(proba,na.rm=T)), by=species]
data.table::setkey(species_proba_per_cell, key='species')

# left join the range size weights to the species probability per cell data
species_proba_per_cell[weights, weight:=weight]
gc() # optional, to force R to free some memory for computations
data.table::setkey(species_proba_per_cell, key='cell')
gc()

# compute the weighted endemism
weighted_endemism <- species_proba_per_cell[,list(WE=sum(proba*weight,na.rm=T)),
                                              by=cell]

# get valid cell numbers (i.e. cells with values)
valid.cells <- !is.na(weighted_endemism[['cell']])

# create an empty raster to accomodate the weighted endemism estimate
predicted_weighted_endemism <- rast_ref

# make sure to mask out the original values
raster::values(predicted_weighted_endemism) <- NA

# fill up raster values with corresponding estimates
predicted_weighted_endemism[predicted_weighted_endemism[['cell']][valid.cells]] <-
  predicted_weighted_endemism[['WE']][valid.cells]
```

```
m <- raster::rasterToPoints(predicted_weighted_endemism) %>%
  as.data.frame()

weplt <- m2 %>%
  ggplot2::ggplot() +
  ggplot2::geom_raster(ggplot2::aes(x=x,y=y,fill=log(GlobalWEmv3+1))) +
  ggplot2::theme_classic() +
  ggplot2::xlab("Longitude") + ggplot2::ylab("Latitude") +
  ggplot2::labs(title="Global distribution of useful plant species endemism",
                subtitle = "based on species occurrence probability") +
  ggplot2::theme(
    text=ggplot2::element_text(family=my_font_family),
    plot.title = ggplot2::element_text(face="bold", size=18, hjust = 0),
    axis.text=ggplot2::element_text(colour="black", size=12),
    axis.title=ggplot2::element_text(colour="black", size=14,face="bold"),
    panel.background = ggplot2::element_rect(fill=NA, color="black", size=12)
  )+
  ggplot2::geom_hline(yintercept=0,linetype="dashed") +
  ggplot2::scale_fill_viridis_c(name='Log weighted\nendemism',
                                option="magma",
                                direction=-1) +
  ggplot2::coord_equal()

weplt
ggplot2::ggsave("H:/Useful_Plants_2/WE.png", weplt)
```


Global distribution of useful plant species endemism
based on species occurrence probability
Latitude
50
0
-50
-100
0
100
Longitude
Log weighted endemism
2
1
0

References

Brown, Matilda J. M., Barnaby E. Walker, Nicholas Black, Rafaël Govaerts, Ian Ondo, Robert Turner, and Eimear Nic Lughadha. 2023. "rWCVP: A Companion r Package to the World Checklist of Vascular Plants." *New Phytologist*.

Calabrese, Justin M., Grégoire Certain, Casper Kraan, and Carsten F. Dormann. 2014. "Stacking Species Distribution Models and Adjusting Bias by Linking Them to Macroecological Models." *Global Ecology and Biogeography* 23 (1): 99–112. https://doi.org/https://doi.org/10.1111/geb.12102.

Corporation, Microsoft, and Steve Weston. 2020. *doParallel: Foreach Parallel Adaptor for the 'Parallel' Package*. https://CRAN.R-project.org/package=doParallel.

Crisp, MICHAEL D, Shawn Laffan, H Peter Linder, and ANNA Monro. 2001. "Endemism in the Australian Flora." *Journal of Biogeography* 28 (2): 183–98.

Dowle, Matt, and Arun Srinivasan. 2021. *Data.table: Extension of 'Data.frame'*. https://CRAN.R-project.org/package=data.table.

Kassambara, Alboukadel. 2023. *Ggpubr: 'Ggplot2' Based Publication Ready Plots*. https://CRAN.R-project.org/package=ggpubr.

Microsoft, and Steve Weston. 2020. *Foreach: Provides Foreach Looping Construct*. https://CRAN.R-project.org/package=foreach.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.