Species community matrices for estimating phylogenetic diversity

Weston Testo

8/24/2021

Here, I detail a flexible approach for estimating Faith's phylogenetic diversity using the `picante` package. This is of course a very well-known and popular package that has long been in the toolkits of many community ecologists and others, and I am sure that there are many ways to do what I describe here. As with other community phylogenetic methods, calculating Faith's PD in `picante` requires a species-community matrix as input. Depending on the type of input data you have, there are lots of ways to make a species-community matrix for this purpose, but I have not found a straightforward approach to developing one from point occurrence records, so I have set out to do so. The main challenges that I have come across are 1) ensuring that the extent and resolution of the grid used to define the matrix can be easily adjusted (there are lots of ways to make grids, and some are better fits for this application than others) and 2) converting the joined occurrence-grid data object to a species-community matrix. Neither of these are particularly complicated tasks, but it took me enough effort that I figured other folks might benefit.

For this walkthrough I will be calculating PD for ferns in Madagascar. Estimating Faith's PD using the `pd( )` function in `picante` requires two inputs: 1) a time-calibrated phylogeny and 2) a species-community matrix. To keep things simple, I will use the tree from Testo & Sundue (2016) link here and occurrence data from the Pteridophyte Collections Consortium's data portal.

Why do it this way?

- Flexibility
  - Grid extent is bounded by the extent of the specified study area
- Shapefiles at hand
  - By using `rnaturalearth`, country polygons can be easily specified
- Fast
  - All code should run in < 30 seconds for a matrix with thousands of cells and hundreds of taxa

---

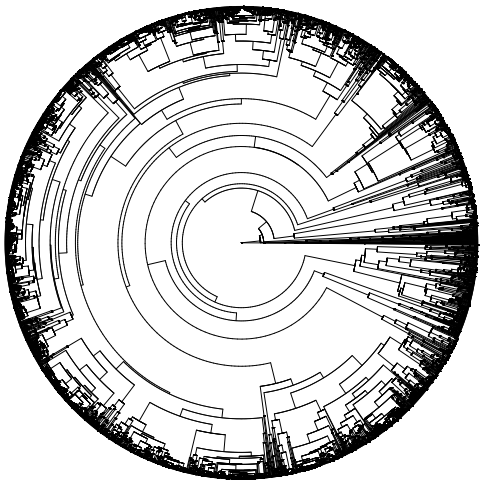
Step-by-step walkthrough

Loading & checking data First, we'll want to load the packages we'll be using:

```
library(ape)
library(geiger)
library(dplyr)
library(sf)
library(ggplot2)
library(picante)
library(rnaturalearth)
library(rgdal)
library(biscale)
library(cowplot)
```

First, let's read in the phylogeny

```
Tree <- read.nexus("4000phylogeny.tre")

plot(Tree, type = "fan", show.tip.label=F, edge.width=0.25)
```



Now we'll load the CSV with our occurrence data. The lat/long data are in decimal degrees (WGS84).

```
Occ <- read.csv("madagascar_ferns.csv")

head(Occ)
```

```
##                   scientificName decimalLatitude decimalLongitude
## 1 Ophioglossum_vulgatum_vulgatum       -25.51666         45.30000
```

```
## 2           Cheilanthes_perrieri       -25.31694         45.48500
## 3         Doryopteris_pedatoides       -25.15000         46.71667
## 4           Drynaria_willdenowii       -25.13694         46.76972
## 5         Selaginella_pectinata        -25.08333         46.78333
## 6          Selaginella_echinata        -25.08333         46.81666
```

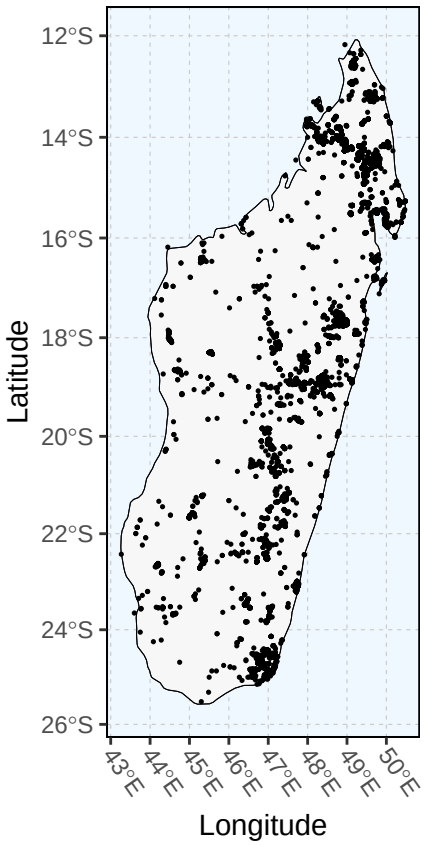
Since we'll be working with the`sf` package, we'll convert our data to a Simple Features object. We'll also reproject the coordinates to an equal-area projection.

```
Occ <- st_as_sf(Occ, coords=c("decimalLongitude", "decimalLatitude"),
                crs='+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0')

behrmann <- "+proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

Occ <- st_transform(Occ, behrmann)
```

Now, we'll read in our map from `rnaturalearth` and subset out the country that we are interested in: Madagascar. We'll also reproject so that we match the occurrence data.

```
Map <- ne_countries(scale = 50, returnclass = "sf",
                            type = "countries") %>%
  filter(sovereignt == "Madagascar")

Map <- st_transform(Map, behrmann)
```

A quick reality check to make sure we don' have any erroneous points...



Things look okay – most collections are in the island's east and north, where we know fern diversity should be highest.

Matching phylogenetic and occurrence data Because we can't estimate phylogenetic diversity for taxa that are not represented in our phylogeny, our next step is to drop any occurrences of species missing from our tree. There are numerous ways to do this, but we'll keep things simple here:

```
MatchedNames <- intersect(Tree$tip.label, Occ$scientificName)

Occ <- Occ %>% filter(scientificName %in% MatchedNames)

length(unique(Occ$scientificName))
```

```
## [1] 302
```

That leaves us with 302 taxa moving forward. We could certainly improve this with a bit more thorough effort at taxonomic resolution, but we'll save that for another day.

Creating a grid In order to make our species-community matrix, we need to make a grid to intersect with our occurrence data. A bit of time perusing the internet will reveal that there are lots of ways to make grids for spatial analyses in R, but some of these methods are better suited for this approach than others. The `st_make_grid( )` function from he `sf` package is a good choice here because it makes a grid whose extent covers the bounding box of our study area. Once we make the grid, we'll use `st_sf` to convert it to an sf object and provide cell ID numbers, which will prove useful later on.

We'll be using a grid cell size of 50 km^2.

```
GridSize <- 50000 ##specify grid dimensions (in this case, in meters)

Grid <- st_make_grid(x = Map, what = "polygons", cellsize = GridSize)

Grid <- st_sf(idcell = 1:length(Grid), geom = Grid) %>%
  st_cast("POLYGON")
```

Joining data to make the species-community matrix Now that we have a grid, the next step is to join it with our species occurrence data. We can do this easily in `sf` using the `st_intersection( )` function [NB: make sure to use `st_intersection` rather than `st_intersects` here, as the latter function just returns logical response telling you if two objects intersect]. We'll then convert this to a data frame and retain just the taxon names and the corresponding grid cell ID values. This will be the basis for making our species-community matrix.

```
IntersectedGrid <- st_intersection(Grid,Occ)

IntersectedGridDF <- as.data.frame(IntersectedGrid)[,c(1,2)]

head(IntersectedGridDF)
```

```
##      idcell          scientificName
## 7         7  Doryopteris_pedatoides
## 21       21    Drynaria_willdenowii
## 21.1     21 Microsorum_scolopendria
## 20       20       Adiantum_poiretii
## 22       22  Ampelopteris_prolifera
## 22.1     22   Platycerium_alcicorne
```

The following code will create the species-community matrix by merging data frames based on grid cell ID values:

```
GridSpeciesMatrix <- merge.data.frame(x = data.frame(idcell = Grid$idcell),
                                      y = IntersectedGridDF,
                                      all.x = TRUE, by.x = TRUE,
                                      sort = TRUE) %>% table()
```

A quick look inside the matrix gives a sense of what we are looking at:

```
GridSpeciesMatrix[105:108 ,481:484]
```

```
##       scientificName
## idcell Elaphoglossum_acrostichoides Elaphoglossum_ambrense
##    105                            0                      0
##    106                            2                      0
##    107                            0                      0
##    108                            0                      0
##       scientificName
## idcell Elaphoglossum_angulatum Elaphoglossum_angustatum
##    105                      1                       0
##    106                      1                       0
##    107                      0                       0
##    108                      0                       0
```

Calculating Phylogenetic Diversity We now have the information we need to calculate PD in `picante`, so let's do it:

```
PdOutput <- pd(GridSpeciesMatrix,Tree)
```

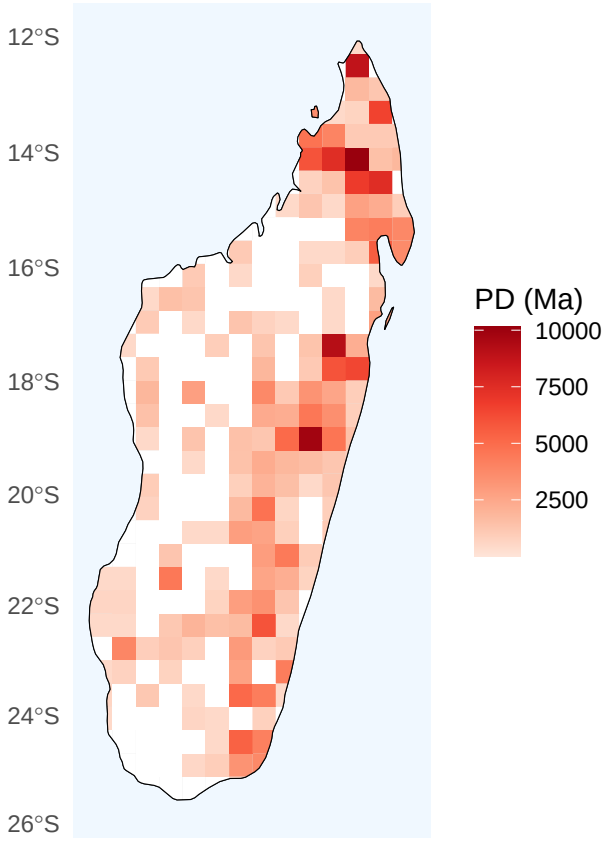
Now we can join the object `PdOutput` with the grid by assigning cell ID values based on row numbers.

```
PdOutput <- PdOutput %>%
  mutate(idcell = row_number())

PdOutput$idcell <- as.integer(PdOutput$idcell)

GridPD <- left_join(Grid,PdOutput, by = "idcell")
```

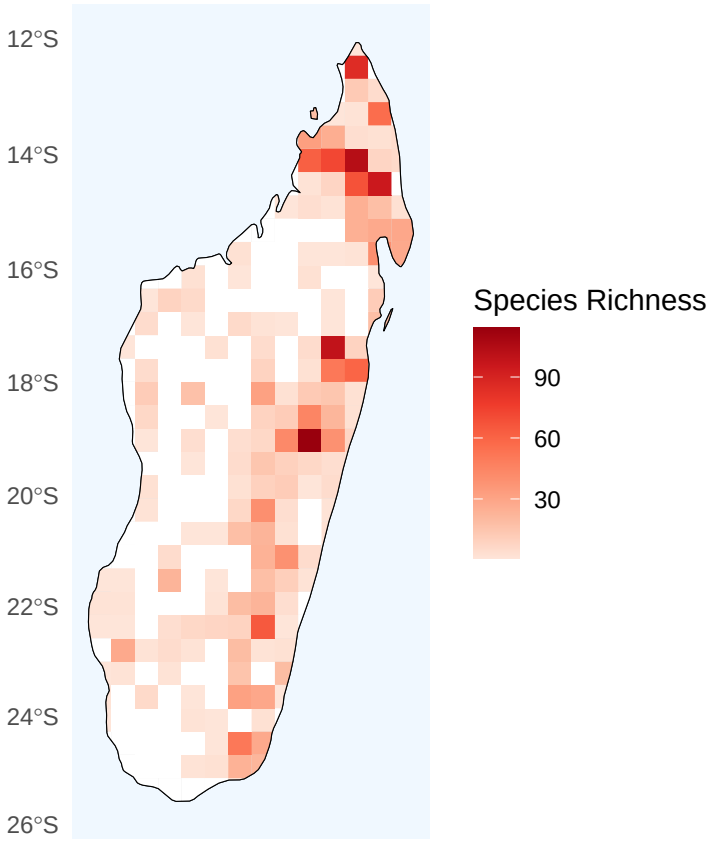
This object includes cells in the ocean, because our grid extent is defined as a rectangular bounding box around our country of origin. There are a few options for how to deal with this - one simple one is to use `st_intersection` again to retain only cells that intersect with our land polygon.

```
ClippedGridPD <- st_intersection(Map, GridPD)

ClippedGridPD <- ClippedGridPD %>% select(idcell, PD, SR, geometry)
```

Taking a look a PD values...



This looks good, but if we take a look at species richness itself, we will see that the two are strongly correlated:



Bivariate mapping with the `biscale` package Because PD is generally closely linked to species richness, I also wanted to take care to visualize the correlation between these two metrics. After spending a lot of time working on a way to come up with a way to effectively map two variables together (mostly a lot time mixing colors), I came across the `biscale` package, which was built for exactly this purpose. You can check it out here if you want to read more. I'll be following the tutorial closely here.

`bi_class( )` assigns our data (grid cell scores of phylogenetic diversity and species richness) into quantiles, which we'll need for visualization purposes. In this case, we'll use the 'Fisher' break style and break both phylogenetic diversity and species richness ino three quantiles.

```
BiscaleData <- bi_class(ClippedGridPD, x = PD, y = SR,
                        style = "fisher", dim = 3)
```

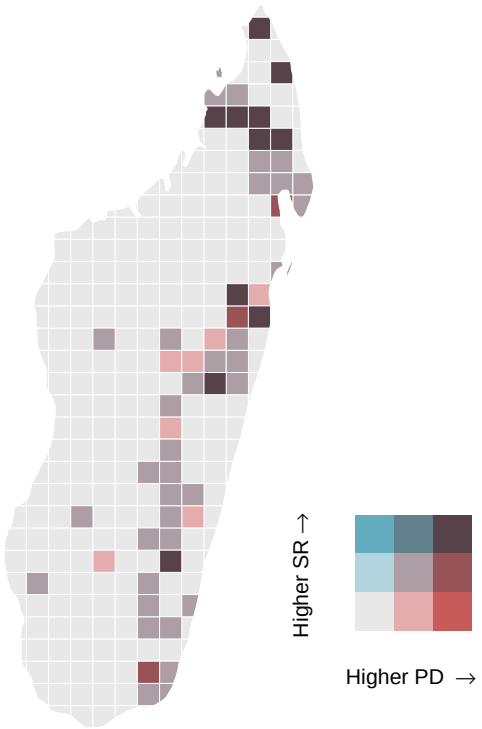
Taking a look at the output of `bi_class`, we can see the quantile assignments for both phylogenetic diversity and species richness in the "bi_class" column.

```
## Simple feature collection with 6 features and 4 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 4620110 ymin: -1807525 xmax: 4842794 ymax: -1708584
## CRS:           +proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_de
##       idcell        PD  SR bi_class                       geometry
## 1.264    389  7492.919  72      3-3 POLYGON ((4673719 -1807525,...
```

```
## 1.265    390 10151.697 103      3-3 POLYGON ((4723719 -1807525,...
## 1.266    391  1462.366   8      1-1 POLYGON ((4773719 -1807525,...
## 1.267    392  1828.388   6      1-1 POLYGON ((4838344 -1757525,...
## 1.268    401     0.000   0      1-1 POLYGON ((4623719 -1753841,...
## 1.269    402  4721.576  33      2-2 POLYGON ((4626361 -1757525,...
```

Now, we are ready to plot our bivariate map. First, we'll use `bi_legend( )` to make a suitable bivariate legend, then use a combination of `ggplot( )` and `ggdraw( )` to make the map.

```
Legend <- bi_legend(pal = "GrPink",
                    dim = 3,
                    xlab = "Higher PD ",
                    ylab = "Higher SR",
                    size = 8)

BivariateMap <- ggplot() +
  geom_sf(data = BiscaleData, mapping = aes(fill = bi_class),
          color = "white", size = 0.1, show.legend = F) +
  bi_scale_fill(pal = "GrPink", dim = 3)+
  bi_theme()


FinalPlot <- ggdraw() +
  draw_plot(BivariateMap, 0, 0, 1, 1) +
  draw_plot(Legend, 0.55, 0.1, 0.275, 0.275)
FinalPlot
```



Looks pretty good! As we already saw, most cells with high phylogenetic diversity also are in the upper quartile of species richness (purplish cells). However, we are now able to pick out some areas of high phylogenetic diversity and low species richness (bright red) – these areas look to mostly correspond to high elevation humid forest.