

Reorganisation following disturbance: multi trait-based methods in R

Part 2: Building the multi-trait space to measure trait-based diversity

Estimated time: 120 minutes

In Part 1, you made predictions and estimated how the abundance of two fish species varied before and after the bleaching event. You should by now have looked at your habitat specific dataset and chosen two species whose trends (pre and post bleaching) you will report on in your poster.

Each species plays their own unique role in terms of ecosystem function, and traits can be used as proxies for the functional role species play in the ecosystem. For example, diet can influence energy flow and provide information on ecosystem processes like herbivory, while body size can provide information on movement, home range and energetic needs of an individual. Because we are interested in the community as a whole, we are going to move from thinking about species individually and start thinking about the trait-based diversity of the community. There are three steps to doing this:

1. 'Functionally' describe your species (assign traits)
2. Measure how different species are from each other in relation to these traits
3. Visualize the functional trait space using multivariate statistics

Today we will look at the functional traits of the fish species surveyed at Lizard Island. We will look at a variety of different traits to understand how collectively the whole fish community is organized.

Last week, you researched the biology and ecology of two species on FishBase, but now we need functional descriptions for all fishes in the dataset. We will use an already collated species-specific trait dataset (Richardson et al. 2018), collated from FishBase and existing published literature.

Because we already have the functional trait descriptions, Step 1 is done, we will jump straight to Step 2 and measure how different these species are with regard to traits. We will then reduce down these multiple traits differences using multivariate statistics into a single dimension, and use this to calculate trait-based diversity (functional richness). We will walk through what this data reduction actually means step by step.

Measuring functional trait dissimilarity between species and creating the distance matrix

Background on R data structures

Everything in your R environment is an object. R can store data in a variety of different data structure object types. A matrix is one of them. Technically, a matrix is actually a vector, a vector being another R data structure type. Vectors are described as the little engines of R – they do all the work. Much like zooxanthellae in coral – one wouldn't exist without the other. The thing to know about vectors is that they must contain data of the same data type e.g. numeric data or character data. This is in contrast to a dataframe which can store different data types.

A matrix then is a rectangular array of numbers, but they are stored with two extra bits of information, the number of rows and the number of columns. This means that while a matrix and a dataframe can look the same – they will behave differently.

We will create a dataframe in R later – but for future programming in R, it can be good to know that a matrix behaves differently to a dataframe, which is why sometimes we get unexpected results or errors if you treat a matrix the same as a dataframe. You can easily find out what type of object you are using in R with the class function e.g. `class(name of r object)`.

If you want to learn more about the different data types and structure types in R there is an excellent online tutorial by [Data Carpentry] (<https://swcarpentry.github.io/r-novice-inflammation/13-supp-data-structures.html>).

Calculating dissimilarity scores

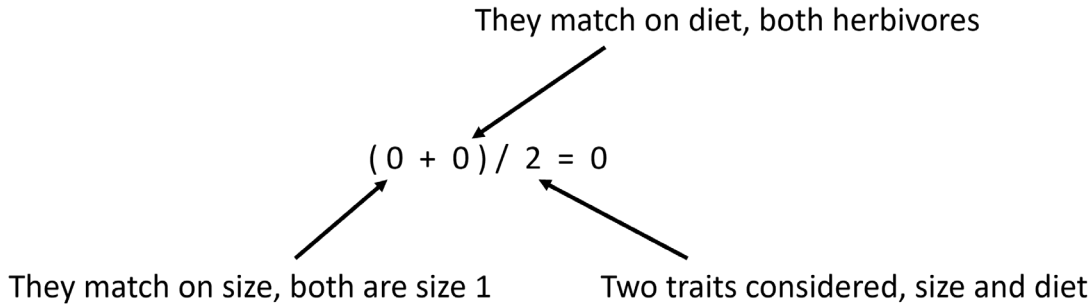
Consider the following data table for a fish community of four species (A-D) and two traits (Size and Diet). Species A and B are exactly the same for Size and Diet (they are similar, not dissimilar), while species C is the same for size but differs on Diet and so on.

Species	Size	Diet
A	1	Herbivore
B	1	Herbivore
C	1	Piscivore
D	2	Piscivore

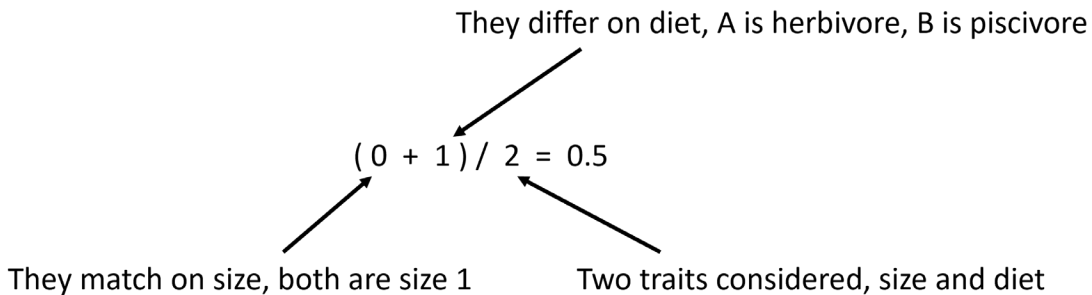
Using this information, you can calculate a pairwise dissimilarity score by assigning a 0 when species match on a trait and 1 when they differ.

The average dissimilarity score for each species pairwise comparison is then calculated, by summing the scores and dividing by the number of traits considered.

TIEE



The dissimilarity score for Species A and B



The dissimilarity score for Species A and C

Q1 Based on the trait table above, fill out the following table, calculating the average dissimilarity scores for each species pair combination.

	A	B	C	D
A	x			
B		x		
C			x	
D				x

The closer the species pairwise score is to 0 the more similar the species are, with 0 being an exact match, and 1 being the most dissimilar they can be for the traits considered.

Next, we will see how a matrix of dissimilarity scores can be calculated in R.

Getting ready to get started

Download the datafiles for Part 2, then get ready to work in RStudio.

You will now put what was covered in the preparation document and during Day 1 into practice (refer to those docs if you are unsure) to get ready to analyze the data. Do the following:

- Open a new script and call it `Day_2_code.R`
- Based on what you learned last week, type the code into your new R script to load the following libraries (`vegan`, `ade4`, `cluster`, `ggplot2`, `mFD`) using the `library()` function then pass it to the R console.

```
• library(vegan)  
• ## Loading required package: permute  
• ## Loading required package: lattice  
• ## This is vegan 2.6-4  
• library(ade4)  
library(ggplot2)  
library(mFD)
```

Creating the trait dissimilarity scores

To make sure you understand what is going on, we will first create a trait matrix using a dummy dataset. Type the following lines into your R script and pass to the R console.

```
• tinytraits <- read.csv("tiny_trait_matrix.csv", row.names = 1)  
## you may get a warning message - that is ok
```

We need to understand what types of trait data we are using and tell R. This can be done directly in R or by loading a `.csv` file with the information needed. Here, we have made a dataframe categorizing our traits called `tr_cat_1.csv`. It can either be constructed directly through R or you can load a `.csv` file as we will do here. The first column in the dataframe lists the trait names and the second column categorizes what type of trait they are. The categories can include:

- N: nominal trait (factor variable)
- O: ordinal traits (ordered variable which is a specific factor with ordered levels. For example, 'size' as its levels are made up of the categories "1", "2", "3", "4", "5", "6" and "7" which are ordered, i.e., level "1" is smaller than level "2" which is smaller than level "3" etc.)
- C: circular traits (integer values, like *months* within a year)

TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

- Q: quantitative traits (numeric values, such as the amount of sugar in a fruit)
- F: fuzzy traits (described with several values defined in several columns in the `tr_cat_1` data frame. For example, there can be multiple uses of fruits (i.e., pastry, jam, eaten raw): in this example each column would represent a type of “use” and values in each column would refer to the percentage for which the fruit is used for this specific purpose)

In this `tr_cat_1.csv` example, we only have ordinal (size) and nominal (diet) traits. Load the `tr_cat_1.csv` dataframe by typing the following lines into your R script and passing to the R console:

```
• tr_cat <- read.csv("tr_cat_1.csv")
tr_cat

• ## trait_name trait_type
## 1 Size 0
## 2 Diet N
```

In this next chunk of code we will generate a pairwise dissimilarity matrix of Gower distances using the `funct.dist` function in the `mFD` package. This matrix gathers distances between species based on their trait’s similarity and difference. You don’t need to worry too much about the details in this code for now - it is essentially doing the process that you just did by hand above.

```
• # Tell R that traits are factors, and their order if needed:
tinytraits[, "Diet"] <- as.factor(tinytraits[, "Diet"])
tinytraits[, "Size"] <- factor(tinytraits[, "Size"], levels = c("1", "2"), ordered = TRUE)

# Compute functional dissimilarities:
traitdis <- mFD::funct.dist(
  sp_tr = tinytraits,
  tr_cat = tr_cat,
  metric = "gower",
  stop_if_NA = TRUE)

## You may get a warning message - that is ok

## Put the dissimilarity matrix (distance object called ‘traitdis’) into a dataframe to make it easier to look at:
```

```
mFD:::dist.to.df(list(traitdis = traitdis))

•   ## Registered S3 method overwritten by 'dendextend':
##   method      from
##   rev.hclust  vegan

•   ##   x1 x2 traitdis
## 1  A  B      0.0
## 2  A  C      0.5
## 3  A  D      1.0
## 4  B  C      0.5
## 5  B  D      1.0
## 6  C  D      0.5
```

We can see that the trait-based dissimilarity distance between species A and species B is equal to 0 as they are the same, i.e. they both have the same combination of traits: Size = 1 and Diet = Herbivore. Conversely, the trait-based dissimilarity distance between species A and species D is equal to one as they are completely different, i.e. they differ across the two traits: species A is Size = 1 and Diet = Herbivore whereas species D is Size = 2 and Diet = Piscivore.

This was a simple example with only two traits and four species to show you how the process of creating a dissimilarity matrix works. Now we will create a trait dissimilarity matrix for the Lizard Island fish species. We will start by reading in the dataset of species traits. This dataframe is called 'traits' and is saved as a .csv file ('traits.csv').

```
•   traits <- read.csv("traits.csv", row.names = 1)
```

Using the functions (head, dim, levels, summary) you learnt in Part 1 – get to know the traits dataframe that you have just loaded in.

Q2 What different trait types are there, and what are the levels within each?

Inspect the trait data. Do you understand what each trait column refers to? Do you understand each of the values a trait can take? You will have to read the Methods section of Richardson et al. (2018) and take a look at the supplemental materials to find out. These can be accessed free of charge and without subscription here:

<https://doi.org/10.25903/5b57c26b0beb7> (see Chapter 4).

We want to make sure that R is treating each of our trait columns as factors. Depending on your R default settings, the trait columns (Diet, Activity, Size, Mobility, Schooling, Position) may have been imported as 'character' data. We will use the below code to make sure R is treating each of the trait columns as factors, and that it knows which are

TIEE

'ordinal' categories. Note in the code below for Size, Mobility, Schooling, and Position the factor levels are listed in the code and then 'ordered = TRUE' tells R that it should treat these as the category orders (e.g. there is an inherent order from Benthic, to Benthic-Pelagic, to Pelagic).

```
• # Tell R which traits are factors and ordered factors as needed:
traits[, "Diet"] <- as.factor(traits[, "Diet"])

traits[, "Activity"] <- as.factor(traits[, "Activity"])

traits[, "Size"] <- factor(traits[, "Size"], levels = c("1", "2", "3", "4", "5", "6", "7"), ordered = TRUE)

traits[, "Mobility"] <- factor(traits[, "Mobility"], levels = c("Sedentary", "Mobile within reef", "Mobile across reefs"), ordered = TRUE)

traits[, "Schooling"] <- factor(traits[, "Schooling"], levels = c("Solitary", "Pairing", "SmallG", "MedG", "LargeG"), ordered = TRUE)

traits[, "Position"] <- factor(traits[, "Position"], levels = c("Benthic", "Benthic-Pelagic", "Pelagic"), ordered = TRUE)
```

Using this code above, and the functions you learnt in Part 1 (e.g. str(traits), levels(traits\$Diet etc.), make sure you know what each trait column refers to and what categories are within each trait.

Discuss this trait data with your peers and ask the demonstrators before moving on. Write down some notes in the table below, as they will come in useful later when interpreting the trait space.

Trait	Meaning	Factor levels	Notes of definitions
Size			
Diet			
Mobility			
Activity	Time when active	Both, diurnal, nocturnal	Diurnal: during the day, nocturnal: during the night
Schooling			
Position			

Transcribe the trait data for the species listed in the table below. We have deliberately selected a few species that use the reef in a very different way. Think about the different role these fishes play on the reef and how they may be affected by habitat change caused by bleaching.

TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

You should also look at the trait information for the two species you have selected to focus on in your poster.

Species	Size	Diet	Mobility	Activity	Schooling	Position
<i>Lutjanus bohar</i>						
<i>Chaetodon kleinii</i>						
Species a						
Species b						

The next step is to calculate the pairwise dissimilarity scores for the fish species at Lizard Island – for all of the traits considered for all the species surveyed. The steps are exactly the same as you did above by hand, but we will get R to do it, as it would take us a very long time to process so much data by hand (and would be prone to human error!)

Load the `tr_cat_2.csv` dataframe which contains the trait categories. We will then use the same code as before but apply it to the full trait dataframe called `traits`.

```
• # Load the traits category dataframe:
tr_cat <- read.csv("tr_cat_2.csv")

# Compute the dissimilarity scores:
traitdis <- mFD::funct.dist(
  sp_tr      = traits,
  tr_cat     = tr_cat,
  metric     = "gower",
  stop_if_NA = FALSE)

# Display the dissimilarity matrix as a dataframe (easier to read):
traitdis_df <- mFD::dist.to.df(list(trait_dis = traitdis))
head(traitdis_df)

• ##          x1          x2 trait_dis
## 1 Abundefdof.bengalensis Abundefdof.sexfasciatus 0.0000000
## 2 Abundefdof.bengalensis Abundefdof.whitleyi 0.0000000
## 3 Abundefdof.bengalensis Acanthochromis.polyacanthus 0.0000000
## 4 Abundefdof.bengalensis Acanthurus.blochii 0.3833333
## 5 Abundefdof.bengalensis Acanthurus.dussumieri 0.4333333
## 6 Abundefdof.bengalensis Acanthurus.grammoptilus 0.3833333
```


This has created a large R object called `traitdis`. That this is large is unsurprising, because you have run pairwise comparisons for 189 species!

The formula to calculate the total number of pairwise combinations is: $n(n-1)/2$

N.B. The warning message we get here says that some trait-based distances between species pairs are equal to 0 (that is to say, that at least two species have the exact same combination of traits). If this relates to only a few species pairs (as in our case), it is ok to continue as planned without gathering species into Functional Entities as the warning suggests.

Q3 How many pairwise combinations do you have with this dataset?

Here you have $n = 189$ species. Use the formula and the R console as a calculator to calculate the number of pairwise combinations, and so dissimilarity scores you should have with this species list.

You can check whether your answer is correct by looking at the length of the `traitdis` object using the length function – it is the same?

In the environment window on the top right you can see the ‘`traitdis`’ object contains 17766 elements, which is each of these pairwise comparisons. At the start of this session, you calculated some pairwise comparisons by hand, you can see how in reality you would not be calculating these by hand as it would take far too long!

Now we will look at the scores: `head(traitdis)`

The first three dissimilarity scores are zero – this means the first species in the traits dataframe, *Abudefduf bengalensis*, is an exact match for all 6 traits compared to the first three species in the traits dataframe (*Abudefduf sexfasciatus*, *Abudefduf whitleyi*, and *Acanthochromis polyacanthus*).

Double check this is the case by looking at first few rows of the traits dataframe `head(traits)`.

Visualising the species dissimilarity matrix

A principal coordinates analysis (PCoA) is a method to visualize dissimilarities in multidimensional data. You have already generated the dissimilarity matrix (pair by pair comparisons), the next step is to visualize the structure of this distance data. Distance being how close together or far apart the species are in multi-dimensional trait space. It is complex data because we have more than 2 traits and every combination of pairs of species.

Understanding the PCoA analysis

The advantage of multivariate methods is that it allows us to visualize individual or group differences, considering multiple factors at once.

NOTE: for more background on these multivariate methods – in your own time watch the videos listed at the end of the Part 1 practical handout and revisit the introductory lecture.

The function to run a principal coordinates analysis is `dudi.pco`. Below we have provided the code that will run a PCoA and return the visualization of the trait space.

The aim here is for you to inspect the output and learn how to interpret these multivariate type plots, you don't have to understand the underlying nuts and bolts and code, but you can use the code comments to know what is happening in each step if you wish.

We will now visualize the species dissimilarity scores for each species pair combination using the trait distance matrix.

The first step is to perform the PCoA on the dissimilarity matrix. This step calculates the axes (eigenvectors and eigenvalues). When you run the below code, R then asks how many synthetic axes you want to reduce all these data down to. We will use 4 based on recommendation in Maire et al. 2015: <https://doi.org/10.1111/geb.12299>).

Run the below code, then type the number 4 into the console under where it says "Select the number of axes:" and hit return.

```
• trait_pco_corr <- ade4::dudi.pco(traitdis, tol = 1e-07)
```

One of the PCoA outputs is the set of coordinates for the 4 synthetic axes. We take these coordinates and use them to create the plot to visualize where each species lies in the trait space.

TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

```
• coordinates <- as.data.frame(trait_pco_corr$li)
summary(coordinates)
str(coordinates)
```

Next, we will plot the PCoA coordinates on the first two axes to visualise the trait space with the traits positioned as vectors.

```
• ## the envfit function can fit factors onto more than 2 axes in multidimensional space, but we will just plot the first two axes to take a quick look since most of the variation among traits will be described by axis 1 and 2:
pc2d <- coordinates[,1:2]
vectors <- vegan::envfit(pc2d, traits, perm=1000, na.rm=TRUE)
```

This next bit of code to make the plot must be run all at the same time, not line by line, for the addition of the vectors labels to be added. Select all of the code together, and press Ctrl + Enter to run it.

```
• {plot(x=coordinates[,1],
      y=coordinates[,2],
      xlab="PCoA1",
      ylab="PCoA2", xlim=c(-0.5, 0.4), ylim=c(-0.4, 0.6), pch=20)
  plot(vectors, p.max=0.05, cex=0.7)}
```

Ok! This is functional trait space – reduced down to 4 axes.

Computing functional space using PCoA and assessing it's quality

To build functional space (using PCoA) which the functional indices will be calculated from, you can also use the *mFD* package. For this tutorial, we have walked you through what PCoA is and how to make one using this reef fish trait data. But we can also use the *mFD* package which allows us to quickly build and plot functional space based on PCoA analysis.

The first step is to build multiple different functional spaces, each with a different number of synthetic axes, and then chose the one with the best “quality” *i.e.* the one with the lowest deviation between trait-based distances and distances in the functional space. This step is done using the *quality.fspaces* function of *mFD* as follows:

```
• # Compute functional spaces:
fspaces_quality <- mFD::quality.fspaces(
  sp_dist      = traitdis,
  maxdim_pcoa  = 10,
  deviation_weighting = "absolute",
  fdist_scaling = FALSE,
  fdendro      = NULL)

# Look at their quality
round(fspaces_quality$"quality_fspaces", 3)

• ##          mad
## pcoa_1d  0.187
## pcoa_2d  0.106
## pcoa_3d  0.065
## pcoa_4d  0.050
## pcoa_5d  0.046
## pcoa_6d  0.046
## pcoa_7d  0.050
## pcoa_8d  0.056
## pcoa_9d  0.062
## pcoa_10d 0.068
```

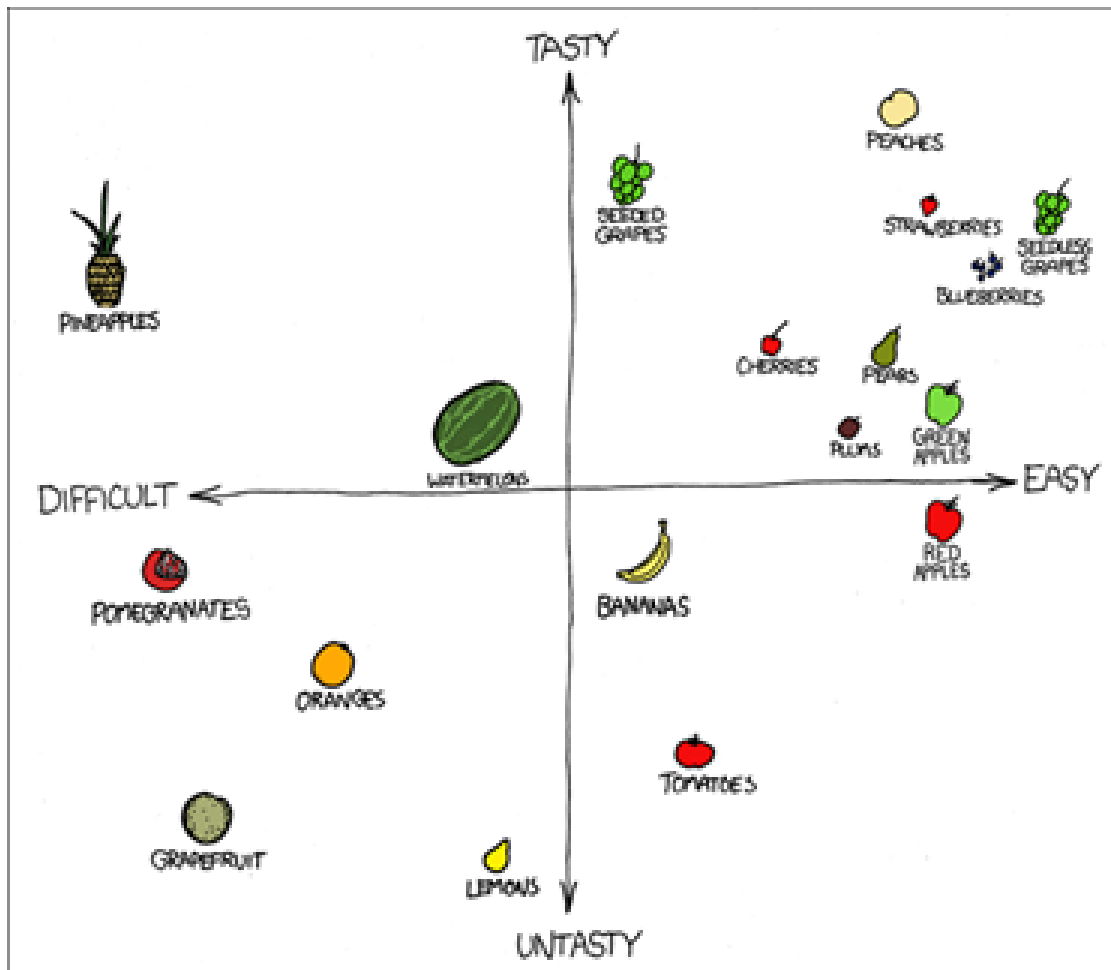
Here, we computed up to ten functional synthetic axes ('maxdim_pcoa' argument) and then asked R to build ten functional spaces using between the range of up to ten synthetic axes. Based on the mean absolute deviation index ("mad") ('deviation_weighting' argument) which describes absolute differences between distances based on the dissimilarity matrix and the distance in the functional space, we can see that the 5D functional space is the best (same mad as 6D functional space so we chose the functional space with the lowest number of synthetic axis between the

two). As the “mad” index reflects the absolute difference between distances between species pairs based on the dissimilarity matrix (based on traits values) and distances between species pairs in the functional space built through a PCoA, the smaller the “mad”, the smaller the deviation between the two distance types. Therefore, the more the functional space distances accurately reflect trait-based distances. So, from now on **the rest of the tutorial will use the first five synthetic axes**. Use the chunk of code above and run this analysis yourself.

We then retrieve the coordinates of each species along the five synthetic axes:

- ```
coordinates_mFD <- fspaces_quality$"details_fspaces" $"sp_pc_coord"
```

The next step is to inspect the trait space to look for trends in traits along the PCoA ‘functional’ axes. Before you do that – it might help to look at this more intuitive example.



A two dimensional plot showing the tastiness and ease of eating of different fruits. Source: XKCD.com

Here, you have a plot summarizing information about fruit. You can see that each fruit has been positioned on two axes (taste and ease of eating) – and they indicate where they fall in relation to those axes. Moving from right to left, fruit go from easy to difficult to eat, and moving up down from tasty to untasty. Some fruit are tasty and easy to eat (peach, grapes), some are untasty and difficult (grapefruit), while some middle on both (banana, watermelon). Fruits that are close together are more similar e.g. blueberries, strawberries and grapes are all really easy to eat and pretty tasty.

Using the `traits.faxes.cor` function helps you to test for correlations between traits and functional axes to look for trends in how the different traits of all species group along the different axes. You can use it as follows:

# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

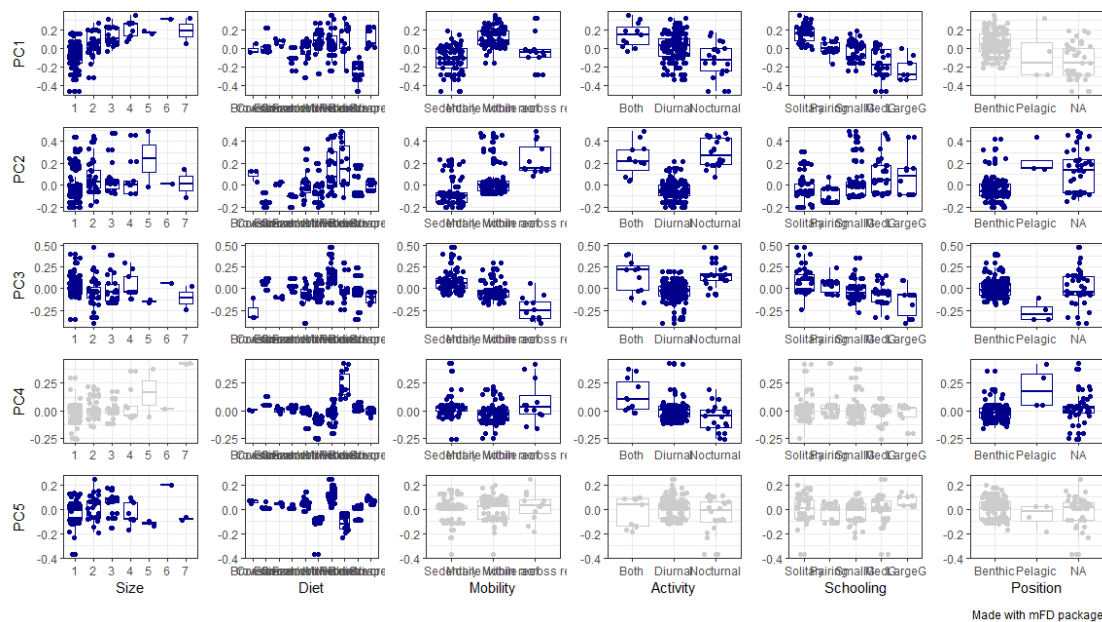
```
• # Compute correlations:
tr_faxes <- mFD::traits.faxes.cor(
 sp_tr = traits,
 sp_faxes_coord = coordinates_mFD[, c("PC1", "PC2", "PC3", "PC4", "PC5")],
 plot = TRUE,
 stop_if_NA = FALSE)

Find those traits that have a statistically significant effect (p-value < 0
.05) on their position along each of the axes:
tr_faxes$"tr_faxes_stat"[which(tr_faxes$"tr_faxes_stat$"p.value" < 0.05),]

• ## trait axis test stat value p.value
1 Size PC1 Kruskal-Wallis eta2 0.331 0.0000
2 Size PC2 Kruskal-Wallis eta2 0.215 0.0000
3 Size PC3 Kruskal-Wallis eta2 0.068 0.0052
5 Size PC5 Kruskal-Wallis eta2 0.146 0.0000
6 Diet PC1 Kruskal-Wallis eta2 0.525 0.0000
7 Diet PC2 Kruskal-Wallis eta2 0.440 0.0000
8 Diet PC3 Kruskal-Wallis eta2 0.462 0.0000
9 Diet PC4 Kruskal-Wallis eta2 0.760 0.0000
10 Diet PC5 Kruskal-Wallis eta2 0.853 0.0000
11 Mobility PC1 Kruskal-Wallis eta2 0.512 0.0000
12 Mobility PC2 Kruskal-Wallis eta2 0.421 0.0000
13 Mobility PC3 Kruskal-Wallis eta2 0.372 0.0000
14 Mobility PC4 Kruskal-Wallis eta2 0.204 0.0000
16 Activity PC1 Kruskal-Wallis eta2 0.083 0.0002
17 Activity PC2 Kruskal-Wallis eta2 0.386 0.0000
18 Activity PC3 Kruskal-Wallis eta2 0.175 0.0000
19 Activity PC4 Kruskal-Wallis eta2 0.072 0.0004
21 Schooling PC1 Kruskal-Wallis eta2 0.552 0.0000
22 Schooling PC2 Kruskal-Wallis eta2 0.177 0.0000
23 Schooling PC3 Kruskal-Wallis eta2 0.162 0.0000
27 Position PC2 Kruskal-Wallis eta2 0.048 0.0016
28 Position PC3 Kruskal-Wallis eta2 0.052 0.0011
29 Position PC4 Kruskal-Wallis eta2 0.048 0.0016

• # Plot correlations:
tr_faxes$"tr_faxes_plot"
```

Relation between traits and PCoA axes



Inspect that plot and look for general trends within the trait space. The traits with a significant effect on their position on the axes are coloured in blue (non-significant effect traits are in grey). For example, on the right of the first synthetic axis (PC1) are 'Solitary' species whereas on the left of PC1 axis are 'Large Group' schooling species.

Lastly, plot the functional space as follows with each subplot being a different combination of the 5 synthetic axes. Run the following command to plot the functional space *i.e.* the position of species along the first five synthetic axes:

```

• functional_space <- mFD::funct.space.plot(
 sp_faxes_coord = coordinates_mFD[, c("PC1", "PC2", "PC3", "PC4", "PC5")],
 faxes = NULL,
 name_file = NULL,
 faxes_nm = NULL,
 range_faxes = c(NA, NA),
 color_bg = "grey95",
 color_pool = "darkgreen",
 fill_pool = "white",
 shape_pool = 21,
 size_pool = 1,
 plot_ch = TRUE,
 color_ch = "black",
 fill_ch = "white",
 alpha_ch = 0.5,

```



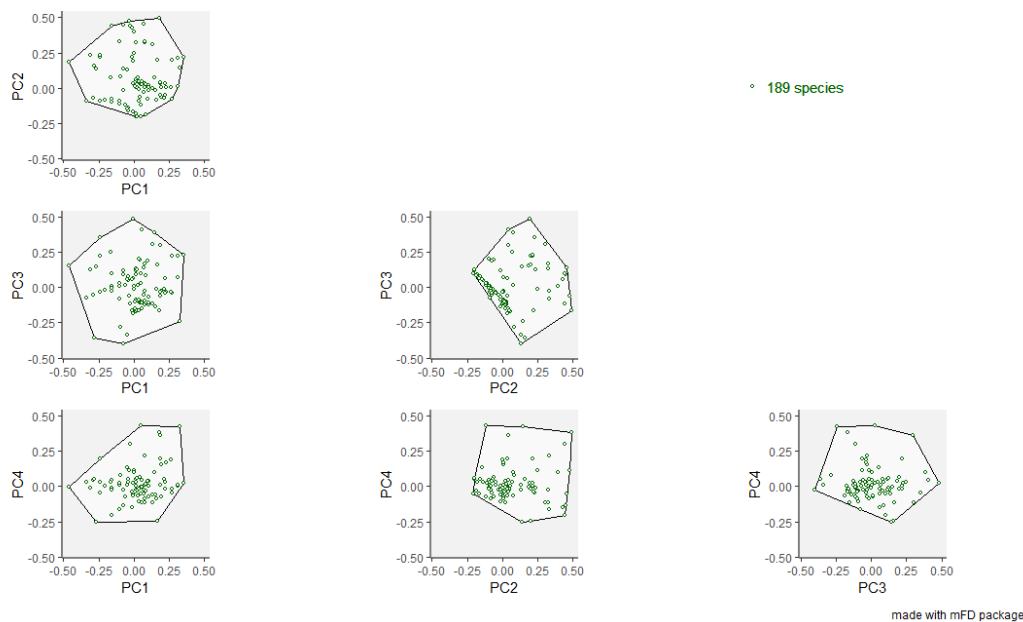
# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

```
plot_vertices = FALSE,
color_vert = "blueviolet",
fill_vert = "blueviolet",
shape_vert = 23,
size_vert = 1,
plot_sp_nm = NULL,
nm_size = 3,
nm_color = "black",
nm_fontface = "plain",
check_input = TRUE)
```

```
Plot the graph with all pairs of axes:
functional_space$patchwork
```

Position of species along pairs of functional axes



**Q4 Write a short (3-4 sentence) summary to describe how the fish species have clustered in this trait space, i.e. how do some of these fish traits vary across the trait space?**

*TIP: for ideas on how you can do this, see the Richardson et al. (2018) paper, section 3.2 'Fish assemblage structure' for an example of how they described their trait space (note: theirs included other habitats and fish species so will be different to yours). This can be accessed free of charge and without subscription here:*

<https://doi.org/10.25903/5b57c26b0beb7> (Chapter 4).

You will now export a plot of your trait space as you can use it in your poster. You can export this mFD version of trait space (and chose one set of plotted axes to illustrate). However, at the time of writing this, the mFD package didn't yet allow us to overlay the trait vectors as we did with the plot made with `ade4` (`dudi.pco`) and `vegan`(`envfit`). So you can alternatively export the plot you created in L319-353. You can use the options in the bottom right window under the Plots tab to Export your image. We recommend you export it as a pdf.

If you use the earlier version of the trait space for your poster, you can use either the plot with the trait vectors overlaid (blue text), or you can replot and not include the trait vectors, so that you can annotate yourself. You can do this by rerunning the plot code to recreate the plot, but dropping the `plot(vectors...)` line. As we saw in the lecture, there are several methods to annotate these multivariate plots in a way that graphically illustrates the key message you want to communicate about the trait space. For your poster, if you wish you can come up with your own way of visualizing the summary sentences you drafted above. To do this, export the trait visualization plot without the trait vectors overlaid (no blue text) and then edit and add visuals to it using a program like PowerPoint, or some other figure editing/creating tool.

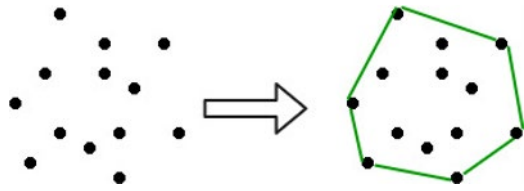
## Estimating and summarizing functional richness

*NOTE: for more information on estimating functional richness read Mouillot et al. (2013) in your own time (for details see Tasks at the end of Part 1 practical handout).*

Mouillot et al. (2013) recommended several ways to assess change in functional structure of ecological communities after disturbance, using the two-dimensional functional space we have just created from the PCoA. We will use the estimate of functional richness, which is the size of the convex hull that includes all species in the community. Fishes are plotted on the synthetic axes (PCoA1 and PCoA2) from the PCoA and the size of the functional trait space is used to estimate functional richness – the larger the area the more functionally rich the community is.

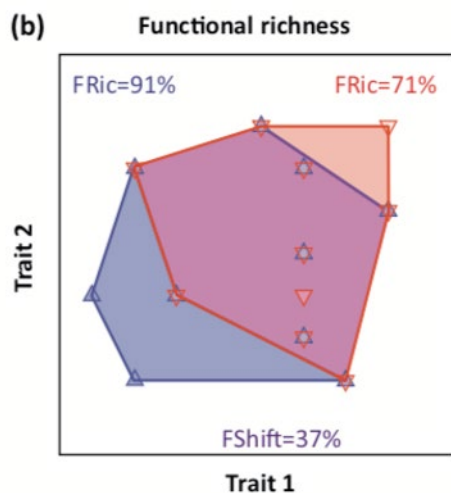
# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024



The convex hull is the smallest polygon that contains all the points.

A change in the size of the convex hull before and after disturbance can indicate whether the functional richness of the community has changed.



Functional richness is measured by the functional space filled by species in the community (the convex hull). Source: Mouillot et al. 2013

Changes in the fish assemblage (i.e. the loss of habitat specialists, like corallivores) can alter functional richness. In this example, functional richness is less after the disturbance (red) as compared to before (blue). Taken from Mouillot et al. (2013).

You can compute the functional richness estimates before and after disturbance for your habitat type using the `alpha.fd.multidim` function. The code below is quite long and complex, you do not need to understand all of the code here, but to summarise what it is doing:

- It takes the fish abundance data set we have been using, and subsets out the branching Porites (if you are working with a different habitat type, e.g. Low\_coral\_cover, Mixed\_coral, Soft\_coral, you would need to edit this code to subset and name the columns accordingly).

# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

- It converts this to a table showing the mean abundance of each species pre- and post-bleaching (the `asb_sp` object).
- It computes the functional richness index using the `alpha.fd.multidim` function.

• *# This code creates a matrix summarizing the reef fish species belonging to an assemblage of fishes that are observed in a specific habitat, here showing one habitat type (Branching porites) for the two survey periods (pre- and post-bleaching event):*

```
Read in the data:
```

```
dat <- read.csv("LI_fish_abundance_pre_post_bleaching.csv")
```

```
Edit this code if you are working with a different habitat type:
```

```
data <- subset(dat, Habitat == "Branching_porites")
```

```
asb_sp <- as.data.frame(matrix(nrow = 2, ncol = ncol(data) - 4))
```

```
colnames(asb_sp) <- colnames(data[, -c(1:4)])
```

```
Edit this code if you are working with a different habitat type:
```

```
rownames(asb_sp) <- c("Branching Porites PreB", "Branching Porites PostB")
```

```
for (i in (1:ncol(asb_sp))) {
```

```
 sp_nm <- colnames(asb_sp)[i] # take first col name (i.e. first species)
 asb_sp[1, sp_nm] <- mean(data[which(data$Bleaching == "Pre"), sp_nm]) # add the pre-bleaching 'PreB' mean species abundance value
 asb_sp[2, sp_nm] <- mean(data[which(data$Bleaching == "Post"), sp_nm]) # add the post-bleaching 'PostB' mean species abundance value
}
```

```
Compute FRic index:
```

```
alpha_fd_indices <- mFD::alpha.fd.multidim(
 sp_faxes_coord = coordinates_mFD[, c("PC1", "PC2", "PC3", "PC4", "PC5")]
,
```

```
 asb_sp_w = as.matrix(asb_sp),
 ind_vect = c("fric"))
```

```
Display results:
```

```
alpha_fd_indices$"functional_diversity_indices"
```

```
sp_richn fric
Branching Porites PreB 100 0.3738774
Branching Porites PostB 100 0.4269412
```

NB. As functional richness is the proportion of functional space filled by a species assemblage, it takes into account the positions of the species in that space with the most extreme trait values (positioned at the lower or upper ends of the PCoA axes of trait space). Thus, the functional richness values can be either computed with abundance data (as we have here) or with only presence/absence data. Here, we have computed the functional richness of fish assemblages in Branching porites habitats before (pre) and after (post) the coral bleaching event, relative to the whole species pool observed across all habitats and all survey periods. This is why we get a warning message that some species are absent when we run code in line 491.

## Plotting a summary of the trait space metrics before and after bleaching

We will use the *mFD* package which uses the *ggplot2* library to visualize the estimates of functional richness. Load in your habitat specific data-set on functional richness. Within this dataframe, there is also an estimate of species richness (the number of species) and total fish abundance. You will now create graphs of both to compare how they change following disturbance.

## What is ggplot2?

You can create graphs in R using base R functions, or by using the *ggplot2* package. Packages in R can be thought of as applications on your phone. When you get a new phone, it is loaded with some standard applications, but not all of the applications that exist. You chose which applications you want to install. It's the same with R, when R is installed, it comes with some base packages that contain functions. Of the hundreds of additional packages available, you chose which you want to work with. *ggplot2* is a powerful data visualization package that is well worth getting to know as it can be used to create beautiful, professional looking plots.

For more information on *ggplot2* you can download a *ggplot2* cheat sheet:

<https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>

Below is a chunk of code to plot the functional richness estimates before and after bleaching for all combination of PCoA axis using the dataframe *FD\_BP*. Edit this code to fit your habitat types and pass to R.

```
• # Build the plot:
p <- mFD::alpha.multidim.plot(
 output_alpha_fd_multidim = alpha_fd_indices,
 plot_asb_nm = c("Branching Porites PreB", "Branching Porites P
ostB"),
 ind_nm = c("fric"),
 faxes = NULL,
```

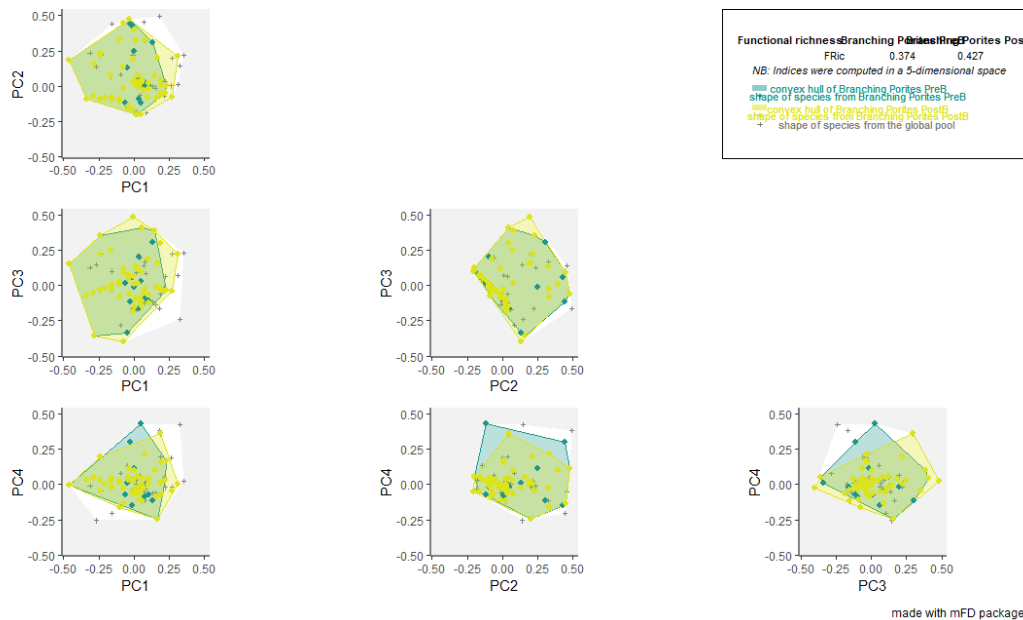
# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

```
faxes_nm = NULL,
range_faxes = c(NA, NA),
color_bg = "grey95",
shape_sp = c(pool = 3, asb1 = 21, asb2 = 21),
size_sp = c(pool = 0.7, asb1 = 1, asb2 = 1),
color_sp = c(pool = "grey50", asb1 = "#1F968BFF", asb2 = "#
DCE319FF"),
color_vert = c(pool = "grey50", asb1 = "#1F968BFF", asb2 = "#
DCE319FF"),
fill_sp = c(pool = NA, asb1 = "#1F968BFF", asb2 = "#DCE319
FF"),
fill_vert = c(pool = NA, asb1 = "#1F968BFF", asb2 = "#DCE319
FF"),
color_ch = c(pool = NA, asb1 = "#1F968BFF", asb2 = "#DCE319
FF"),
fill_ch = c(pool = "white", asb1 = "#1F968BFF", asb2 = "#D
CE319FF"),
alpha_ch = c(pool = 1, asb1 = 0.3, asb2 = 0.3),
shape_centroid_fdis = c(asb1 = 22, asb2 = 24),
shape_centroid_fdiv = c(asb1 = 22, asb2 = 24),
shape_centroid_fspe = 23,
color_centroid_fspe = "black",
size_sp_nm = 3,
color_sp_nm = "black",
plot_sp_nm = NULL,
fontface_sp_nm = "plain",
save_file = FALSE,
check_input = TRUE)

Display the plot:
p$"fric"$"patchwork"
```

Functional Richness of 'Branching Porites PreB' and 'Branching Porites PostB'



There are several ways in which you can edit your plot (<https://cmlmagneville.github.io/mFD/reference/alpha.multidim.plot.html>) – when you are happy with how it looks, export your figure for use in your poster.

So far, we have calculated functional richness per habitat type. We now need to calculate this index for each individual transect. Again, it is not important to understand all of this code, we will not be asking you to write code like this yourselves, but we have included it here so that you can see the data processing required for this type of analysis. If you read the comments (the lines that start with a hashtag #) this will tell you in plain language what each section of the code is doing - you can just copy and paste these sections of code into your script.

This code is taking the species data and calculating the functional richness FRic index for each transect - copy/paste and then run this whole section of code and you will see these FRic indices:

```

• # Load in species data
asb_sp <- read.csv("LI_fish_abundance_pre_post_bleaching.csv")

Create unique ID for each transect to use as a rowname
asb_sp$ID <- paste(asb_sp$Habitat, asb_sp$Site, asb_sp$Replicate, asb_sp$Bleaching, sep="_")
rownames(asb_sp) <- asb_sp$ID

Computer FRic index
alpha_fd_indices <- mFD::alpha.fd.multidim(

```

# TIEE

Teaching Issues and Experiments in Ecology - Volume 20, June 2024

```
sp_faxes_coord = coordinates_mFD[, c("PC1", "PC2", "PC3", "PC4", "PC5")],
asb_sp_w = as.matrix(asb_sp[,5:193]),
ind_vect = c("fric"))
```

```
Display the results:
alpha_fd_indices$"functional_diversity_indices"
```

The next bit of code is adding the total abundance to the table too:

```
• # Calculate the total abundance by taking a sum of the fish species count values in each row
asb_sp$Abun_tot <- rowSums(asb_sp[,5:193])

Take out just the relevant columns of abundance and transect information (i.e. removing all individual species columns)
asb_sp <- asb_sp[, c("Habitat", "Site", "Replicate", "Bleaching", "Abun_tot")]

Add the FRic index column to this
asb_sp <- cbind(asb_sp, alpha_fd_indices$"functional_diversity_indices")

Take a look at the table this has produced
asb_sp
```

This final bit of code is subsetting this out so that you have a dataframe for each habitat type, and writing this as a .csv file that you can use in the next practical (although noting that the subsets will be retained as objects in your R Project environment, so saving here is an extra optional step):

```
• # Subset out each individual habitat type
FD_BP <- subset(asb_sp, Habitat == "Branching_porites")
FD_LC <- subset(asb_sp, Habitat == "Low_coral_cover")
FD_MC <- subset(asb_sp, Habitat == "Mixed_coral")
FD_SC <- subset(asb_sp, Habitat == "Soft_coral")

Optional: Write each subset of habitat data to a .csv file. In this step, you can practice saving your computed data which you can then use outside of R, or upload back into R for future use.
write.csv(FD_BP, "FD_BP.csv")
write.csv(FD_LC, "FD_LC.csv")
```



```
write.csv(FD_MC, "FD_MC.csv")
write.csv(FD_SC, "FD_SC.csv")
```

**Q5 Create boxplot graphs showing the estimates of functional richness, species richness, and total abundance of fishes before and after bleaching, and export these for your poster.**

**OK - well done!** You have created your main results figures for your poster. You have learnt how to visually summarize your data using a boxplot and multi-dimensional functional spaces - allowing you to see and describe trends before and after bleaching in these fish communities. Next week we will learn how to test whether these differences are statistically significant.

**Before part 3 you should do 3 things:**

- Use ggplot2 to create graphs of the estimates of your chosen two species abundance counts before and after bleaching. Note: you will have to use the data that you subset in part 1 to create the 'dadata' dataframe.
- Watch this video on a two-sample t-test:  
<https://www.youtube.com/watch?v=RIhnNbPZC0A>
- Read this [10-minute guidance document](#) on Best Practices for Writing R Code and then revisit your code and tidy it up as per the guidance points
- Think about what will go into your your poster, and start drafting bullet points that you will want to make in Background and Data analysis sections.

## References

- Green, A.L., D.R. Bellwood, and H. Choat. 2009. Monitoring functional groups of herbivorous reef fishes as indicators of coral reef resilience. A practical guide for coral reef managers in the Asia Pacific Region. IUCN, Gland, Switzerland. Available here:  
[http://cmsdata.iucn.org/downloads/resilience\\_herbivorous\\_monitoring.pdf](http://cmsdata.iucn.org/downloads/resilience_herbivorous_monitoring.pdf).
- Magneville, C., N. Loiseau, C. Albouy, N. Casajus, T. Claverie, A. Escalas, F. Leprieur, E. Maire, D. Mouillot, and S. Villeger. 2022. `_mFD`: mFD: an R package to

- compute and illustrate the multiple facets of functional diversity, *Ecography* 2022: <https://onlinelibrary.wiley.com/doi/full/10.1111/ecog.05904> (open access).
- Mouillot, D., D.R. Bellwood, C. Baraloto, J. Chave, R. Galzin, M. Harmelin-Vivien, M. Kulbicki, S. Lavergne, S. Lavorel, N. Mouquet, C.E. Paine, J. Renaud, and W. Thuiller. 2013. Rare species support vulnerable functions in high-diversity ecosystems. *PLoS Biol* 11:e1001569. <https://doi.org/10.1371/journal.pbio.1001569> (open access).
- Parravicini, V., M.G. Bender, S. Villéger, F. Leprieur, L. Pellissier, F.G.A. Donati, S.R. Floeter, E.L. Rezende, D. Mouillot, and M. Kulbicki. 2021. Coral reef fishes reveal strong divergence in the prevalence of traits along the global diversity gradient. *Proceedings of the Royal Society of London B: Biological Sciences* 288:20211712 doi: 10.1098/rspb.2021.1712. (Author copy available here: [https://www.researchgate.net/publication/355454180\\_Coral\\_reef\\_fishes\\_reveal\\_strong\\_divergence\\_in\\_the\\_prevalence\\_of\\_traits\\_along\\_the\\_global\\_diversity\\_gradient](https://www.researchgate.net/publication/355454180_Coral_reef_fishes_reveal_strong_divergence_in_the_prevalence_of_traits_along_the_global_diversity_gradient)).
- Pratchett, M.S., A.S. Hoey, S.K. Wilson, V. Messmer, and N.A. Graham. 2011. Changes in biodiversity and functioning of reef fish assemblages following coral bleaching and coral loss. *Diversity* 3:424-452. <https://doi.org/10.3390/d3030424> (open access).
- Wilson, S.K., R. Fisher, M.S. Pratchett, N.A. Graham, N.K. Dulvy, R.A. Turner, A. Cakacaka, N.V. Polunin, and S.P. Rushton. 2008. Exploitation and habitat degradation as agents of change within coral reef fish communities. *Global Change Biology* 14:2796-2809. doi: 10.1111/j.1365-2486.2008.01696.x; Available here: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2486.2008.01696.x>