Biodiversity Facets tutorial

Matthias Grenié & Marten Winter

Monday July 5th 2021

Contents

Setup

RStudio

Double-click on the `.Rproj` file which is as the root of the folder you downloaded. This should open a new session in RStudio that will start at the root of the project folder.

Following the tutorial

To follow the tutorial you have plenty of options:

1. It is accessible online at https://rekyt.github.io/biodiversity_facets_tutorial/ click on `Tutorial` in the navigation bar to access the tutorial.
2. You can access open the `diversity_facets_tutorial.Rmd` in RStudio to follow along.
3. You can also open the `diversity_facets_tutorial.R` in RStudio to get only the code to execute exactly the codes here.

Installing needed packages

We will use some specific packages in the rest of the tutorial. To make sure you have them please run the following command:

```
install.packages(
  c("ade4",
    "ape",
    "FD",
    "fundiversity",
    "ggplot2",
    "ggspatial",
    "performance",
    "picante",
    "rnaturalearth",
    "sf")
)
```

Context of the study

We will be using the data from the following study (Döbert et al. 2017):

> Döbert, T.F., Webber, B.L., Sugau, J.B., Dickinson, K.J.M. and Didham, R.K. (2017), Logging increases the functional and phylogenetic dispersion of understorey plant communities in tropical lowland rain forest. J Ecol, 105: 1235-1245. https://doi.org/10.1111/1365-2745.12794

Logging is the major cause of forest degradation in the Tropics. The effect of logging on taxonomic diversity is well known but more rarely studied on other facets of biodiversity such as functional diversity and phylogenetic diversity. Functional diversity and phylogenetic diversity should better reflect the impact of logging on ecosystem. For example logging can decrease the functional "redundancy" observed in ecosystems, meaning that some functional traits could be lost.

The tropical lowland rain forests on the island of Borneo are floristically among the most diverse systems on the planet, yet large-scale timber extraction and conversion to commercial tree plantations continue to drive their rapid degradation and loss. As is the case for the majority of tropical forests, the effects of logging on habitat quality in these forests have rarely been assessed, despite the critical implications for biodiversity conservation. Moreover, studies investigating the effects of logging on plant community dynamics across both tropical and temperate forest ecosystems have rarely focused on the understorey, despite its crucial relevance for successional trajectories.

Our goal with this tutorial is to reproduce the analyses from the paper and analyze how logging impacts the different facets of the diversity of the understorey vegetation, and to reveal to what extent similar facets give similar answers. The general goal is to familiarize yourself with the data and functions needed to compute diversity indices. As well the general principles behind them.

Loading the data

Fortunately for us, the authors of the study have shared openly the data they used in their article (Döbert et al. 2018). They are available through the Dryad platform at the following link: https://doi.org/10.5061/dryad.f77p7

> Döbert, Timm F. et al. (2018), Data from: Logging increases the functional and phylogenetic dispersion of understorey plant communities in tropical lowland rainforest, Dryad, Dataset, https://doi.org/10.5061/dryad.f77p7

The fact that these data researchers provided the full dataset including all data and meta-data will help us reproduce the exact same analyses as well as additional analyses not in their paper.

Getting the data

To get the data you can follow the above-mentioned link https://doi.org/10.5061/dryad.f77p7 and click on the "Download Dataset" button available on the top right of the webpage. It will download a .zip file that you can unzip in the folder you created for the project. This will create a folder named `doi_10.5061_dryad.f77p7__v1` that contains all needed data files.

NOTE: We've also included a copy of the unzipped file in the `data` folder of the practical project.

Summarizing the data

The zip file contains 5 files (also available in the `data/doi_10.5061_dryad.f77p7__v1/` folder):

- `README.txt` which is a text file that describes the content of the other files with great precision. It details all the columns available in the other files.
- `PlotData.csv` is a comma-separated file that describes characteristics for each of the sampled vegetation plots including logging metrics, environmental variables as well as taxonomic, functional and phylogenetic diversity indices (to which we'll compare the indices we compute ourselves).
- `PlotSpeciesData.csv` is a comma-separated file that contains a matrix of biomass values for the plant taxa sampled across the sampled vegetation plots.
- `phylo_tree.nwk` the phylogenetic tree describing the relationships between species.
- `SpeciesTraitData.csv` contains the complete list of species sampled across all vegetation plots, with their associated traits both continuous and discrete.

We will load each of the file (apart from the phylogenetic tree) in your workspace now with the `read.csv()` function:

```
plot_data         = read.csv("data/doi_10.5061_dryad.f77p7__v1/PlotData.csv",
                             na.strings = c("NA", "na"),
                             stringsAsFactors = TRUE)
plot_species_data = read.csv("data/doi_10.5061_dryad.f77p7__v1/PlotSpeciesData.csv")
species_traits    = read.csv("data/doi_10.5061_dryad.f77p7__v1/SpeciesTraitData.csv",
                             na.strings = c("NA", "na"), stringsAsFactors = TRUE)
```

To describe the data we will use the `str()`, `summary()`, and `dim()` functions.

```
str(plot_data)
summary(plot_data)

str(plot_species_data[, 1:5])
summary(head(plot_species_data)[,1:5])
dim(plot_species_data)

# Transform one column for further analyzed
species_traits$seed = ordered(species_traits$seed)
```

```
str(species_traits)
summary(species_traits)
```

Questions for you:

- **Q1**: How many plots were sampled?
- **Q2**: How many species are there in the dataset?
- **Q3**: How many traits are available?
- **Q4**: How many of them are continuous? How many of them are discrete?
- **Q5**: What is the most numerous family among all observed species?
- **Q6**: What is the most numerous genus?

Description of the environment variables

Forest loss proportion is one of the main driver variable. The data has been acquired across different block with different proportion of logging and compared to unlogged forest.

```
boxplot(forestloss17 ~ block, data = plot_data,
        xlab = "Block of plot", ylab = "Forest loss (%)",
        main = "Forest loss in funciton of block of data")
```

Functional diversity

Now that we loaded all the datasets we can proceed to compute functional diversity per plots.

Computing Biomass-weighted mean traits per plot

One first way to compute functional diversity is to compute mono-dimensional trait diversity (Lavorel and Garnier 2002). We can compute the average trait observed at each plot to described the effect of logging on the understorey vegetation. Because we're interested in the average trait possessed by the community we can compute the community-weighted mean trait CWM_i as follow:

$$CWM_i = \sum_{j=1}^{S} b_{ij} \times t_j \tag{1}$$

CWM_i is the community-weighted mean trait in plot i, S is the total number of species, b_{ij} is the biomass of species j in plot i, and t_j is the trait of species j.

To do so we will use the function `functcomp()` in the `FD` package (Laliberté and Legendre 2010). But we first need to organize our data.

```
# Make site-species data.frame
sp_com           = plot_species_data[, -1]
rownames(sp_com) = plot_species_data$X
sp_com = as.matrix(sp_com)

# Make synthesized trait data.frame
traits = species_traits[, -c(1:5)]
rownames(traits) = species_traits$species.code
```

Now that the data is organized we can compute the CWM per site for all traits:

```
# Get only continuous CWM
quanti_cwm = FD::functcomp(traits[, c("height", "sla", "wood.dens")],
```

```
                              sp_com, CWM.type = "dom")
quanti_cwm$plot.code = rownames(quanti_cwm)
```

The function outputs the CWM as expressed above for continuous traits. We will then merge this information with the CWM values.

```
# Merge environmental data with CWM
cwm_env = merge(
  quanti_cwm,
  plot_data[, c("plot.code", "block", "forestloss17", "roaddensprim")],
  by = "plot.code"
)
```

We can now visualize the relationship between the CWM and the environmental gradients.

```
par(mfrow = c(2, 2))
plot(cwm_env$forestloss17, cwm_env$height,
     xlab = "Forest loss (%)", ylab = "Biomass-weighted height",
     main = "CWM Height vs. forest loss")
plot(cwm_env$forestloss17, cwm_env$sla,
     xlab = "Forest loss (%)", ylab = "Biomass-weighted SLA",
     main = "CWM SLA vs. forest loss")
plot(cwm_env$forestloss17, cwm_env$wood.dens,
     xlab = "Forest loss (%)", ylab = "Biomass-weighted wood density",
     main = "CWM Wood density vs. forest loss")
plot(cwm_env$roaddensprim, cwm_env$height,
     xlab = "Road density (km.km^-2)", ylab = "Biomass-weighted height",
     main = "CWM Height vs. road density")
```

Questions for you:

- **Q7**: How would you describe the relationship between the different CWMs and forest loss?
- **Q8**: Can you test the correlation using the function `cor.test()` and does it support your previous statements?
- **Q9**: How would you describe the understorey vegetation changes with increasing forest loss?

Recompute the CWM by proportion of each category of each trait along the environmental gradient.

```
non_quanti_cwm = FD::functcomp(traits[, -c(5:7)],
                               sp_com, CWM.type = "all")
non_quanti_cwm$plot.code = rownames(non_quanti_cwm)

non_quanti_cwm = merge(
  non_quanti_cwm,
  plot_data[, c("plot.code", "block", "forestloss17", "roaddensprim")],
  by = "plot.code"
)
```

We used the same function as above `functcomp()` with the option `CWM.type = "all"`. The function computes the sum of biomass of each category for categorical traits.

```
par(mfrow = c(1, 1))
plot(non_quanti_cwm$forestloss17, non_quanti_cwm$woody_no,
     xlab = "Forest loss (%)", ylab = "Sum of biomass of non-woody species",
     main = "Biomass of non-woody species vs. forest loss")
```

Questions for you

- **Q10**: How does this observation compare to above description of the change of understorey vegetation along the forest loss gradient?

Building the functional space

Before computing the functional diversity indices we need first to place the species on a functional space. The way to do is to visualize the species cloud onto the synthetic axes that represent their trait values. Because we cannot visualize that different traits (our vision is still limited to only 3 dimensions!) we need to use dimension reduction techniques such as *Principal Component Analysis* (PCA). Dimension reduction techniques combines the different variables to give synthetic axes accounting for the correlations between the different input variables Because we have a dataset that contain both continuous and categorical trait data, we cannot use PCA and we will have to use a slighly different statistical tool called *Principal Coordinates Analysis* (PCoA, also named Metric Dimensional Scaling) that follow similar principles.

To compute the PCoA we first need to compute a distance matrix that expresses the difference between each pair of species. Because we have a mixture of continuous and categorical traits, we cannot use the Euclidean distance and have to resort to use the Gower's dissimilarity metric through the `daisy()` function with the package `cluster`.

```
gower_dissim = cluster::daisy(traits)
```

To perform the PCoA we will be using the `ade4` package with the function `dudi.pco()`:

```
trait_pcoa = ade4::dudi.pco(ade4::quasieuclid(gower_dissim), nf = 3,
                            scannf = FALSE)
trait_pcoa
```

The `trait_pcoa` object contains the coordinates of each species along the different PCoA axes (we chose 5 to have a limit). We can visualize the results with the following command:

```
ade4::scatter(trait_pcoa, clab.row = 0)
```

We see two well separated groups indicating strong differences along the two first axes of the PCoA. We can visualize the meaning of the groups. We can try to better understand this group by looking at the distribution of traits along these groups:

```
ade4::s.class(trait_pcoa$li[,1:2], fac = traits$pgf)
```

Questions for you

- **Q11**: Using the metadata available in the `README.txt` file, what is the meaning `pgf` column means?
- **Q12**: How do you interpret the PCoA results given your answer to the previous question?

Computing functional diversity indices

Now that we have species positioned in a multidimensional space we can actually compute distinct functional diversity indices. For that we'll be using the `fundiversity` package that offers both flexibility and consistency to compute the indices.

We will first compute Functional Richness (FRic) with the `fd_fric()` function:

```
site_fric = fundiversity::fd_fric(trait_pcoa$li, sp_com, stand = FALSE)
```

Then we will also compute Rao's Quadratic Entropy (Rao's Q) and Functional Evenness (FEve):

```
site_raoq = fundiversity::fd_raoq(trait_pcoa$li, sp_com)
site_feve = fundiversity::fd_feve(trait_pcoa$li, sp_com)

site_fd = merge(
  merge(site_fric, site_raoq, by = "site"),
```

```
  site_feve,
  by = "site"
)
site_fd$plot.code = site_fd$site
site_fd = site_fd[, -1]
```

We can now compare the observed relationship with forest loss:

```
site_env_fd = merge(site_fd,
                    plot_data[, c("plot.code", "forestloss17", "roaddensprim")],
                    by = "plot.code")

par(mfrow = c(2, 2))
plot(site_env_fd$forestloss17, site_env_fd$FRic,
     xlab = "Forest loss (%)", ylab = "Functional Richness (FRic)",
     main = "Functional Richness vs. forest loss")
plot(site_env_fd$forestloss17, site_env_fd$Q,
     xlab = "Forest loss (%)", ylab = "Rao's Quadratic Entropy",
     main = "Q vs. forest loss")
plot(site_env_fd$forestloss17, site_env_fd$FEve,
     xlab = "Forest loss (%)", ylab = "Functional Evenness (FEve)",
     main = "FEve vs. forest loss")
plot(site_env_fd$roaddensprim, site_env_fd$FRic,
     xlab = "Primary Road Density (km.km^-2)", ylab = "Functional Richness (FRic)",
     main = "FRic vs. road density")
```

Questions for you

- **Q13**: How would you describe the relationships between functional diversity and forest loss and road density?
- **Q14**: Using the plot generated by the code beneath how could you describe the relationships between the three different functional diversity indices we computed?

```
panel.cor = function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use = "complete.obs"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(~FEve + Q + FRic, data = site_env_fd, lower.panel = panel.smooth,
      upper.panel = panel.cor, gap = 0, row1attop = FALSE)
```

One issue we're having with our functional diversity indices is also that some of them correlate with species richness:

```
site_rich_fd = merge(
  site_fd,
  plot_data[, c("plot.code", "ntaxa")],
  by = "plot.code"
)
```

```
pairs(ntaxa ~ FRic + FEve + Q, data = site_rich_fd, upper.panel = panel.cor)
```

Because we are using indices computed with biomass values the indices should be more related to the total biomass values than species richness. Let's get the total biomass values per site and correlate it with functional diversity indices.

```
site_biomass = rowSums(sp_com)
site_biomass = stack(site_biomass)

site_biomass$plot.code = site_biomass$ind
site_biomass$tot_biomass   = site_biomass$values

site_biomass = site_biomass[, c("plot.code", "tot_biomass")]

site_rich_fd = merge(
  site_rich_fd,
  site_biomass,
  by = "plot.code"
)

pairs(tot_biomass ~ FRic + FEve + Q, data = site_rich_fd,
      upper.panel = panel.cor)
```

Questions for you

- **Q15**: How does the relationship between indices with species richness compare with the one observed with total biomass values? (You can use the function `cor.test()` if you want to test the association)

Null modeling

The principle of null modelling is to create random communities following certain rules to get an expected distribution of diversity metrics while keeping some properties of the data constant. In our case, we know that functional diversity is directly linked to the number of species, so we want to keep the species richness constant while changing the distribution of functional diversity.

Because the site-species matrix contains biomass values which are not discrete, the classical swapping algorithms will not work to maintain total biomass per site and species overall biomass. The solution is then to perform a null model based on trait values only. In this way it will give us a null distribution of trait values while maintaining the same richness per plot and the same relative biomass distribution.

To do so we'll shuffle the trait table along species. **Caution**: in our case we do not want to break the links that exist between trait values, so we will be shuffling entire rows of traits and not trait individually. This would result in a different null model otherwise.

Because we were using the PCoA axes as our "synthetic traits" above we'll perform the shuffling between species names on these PCoA axes.

```
# Set random seed so that everybody gets the same null traits
set.seed(20210705)

# Number of null simulations
# CAUTION: increasing this number may increase future computation time by a lot
n_null = 99

# Repeat the operation as many times as set aboev
null_traits = lapply(seq.int(n_null), function(x) {
  null_trait = trait_pcoa$li
```

```
  # Shuffle species names
  null_species = sample(rownames(trait_pcoa$li), nrow(trait_pcoa$li))

  # Replace species name in table
  rownames(null_trait) = null_species

  # Do not forget to return the modified table!
  return(null_trait)
})

str(null_traits, max.l = 0)
head(null_traits[[1]])
```

We now obtain a distribution of null traits on which we still need to compute functional diversity indices. We'll apply similar steps as above to perform the functional diversity computation. But in this case we'll have to apply the step for each distribution of null trait.

```
# Beware this make take a long time
null_fd = lapply(seq(length(null_traits)), function(y) {

  x = null_traits[[y]]

  null_fric = fundiversity::fd_fric(x, sp_com, stand = FALSE)
  null_raoq = fundiversity::fd_raoq(x, sp_com)
  null_feve = fundiversity::fd_feve(x, sp_com)

  # Combine all null functional diversity values
  null_all = merge(
    merge(null_fric, null_raoq, by = "site"), null_feve, by = "site"
  )

  # Null Index to separate between all null simulations
  null_all$null_id = y

  return(null_all)
})

null_fd_all = do.call(rbind.data.frame, null_fd)
head(null_fd_all)
```

We now observe a list of null functional diversity metrics for each site. Because computing functional diversity on null traits is computationally intensive, running more simulations can take a long time. We've included a version of the null functional diversity values with 999 simulations in the `data/` folder. We're now going to use this precomputed version to get a better approximation of the expected distribution under the null hypothesis.

```
null_fd_999 = readRDS("data/null_fd_999.Rds")

head(null_fd_999)
```

With this null distribution we can now compare the observed values of functional diversity with the null ones. Let's for example focus on the site `"a100f177r"`:

```
# The observed value of FRic for the site
subset(site_fd, plot.code == "a100f177r")$FRic
```

```r
# The null distribution of FRic for the same site
summary(subset(null_fd_999, site == "a100f177r")$FRic)
```

We can visualize this comparison with an histogram:

```r
par(mfrow = c(1, 1))
# Visualize histogram of null values
hist(subset(null_fd_999, site == "a100f177r")$FRic,
     breaks = 20,
     xlab = "null Functional Richness",
     ylab = "Frequency",
     main = "FRic comparison for site 'a100f177r'")
abline(v = subset(site_fd, plot.code == "a100f177r")$FRic,
       col = "darkred", lwd = 2)
```

Question for you

- **Q16**: How would describe verbally the position of the observed value of FRic for site "a100f177r" compared to the null distribution?

To get a proper estimate of the relartive position of the observed value compared to the null distribution we have to build the Empirical Cumulative Distribution Function (ECDF) that will give us the exact quantile of the observed value. We will do so with the `ecdf()` function:

```r
# Build the ECDF
one_null_fric_ecdf = ecdf(subset(null_fd_999, site == "a100f177r")$FRic)

# Then actually use it
obs_fric = subset(site_fd, plot.code == "a100f177r")$FRic

one_null_fric_ecdf(obs_fric)
```

Question for you

- **Q17**: What the quantile of the observed FRic value in the end?

This gives us an empirical comparison of the observed value with the null distribution. However, in macroecology we prefer to even standardize further through the use of Standardized Effect Sizes (SES). As it is done in the article we are using for our analyses. These are simpler to compute than ECDF and simplify the interpretation. SESs are computed in the following way:

$$SES_i = \frac{\overline{y_{\text{null},i}} - y_{\text{obs},i}}{\text{SD}_{\text{null},i}}$$

with SES_i the standardized effect size of the index at site i, $\overline{y_{\text{null},i}}$ the average observed value along the null distribution of the index at site i, $y_{\text{obs},i}$, and $\text{SD}_{\text{null},i}$ the standard deviation of the null distribution of the index at site i. This index is negative when the observation is smaller than the average of the null distribution, and positive otherwise. In the literature an SES value under -2 or above 2 is generally considered as significant.

However, note that there are controversies in the literature about the use of SESs compared to the use of the ECDF because we're only leveraging on the use of the mean and standard deviation of the null distribution instead of using the entirety of the distribution.

Now we need to compute the average and standard deviation of the null distribution for each index and each site. We will do so using the `aggregate()` function.

```r
# Compute average and standard deviation of null distribution
mean_null_fd = aggregate(
```

```
  cbind(mean_FRic = FRic, mean_Q = Q, mean_FEve = FEve) ~ site,
  data = null_fd_999, FUN = mean, na.rm = TRUE
)
sd_null_fd   = aggregate(
  cbind(sd_FRic = FRic, sd_Q = Q, sd_FEve = FEve) ~ site, data = null_fd_999,
  FUN = sd, na.rm = TRUE
)

# Merge null mean & sd with observed values
obs_null_fd = merge(
  site_fd,
  merge(mean_null_fd, sd_null_fd, by = "site"),
  by.x = "plot.code", by.y = "site"
)

# Compute SES
obs_null_fd$ses_FRic = (obs_null_fd$mean_FRic - obs_null_fd$FRic)/obs_null_fd$sd_FRic
obs_null_fd$ses_Q = (obs_null_fd$mean_Q - obs_null_fd$Q)/obs_null_fd$sd_Q
obs_null_fd$ses_FEve = (obs_null_fd$mean_FEve - obs_null_fd$FEve)/obs_null_fd$sd_FEve

# Cleaner table
ses_fd = obs_null_fd[, c("plot.code", "FRic", "Q", "FEve", "ses_FRic", "ses_Q",
                         "ses_FEve")]
```

Question for you

- **Q18**: Using the `subset()` function with the greater (or equal) than `>=` and the lower (or equal) than `<=`, can you determine how many sites show a significant deviation from the null observation? (absolute SES >= 2)
- **Q19**: Using similar code as used for observed values, what are the relationships between SES values and forest loss?

Mapping functional diversity

One of the joy of doing macro-ecology is to work with spatial data. Spatial data means that we have to draw maps and this can help uncover structures in our data. In this section of the tutorial we're going to use both the observed and SES functional diversity indices to draw maps and compare them to maps of species richness to visualize the geographical structure of the dataset. We we'll be using the packages `sf` for creating and manipulating spatial data, `rnaturalearth` to get background maps, and `ggplot2` to show them. **Nota Bene**: The goal of this particular section is to make nice visualizations of our data and see potential structure, it is not to teach the particular concept around spatial data and spatial visualization that have their own challenges. If you had trouble installing the `sf` package which may be quite capricious or if you feel lost in the meaning of the code of this section, it's fine, you can skip it.

Looking back at the plot level data we have the coordinates of the plot in UTM coordinates:

```
head(plot_data[, c(1, 4, 5)])

plot_sf = sf::st_as_sf(
  plot_data[, c(1:7)],
  coords = c("north", "east"),
  crs = sf::st_crs("+proj=utm +zone=50 +datum=WGS84 +units=m +no_defs")
)
```

We can represent a basic map to see the location of the plot at world scale:

```
library("ggplot2")

ggplot() +
  geom_sf(data = rnaturalearth::ne_countries(returnclass = "sf")) +
  geom_sf(data = plot_sf, aes(color = forestloss17)) +
  scale_color_viridis_c() +
  coord_sf(crs = sf::st_crs("+proj=eck4")) +  # Set projection
  labs(title = "Map of the concerned plots at world scale") +
  theme_bw()
```

We see that all of our plots are indeed in Malaysia so we can focus there:

```
ggplot() +
  geom_sf(data = rnaturalearth::ne_countries(continent = "Asia",
                                             returnclass = "sf")) +
  geom_sf(data = plot_sf, aes(color = forestloss17)) +
  scale_color_viridis_c() +
  coord_sf(crs = sf::st_crs(3376), xlim = c(-1072025.83, 1053446.00),
           ylim = c(85496.43, 767752.41)) +
  labs(title = "Map of plots focused on Malaysia")
ggspatial::annotation_scale() +
  theme_bw()
```

We can even zoom even more onto the plots to see them better:

```
ggplot() +
  geom_sf(data = rnaturalearth::ne_countries(country = "Malaysia",
                                             returnclass = "sf")) +
  geom_sf(data = plot_sf, aes(color = forestloss17)) +
  scale_color_viridis_c() +
  coord_sf(crs = sf::st_crs(3376), xlim = c(800000, 890000),
           ylim = c(500000, 550000)) +
  labs(title = "Map of plots zoomed-in on Sabah region") +
  ggspatial::annotation_scale() +
  ggspatial::annotation_north_arrow(location = "br") +
  theme_bw()
```

We can even add background information to better distinguish the plots in context (beware this will download map tiles from the internet):

```
ggplot() +
  ggspatial::annotation_map_tile(zoomin = -1) +
  geom_sf(data = plot_sf, aes(color = forestloss17)) +
  scale_color_viridis_c() +
  coord_sf(crs = sf::st_crs(3376), xlim = c(800000, 890000),
           ylim = c(500000, 550000)) +
  labs(title = "Map of plots zoomed-in on Sabah region") +
  ggspatial::annotation_scale() +
  ggspatial::annotation_north_arrow(location = "br") +
  theme_bw()
```

Because of the group of plots on the West we can't clearly see the distinction between plots let's focus on the ones that show a gradient in forest loss:

```
ggplot() +
  ggspatial::annotation_map_tile(zoomin = -1) +
```

```
  geom_sf(data = subset(plot_sf, block != "og"),
          aes(color = forestloss17)) +
  scale_color_viridis_c() +
  coord_sf(crs = sf::st_crs(3376), xlim = c(875000, 890000),
           ylim = c(518500, 531000)) +
  labs(title = "Map of all plots but block 'og'") +
  ggspatial::annotation_scale() +
  ggspatial::annotation_north_arrow(location = "br") +
  theme_bw()
```

And we can now visualize the map of the SES of functional diversity indices

```
ggplot() +
  geom_sf(
    data = merge(subset(plot_sf, block != "og"), ses_fd, by = "plot.code"),
    aes(color = ses_Q)
  ) +
  scale_color_distiller(type = "div", palette = "RdYlBu",
                        name =  "SES of Rao's Quadratic Entropy") +
  coord_sf(crs = sf::st_crs(3376), xlim = c(875000, 890000),
           ylim = c(518500, 531000)) +
  labs(title = "Map of all plots but block 'og'") +
  ggspatial::annotation_scale() +
  ggspatial::annotation_north_arrow(location = "br") +
  theme_gray()
```

Even with all this effort it is not clear how the SES varies between sites. But at least you're more informed about where the data we're studying comes from.

Phylogenetic diversity

Now that we computed functional diversity, its SES, and put it on the map. We can proceed similarly with phylogenetic diversity. For this whole section we will use the `ape` package to manipulate phylogenetic trees and the `picante` package to compute phylogenetic diversity indices.

Getting the phylogenetic tree

We included a copy of the phylogenetic tree used in the article (it is given in the Supplementary Information). It is named `phylo_tree.nwk` in the `data/` folder.

We can read it with the `read.tree()` function in the `ape` package:

```
phylo_tree = ape::read.tree("data/doi_10.5061_dryad.f77p7__v1/phylo_tree.nwk")

phylo_tree
str(phylo_tree)
```

Questions from you

- **Q20**: How many taxa are in the phylogenetic tree?
- **Q21**: How does this number compare to the number of taxa found in the dataset?

You can visualize the taxa in the phylogenetic tree in the `tip.label` slot of the phylogenetic tree:

```
phylo_tree$tip.label
```

- **Q22**: What do you notice with the species names? Especially compared to the ones available in `species_traits`.

To solve the naming issue we'll have to match the names used in the phylogenetic tree to the species code used in the site-species matrix. For that we'll match the epitheton to the first code available. You do not need to understand this code and can just copy-paste it to execute it because we're going to use it further down.

```
# Create an indexed list of names
phylo_names = species_traits[, c("species.code", "species")]
phylo_names$code_id = seq(nrow(phylo_names))

# Get the first species code based on species epithet
code_id_to_use = aggregate(code_id ~ species, phylo_names,
                           FUN = function(x) head(x, 1))

# Get back the data.frame of species names with the actual species.code
code_species = merge(
  code_id_to_use, phylo_names[, c("code_id", "species.code")], by = "code_id"
)

# Tidying code for edge cases
code_species$species = gsub(" ", "", code_species$species)
code_species$species = paste0(
  tolower(substr(code_species$species, 1, 1)),
  substr(code_species$species, 2, nchar(code_species$species))
)

code_species = code_species[, c("species.code", "species")]

dim(code_species)
```

We can now check that we have all the names of the phylogenetic tree available as codes:

```
length(intersect(phylo_tree$tip.label, code_species$species))
```

Now that we have a clear correspondance between species code and phylogenetic name we can proceed to the computation of phylogenetic diversity indices. This won't let use reproduce exactly the same analyses as in the paper but this is the best we can do, given the data at our disposal. If all the species were determined another possibility could have to re-create a phylogenetic tree from genetic sequences available from genetic databases. This approach however needs specific skills and is a story for another time!

Visualizing the phylogenetic tree

We can visualize the phylogenetic tree to better understand the relationship between species. With more than 600 taxa, the visualization can be quite challenging and some ajustements should be made to ease the vizualition.

The easiest way to show the phylogenetic tree is to use the `plot.tree()` function available through the `ape` package.

```
ape::plot.phylo(phylo_tree)
```

By default the function shows the phylogram type of phylogenetic tree and plot all the labels for all species. Let's make it easier to read:

```
ape::plot.phylo(phylo_tree, type = "fan", show.node.label = TRUE,
                show.tip.label = FALSE, cex = 0.6)
```

It is still difficult too read but we can already look at how botanical are related to one another.

Computing phylogenetic diversity indices

To compute phylogenetic diversity analyses we need to combine the phylogenetic tree with the site-species matrix. We need to subset the communities by selecting only species with a defined code from the previous section.

```
# Initial site-species matrix
head(sp_com[, 1:5])
dim(sp_com)

# Subset of site-species matrix compatible with phylogenetic tree
sub_phylo_com = sp_com[, as.character(code_species$species.code)]
dim(sub_phylo_com)
```

To measure phylogenetic diversity we will compute the Mean Pairwise Distance (MPD, Webb (2000)) using the `picante` package. The MPD is an index that represents the average distance between all pairs of species occurring in the community. It can also be weighted by the abundance or the biomass of considered species so that more weight is given to species that show the greatest abundance.

The first data needed to compute the MPD is the phylogenetic distance between pair of species. We'll use the cophenetic distance which represent the same relationships as a phylogenetic tree but through a distance matrix. We can use the function `cophenetic.phylo()` in the `ape` package to obtain cophenetic distances.

```
# Compute cophenetic distances from the phylogenetic tree
cophen_dist = ape::cophenetic.phylo(phylo_tree)

str(cophen_dist)

# We need to change the names to species codes
corres_codes = data.frame(
  species = rownames(cophen_dist)
)
corres_codes = merge(corres_codes, code_species, by = "species")
rownames(cophen_dist) = corres_codes$species.code
colnames(cophen_dist) = corres_codes$species.code
```

Then to compute MPD we use the `mpd()` function in the `picante` package.

```
# Observed Mean Pairwise Distance
# Unweighted
mpd_val_uw = picante::mpd(sub_phylo_com, cophen_dist, abundance.weighted = FALSE)
# Weighted
mpd_val_w = picante::mpd(sub_phylo_com, cophen_dist, abundance.weighted = TRUE)

# Make a nice data.frame with observed MPD values
obs_mpd = data.frame(
  plot.code = rownames(sub_phylo_com),
  mpd_unweighted = mpd_val_uw,
  mpd_weighted = mpd_val_w
)

# Add forest loss proportion and richness for each site
obs_mpd = merge(obs_mpd, plot_data[, c("plot.code", "forestloss17", "ntaxa")])
```

Questions for you

- **Q23**: What are the relationships between weighted and unweighted version of the MPD?
- **Q24**: What are the relationships between MPD and taxa richness? And with forest loss? Plot these relationships to visualize them and use the `cor.test()` function to validate your observations.

Null modeling

Because of the expected relationship between MPD and species richness, we have to perform null models in a similar fashion to what we've done for functional diversity indices. Because, as with functional diversity, we want to keep null sites with same total biomass and same total biomass per species as observed sites, we can perform a "swap" null model. We will use a null model that shuffle the names of the species at the tip of the phylogenetic tree.

Fortunately, compared to functional diversity, the null models are all integrated in the `ses.mpd()` function in the `picante` package. The null model we'll use is the `"taxa.labels"` one. **Caution**: null models can be computationally challenging; for the sake of the example we'll do only 99 iterations but as for functional diversity a version of the null models with 999 iterations is saved in the data folder.

```
# Set random seed for repeatability of analysis
set.seed(20210705)

# Compute null permutation of MPD
ses_mpd = picante::ses.mpd(
  sub_phylo_com, cophen_dist, null.model = "taxa.labels",
  abundance.weighted = TRUE, runs = 99
)
head(ses_mpd)
```

The function `ses.mpd()` computes many values. You can get the detail by looking at the help of the functions with `?picante::ses.mpd` in the `Value` section.

We'll now load the version with 999 iterations.

```
ses_mpd_999 = readRDS("data/null_mpd_999.Rds")
```

Questions for you

- **Q25**: Explain what does the column `mpd.obs.z` means? How does this compare with the SES values we computed for functional diversity indices?
- **Q26**: How does the standardized value relates with taxa richness?
- **Q27**: What are the relationships between MPD values considering null models and forest loss? Visualize the relationships with the `plot()` function, validate your observations with the `cor.test()` function.

Mapping phylogenetic diversity

In order to see if there is a geographical pattern in phylogenetic diversity we can plot maps of MPD.

```
ses_mpd_999$plot.code = rownames(ses_mpd_999)

ggplot() +
  geom_sf(
    data = merge(subset(plot_sf, block != "og"), ses_mpd_999, by = "plot.code"),
    aes(color = mpd.obs.z)
  ) +
  scale_color_distiller(type = "div", palette = "RdYlBu",
                        name =  "SES of MPD") +
  coord_sf(crs = sf::st_crs(3376), xlim = c(875000, 890000),
           ylim = c(518500, 531000)) +
  labs(title = "Map of all plots but block 'og'") +
```

```
  ggspatial::annotation_scale() +
  ggspatial::annotation_north_arrow(location = "br") +
  theme_gray()
```

By eye at least, the pattern doesn't seem obvious on the map. And the observed SES values seems to vary widely within each forest block.

Comparing facets

One burning question in the scientific literature and that is quite debated still is the relationship between taxonomic, functional, and phylogenetic diversity (Pavoine et al. 2013).

We can leverage on the computation we have to test the relationships between all facets of diversity. /!\ **NOTE**: Because we had trouble with the phylogenetic tree, we're not strictly comparing the same subset of data, we're going to compare them anyway for the sake of the example. The proper way would be to subset the similar sets of species and recompute functional diversity.

```
# Combine taxonomic, functional, and phylogenetic diversity
all_diversity = merge(
  plot_data[, c("plot.code", "ntaxa")],
  merge(
    ses_fd, ses_mpd_999[, -1], by = "plot.code"
  )
)

# Comparison of observed values
pairs(all_diversity[, c("ntaxa", "FRic", "Q", "FEve", "mpd.obs")],
      upper.panel = panel.cor)

# Comparison of SESs
pairs(all_diversity[, c("ntaxa", "ses_FRic", "ses_Q", "ses_FEve", "mpd.obs.z")],
      upper.panel = panel.cor)
```

Question for you

- **Q28**: How are related are observed values of functional diversity and phylogenetic diversity? What about the SESs?

Modelling the effect of logging

Finally! We now have all we need to properly build a statistical model of the relationship between diversity facets and the different co-variables. We'll be building linear models with `lm()` with environmental variables and possible co-variables that may confound the effect of logging.

We first combine the diversity metrics to the environmental co-variables:

```
plot_div_env = merge(
  all_diversity,
  plot_data[, c(1, 6:21)],
  by = "plot.code"
)

dim(plot_div_env)
head(plot_div_env)
```

Single-predictor models

Then we can build models of disturbance variables on diversity metrics. Let's build individual models with the main disturbance variables: local forest loss `forestloss17`, primary road density `roaddensprim`, and distance to primary roads `roaddistprim`.

First a model on taxa richness:

```
mod_taxa_loss = lm(ntaxa ~ forestloss17, data = plot_div_env)

mod_taxa_loss
summary(mod_taxa_loss)
```

We can plot the regression line from the model with the data with the following:

```
par(mfrow = c(1, 1))
plot(mod_taxa_loss$model$forestloss17, mod_taxa_loss$model$ntaxa,
     xlab = "Forest Loss (%)", ylab = "Taxa Richness")
abline(coef = coef(mod_taxa_loss), col = "darkred", lwd = 1)
```

Questions for you

- **Q29**: How would you qualify the effect of forest loss on the taxa richness?
- **Q30**: With the same formula build similar models with the other predictors `roaddensprim` and `roaddistprim`. How do they compare with forest loss?

We can now build similar models for both functional and phylogenetic diversity. Because we do not want to consider the potential confounding factor of taxa richness we can consider directly the SES values we carefully built with our null models.

```
mod_fd_loss = lm(ses_Q ~ forestloss17, data = plot_div_env)
mod_pd_loss = lm(mpd.obs.z ~ forestloss17, data = plot_div_env)
```

Multi-predictors models

Because of the many possible confounding variables (different local environmental conditions, difference in vegetation types) we should build a model with many more predictors.

Let's build a complete model with all environmental predictors:

```
mod_taxa_all = lm(
  ntaxa ~ elevation + forestloss17 + forestloss562 + roaddenssec +
    roaddistprim + soilPC1 + soilPC2,
  data = plot_div_env
)
mod_fd_all = lm(
  ses_Q ~ elevation + forestloss17 + forestloss562 + roaddenssec +
    roaddistprim + soilPC1 + soilPC2,
  data = plot_div_env
)
mod_pd_all = lm(
  mpd.obs.z ~ elevation + forestloss17 + forestloss562 + roaddenssec +
    roaddistprim + soilPC1 + soilPC2,
  data = plot_div_env
)
```

Question for you

- **Q31**: What can you say about the effect of the disturbances on the different diversity metrics? What are the explanatory power of our models?

To explain the issues we have with the models we can look at the model diagnostics:

```
par(mfrow = c(2, 2))
plot(mod_taxa_all)
# Or even better
performance::check_model(mod_taxa_all)
```

To go further (but it's beyond the scope of this tutorial) we could follow the tracks of the paper:

1. Build a generalized linear model (GLM) when working with taxa richness as it is a count data.
2. Leverage the fact that we have a block design in our sampling data and use mixed-models to account for that (the observations are not fully independent of one another and are structured in groups).
3. Account for some non-linear effects of some predictors (forest loss have a strong quadratic component).
4. Perform model averaging or model selection to prune the model to the most important predictors.

References

Döbert, Timm F., Bruce L. Webber, John B. Sugau, Katharine J. M. Dickinson, and Raphael K. Didham. 2017. "Logging Increases the Functional and Phylogenetic Dispersion of Understorey Plant Communities in Tropical Lowland Rain Forest." *Journal of Ecology* 105 (5): 1235–45. https://doi.org/10.1111/1365-2745.12794.

———. 2018. "Data from: Logging Increases the Functional and Phylogenetic Dispersion of Understorey Plant Communities in Tropical Lowland Rainforest." Dryad. https://doi.org/10.5061/DRYAD.F77P7.

Laliberté, Etienne, and Pierre Legendre. 2010. "A Distance-Based Framework for Measuring Functional Diversity from Multiple Traits." *Ecology* 91 (1): 299–305. https://doi.org/10.1890/08-2244.1.

Lavorel, S., and E. Garnier. 2002. "Predicting Changes in Community Composition and Ecosystem Functioning from Plant Traits: Revisiting the Holy Grail." *Functional Ecology* 16 (5): 545–56. https://doi.org/10.1046/j.1365-2435.2002.00664.x.

Pavoine, Sandrine, Amandine Gasc, Michael B. Bonsall, and Norman W. H. Mason. 2013. "Correlations Between Phylogenetic and Functional Diversity: Mathematical Artefacts or True Ecological and Evolutionary Processes?" *Journal of Vegetation Science* 24 (5): 781–93. https://doi.org/10.1111/jvs.12051.

Webb, null. 2000. "Exploring the Phylogenetic Structure of Ecological Communities: An Example for Rain Forest Trees." *The American Naturalist* 156 (2): 145–55. https://doi.org/10.1086/303378.