

Exploring the latitudinal gradient of utilised plants species richness and endemism

Ian Ondo

2023-07-24

Contents

Package notes:

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

- **ComplexHeatmap** (Gu 2022)
- **terra** (Hijmans 2023)
- **mgcv** (S. N. Wood 2017)
- **purrr** (Wickham and Henry 2023)
- **circlize** (Gu et al. 2014)

Introduction

In this vignette, we will investigate the variation in the latitudinal gradient profiles of utilised species richness and endemism across different plants uses (Pironon S. 2023) using Generalised Additive Models (GAMs) (S. N. Wood 2011).

We need species richness/endemism maps of per use category and for all uses confounded. If you do not have these maps yet, please have a look at the vignette **Mapping species distribution, richness and endemism**. We will demonstrate how to run the analyses with the species richness maps, but the process is similar for the endemism.

Step 1: Prepare the data for model fitting

Let's assume that we stored our species richness maps predictions (all uses confounded and per uses) as Tagged Image Format (TIF) files in a folder named "usefulplants_species_richness_maps", then:

```
# load all maps into R
all_maps_files <- list.files("path/to/usefulplants_species_richness_maps",
                             pattern="\\.tif$", full.names=TRUE)
all_maps_rast <- terra::rast(all_maps_files)

# get the total species richness only
raster_ref <- all_maps_rast[["GlobalSR"]] # change with the name of the "all uses" map

# get the species richness per plants uses only
raster_uses <- subset(all_maps_rast, subset="GlobalSR", negate=TRUE)

# vector of latitude values
ylat <- terra::yFromRow(raster_ref, row=1:nrow(raster_ref))
```

```

# number of valid values by row
nna <- apply(as.matrix(raster_ref[[1]]), wide=TRUE, 1, function(x) sum(!is.na(x)))
lat <- base::rep(ylat, times=nna)

# get the valid cells per raster row
row_valid_cells <- apply(as.matrix(raster_ref[[1]]), wide=TRUE, 1,
                        function(x) which(!is.na(x)))

```

Step 2: Fit Generalised Additive Models

We are going to use the function `bam` from the `mgcv` package to estimate our latitudinal gradient patterns. It provides faster inference for large datasets.

```

to_keep <- complete.cases(all_maps_rast) # keep track of valid rows
all_maps_rast$lat <- lat # add the latitude vector

# fit GAMs for each plants species richness distribution as a function of the latitude
my_gam_pred <- all_maps_rast[to_keep,] %>%
  # remove latitude as response variable
  dplyr::select(-lat) %>%
  # fit a GAM on with a penalised cubic regression spline as smoother for the latitude
  purrr::map(~ mgcv::bam(.x ~ s(lat, bs="cr"),
                        family = "Gamma",
                        data=all_maps_rast[to_keep,],
                        discrete=TRUE)) %>%
  # predict for unique latitude values
  purrr::map(~(mod) predict(mod,
                          newdata=data.frame(lat=unique(lat)),
                          type="response")) %>%
do.call(cbind,.)

my_gam_pred

```

Step 3: Prepare the data for the heatmap visualisation

We now have smoothed species richness predictions per latitude for each plants uses and all uses at once. Next step is to compute differences between each individual plant use categories and the “all uses” estimates taken as a baseline. In our example the “all uses” predictions (and maps) are called “GlobalSR”, make sure to update the following code to distinguish yours accordingly.

```

# matrix of deviation of plants uses' species richness from the "all uses" baseline pattern
delta_SR_matrix_scaled <- my_gam_pred %>%
  as.data.frame() %>%
  # scaled the predictions to obtain proportions per latitudinal values
  dplyr::mutate_all(~ ./sum(., na.rm=TRUE)) %>%
  # calculate differences
  dplyr::mutate(dplyr::across(-dplyr::starts_with("Global"),
                             ~ . - GlobalSR, .names = "delta{col}")) %>%
  dplyr::select(dplyr::starts_with("delta")) %>% # select plants use categories only
  dplyr::rename_all(~ gsub("delta|SR", "", .)) %>% # remove suffices
  as.matrix()

```

```

# retrieve the "all uses" (scaled) predictions only
my_gam_pred_SR_ref <- my_gam_pred %>%
  as.data.frame() %>%
  dplyr::mutate_at(dplyr::vars(GlobalSR), ~ ./sum(.,na.rm=TRUE)) %>%
  dplyr::pull(GlobalSR)

```

Step 4: Create a heatmap of the variation of the species richness among plants uses along the latitudinal gradient

We have a matrix of relative differences in species richness per latitude between each plants use categories and all confounded uses. We are going to use the package **Complexheatmap** to better visualise these differences.

```

# remove missing values and predictions in Antartica
to_remove <- !complete.cases(delta_SR_matrix_scaled) | (unique(lat) < -50)

ylat <- gam_pred_use_sr$Lat[!to_remove]

# create a pre-heatmap to compute the dendrogram
my.pre.heat.map <- ComplexHeatmap::Heatmap(
  delta_SR_matrix_scaled[!to_remove,]
)

ht <- ComplexHeatmap::draw(my.pre.heat.map)
col_order <- ComplexHeatmap::column_order(ht)

# rotate the dendrogram
column_dendo = as.dendrogram(ComplexHeatmap::column_dend(ht))

# to visualise
# plot(column_dendo %>%
#   dendextend::set("labels_colors")
# )

# optional: preferential ordering of categories
my.order <- c("AnimalFood", "Materials", "Medicines", "GeneSources",
  "EnvironmentalUses", "HumanFood", "SocialUses", "Poisons",
  "InvertebrateFood", "Fuels")

#to visualise
# plot(column_dendo %>%
#   dendextend::set("labels_colors") %>%
#   dendextend::rotate(my.order)
# )

column_dendogrm <- column_dendo %>%
  dendextend::set("labels_colors") %>%
  dendextend::rotate(col_order)

new_col_order <- labels(column_dendogrm)

```

Define top annotations

```
# optional font parameter
windowsFonts("sans" = windowsFont("sans"))
my_font_family="sans"

# get the paths to useful plants icons
icons_paths <- list.files(system.file("man/figures",package="UsefulPlants"),
                          pattern="(Uses|Food|s)\\.png$", full.names=TRUE)

anno_joyplot_list <- my_gam_pred %>%
  dplyr::select(-GlobalSR,-Lat) %>%
  dplyr::mutate_all(~ ./sum(.,na.rm=TRUE)) %>%
  dplyr::filter(!to_remove) %>%
  purrr::map(~ data.frame(x=ylat,y=.) )

# restricted vector of latitudes
ylat <- my_gam_pred$Lat[!to_remove]

# top heatmap annotations
my.top.annotations <- ComplexHeatmap::HeatmapAnnotation(

  "Uses" = ComplexHeatmap::anno_image(icons_paths,
                                       height = grid::unit(1.8, "cm"),
                                       width = grid::unit(3, "cm"),
                                       border=FALSE),

  "Species richness\nper use" = ComplexHeatmap::anno_joyplot(anno_joyplot_list,
                                                             gp = grid::gpar(lwd=1, col='black',
                                                             fill=NA,
                                                             transparency = 0.25),
                                                             scale=0.6,
                                                             width = grid::unit(2, "cm"),
                                                             which="column",
                                                             axis_param = list(
                                                               side = "right",
                                                               at = c(66.5, 23.5, 0, -23.5),
                                                               gp = grid::gpar(fontface=c(3,3,4,3),
                                                               fontsize=14,
                                                               fontfamily=my_font_family),
                                                               labels=c("Arctic Circle",
                                                               "Tropic of Cancer",
                                                               "Equator",
                                                               "Tropic of Capricorn"),
                                                               labels_rot = 0)
                                                             ),
  annotation_name_gp = grid::gpar(fontfamily=my_font_family, fontsize = 20),
  annotation_name_side = "left",
  border=TRUE,
  show_annotation_name = c(Uses = FALSE),
  which='column'
)
```

Define left annotations

```
my.left.row.annotations = ComplexHeatmap::HeatmapAnnotation(  
  foo = ComplexHeatmap::anno_mark(at = match(c(66.5, 23.5, 0, -23.5), round(ylat, 1)),  
    labels = c("Arctic\nCircle",  
              "Tropic of\nCancer",  
              "Equator",  
              "Tropic of\nCapricorn"),  
    side="left",  
    labels_gp = grid::gpar(fontface=c(3,3,4,3),  
                          fontsize=18,  
                          fontfamily=my_font_family),  
    link_gp=grid::gpar(linewidth=1.5)),  
  
  "Total utilised\nspecies richness" = ComplexHeatmap::anno_lines(  
    my_gam_pred_SR_ref[!to_remove,],  
    which="row",  
    gp = grid::gpar(lwd=c(1,0.5,0.5),  
                  col='black',  
                  fontsize=18,  
                  fill=1, lty=c("solid", "dashed", "dashed"),  
                  transparency = 0.25),  
  
    pch=16,  
    ylim = c(0, max(my_gam_pred_SR_ref[!to_remove, 3])),  
    width = grid::unit(5.5, "cm"),  
    border=TRUE,  
    axis_param = list(  
      direction="reverse",  
      gp = grid::gpar(fontsize=14),  
      at = c(0, max(my_gam_pred_SR_ref[!to_remove, 3])),  
      labels=c('0', paste0(round(max(gam_pred_use_sr[!to_remove, 6])), '')),  
      labels_rot = 0)  
    ),  
  
  annotation_name_gp = grid::gpar(fontfamily=my_font_family, fontsize=20),  
  
  annotation_name_side = "bottom",  
  
  annotation_name_rot = 0,  
  
  annotation_name_offset = grid::unit(1, "cm"),  
  
  which='row'  
)
```

Define right annotations

```
my.right.row.annotations = ComplexHeatmap::HeatmapAnnotation(  
  foo = ComplexHeatmap::anno_mark(at = match(seq(80, -40, by=-20), round(ylat, 1)),  
    labels = c(paste0(seq(80, 20, by=-20), '°N'),  
              '0°',  
              paste0(seq(20, 40, by=20), '°S')),  
    link_width= grid::unit(2, "mm"),
```

```

        side="right",
        labels_gp = grid::gpar(fontfamily=my_font_family,
                                fontsize=18),
        link_gp=grid::gpar(lwd=1.5)),

    annotation_name_gp = grid::gpar(fontfamily=my_font_family, fontsize=24),
    which='row'
)

```

Define bottom annotations

In the following code, *sp.names* is the vector of species names used to build the each (plants use) species richness map with.

```

# get plants uses sample size
plants_use_sample_size <- UsefulPlants::usefulplants |>
  `[`(sp.names)
  dplyr::select(3:12) |>
  colSums()

my.bottom.annotations = ComplexHeatmap::HeatmapAnnotation(
  "Number of\nspecies" = ComplexHeatmap::anno_barplot(plants_use_sample_size,
    add_numbers=TRUE,
    axis=FALSE,
    gp = grid::gpar(fill= 1,
                    fontsize=16),
    numbers_gp = grid::gpar(fontsize=16),
    height = grid::unit(3, "cm"),
    numbers_rot =0,
    border=TRUE,
    axis_param = list(side = "right",
                      gp=grid::gpar(fontsize=16))
  ),
  annotation_name_gp = grid::gpar(fontfamily=my_font_family, fontsize = 20),
  annotation_name_rot = 0,
  annotation_name_side = "right",
  which='column'
)

```

Create the heatmap

```

## create a palette for visualisation
my_pal <- c(rgb(67, 14, 89, maxColorValue = 255),
  rgb(97, 50, 95, maxColorValue = 255),
  rgb(128, 88, 100, maxColorValue = 255),
  rgb(181, 189, 91, maxColorValue = 255),
  rgb(96, 133, 49, maxColorValue = 255),
  rgb(58, 105, 31, maxColorValue = 255),
  rgb(24, 79, 15, maxColorValue = 255))
color.pal <- grDevices::colorRampPalette(my_pal, space="Lab")
brks <- seq(-2.5e-4, 2.5e-4, by=10^-6) # update for the range of values in your data
color_pal <- circlize::colorRamp2(breaks=brks, colors=color.pal(length(brks)), space="LAB")

```

```

my.heat.map <- ComplexHeatmap::Heatmap(
  delta_SR_matrix_scaled[!to_remove,],
  col = color_pal, # use customised color palette
  show_row_names = FALSE,
  show_column_names = FALSE, # show column names
  cluster_columns = column_dendogr,
  column_dend_height = grid::unit(4, "cm"),
  show_column_dend = TRUE,
  cluster_rows = FALSE, # turn off row clustering
  column_dend_side = "top", # put the dendogram on the top
  top_annotation = my.top.annotations,
  left_annotation = my.left.row.annotations,
  right_annotation = my.right.row.annotations,
  bottom_annotation = my.bottom.annotations,
  heatmap_legend_param = list(
    title = expression(atop(atop(Delta~scaled,""),species~richness)),
    at = c(-2.5e-4,0,2.5e-4),
    legend_height = grid::unit(5,'cm'),
    legend_width = grid::unit(0.7,'cm'),
    labels_gp = grid::gpar(fontfamily = my_font_family,
                           fontsize=14),
    title_gap = grid::unit(1.5, "cm"),
    title_gp = grid::gpar(fontfamily = my_font_family,
                           fontsize=16, cex=1.2)
  ),
  gap = grid::unit(0, "mm"),
  row_gap = grid::unit(0, "mm")
)

```