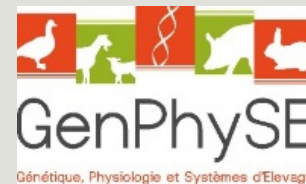


# Analysis of community composition data using phyloseq

---

MAHENDRA MARIADASSOU, MARIA BERNARD, GERALDINE PASCAL,  
LAURENT CAUQUIL, STEPHANE CHAILLOU



# Goals

---

- Learn about and become familiar with **phyloseq** R package for the analysis of microbial census data
- Exploratory Data Analysis
  - **$\alpha$ -diversity**: how diverse is my community?
  - **$\beta$ -diversity**: how different are two communities?
  - Use a distance matrix to study structures:
    - **Hierarchical clustering**: how do the communities cluster?
    - **Permutational ANOVA**: are the communities structured by some known environmental factor (pH, height, etc)?
  - Visual assessment of the data
    - **bar plots**: what is the composition of each community?
    - **Multidimensional Scaling**: how are communities related?
    - **Heatmaps**: are there interactions between species and (groups of) communities?

# Training Data

---

Training data come from a real analysis provided by Stéphane Chaillou et al.

The aim was to compare meat (4 types) and seafood (4 types) bacterial communities.

We use here, an extract of these public data :

- 64 samples of 16S V1-V3
- From 8 environment types :
  - Meat → Ground Beef, Ground veal, Poultry sausage, Diced bacon
  - Seafood → Cooked schrimps, Smoked salmon, Salmon filet, Cod filet
- 508 OTUs affiliated with the Greengenes database

# Training Data

---

→ Take a look at the data

- data/chaillou/otu\_table.tsv

	CDT0#LOT06	CDT0#LOT09	CDT0#LOT07	CDT0#LOT02
otu_00601	0	0	0	4
otu_01717	0	4	37	5

- data/chaillou/tax\_table.tsv

	Kingdom	Phylum	Class	Order	...
otu_00001	Bacteria	Proteobacteria	Gammaproteobacteria	Pseudomonadales	...
otu_00004	Bacteria	Proteobacteria	Gammaproteobacteria	Pseudomonadales	...

# Training Data

---

➔ Take a look at the data

▪ data/chaillou/sample\_data.tsv

	SampleID	EnvType	Description	FoodType
BHT0.LOT01	BHT0#LOT01	BoeufHache	LOT1	Meat
BHT0.LOT03	BHT0#LOT03	BoeufHache	LOT3	Meat

▪ data/chaillou/tree.nwk

```
((((( ((( ((( ((( ((( (otu_00520:0.016,otu_00555:0.01122):0.01094,((otu_00568:0.00301,otu_00566:0.01354):0.00617,otu_00569:0.00821):0.00998):0.00828,otu_00545:0.03879):0.02824,((otu_00527:0.02225,otu_00521:0.00934): ...
```

# Phyloseq

---

# Phyloseq

---

- About phyloseq
- phyloseq data structure
- Other accessors
- Manipulating a phyloseq object: Filtering
- Manipulating a phyloseq object: Smoothing
- Manipulating a phyloseq object: Abundance counts
- Importing a phyloseq object

# About Phyloseq

---

- R package (McMurdie and Holmes, 2013) to analyze community composition data in a phylogenetic framework

He uses other R packages:

- Community ecology functions from vegan, ade4, picante
- Tree manipulation from ape
- Graphics from ggplot2
- (Differential analysis from DESeq2)



# About Phyloseq

---

- Find help:

phyloseq comes with two vignettes

```
vignette("phyloseq-basics")
```

```
vignette("phyloseq-analysis")
```

The first one gives insights about data structure and data manipulation (Section 2), the second one about data analysis (Section 3 to 5).



# Phyloseq : Let's get started

---

→ load the phyloseq package

```
library(phyloseq)
```

→ and some additional custom functions

```
source("https://raw.githubusercontent.com/mahendra-  
mariadassou/phyloseq-extended/master/load-extra-functions.R" )
```

# Phyloseq : Data import

---

The biom format natively supports

- OTU count tables (required)
- OTU description
- sample description

but each component can be stored in TSV files.

Other optional components must be stored in separate files

- phylogenetic tree in Newick format
- sequences in fasta format

# Phyloseq : Data import

---

➔ Import biom file (from FROGS or anywhere else)

```
biomfile <- "data/chailou/chailou.biom"
```

```
food1 <- import_biom(biomfile)
```

```
food1
```

```
## phyloseq-class experiment-level object
```

```
## otu_table() OTU Table: [ 508 taxa and 64 samples ]
```

```
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
```

# Phyloseq : Data import

---

➔ Add samples metadata (optional, these can be already stored in biom file)

```
sampladata <-read.csv("data/chailou/sample_data.tsv", sep="\t", row.names = 1)
```

```
sample_data(food1) <- sampladata
```

```
food1
```

```
## phyloseq-class experiment-level object
```

```
## otu_table() OTU Table: [ 508 taxa and 64 samples ]
```

```
## sample_data() Sample Data: [ 64 samples by 4 sample variables ]
```

```
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
```

# Phyloseq : Data import

---

→ Add phylogenetic tree (optional)

```
treefile <- read.tree("data/chailou/tree.nwk")
```

```
phy_tree(food1) <- treefile
```

```
food1
```

```
## phyloseq-class experiment-level object
```

```
## otu_table() OTU Table: [ 508 taxa and 64 samples ]
```

```
## sample_data() Sample Data: [ 64 samples by 4 sample variables ]
```

```
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
```

```
## phy_tree() Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
```

# Phyloseq : Data import

---

→ More direct way, if sample metadata are in biom file

```
biomfile2 <- "data/chailou/chailou_with_sam_data.biom"
treefile <- read.tree("data/chailou/tree.nwk")
food <- import_biom(biomfile2, treefile, parseFunction =
  parse_taxonomy_greenegenes)
food
## phyloseq-class experiment-level object
## otu_table()      OTU Table:          [ 508 taxa and 64 samples ]
## sample_data()   Sample Data:        [ 64 samples by 4 sample variables ]
## tax_table()     Taxonomy Table:     [ 508 taxa by 7 taxonomic ranks ]
## phy_tree()      Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
```

# Phyloseq Data Structure

---

➔ What information do we expect to find in food?

`food`

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 508 taxa and 64 samples ]
## sample_data() Sample Data: [ 64 samples by 3 sample variables ]
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
```



# Phyloseq Data Structure

---

Our phyloseq object `food` is made up of four parts:

- OTU Table
- Sample Data
- Taxonomy Table
- Phylogenetic Tree

Let's have a quick look at each using the hinted at functions:

- `otu_table`,
- `sample_data`,
- `tax_table`,
- `phy_tree`.

# Phyloseq Data Structure

---

OTU\_table object is a **matrix-like** object

```
head(otu_table(food))
```

```
##OTU Table:          [6 taxa and 64 samples]
```

```
##                taxa are rows
```

```
##           DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06 DLT0.LOT01 DLT0.LOT04 DLT0.LOT10
##otu_00520          0          0          0          0          0          0          0          0
##otu_00555          4          0          8          3          15         4          2          0
##otu_00568          0          0          0          0          0          0          0          0
##otu_00566          0          3          0          0          0          0          0          0
##otu_00569         12          3         26         16         38          0          4          0
##otu_00545          0          0          0          0          0          0          0          0
##           MVT0.LOT05 MVT0.LOT01 MVT0.LOT06 MVT0.LOT07 MVT0.LOT03 MVT0.LOT09 MVT0.LOT08 MVT0.LOT10
##otu_00520          0          0          0          0          0          0          0          0
##otu_00555         12         17         25         31         11         40         12          4
##otu_00568          0          0          0          0          0          0          0          0
##otu_00566          0          0          0          0          0          0          0          0
##otu_00569         10         15         40         35         11        119          0          0
```

# Phyloseq Data Structure

---

tax\_table object is a **matrix-like** object

```
head(tax_table(food))
```

```
##Taxonomy Table:      [6 taxa by 7 taxonomic ranks]:
```

```
##           Kingdom      Phylum           Class           Order           Family
##otu_00520 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##otu_00555 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##otu_00568 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##otu_00566 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##otu_00569 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##otu_00545 "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales" "Enterobacteriaceae"
##           Genus           Species
##otu_00520 "Raoultella"           "Ornithinolytica"
##otu_00555 "Hafnia-Obesumbacterium" "Alveii"
##otu_00568 "Serratia"           "Fonticola"
##otu_00566 "Serratia"           "Liquefaciens"
##otu_00569 "Serratia"           "Proteamaculans"
```

→ Try on food1

# Phyloseq Data Structure

---

Sample\_data object is a `data.frame-like` object

```
head(sample_data(food))
```

```
##Sample Data:          [6 samples by 3 sample variables]:
```

```
##           EnvType FoodType Description
##DLT0.LOT08 DesLardons    Meat      LOT8
##DLT0.LOT05 DesLardons    Meat      LOT5
##DLT0.LOT03 DesLardons    Meat      LOT3
##DLT0.LOT07 DesLardons    Meat      LOT7
##DLT0.LOT06 DesLardons    Meat      LOT6
##DLT0.LOT01 DesLardons    Meat      LOT1
```

# Phyloseq Data Structure

---

Phy\_tree object is a `phylo_class(tree)` object

```
phy_tree(food)
```

```
##Phylogenetic tree with 508 tips and 507 internal nodes.
```

```
Tip labels: otu_00520, otu_00555, otu_00568, otu_00566, otu_00569,  
otu_00545, ...
```

```
Rooted; includes branch lengths.
```

# Phyloseq Data Structure

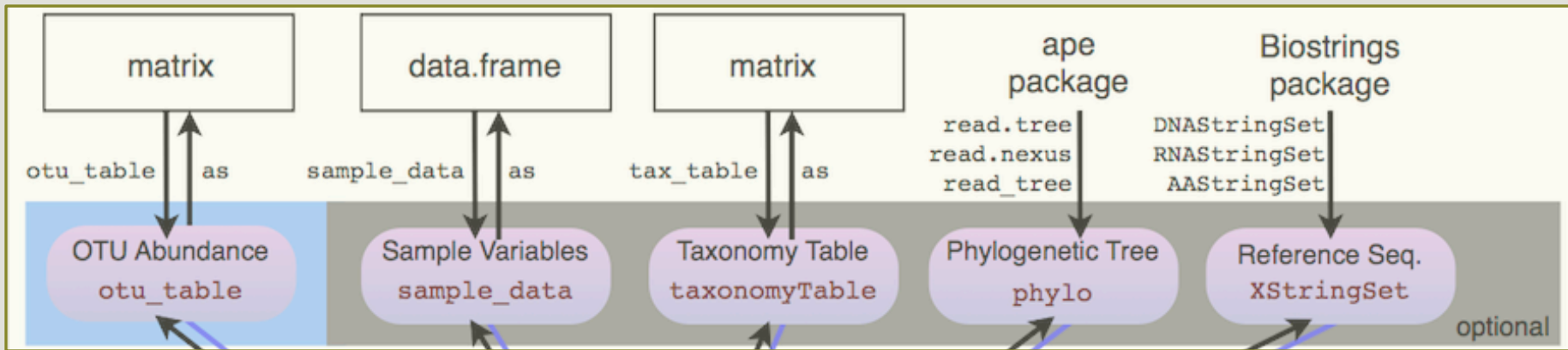
---

A phyloseq object is made of up to 5 **components** (or slots):

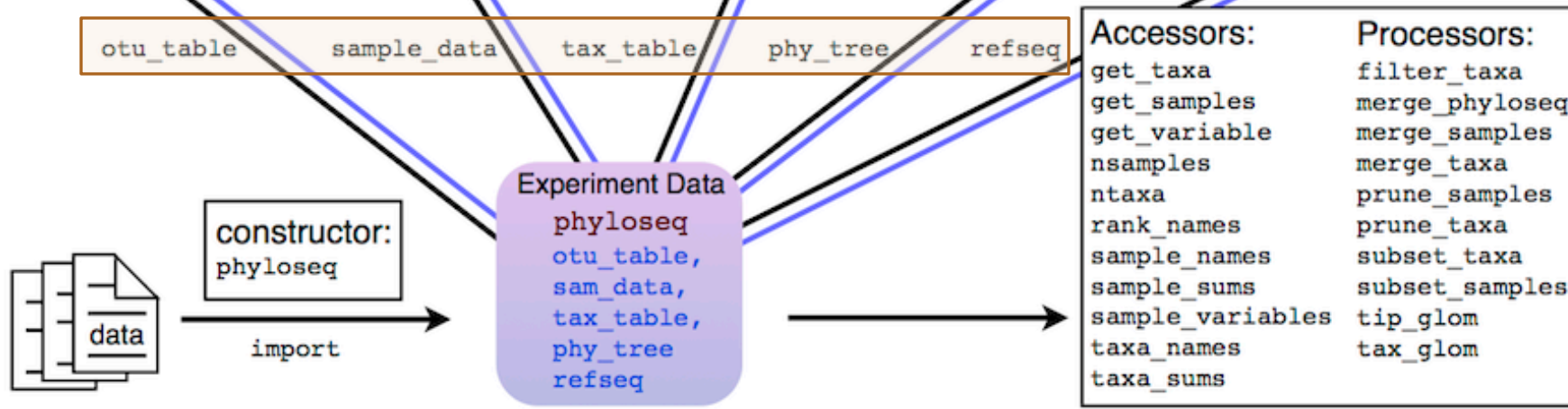
- **otu\_table**: an OTU abundance table;
- **sample\_data**: a table of sample metadata, like sequencing technology, location of sampling, etc;
- **tax\_table**: a table of taxonomic descriptors for each OTU, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
- **phy\_tree**: a phylogenetic tree of the OTUs;
- **refseq**: a set of reference sequences (one per OTU), not present in food.

# Phyloseq Data Structure

Phyloseq objects



Functions to access/edit objects



# Phyloseq : accessors

---

Phyloseq also offers the following **accessors** to extract parts of a phyloseq object.:

- `ntaxa / nsamples`
- `sample_names / taxa_names`
- `sample_sums / taxa_sums`
- `rank_names`
- `sample_variables`
- `get_taxa`
- `get_samples`
- `get_variable`

➔ Try them on your own (on **food**) and guess what they do.



# Phyloseq : accessors

---

```
ntaxa(food)
```

```
## [1] 508
```

`ntaxa` returns the number of taxa;

```
nsamples(food)
```

```
## [1] 64
```

`nsamples` returns the number of samples;

# Phyloseq : accessors

---

```
head(sample_names(food))
```

```
## [1] "DLT0.LOT08" "DLT0.LOT05" "DLT0.LOT03" "DLT0.LOT07" "DLT0.LOT06"  
## [6] "DLT0.LOT01"
```

```
head(taxa_names(food))
```

```
## [1] "otu_00520" "otu_00555" "otu_00568" "otu_00566" "otu_00569"  
"otu_00545"
```

Names of the samples and taxa included in the phyloseq object.

# Phyloseq : accessors

---

```
head(sample_sums(food))
```

```
## DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06 DLT0.LOT01  
##          11812          11787          11804          11806          11832          11857
```

```
head(taxa_sums(food))
```

```
## otu_00520 otu_00555 otu_00568 otu_00566 otu_00569 otu_00545  
##          55          395          22          13          1998          210
```

Total count of each sample (i.e. sample library sizes) or of each taxa (i.e. overall abundances across all samples)

# Phyloseq : accessors

---

```
rank_names(food)
```

```
## [1] "Kingdom" "Phylum" "Class" "Order" "Family" "Genus"  
"Species"
```

Names of the taxonomic levels available in the tax\_table slot.

→ Try on food1

```
head(sample_variables(food))
```

```
## [1] "EnvType" "FoodType" "Description"
```

Names of the contextual data recorded on the samples.

# Phyloseq : accessors

---

→ Find the

- library size of samples `MVT0.LOT01`, `MVT0.LOT07`, `MVT0.LOT09`
- overall abundance of OTUs `otu_00520`, `otu_00569`, `otu_00527`

Hint: What's the class of `sample_sums(food)` and `taxa_sums(food)`? How do you index them?

```
## sample library sizes
sample_sums(food)[c("MVT0.LOT01", "MVT0.LOT07", "MVT0.LOT09")]
## MVT0.LOT01 MVT0.LOT07 MVT0.LOT09
##          11743          11765          11739

## OTU overall abundances
taxa_sums(food)[c("otu_00520", "otu_00569", "otu_00527")]
## otu_00520 otu_00569 otu_00527
##          55          1998          58
```

# Phyloseq : accessors

---

```
head(get_variable(food, varName = "EnvType"))
```

```
## [1] DesLardons DesLardons DesLardons DesLardons DesLardons DesLardons  
## 8 Levels: BoeufHache VeauHache DesLardons MerguezVolaille ...  
Crevette
```

values for variable `varName` in sample data

```
head(get_sample(food, i = "otu_00520"))
```

```
## DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06 DLT0.LOT01  
## 0 0 0 0 0 0
```

abundance values of `OTU i` in all samples (row of OTU table).

# Phyloseq : accessors

---

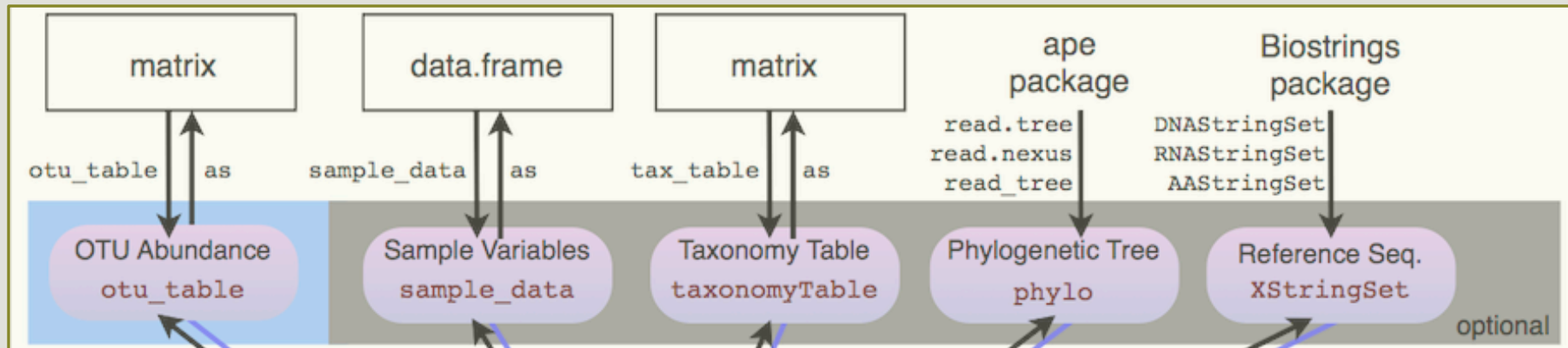
```
head(get_taxa(food, i = "MVT0.LOT07"))
```

```
## otu_00520 otu_00555 otu_00568 otu_00566 otu_00569 otu_00545  
##          0          31          0          0          35          0
```

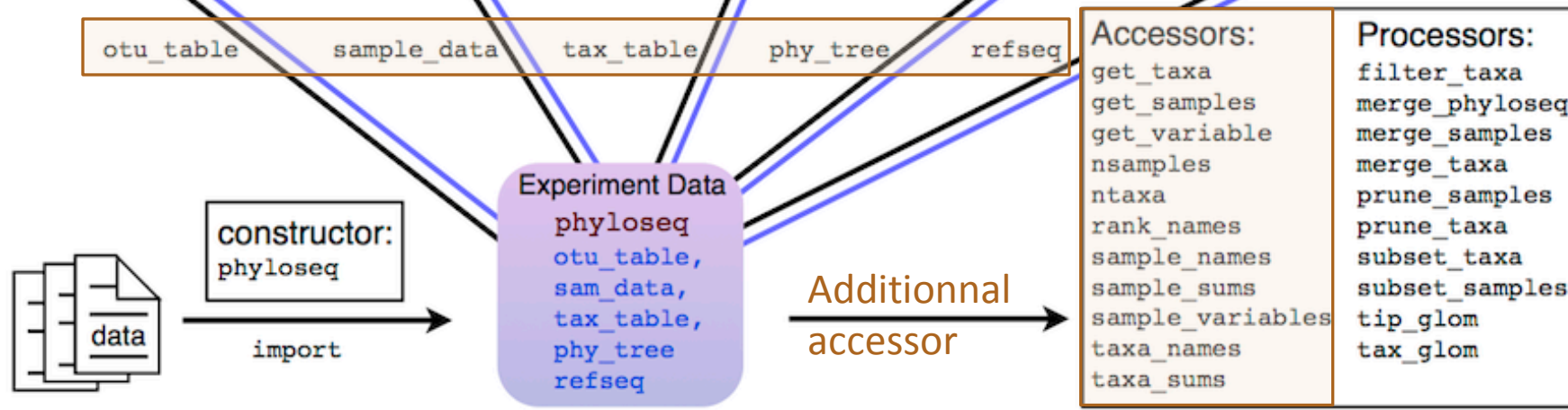
abundance values of all OTUs in [sample i](#) (column of OTU table)

# Phyloseq Data Structure

Phyloseq objects



Functions to access/edit objects





# Phyloseq : accessors/editors

---

→ How to modify part of phyloseq objects?

- Change, the taxonomic rank names in `food1` object. Hint : use the (high level) accessors?

```
## desired rank names
new_rank <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species")
## modify
colnames(tax_table(food1)) <- new_rank
## check
rank_names(food1) ## or head(tax_table(food1))
## [1] "Kingdom"      "Phylum"      "Class"        "Order"        "Family"       "Genus"
      "Species"
```

# Phyloseq : accessors/editors

---

→ How to modify part of phyloseq objects?

- Change, the EnvType variable order to have meat products first and seafood second.

```
## desired order
correct.order <- c("BoeufHache", "VeauHache", "DesLardons« , "MerguezVolaille",
"SaumonFume", "FiletSaumon« , "FiletCabillaud", "Crevette")
## modify
sample_data(food)$EnvType <- factor(sample_data(food)$EnvType, levels = correct.order)
## check
levels(get_variable(food, "EnvType"))
## [1] "BoeufHache" "VeauHache" "DesLardons" "MerguezVolaille"
## [5] "SaumonFume" "FiletSaumon" "FiletCabillaud" "Crevette"
```

# Phyloseq : accessors/editors

---

→ How to modify part of phyloseq objects?

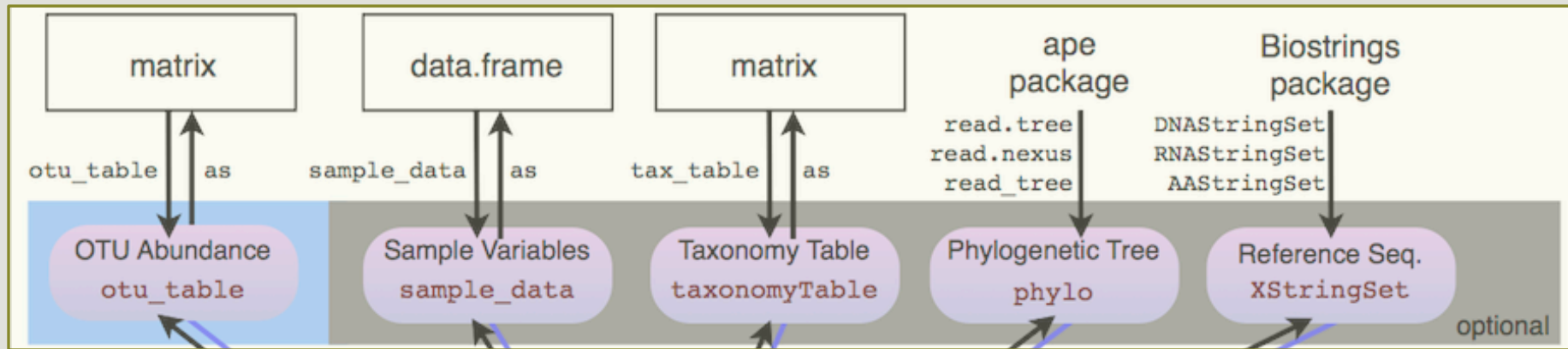
- modify the count of OTU "otu\_00520" in sample "DLT0.LOT08", or its species affiliation?

```
## change abundance of "otu_00520" in sample "DLT0.LOT08" to 0  
otu_table(food)["otu_00520", "DLT0.LOT08"] <- 0
```

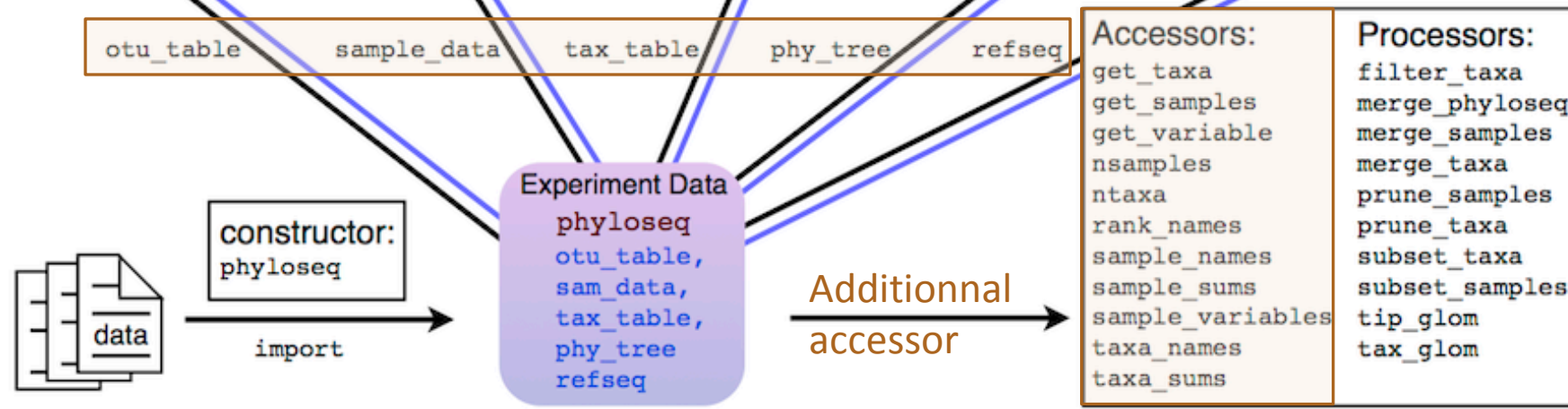
```
## change species affiliation of "otu_00520" to "Ornithinolytica"  
tax_table(food)["otu_00520", "Species"] <- "Ornithinolytica"
```

# Phyloseq Data Structure

Phyloseq objects



Functions to access/edit objects



# Phyloseq : Filtering Functions

---

## ■ Prune

- `prune_taxa / prune_samples` prunes unwanted taxa / samples from a phyloseq object based on a vector of taxa to keep
- The taxa are passed as a vector, `taxa`, of character ("otu\_1", "otu\_4") or of logical (TRUE, FALSE, FALSE, TRUE)
- Example : `prune_taxa(taxa, physeq_obj)` would keep only OTUs otu\_1, otu\_4

## ■ Subset

- `subset_taxa / subset_samples` subsets unwanted taxa / samples from a phyloseq object based on conditions that must be met
- The conditions (any number) can apply to any descriptor (e.g. taxonomy) of the OTUs included in the phyloseq object, `physeq`
- `subset_taxa(physeq, Phylum == "Firmicutes")` would keep only Firmicutes.

# Phyloseq : Filtering Functions

---

→ Prune\_samples: Keep only the first 10 samples of your phyloseq object **food**

```
## samplesToKeep
samplesToKeep <- sample_names(food)[1:10]
## prune_samples
prune_samples(samplesToKeep, food)
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 508 taxa and 10 samples ]
## sample_data() Sample Data: [ 10 samples by 3 sample variables ]
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
```

# Phyloseq : Filtering Functions

---

➔ `subset_samples` :Keep only samples that corresponds to "DesLardons" and "MerguezVolaille".

```
subset_samples(food, EnvType %in% c("DesLardons", "MerguezVolaille"))  
## phyloseq-class experiment-level object  
## otu_table() OTU Table: [ 508 taxa and 16 samples ]  
## sample_data() Sample Data: [ 16 samples by 3 sample variables ]  
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]  
## phy_tree() Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
```

# Phyloseq : Filtering Functions

---

Advanced exercise : Subset and Multiple conditions combined Keep only samples with Phylum affiliation equal to Firmicutes and Class affiliation to Bacilli.

Hint : AND is coded by « & » and OR is coded by « | »

```
small.food <- subset_taxa(food, Phylum == "Firmicutes" & Class == "Bacilli")
head(tax_table(small.food)[ , c("Phylum", "Class", "Order")])
## Taxonomy Table: [6 taxa by 3 taxonomic ranks]:
## Phylum Class Order
## otu_00583 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_00574 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_00581 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_00591 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_00582 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_00586 "Firmicutes" "Bacilli" "Lactobacillales"

## Unique combinations (Phylum, Class)
unique(tax_table(small.food)[ , c("Phylum", "Class")])
## Taxonomy Table: [1 taxa by 2 taxonomic ranks]:
## Phylum Class
## otu_00583 "Firmicutes" "Bacilli"
```



# Phyloseq : Smoothing Functions

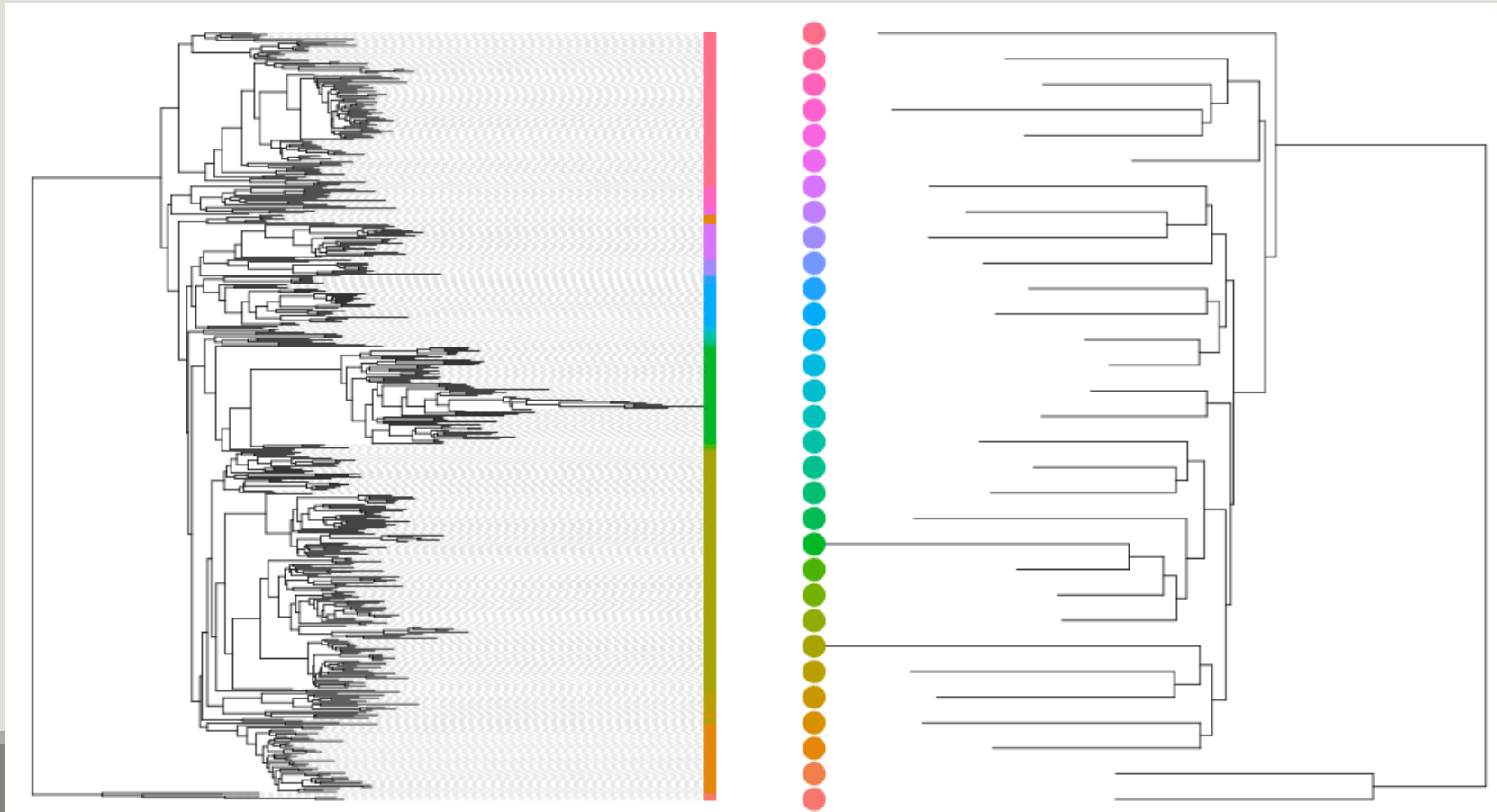
---

`tax_glom` agglomerates OTUs at a given taxonomic level. Finer taxonomic information is lost.

```
mergedData <- tax_glom(food, "Phylum")  
## check with ntaxa, tax_table  
ntaxa(mergedData) ## number of different phyla  
## [1] 11  
tax_table(mergedData)[1:2, c("Phylum", "Order", "Class")]  
## Taxonomy Table: [2 taxa by 3 taxonomic ranks]:  
## Phylum Order Class  
## otu_01101 "Proteobacteria" NA NA  
## otu_01152 "Actinobacteria" NA NA
```

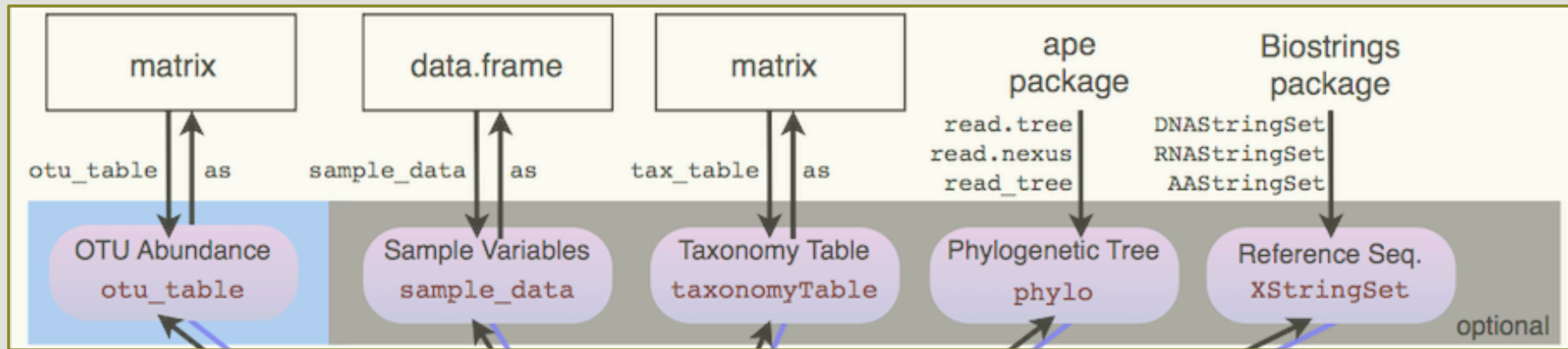
# Phyloseq : Smoothing Functions

The effect of `tax_glom` is most obvious and best understood on the phylogenetic tree (OTUs are colored by phylum).

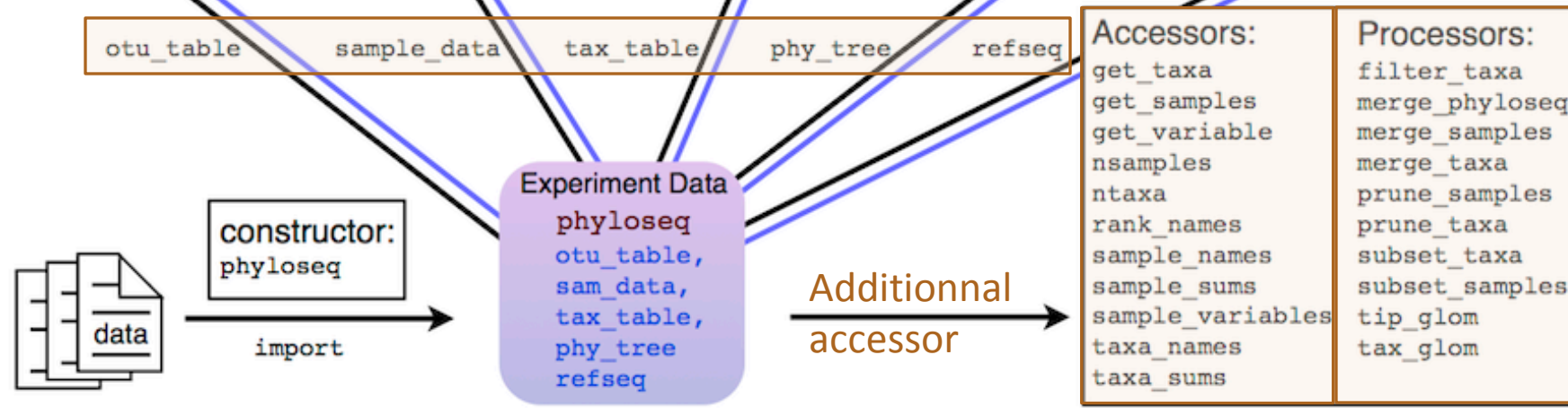


# Phyloseq Data Structure

Phyloseq objects



Functions to access/edit objects



Additional editor

# Phyloseq : Abundance manipulation

---

`rarefy_even_depth` downsamples/normalizes all samples to the same depth and prunes OTUs that disappear from all samples as a result.

→ Try on food

```
foodRare <- rarefy_even_depth(food, rngseed = 1121983)
## `set.seed(1121983)` was used to initialize repeatable random
## subsampling.
## Please record this for your records so others can reproduce.
## Try `set.seed(1121983); .Random.seed` for the full vector
## ...
## 10TUs were removed because they are no longer
## present in any sample after random subsampling
## check with sample_sums
sample_sums(foodRare)[1:5]
## DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06
##          11718          11718          11718          11718          11718
```

# Phyloseq : Abundance manipulation

---

`transform_sample_counts` applies a function to the abundance vector of each sample. It can be useful for normalization.

```
count_to_prop <- function(x) {return( x / sum(x) )}
```

→ transforms counts to proportions.

```
foodTrans <- transform_sample_counts(food, count_to_prop)
```

```
sample_sums(foodTrans)[1:5] ## should be 1
```

```
## DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06
```

```
##          1          1          1          1          1
```

# Phyloseq : in brief

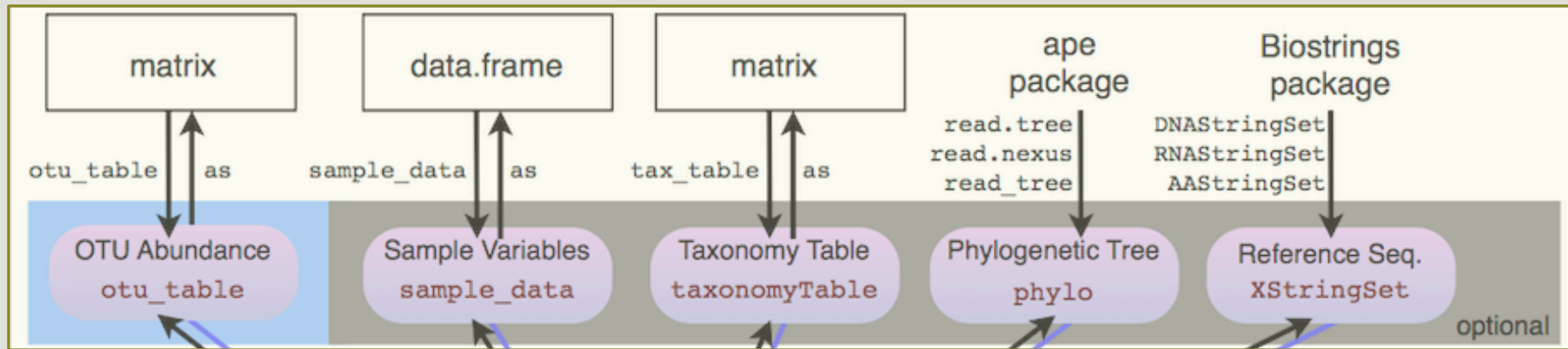
---

A nice data structure to store the **count table**, **taxonomic information**, **contextual data** and **phylogenetic tree** as different components of a single R object .

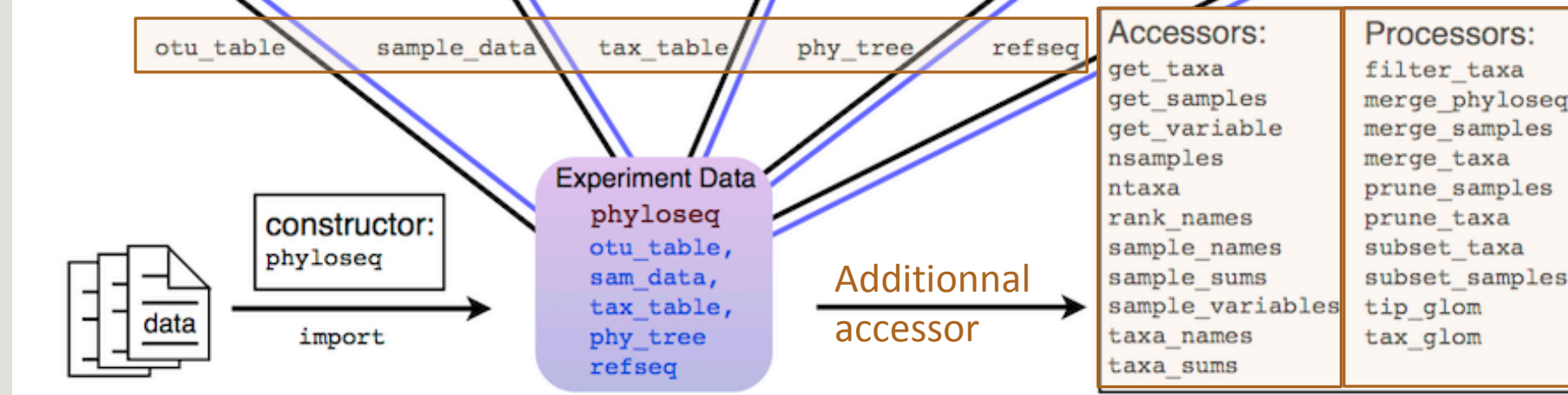
- Functions to **import** data from biom files, qiime output files or plain tabular files.
- **Accessors** to access different component of your dataset.
- Samples and taxa names are **coherent** between the different components.
- **Filters** to keep only part of the dataset.
- **Smoothers** to aggregate parts of the dataset.
- **Manipulators** to rarefy and transform samples.

# Phyloseq in brief

Phyloseq objects



Functions to access objects



Additional editor

# Biodiversity analysis

---



# Biodiversity analysis

---

- Exploring the samples composition
- Notions of biodiversity
- $\alpha$ -diversity analysis
- $\beta$ -diversity analysis

# Biodiversity analysis

---

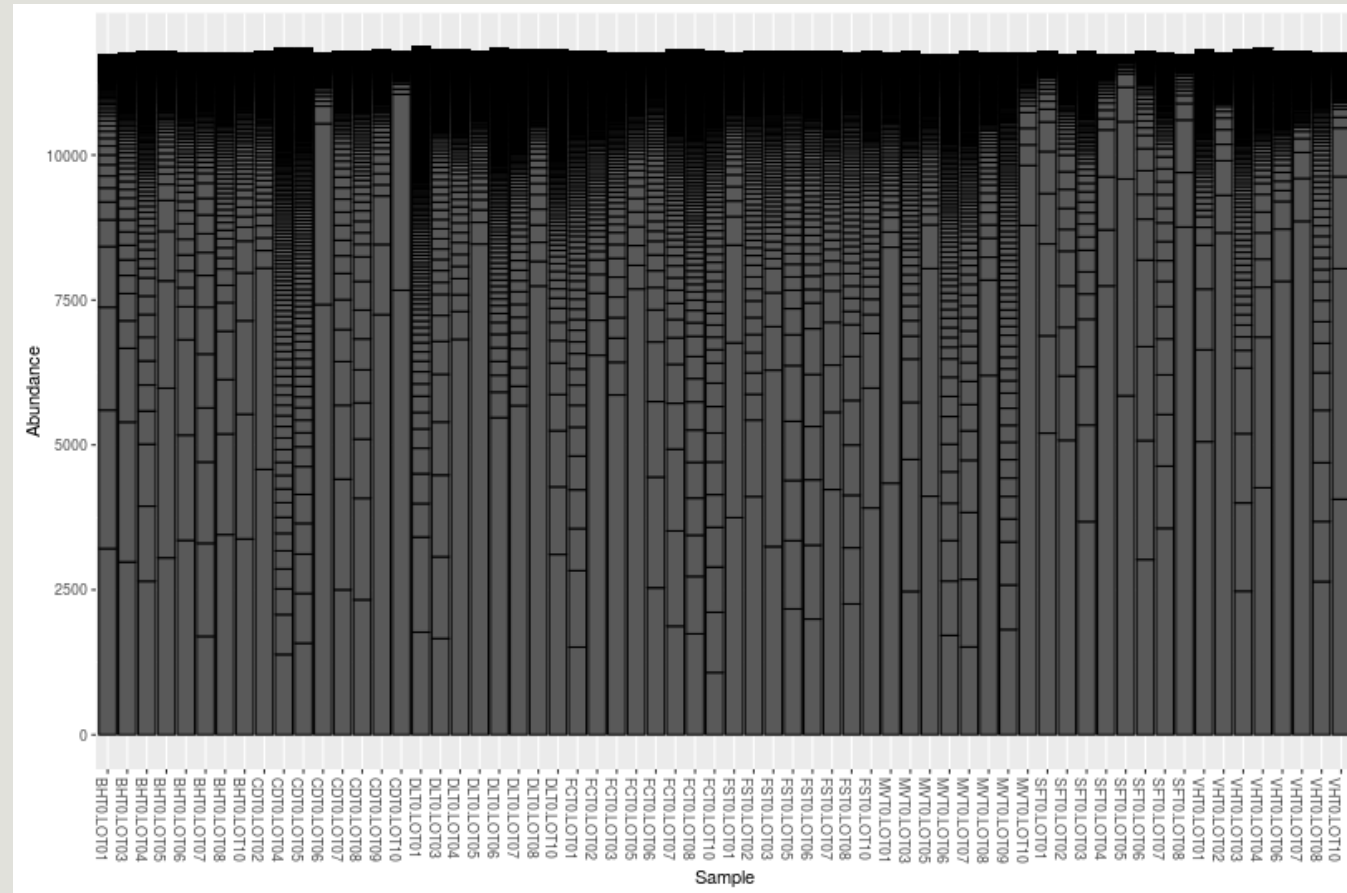
VISUALIZATION

# Exploring biodiversity : visualization

Visualization of abundance per sample thanks to `plot_bar`.

```
p <- plot_bar(food)
```

```
plot(p) ## Base graphic, ugly
```



# Exploring biodiversity : visualization

➔ Thanks to ggplot2, organize samples and color OTU by Phylum

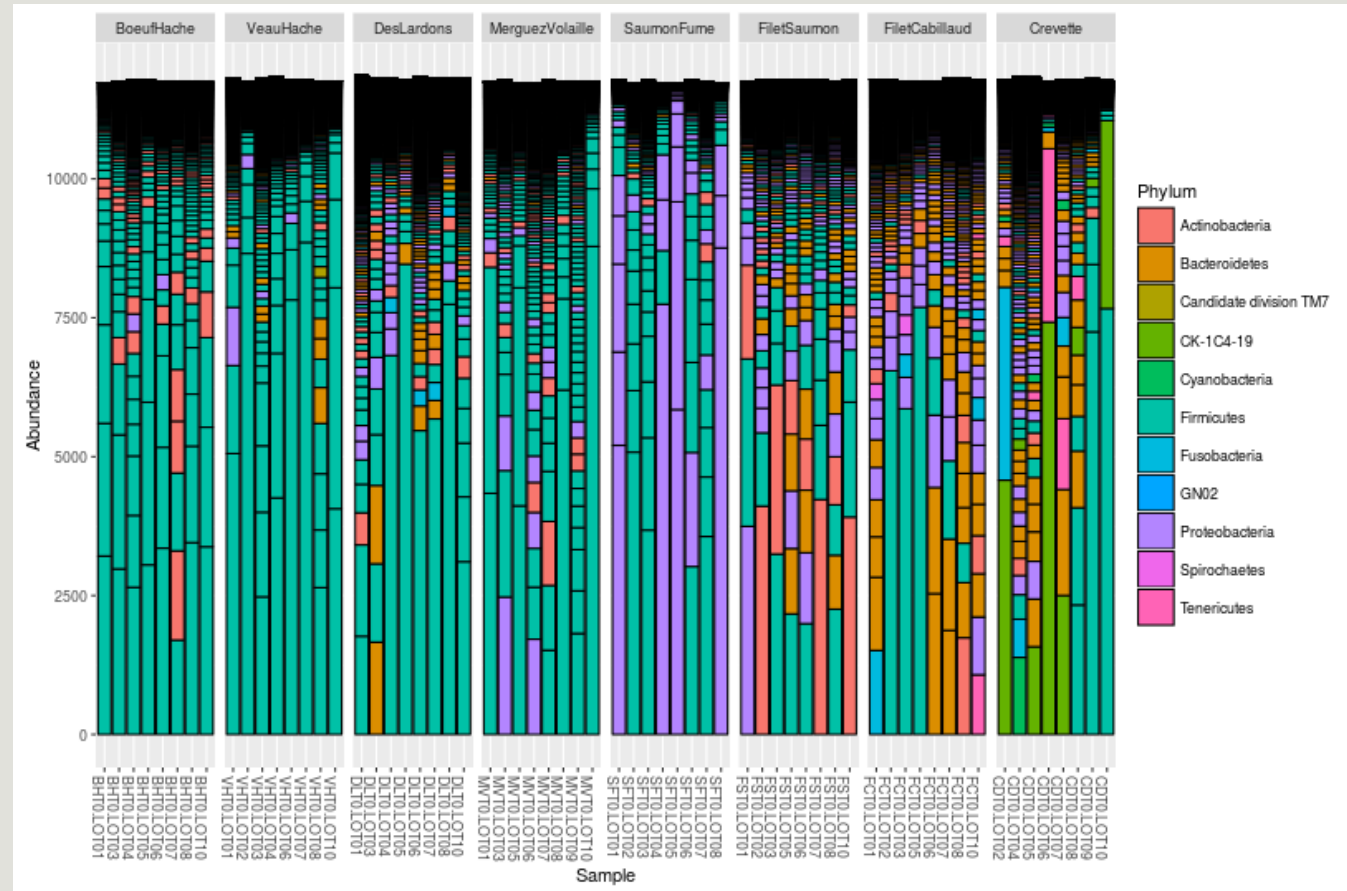
```
## aes, fill bar according to phylum
```

```
p <- plot_bar(food, fill = "Phylum")
```

```
## add facets
```

```
p <- p + facet_wrap(~EnvType, scales = "free_x", nrow = 1)
```

```
plot(p)
```



# Exploring biodiversity : visualization

---

## → Limitations:

- Plot bar works at the OTU-level...
- ...which may lead to graph cluttering and useless legends
- No easy way to look at a subset of the data
- Works with absolute counts (beware of unequal depths)



# Exploring biodiversity : visualization

---

- Customization: `plot_composition` function
  - `subset OTUs` at a given taxonomic level
  - `aggregate OTUs` at another taxonomic level
  - Show `only a given number` of OTUs (default the 10 most abundant).
  - Works with relative abundances

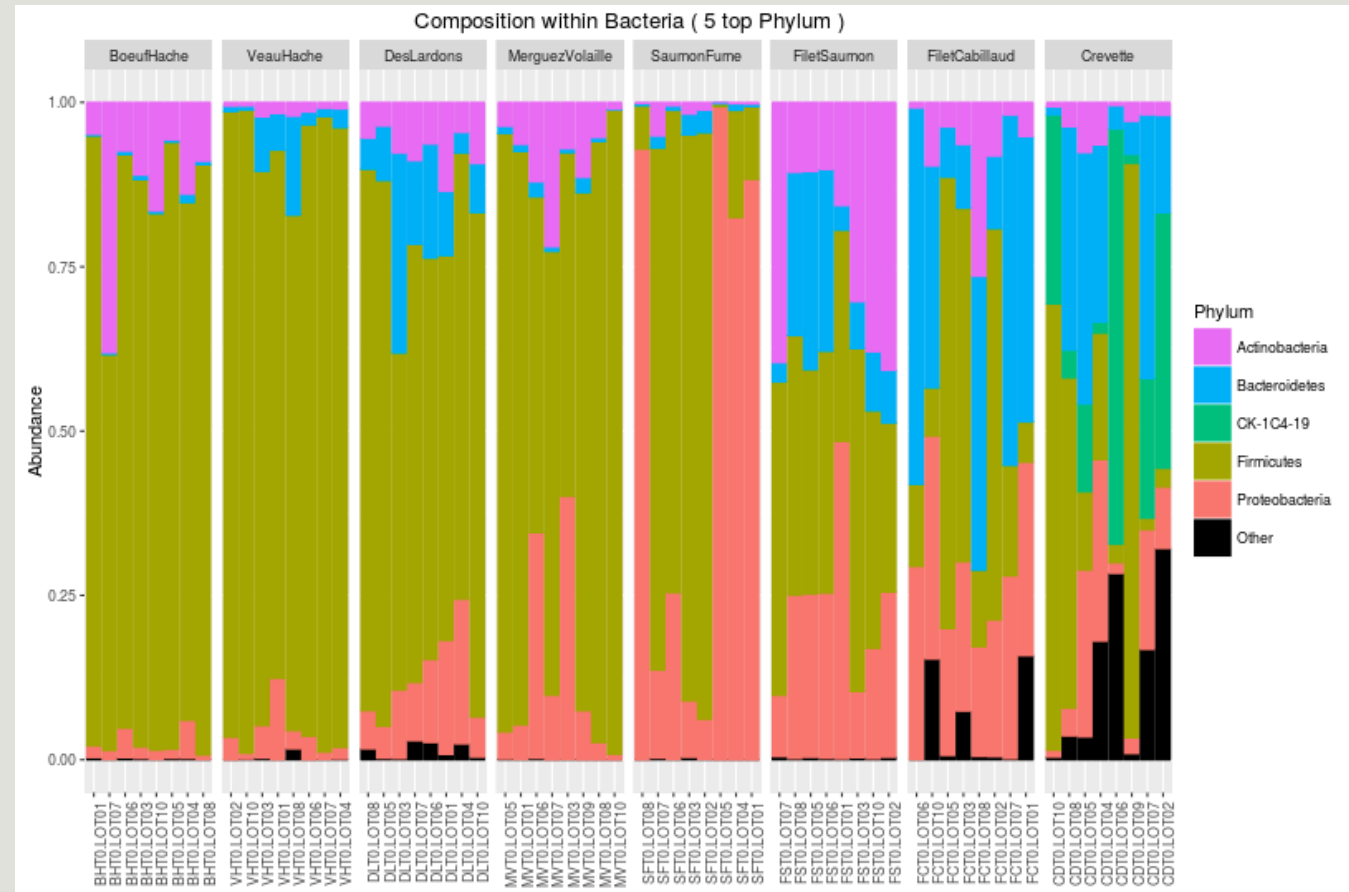
```
p <- plot_composition(physeq, "tax_level_selection",  
  "tax_name_selection", "tax_level_aggregation",  
  numberOfTaxa = NB_OTU, fill = "tax_color")
```



# Exploring biodiversity : visualization

→ Select Bacteria (at Kingdom level) and aggregate by Phylum, select the 5 most abundant.

```
## plot_composition  
  
p <- plot_composition(food, "Kingdom",  
"Bacteria", "Phylum", numberOfTaxa = 5,  
fill = "Phylum")  
  
## plot facetting by EnvType  
  
p <- p + facet_wrap(~EnvType, scales =  
"free_x", nrow = 1)  
  
plot(p)
```





# Exploring biodiversity : visualization

→ Select Proteobacteria (at Phylum level) and aggregate by Family and select the 9 most abundant.

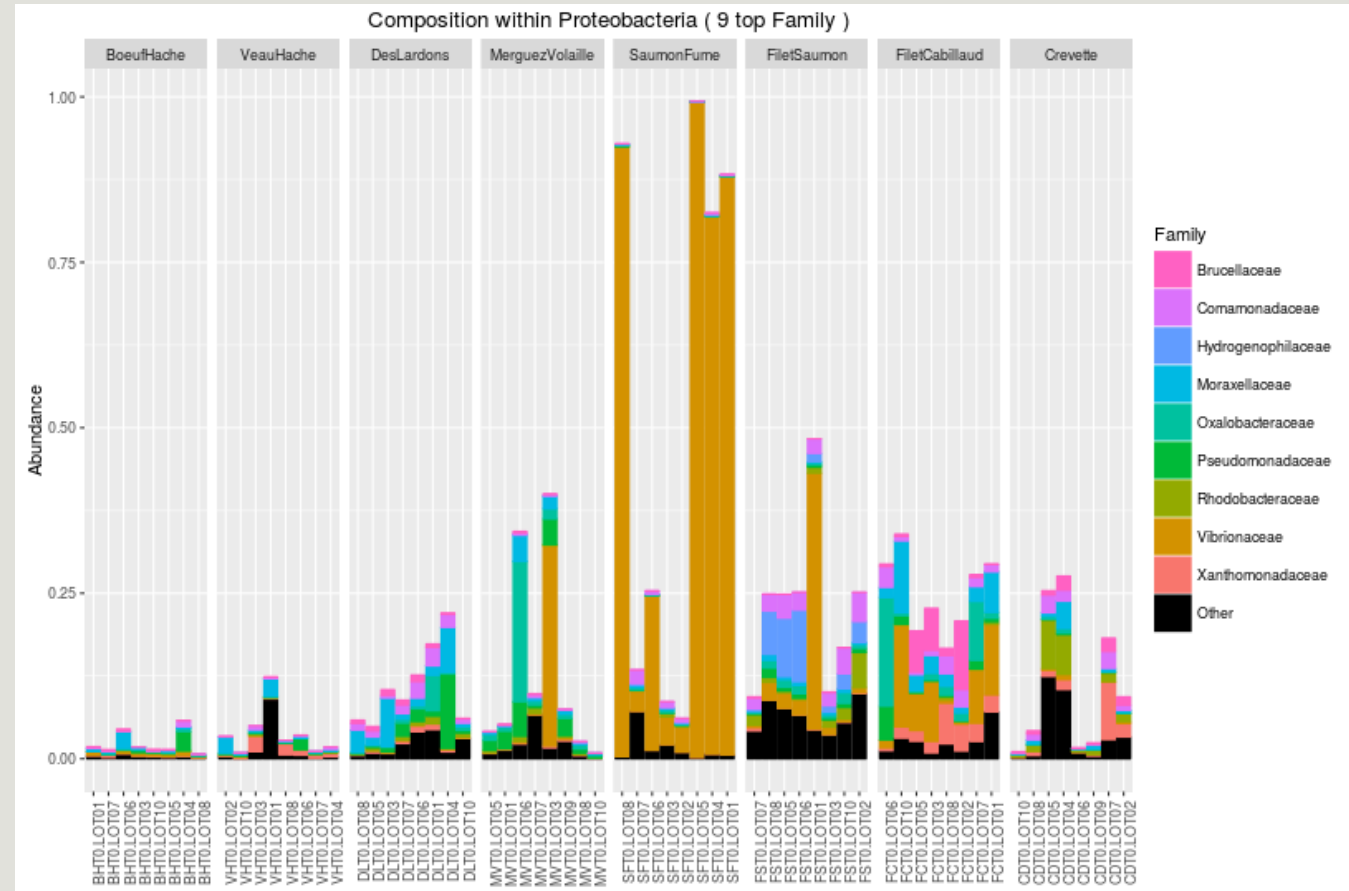
```
##plot_composition
```

```
p <- plot_composition(food, "Phylum", "Proteobacteria", "Family", numberOfTaxa = 9, fill = "Family")
```

```
## plot facetting
```

```
p <- p + facet_wrap(~EnvType, scales = "free_x", nrow = 1)
```

```
plot(p)
```





# Biodiversity analysis

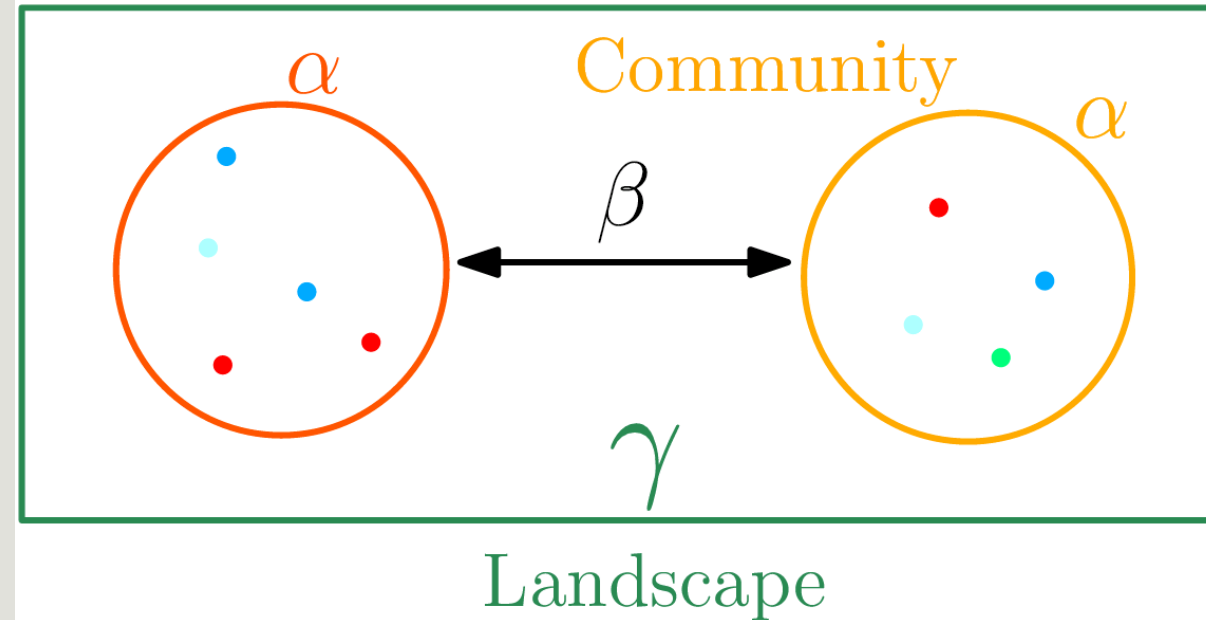
---

DIVERSITY INDICES

# Exploring biodiversity : statistical indices

Second Step, compute and compare diversity indices. Three flavors of diversity:

- **$\alpha$ -diversity**: diversity **within** a community;
- **$\beta$ -diversity**: diversity **between** communities;
  - $\beta$ -dissimilarities/distances
    - Dissimilarities between pairs of communities
    - Often used as a first step to compute-diversity
- $\gamma$ -diversity: diversity at the landscape scale (blurry for bacterial communities);



# Exploring biodiversity : statistical indices

---

## **Qualitative (Presence/Absence) vs. Quantitative (Abundance )**

- Qualitative gives less weight to dominant species;
- Qualitative is more sensitive to differences in sampling depths;
- Qualitative emphasizes difference in taxa diversity rather than differences in composition.

## **Compositional vs. Phylogenetic**

- Compositional does not require a phylogenetic tree;
- is more sensitive to erroneous OTU picking;
- gives the same importance to all OTUs.

# Biodiversity analysis

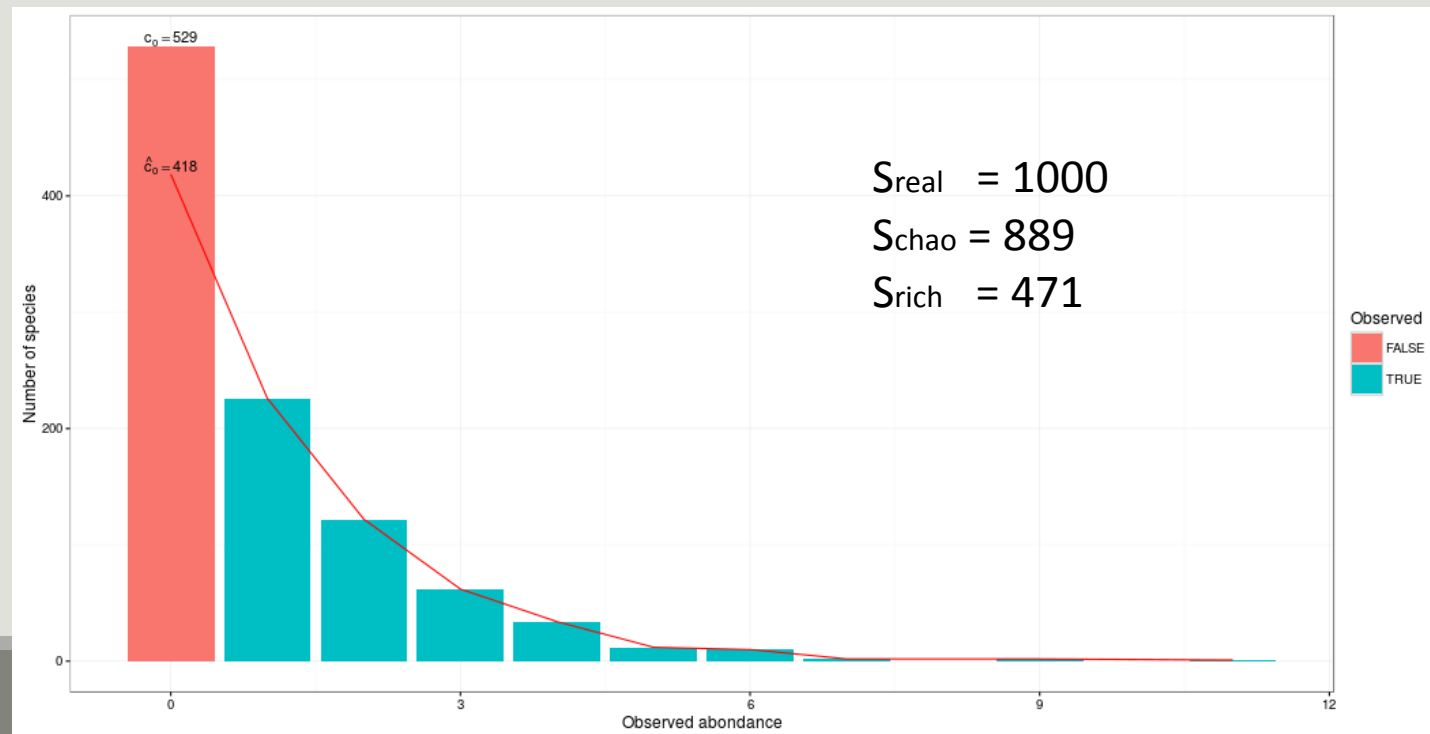
---

$\alpha$ -DIVERSITY INDICES

# Exploring biodiversity : $\alpha$ -diversity

$\alpha$ -diversity is equivalent to the richness : number of species

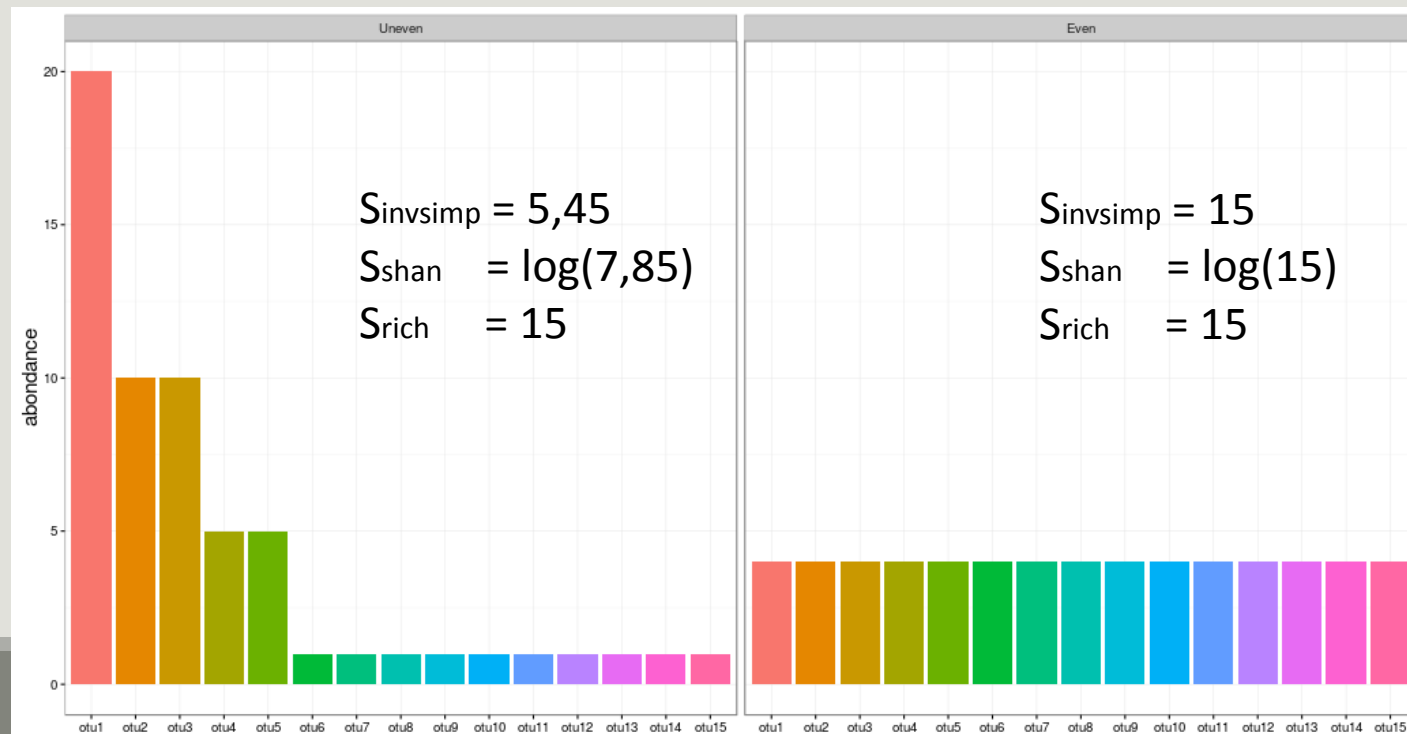
Richness	Chao
Number of observed species	Richness + (estimated) number of unobserved species



# Exploring biodiversity : $\alpha$ -diversity

$\alpha$ -diversity is equivalent to the richness : number of species

Shannon	Inv-Simpson
Evenness of the species abundance distribution	Inverse probability that two sequences sampled at random come from the same species



# Exploring biodiversity : $\alpha$ -diversity

---

$\alpha$ -diversity indices available in phyloseq :

- Species **richness** : number of observed OTUs
- **Chao1** : number of observed OTU + estimate of the number of unobserved OTUs
- **Shannon** entropy / **Jensen** : the width of the OTU relative abundance distribution. Roughly, it reflects our (in)ability to predict the OTU of a randomly picked bacteria.
- **Simpson** : 1 - probability that two bacteria picked at random in the community belong to different OTU.
- **Inverse Simpson** : inverse of the probability that two bacteria picked at random belong to the same OTU.

# Exploring biodiversity : $\alpha$ -diversity

---

Richness are plotted with `plot_richness`.

→ Try it on food

```
plot_richness(food)
```

→ Custom it on EnvType, color EnvType, select measures as "Observed", "Chao1", "Shannon", "Simpson", "InvSimpson", and plot as boxplot

Note the x = "EnvType " passed on to the `aes` mapping of a ggplot.

```
## plot_richness
```

```
p <- plot_richness(food, color = "EnvType", x = "EnvType", measures =  
  c("Observed", "Chao1", "Shannon", "Simpson", "InvSimpson"))
```

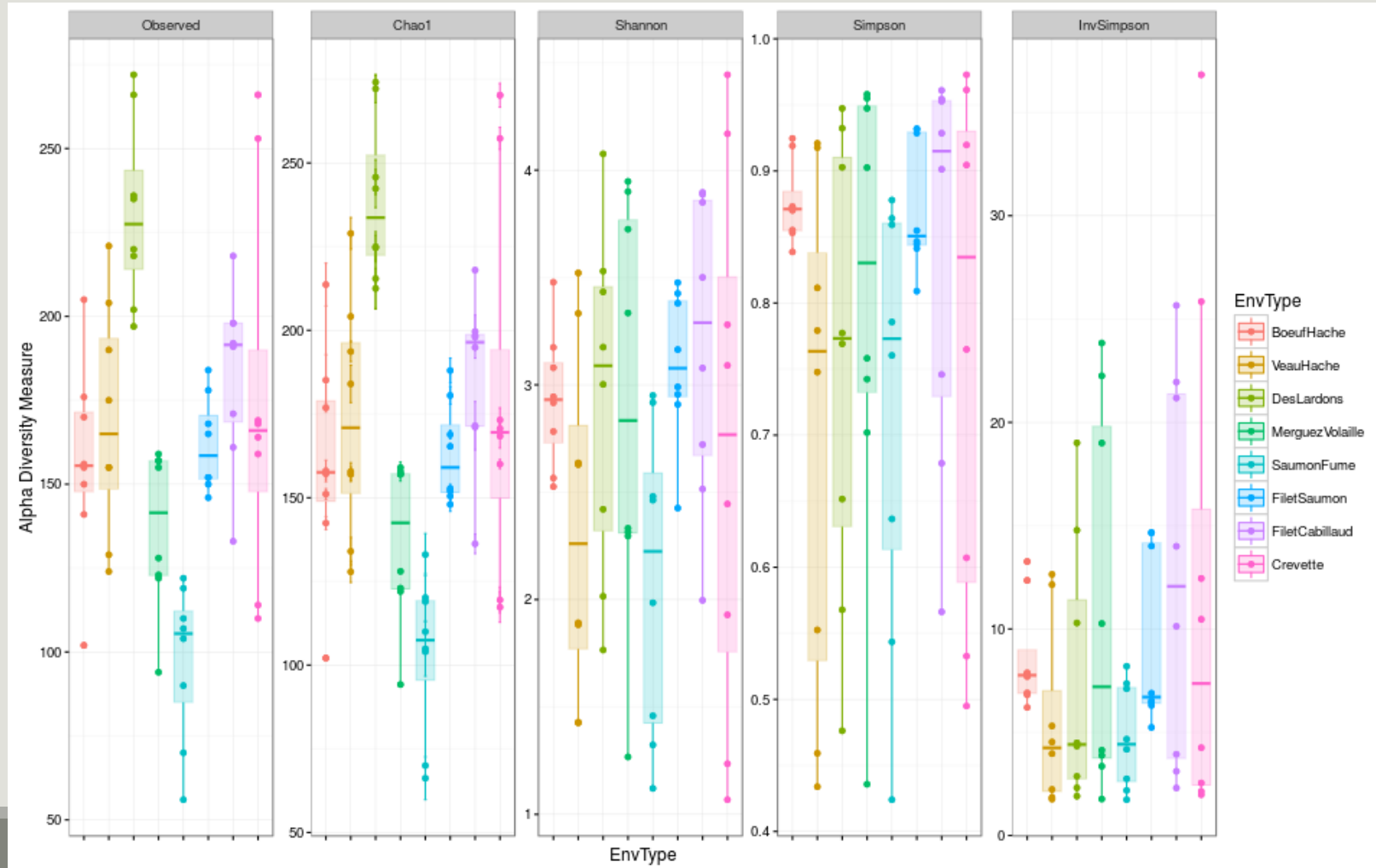
```
## plot as boxplot
```

```
p <- p + geom_boxplot(aes(fill = EnvType), alpha=0.2)
```

```
plot(p)
```

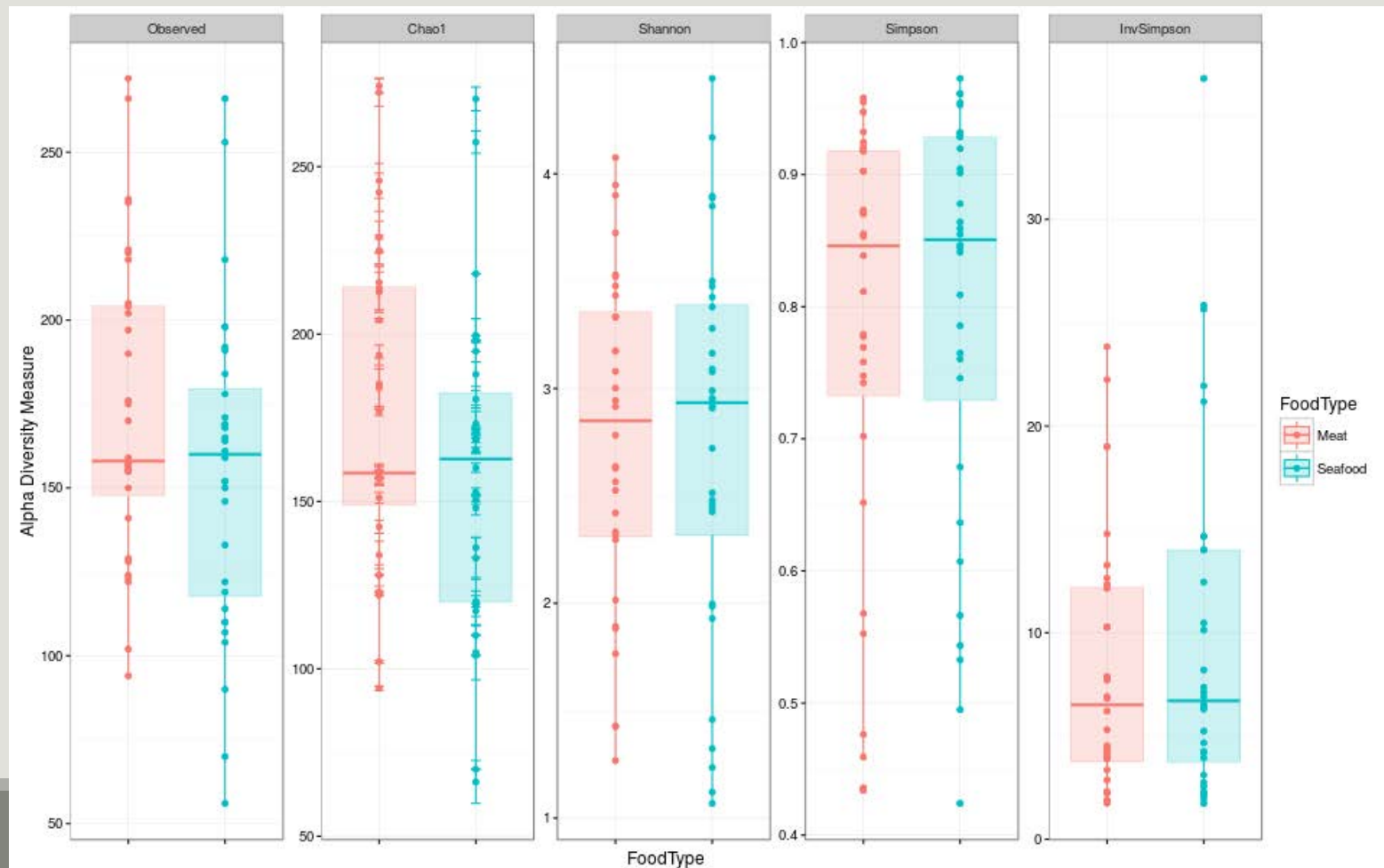


# Exploring biodiversity : $\alpha$ -diversity



# Exploring biodiversity : $\alpha$ -diversity

→ Try with FoodType instead of EnvType



# Exploring biodiversity : $\alpha$ -diversity

---

Numeric values of  $\alpha$ -diversities are given by `estimate_richness` (used internally by `plot_richness`)

```
alpha.diversity <- estimate_richness(food, measures = c("Observed",  
"Chao1", "Shannon"))
```

```
head(alpha.diversity)
```

```
##           Observed      Chao1  se.chao1  Shannon  
## DLT0.LOT08      210  210.0000    0.0000  2.016038  
## DLT0.LOT05      221  254.7857   13.3895  1.798009  
## DLT0.LOT03      226  226.0000    0.0000  3.455284  
## DLT0.LOT07      221  221.0000    0.0000  2.982161  
## DLT0.LOT06      278  278.0000    0.0000  3.209521  
## DLT0.LOT01      281  281.0000    0.0000  4.106852
```

```
write.table(alpha.diversity, "myfile.txt")
```

# Exploring biodiversity : $\alpha$ -diversity

---

A quick ANOVA : tests if observed richness is significantly different in function of EnvType

```
data <- cbind(sample_data(food), alpha.diversity)
food.anova <- aov(Observed ~ EnvType, data)
summary(food.anova)
##           Df  Sum Sq  Mean Sq  F value    Pr(>F)
## EnvType    7   86922   12417     12.49  1.63e-09 ***
## Residuals 56   55686     994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is a significant effect of environment type on richness

# Exploring biodiversity : $\alpha$ -diversity

---

A quick ANOVA : tests if Shannon indices is significantly different in function of EnvType

```
food.anova <- aov(Shannon ~ EnvType, data)
summary(food.anova)
##           Df  Sum Sq  Mean Sq  F value  Pr(>F)
## EnvType     7    7.98    1.139    1.767    0.112
## Residuals  56   36.12    0.645
```

EnvType effect on Shannon diversity is not significant

# Exploring biodiversity : $\alpha$ -diversity

---

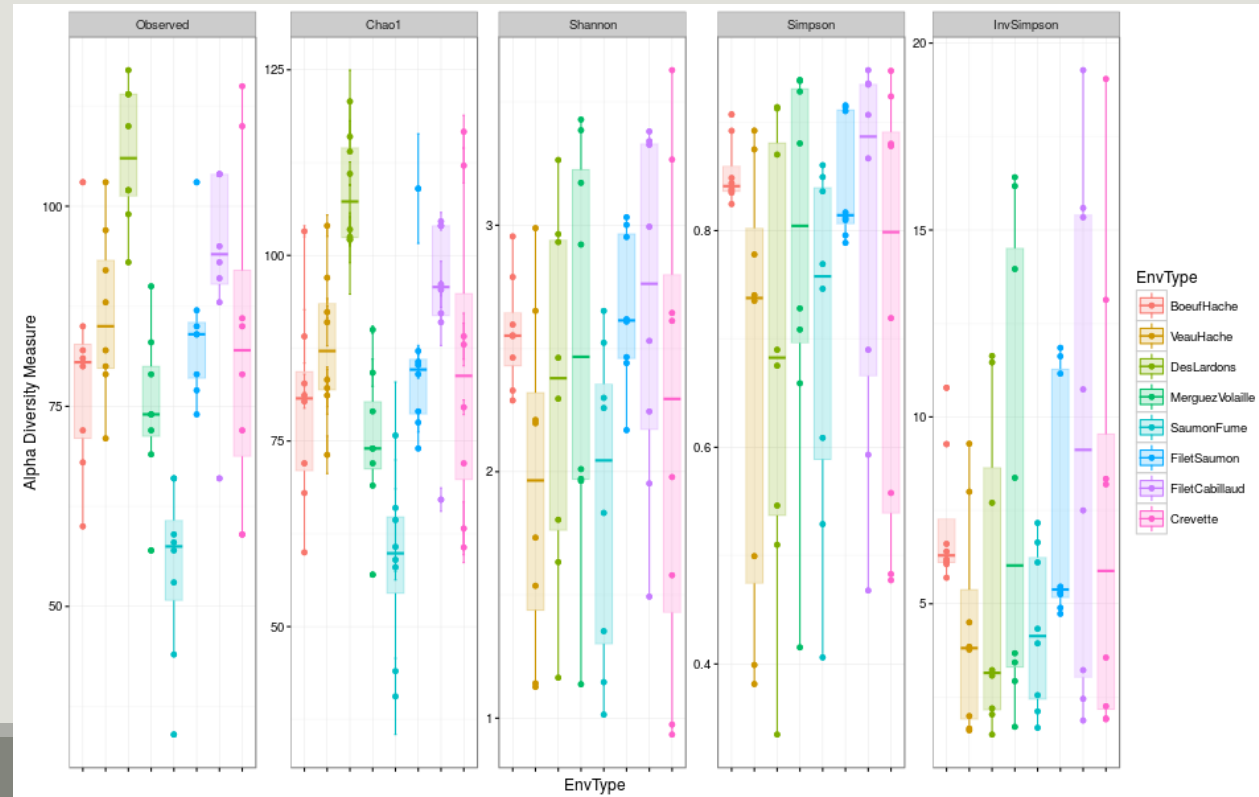
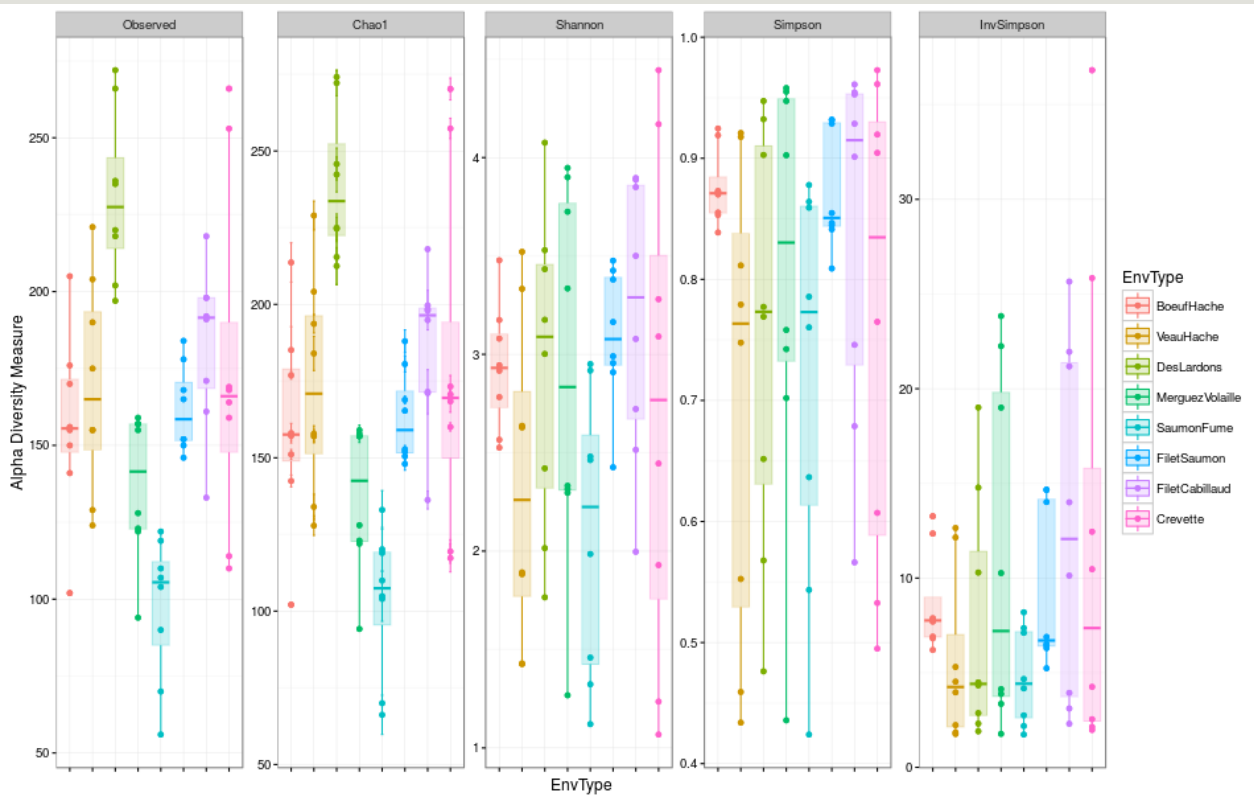
## Interpretation

- Many taxa observed in **Deslardons** (high Chao1, high Observed)...
- ...but low Shannon and Inverse-Simpson
- ➔ communities dominated by a few abundant taxa
  
- Environments differ a lot in terms of richness...
- ...but not so much in terms of Shannon diversity
- ➔ Effective diversities are quite similar

# Exploring biodiversity : $\alpha$ -diversity

**WARNING** : Many diversities (richness, Chao) **depend a lot on rare OTUs**. Do not trim rare OTUs before computing them as it can drastically alter the result.

$\alpha$ -diversity: without (left) and with (right) trimming on rare OTU ( total abundance < 500)



# Biodiversity analysis

---

β-DIVERSITY INDICES

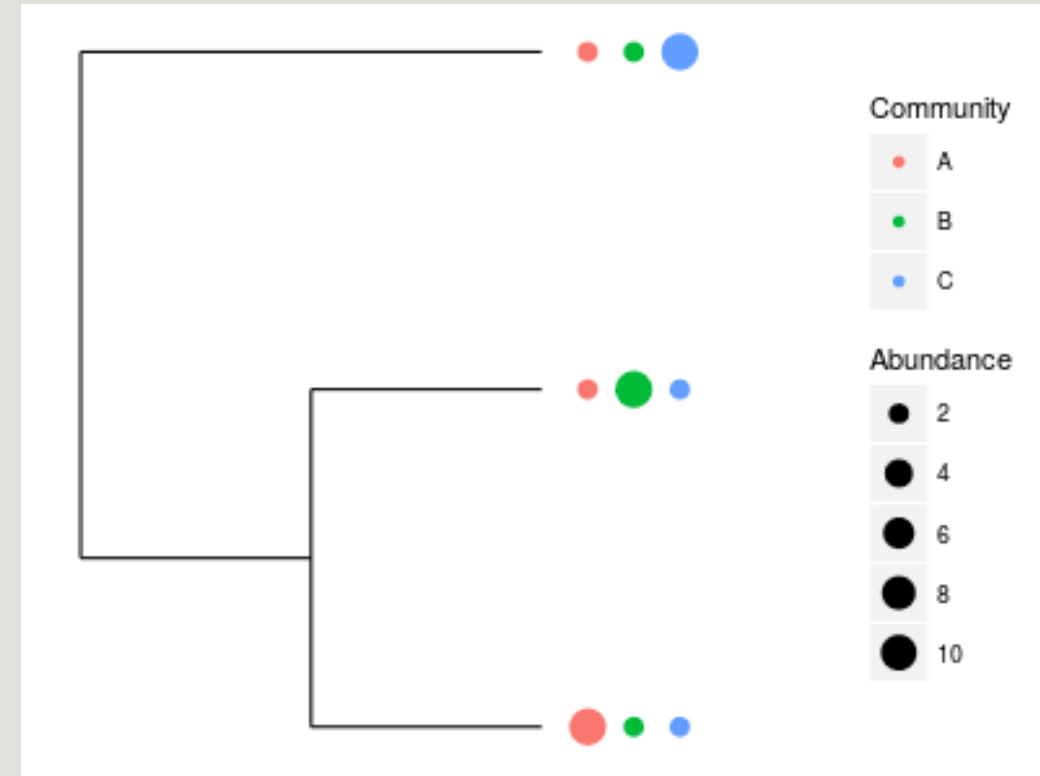


# Exploring biodiversity : $\beta$ -diversity

Many diversities (both compositional and phylogenetic) offered by Phyloseq through the generic distance function.

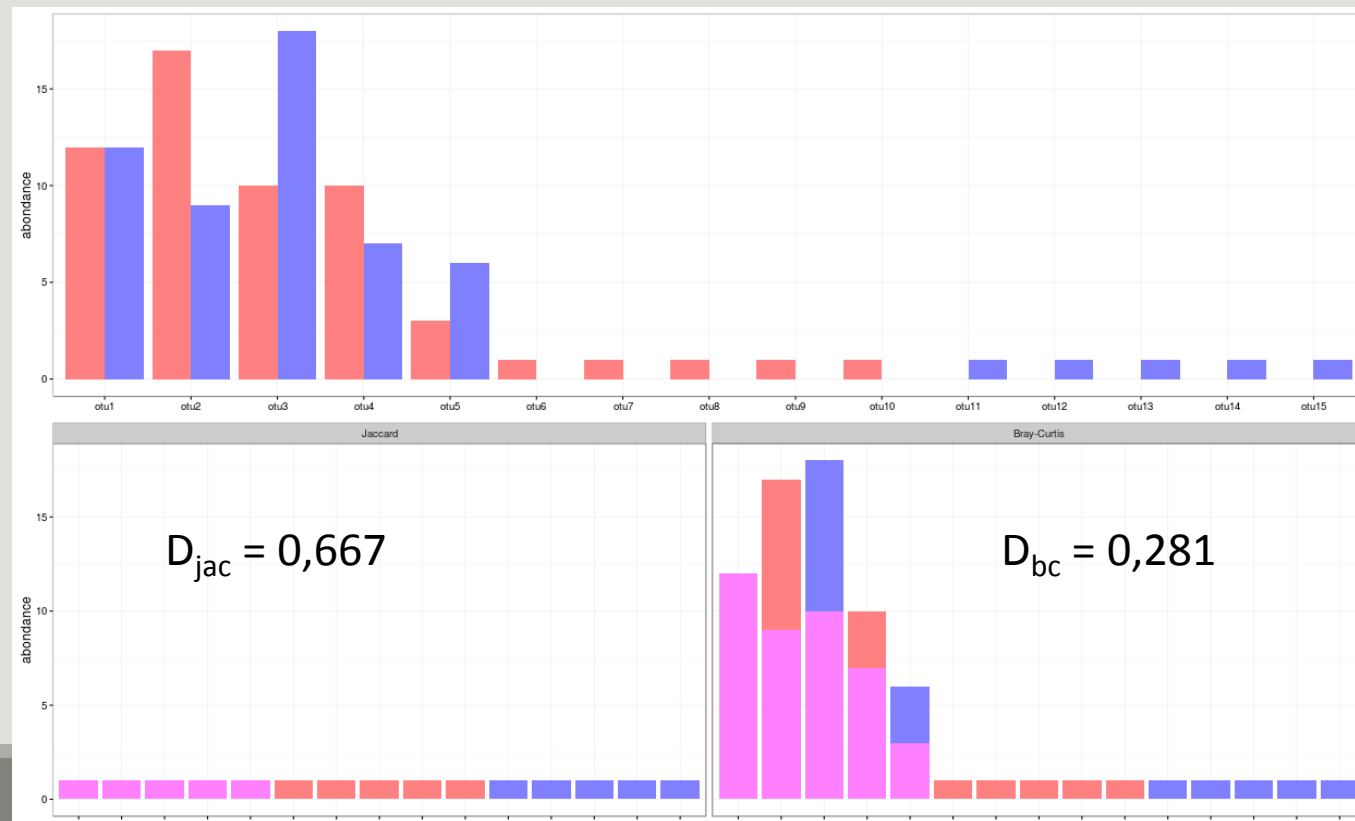
Different dissimilarities capture different features of the communities:

- qualitatively, communities are very similar
- quantitatively they are very different
- phylogenetically two communities seems to be closer than the third one.



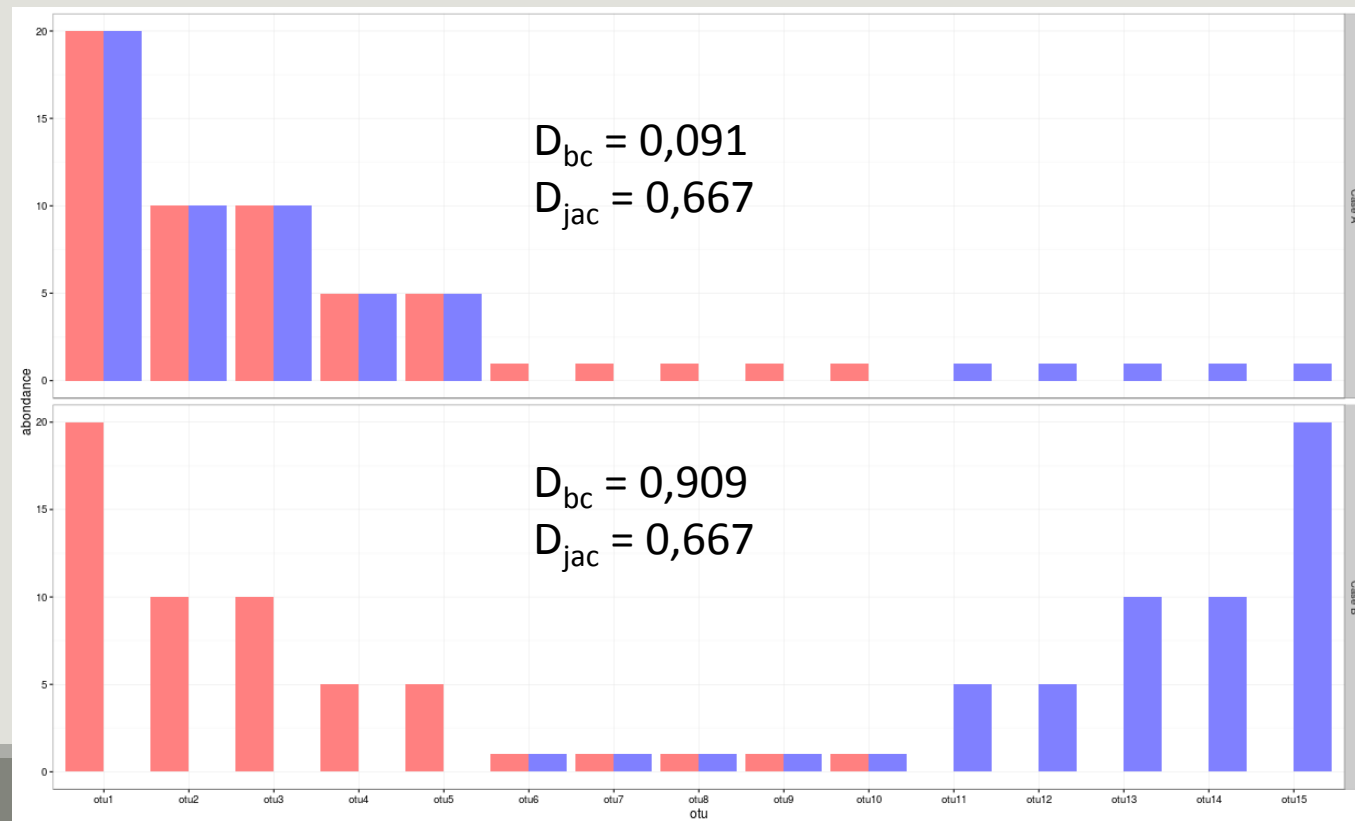
# Exploring biodiversity : $\beta$ -diversity

Jaccard	Bray-Curtis
Fraction of <u>species</u> specific to either 1 or 2	Fraction of the <u>community</u> specific to 1 or to 2



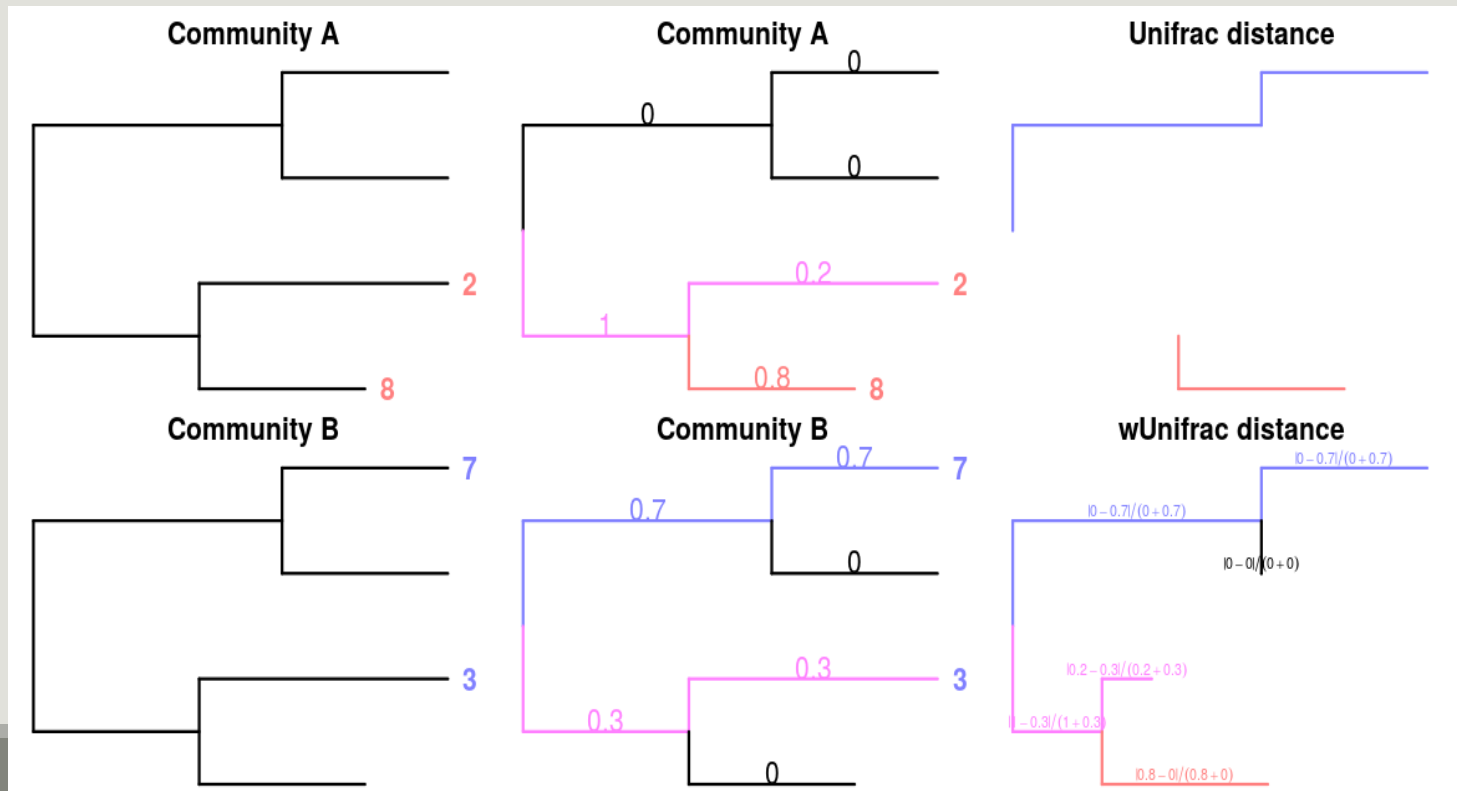
# Exploring biodiversity : $\beta$ -diversity

Jaccard	Bray-Curtis
Fraction of <u>species</u> specific to either 1 or 2	Fraction of the <u>community</u> specific to 1 or to 2



# Exploring biodiversity : $\beta$ -diversity

Unifrac	Weigthed-Unifrac
Fraction of <u>the tree</u> specific to either 1 or 2	Fraction of the <u>diversity</u> specific to 1 or to 2



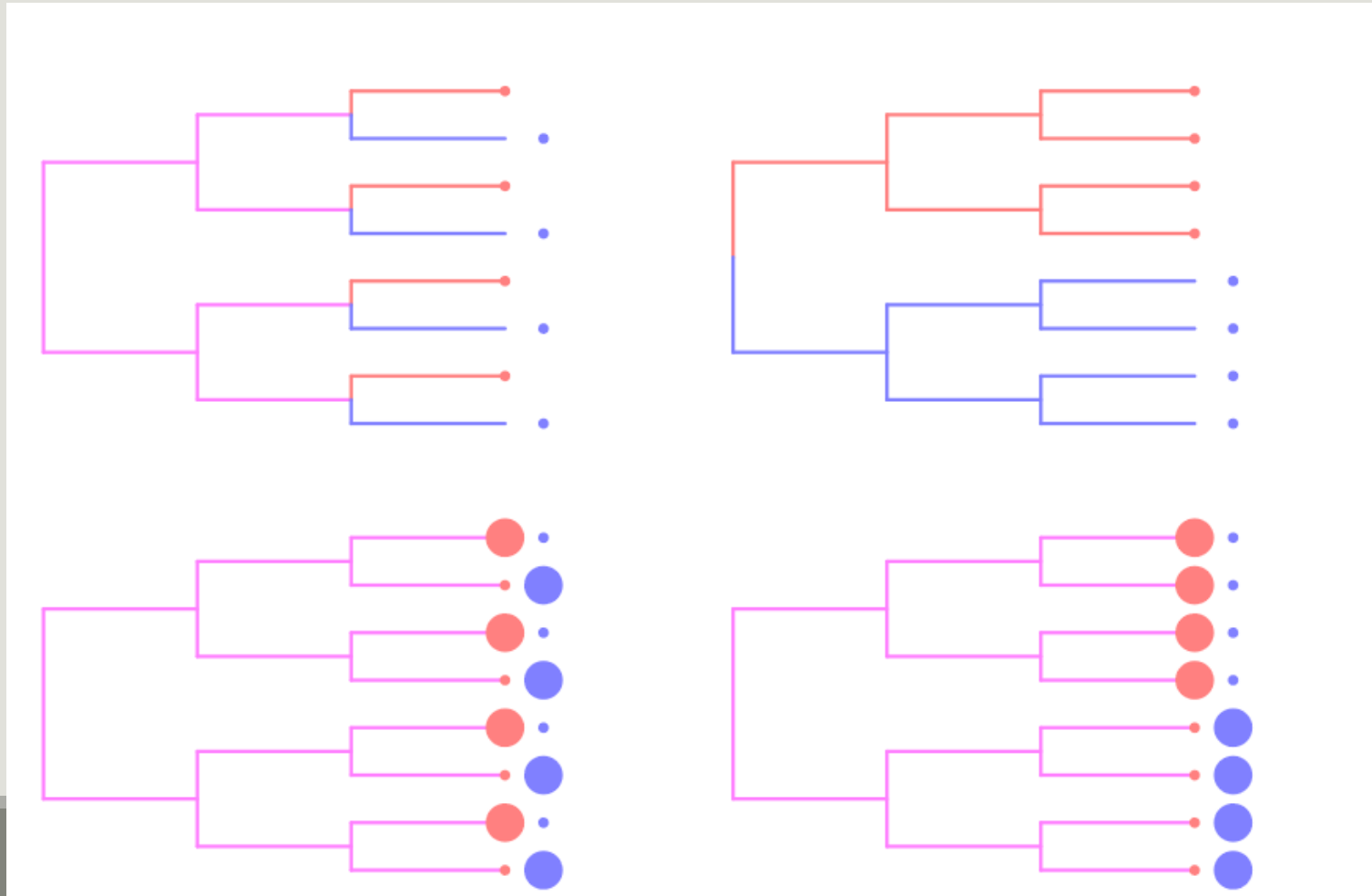
If all branch length are equal to 1, only branch present in at least one community are taken into account :

$$Unifrac = \frac{\sum specific\_branch\_length}{\sum all\_branch\_length} = 0.6$$

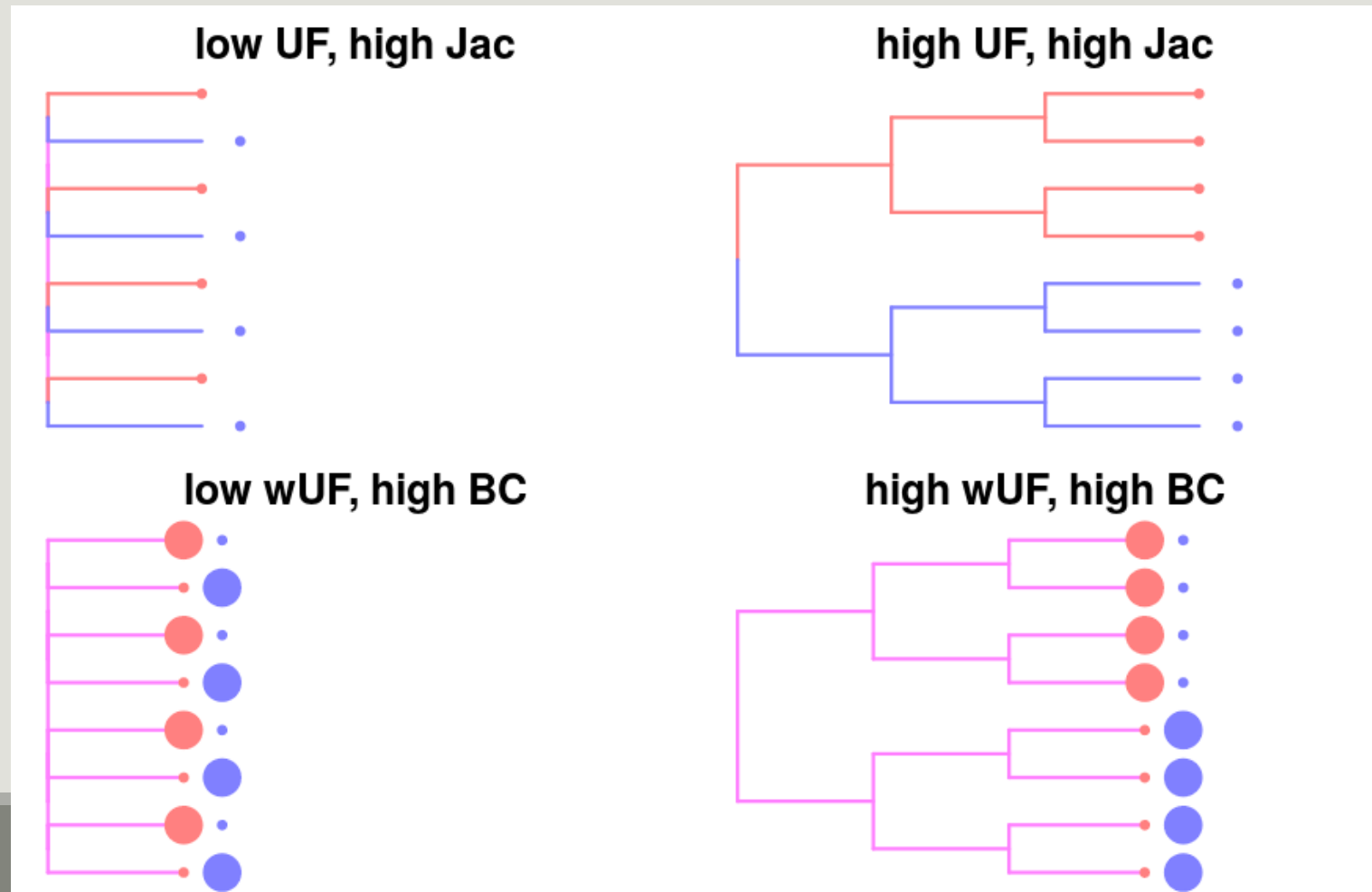
$$WUnifrac = \frac{\sum reduced\_branch\_length}{\sum non\_reduced\_branch\_length} = 0.74$$

# Exploring biodiversity : $\beta$ -diversity

→ What do you conclude in terms of Jaccard, Bray Curtis, Unifrac and weighted Unifrac values?



# Exploring biodiversity : $\beta$ -diversity



# Exploring biodiversity : $\beta$ -diversity

---

Dissimilarities are computed with distance

```
dist.bc <- distance(food, method = "bray") ## Bray-Curtis
```

All available distances are available with

```
distanceMethodList
```

```
## $UniFrac
```

```
## [1] "unifrac" "wunifrac"
```

```
## $DPCoA
```

```
## [1] "dpcoa"
```

```
## $JSD
```

```
## [1] "jsd"
```

```
## $vegdist
```

```
## [1] "manhattan" "euclidean" "canberra" "bray" "kulczynski" "jaccard" "gower"  
"altGower" "morisita" "horn"
```

```
## [11] "mountford" "raup" "binomial" "chao" "cao"
```

```
## $betadiver
```

```
## [1] "w" "-1" "c" "wb" "r" "I" "e" "t" "me" "j" "sor" ...
```

# Exploring biodiversity : $\beta$ -diversity

---

- Bray-Curtis, Jaccard and Kulczynski good at detecting underlying ecological gradients
  - Morisita-Horn, Cao and Jensen-Shannon good at handling different sample sizes
  - All take value in [0; 1] except JSD and Cao.
- Compute Jaccard, Bray Curtis, Unifrac and weighted Unifrac distance

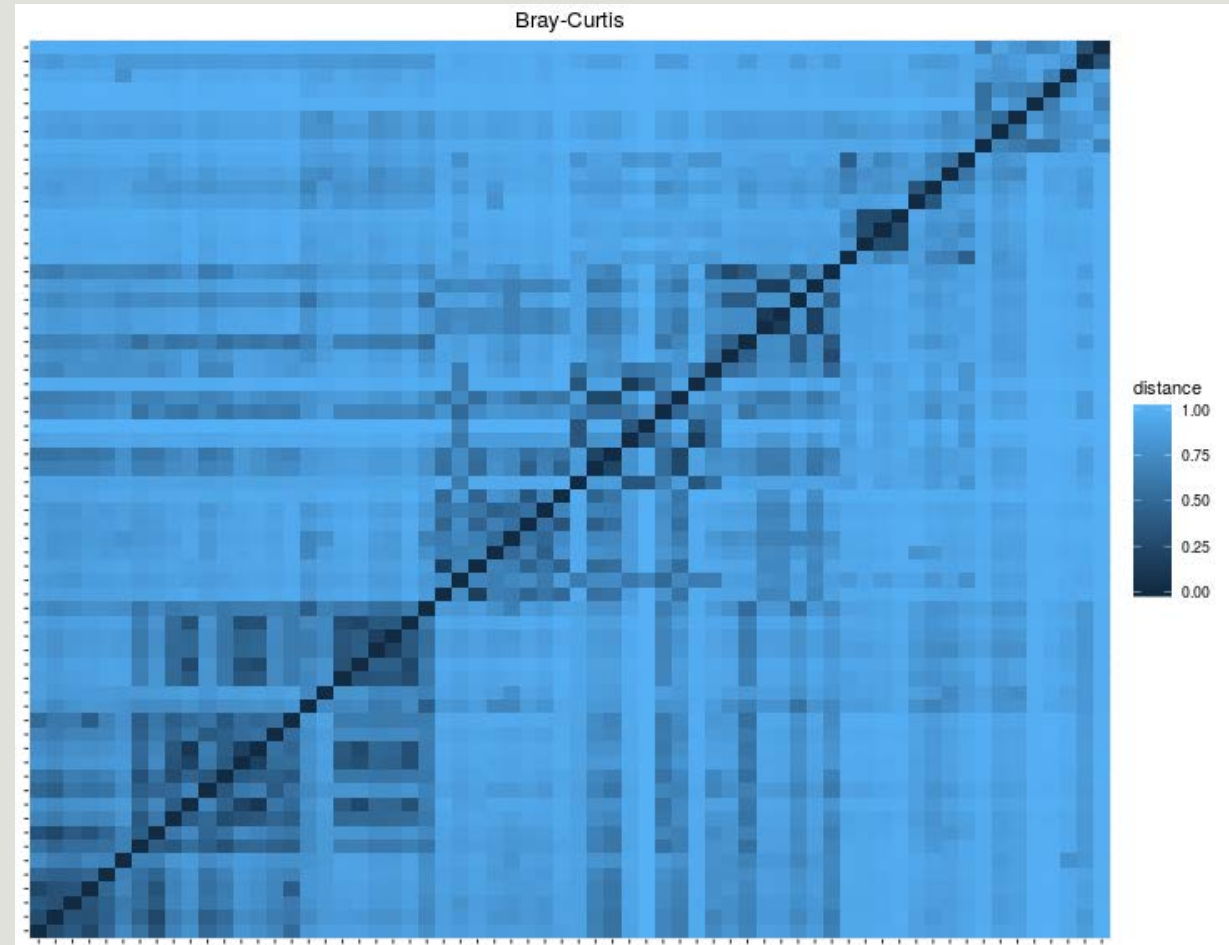


# Exploring biodiversity : $\beta$ -diversity

$\beta$ -diversity indices can be visualized thanks to heatmap.

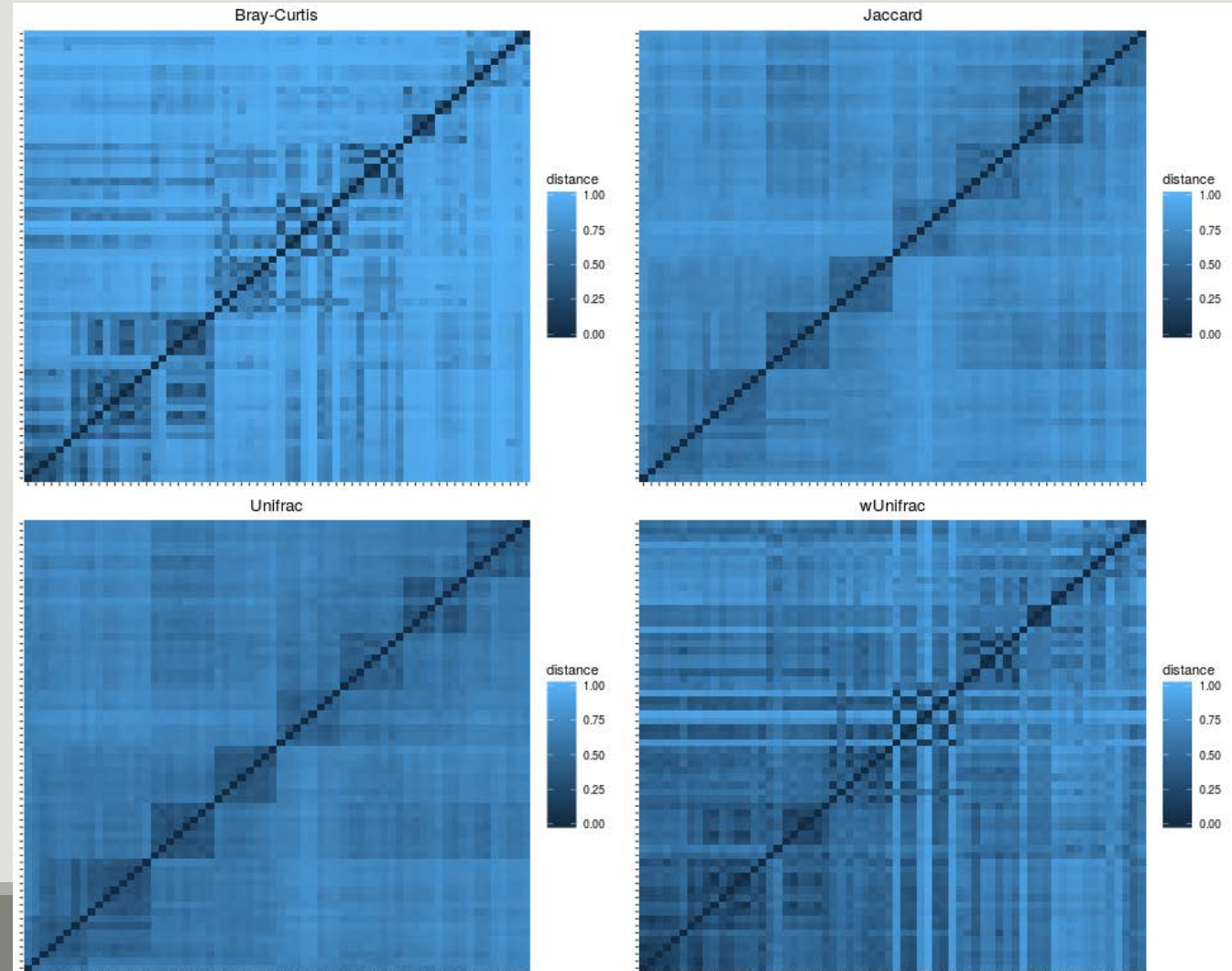
```
## custom function to implement  
p <- plot_dist_as_heatmap(dist.bc,  
title = "Bray-Curtis")  
plot(p)
```

→ Try it on the other distance matrices



# Exploring biodiversity : $\beta$ -diversity

- Jaccard lower than Bray-Curtis  
→ abundant taxa are not shared
- Jaccard higher than Unifrac  
→ communities' taxa are distinct but phylogenetically related
- Unifrac higher than weighted Unifrac  
→ abundant taxa in both communities are phylogenetically close.



# Exploring biodiversity : $\beta$ -diversity

---

- In general, **qualitative** diversities **are more sensitive to factors that affect presence/absence** of organisms (such as pH, salinity, depth, etc) and therefore useful to study and define bioregions (regions with little or no overlap between them)...
- ... whereas **quantitative** distances **focus on factors that affect relative changes** (seasonal changes, nutrient availability, concentration of oxygen, depth, etc) and therefore useful to monitor communities over time or along an environmental gradient.

Different distances capture different features of the samples.

There is no "one size fits all"

# Exploring the structure

---

# Exploring the structure

---

ORDINATION

# Exploring the structure : ordination

---

- Each community is described by OTUs abundances
- OTUs abundance maybe correlated
- PCA finds linear combinations of OTUs that
  - are uncorrelated
  - capture well the variance of community composition

But variance is not a very good measure of  $\beta$ -diversity

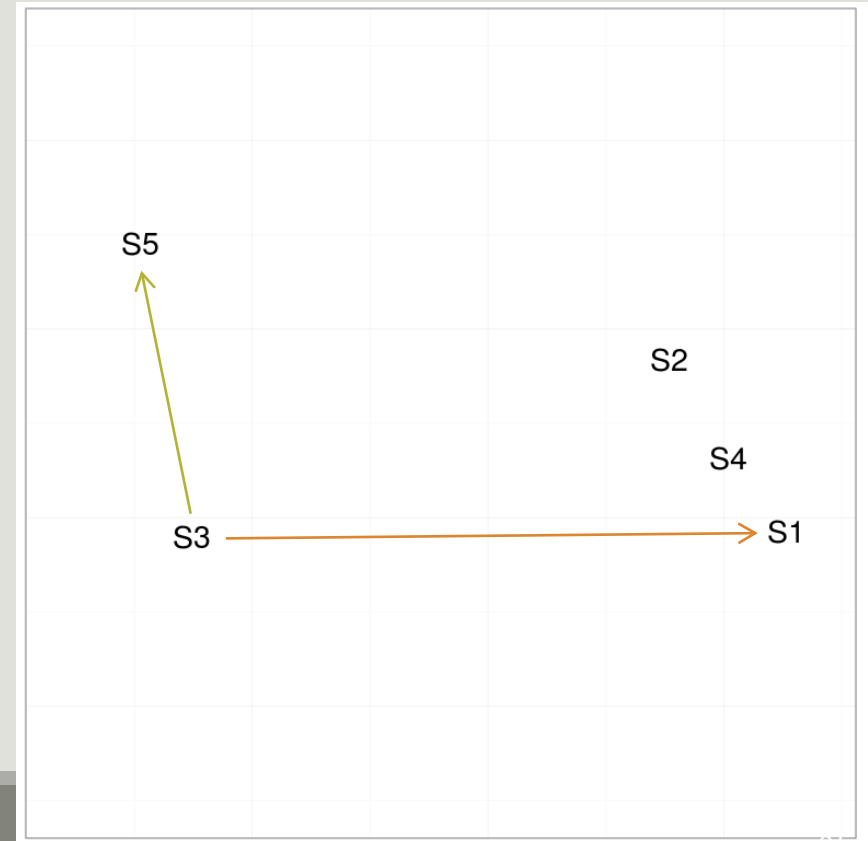
# Exploring the structure : ordination

The Multidimensional Scaling (MDS or PCoA) is equivalent to a principal component Analysis (PCA) but preserves the  $\beta$ -diversity instead of the variance.

The MDS try to represent samples in two dimensions

➔ The samples ordination.

	Distance Matrix				
	S1	S2	S3	S4	S5
S1	0.00	2.21	6.31	0.99	7.50
S2	2.21	0.00	5.40	1.22	5.74
S3	6.31	5.40	0.00	5.75	3.16
S4	0.99	1.22	5.75	0.00	6.64
S5	7.50	5.74	3.16	6.64	0.00



# Exploring the structure : ordination

---

Ordination is done through the `ordinate` function.

You can pass the distance either by name (and `phyloseq` will call `distance`) :

```
ord <- ordinate(food, method = "MDS", distance = "bray")
```

or by passing a distance matrix directly (useful if you already computed it)

```
dist.bc <- distance(food, method = "bray")
```

```
ord <- ordinate(food, method = "MDS", distance = dist.bc)
```



# Exploring the structure : ordination

The graphic is then produced with `plot_ordination`

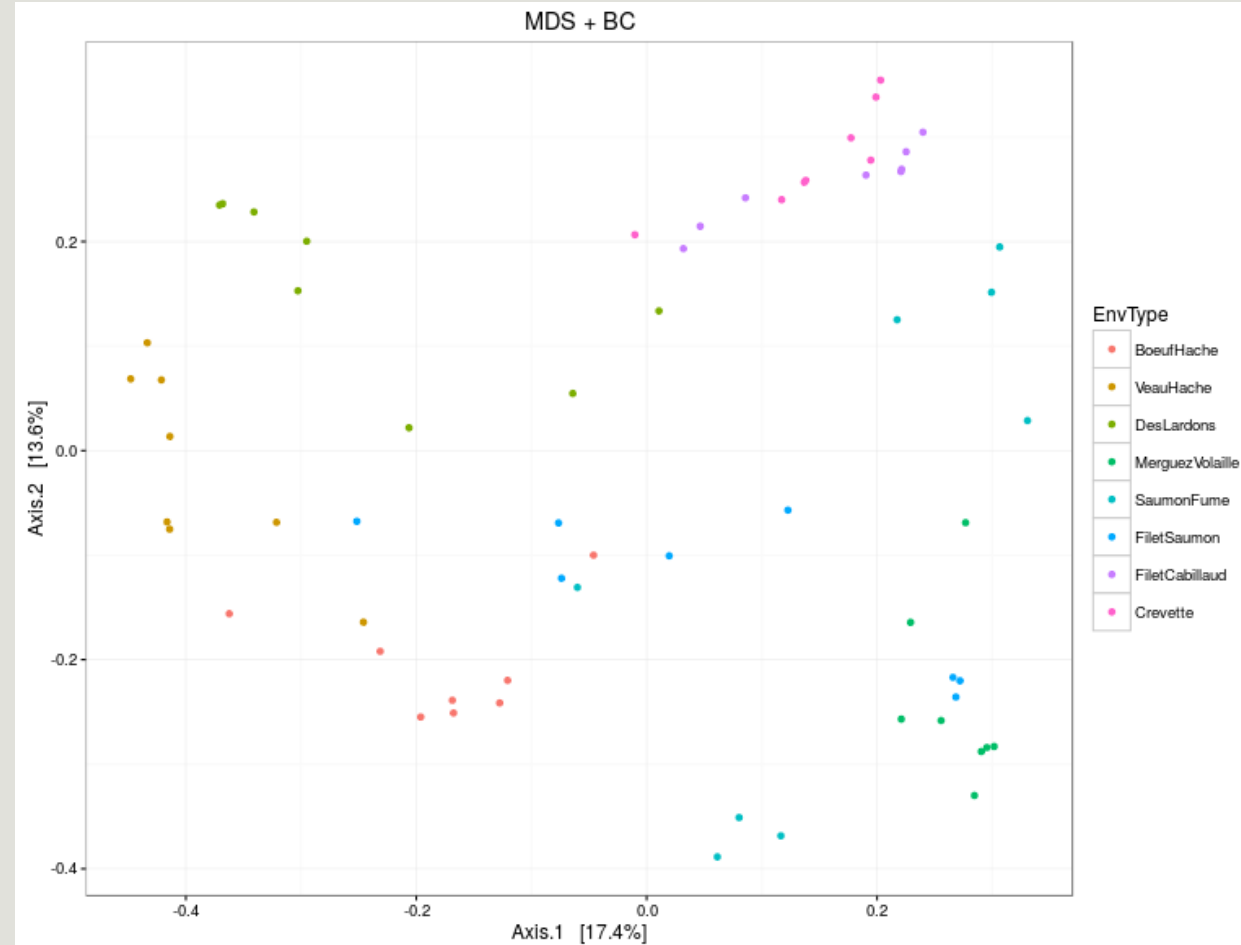
→ Try it on `food` with `ord`

→ Custom color by `EnvType`

```
p <- plot_ordination(food, ord,
color = "EnvType")

## add title and plain background
p <- p + theme_bw() +
ggtitle("MDS + BC")

plot(p)
```





# Exploring the structure : ordination

Custom `plot_samples` function built around `plot_ordination` to represent groups (extra replicate parameter)

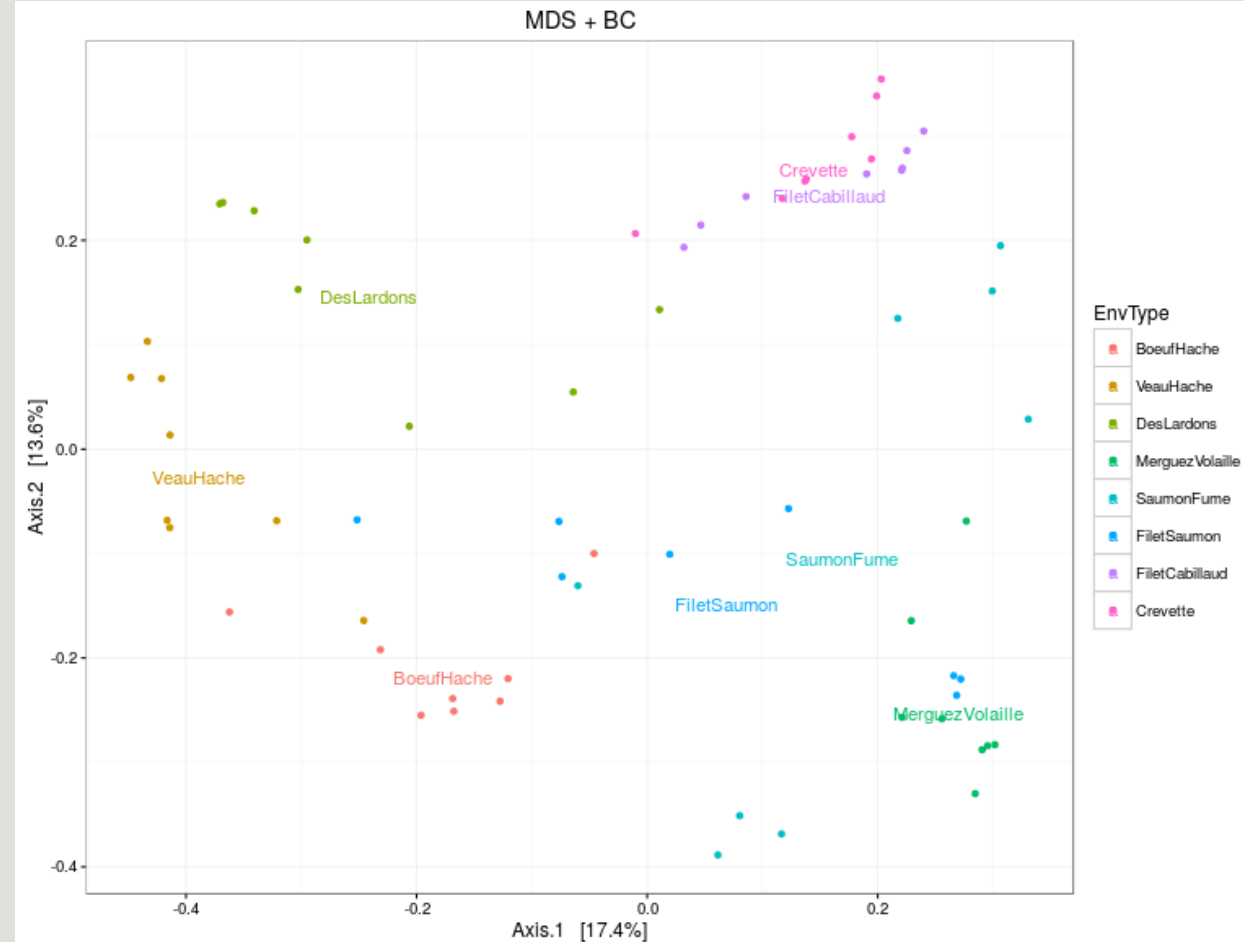
➔ Try it on food and ord, and choose `EnvType` for color and replicate

```
p <- plot_samples(food, ord,  
color = "EnvType", replicate =  
"EnvType")
```

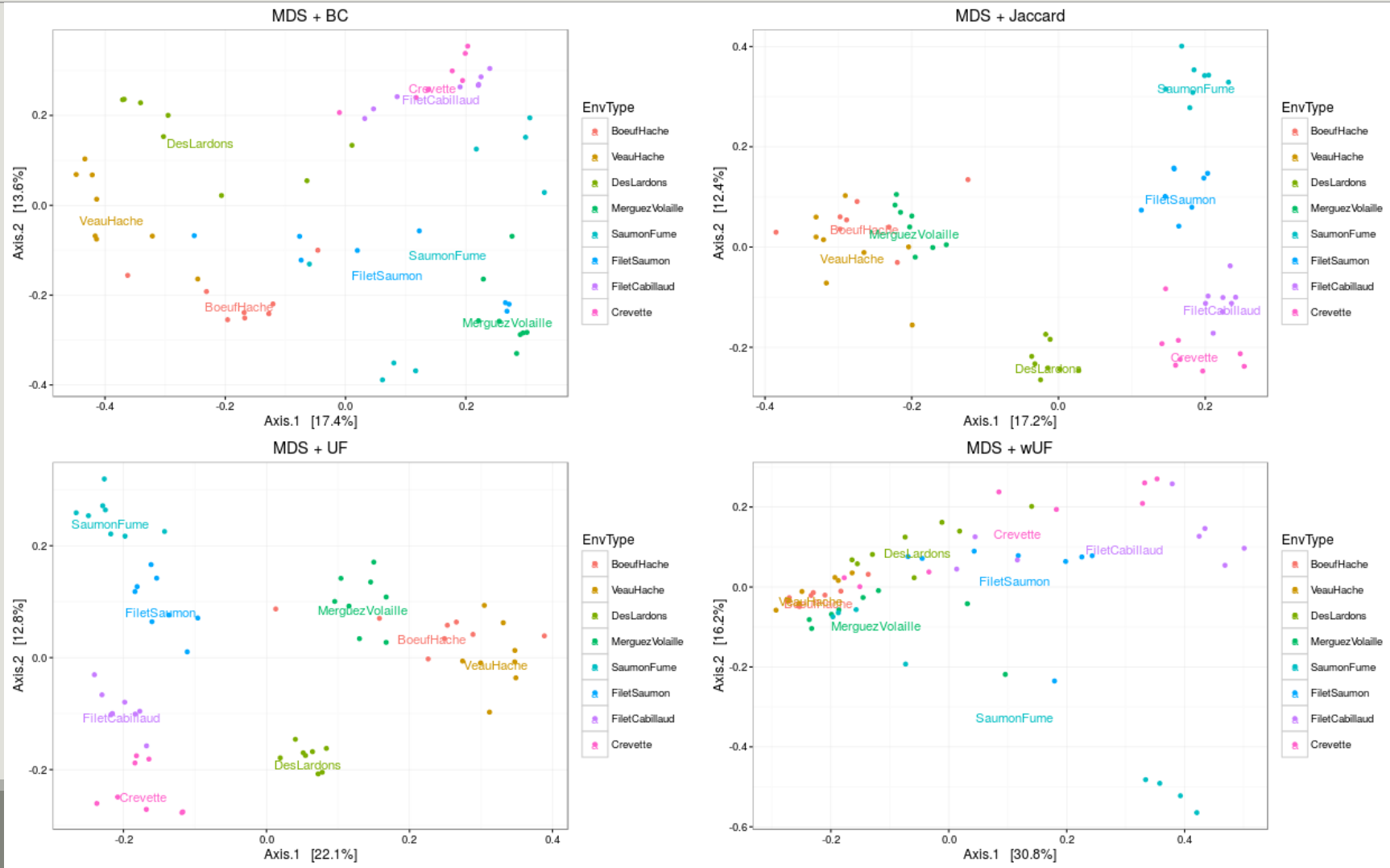
```
## add title
```

```
p <- p + theme_bw() +  
ggtitle("MDS + BC")
```

```
plot(p)
```



# Exploring the structure : ordination



# Exploring the structure : ordination

---

- Qualitative distances (Unifrac, Jaccard) separate meat products from seafood ones
  - detected taxa segregate by origin.
- DesLardons is somewhere in between
  - contamination induced by sea salt.
- Quantitative distances (wUnifrac) exhibit a gradient meat – seafood (on axis 1) with DesLardons in the middle and a gradient SaumonFume - everything else on axis 2.
- Large overlap between groups in terms of relative composition but less so in term of species composition (a side effect of undersampling?)
- Note the difference between wUniFrac and Bray-Curtis for the distances between BoeufHache and VeauHache
- Warning The 2-D representation captures only part of the original distances.

# Exploring the structure

---

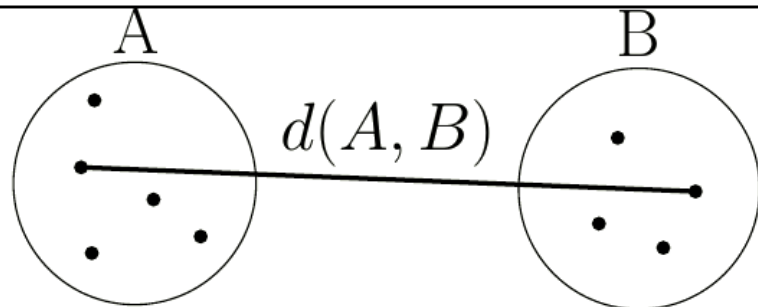
CLUSTERING

# Exploring the structure : clustering

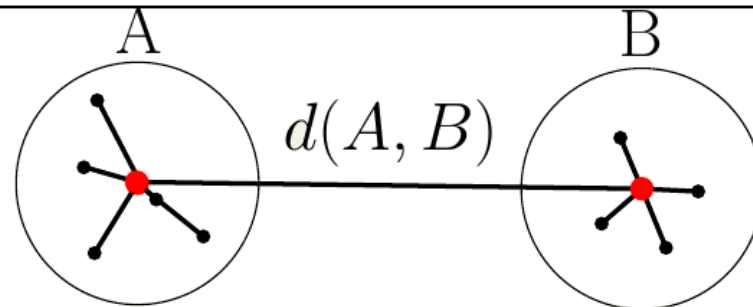
The clustering aims to represent samples in a tree based on a distance matrix and a linkage function:

- Complete linkage: tends to produce compact, spherical clusters and guarantees that all samples in a cluster are similar to each other.
- Ward: tends to also produces spherical clusters but has better theoretical properties than complete linkage.
- single: friend of friend approach, tends to produce banana-shaped or chains-like clusters.

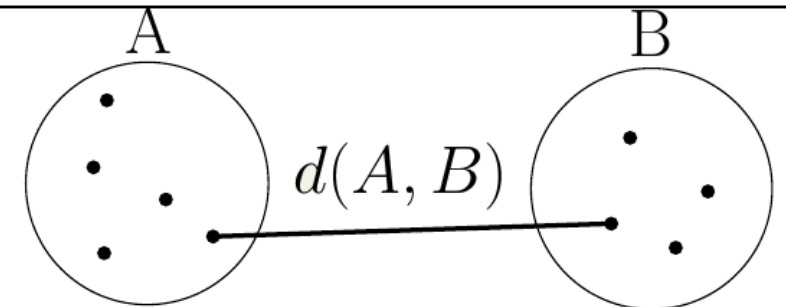
Complete



Ward



Single



# Exploring the structure : clustering

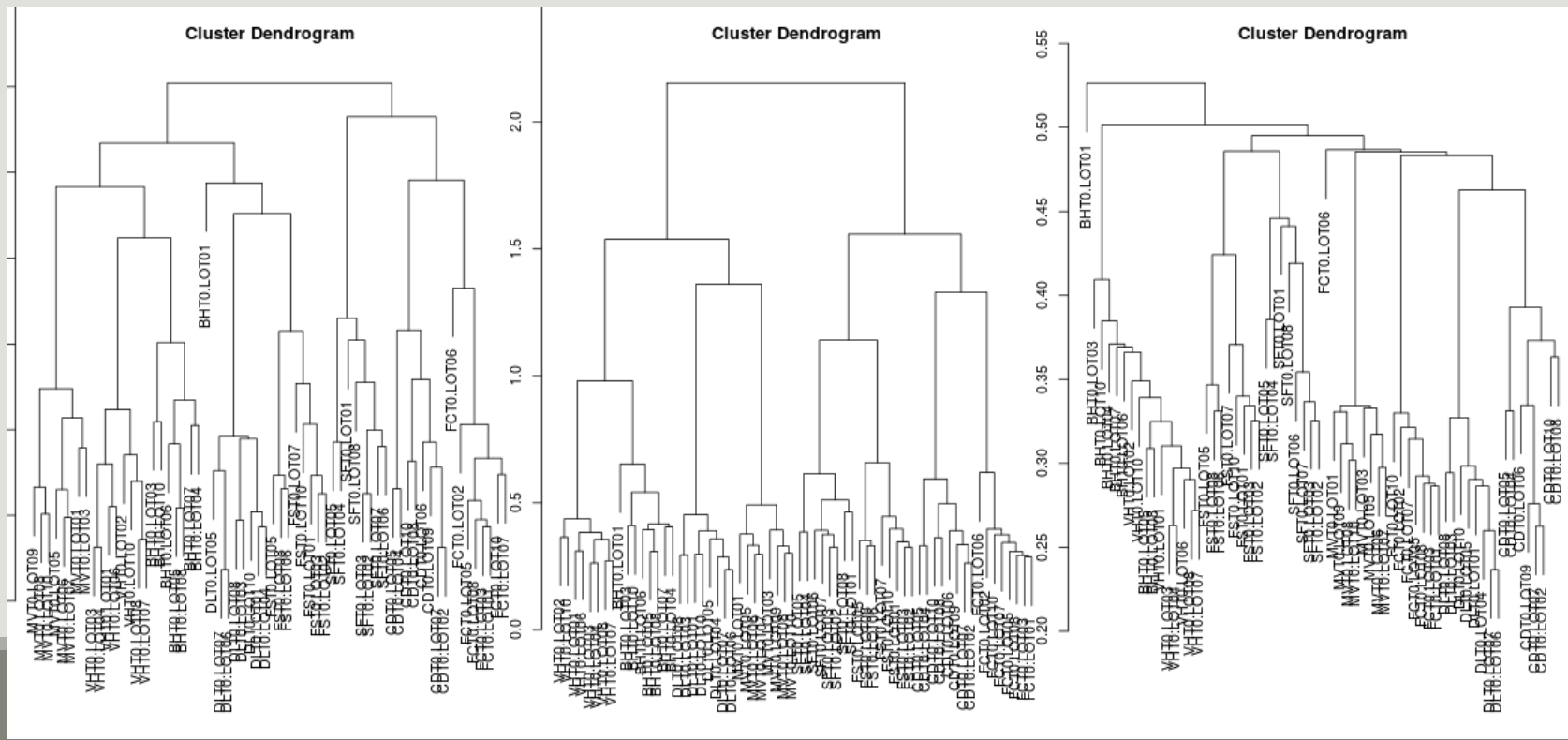
---

- Choose a distance (among Jaccard, Bray-Curtis, Unifrac, etc)
- Choose a linkage function
  - complete (“complete”): tends to produce **compact**, spherical clusters and guarantees that all samples in a cluster are similar to each other.
  - ward (“ward.D2”): tends to also produces **spherical** clusters but has better theoretical properties than complete linkage.
  - single (“single”): friend of friend approach, tends to produce **banana-shaped** or chains-like clusters.
- Feed to **hclust** and plot

```
clustering <- hclust(distance.matrix, method = "linkage.function")  
plot(clustering)
```

# Exploring the structure : clustering

```
par(mfcol = c(1, 3)) ## To plot the three clustering trees side-by-side
plot(hclust(dist.uf, method = "complete"))
plot(hclust(dist.uf, method = "ward.D2"))
plot(hclust(dist.uf, method = "single"))
```







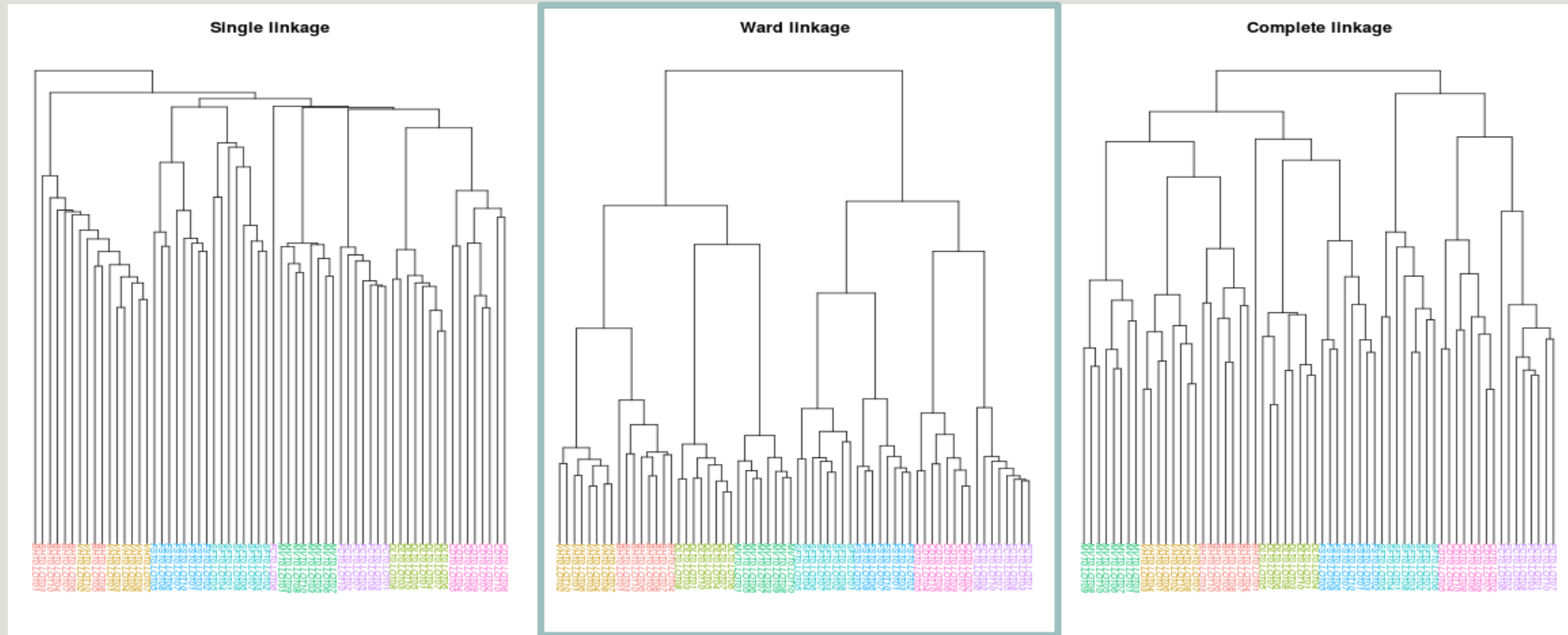
# Exploring the structure : clustering

---

With some effort (see companion R script), we can produce better dendrograms and color sample by food type.

```
## Env types
envtype <- get_variable(food, "EnvType")
## automatic color palette: one color per different sample type
palette <- hue_pal()(length(levels(envtype)))
## Map sample type to color
tipColor <- col_factor(palette, levels = levels(envtype))(envtype)
## Change hclust object to phylo object and plot
par(mar = c(0, 0, 2, 0))
dist.uf <- distance(food, method = "unifrac")## if not already done
clust.uf <- as.phylo(hclust(dist.uf, method = "complete"))
plot(clust.uf, tip.color = tipColor, direction = "downwards",
main = "Ward linkage"))
```

# Exploring the structure : clustering



- Crevette
- FiletCabillaud
- FiletSaumon
- SaumonFume
- MerguezVolaille
- DesLardons
- BoeufHache
- VeauHache

# Exploring the structure : clustering

---

## Remarks

- Consistent with the ordination plots, clustering works quite well for the UniFrac distance for some linkage (Ward)
- Clustering is based on the whole distance whereas ordination represents parts of the distance (the most it can with 2 dimensions)

# Exploring the structure

---

HEATMAP

# Exploring the structure : Heatmap

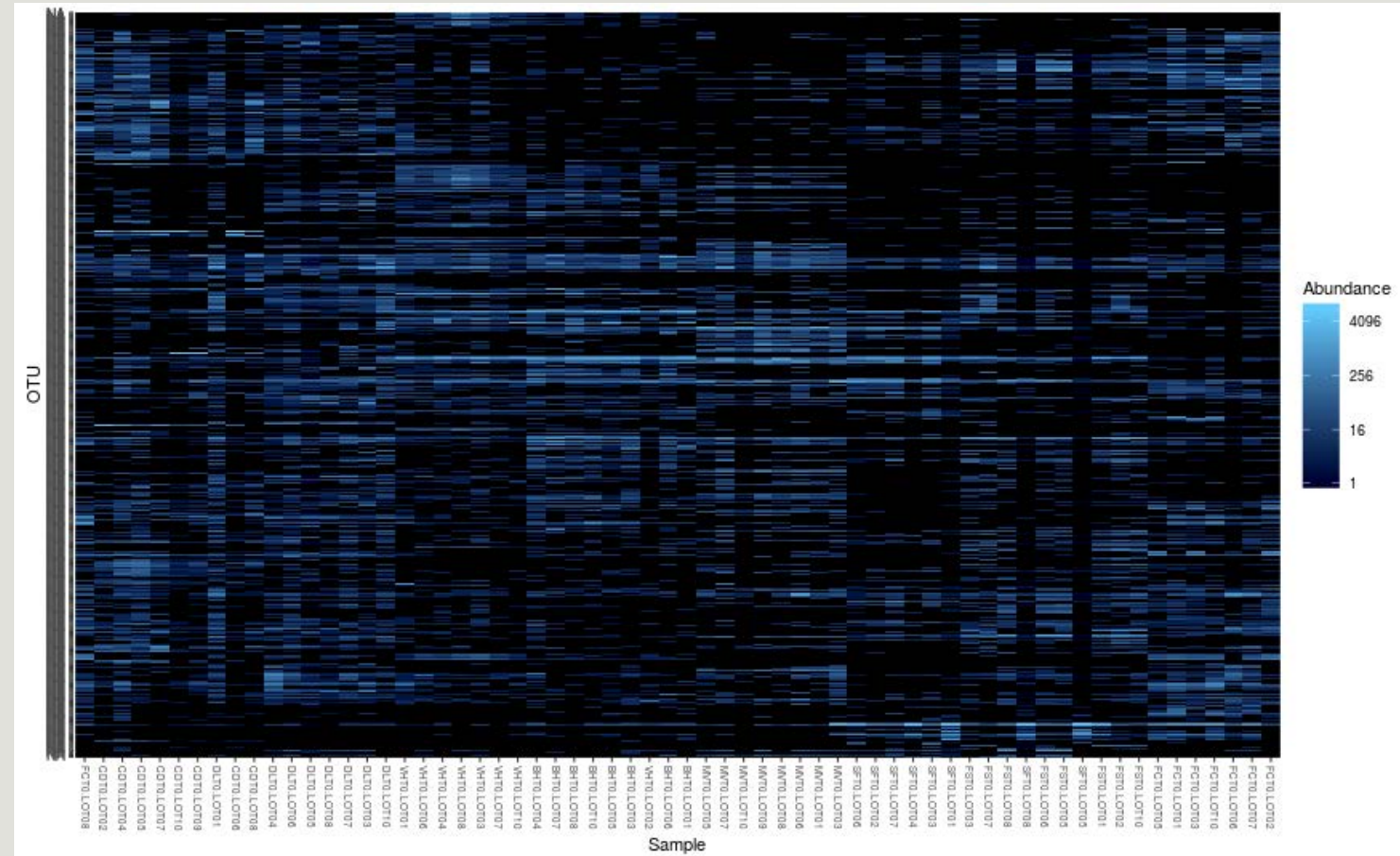
---

- The heatmap is an other representation of the distance matrix.
- It try to reveal it there is a structure between group of OTU and group of samples.
- `plot_heatmap` is a versatile function to visualize the count table.
  - Finds a meaningful order of the samples and the OTUs
  - Allows the user to choose a custom order
  - Allows the user to change the color scale
  - Produces a `gplot2` object, easy to manipulate and customize

# Exploring the structure : Heatmap

→ Try it on food

```
p <- plot_heatmap(food)
```

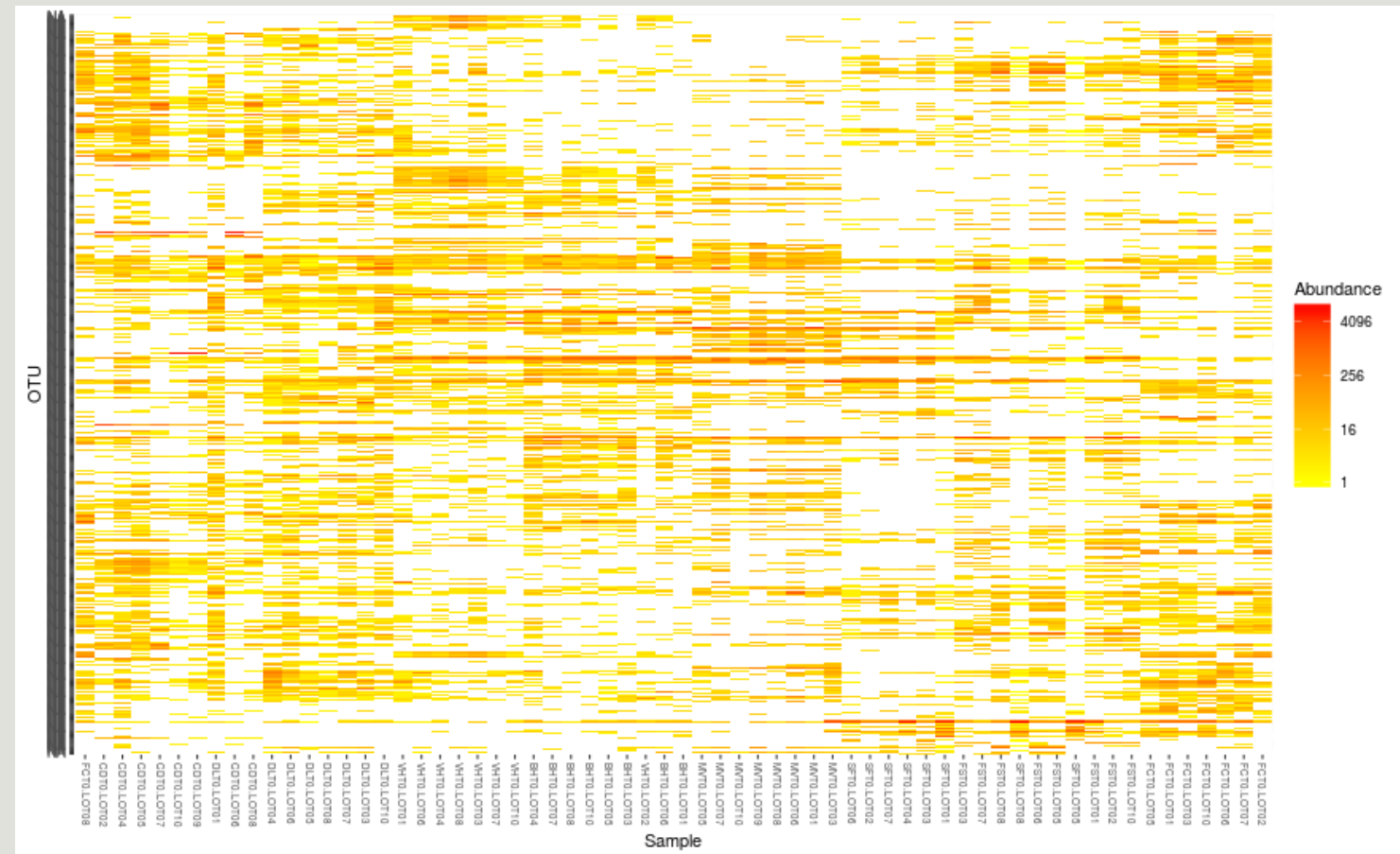


# Exploring the structure : Heatmap

→ Change color scale in `plot_heatmap`

```
p <- plot_heatmap(food, low =  
  "yellow", high = "red",  
  na.value = "white")
```

```
plot(p)
```



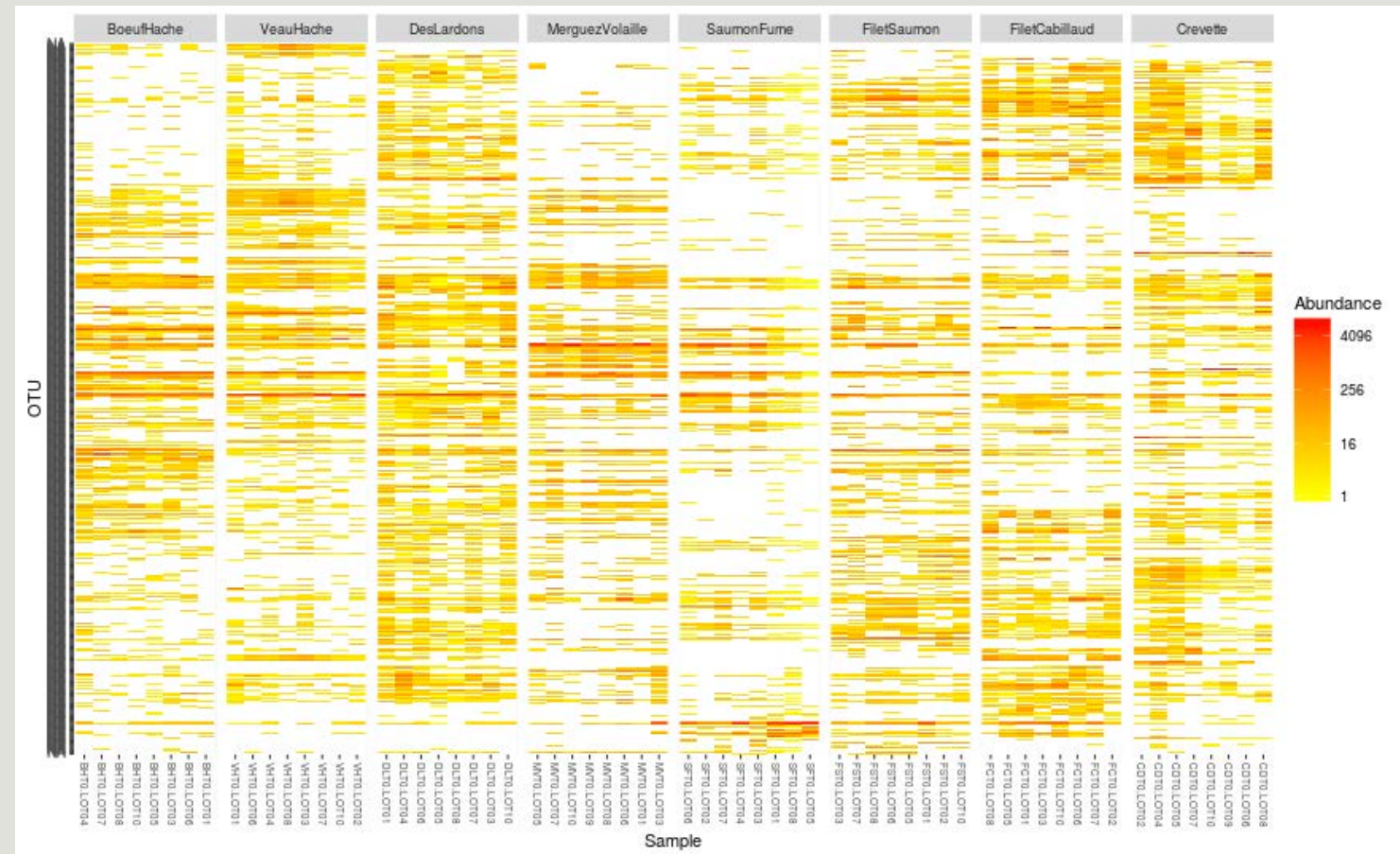
# Exploring the structure : Heatmap

```
p <- plot_heatmap(food, low =  
  "yellow", high = "red",  
  na.value = "white")
```

➔ Facet your plot by EnvType

```
p <- p + facet_grid(~EnvType,  
  scales = "free_x")
```

```
plot(p)
```





# Exploring the structure : Heatmap

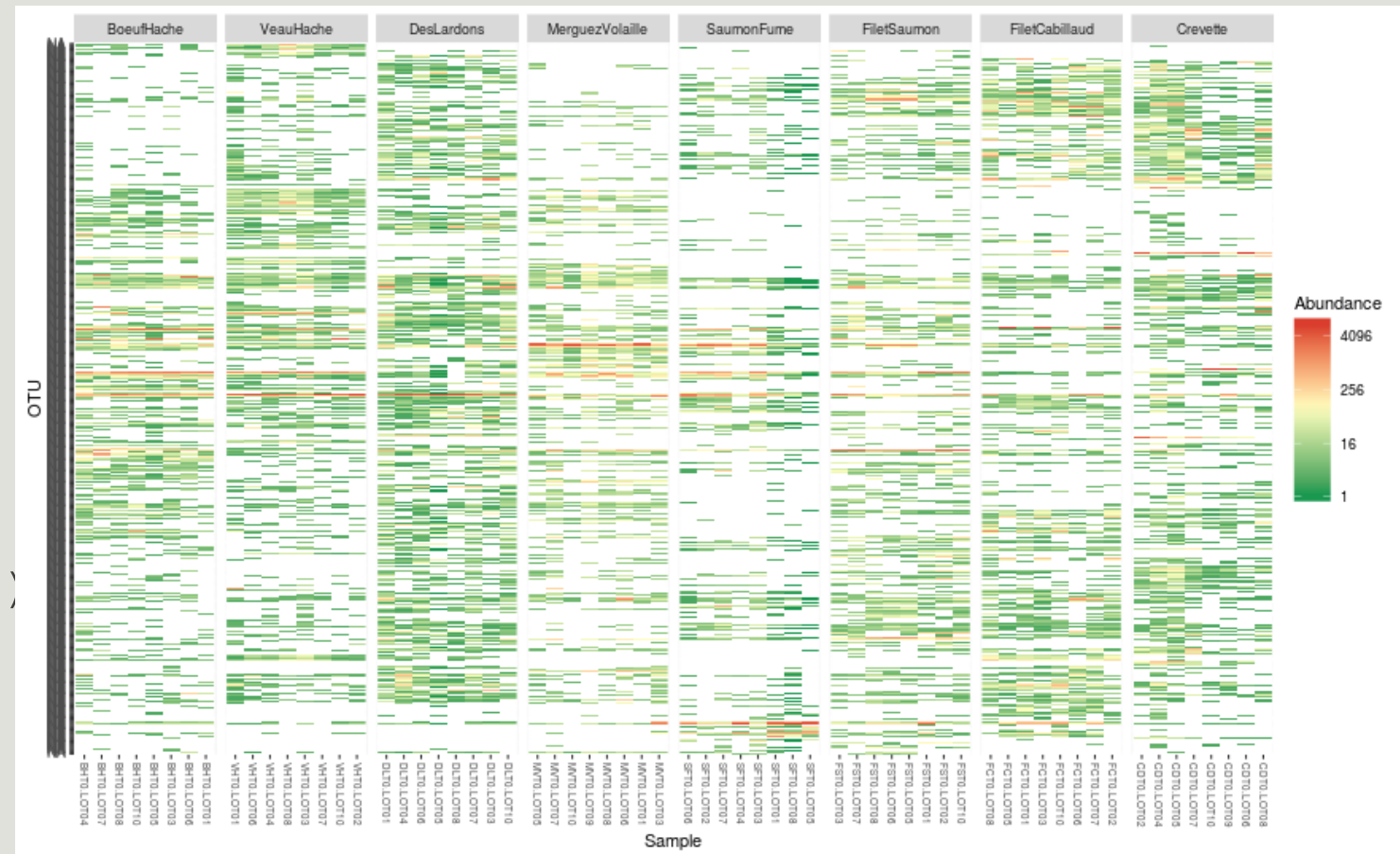
→ Use `scale_fill_gradient` layers of `ggplot2` to change color scale

```
p <- plot_heatmap(food)

p <- p +
  scale_fill_gradient2(low =
    "#1a9850", mid = "#ffffbf",
    high = "#d73027", na.value =
    "white", trans = log_trans(4),
    midpoint = log(100, base = 4))

p <- p + facet_grid(~EnvType,
  scales = "free_x")

plot(p)
```



# Exploring the structure : Heatmap

---

- Block-like structure of the abundance table
- Interaction between (groups of) taxa and (groups of) samples
- Core and condition-specific microbiota
- ➔ Classification of taxa and use of custom taxa order to highlight structure

# Diversity partitioning

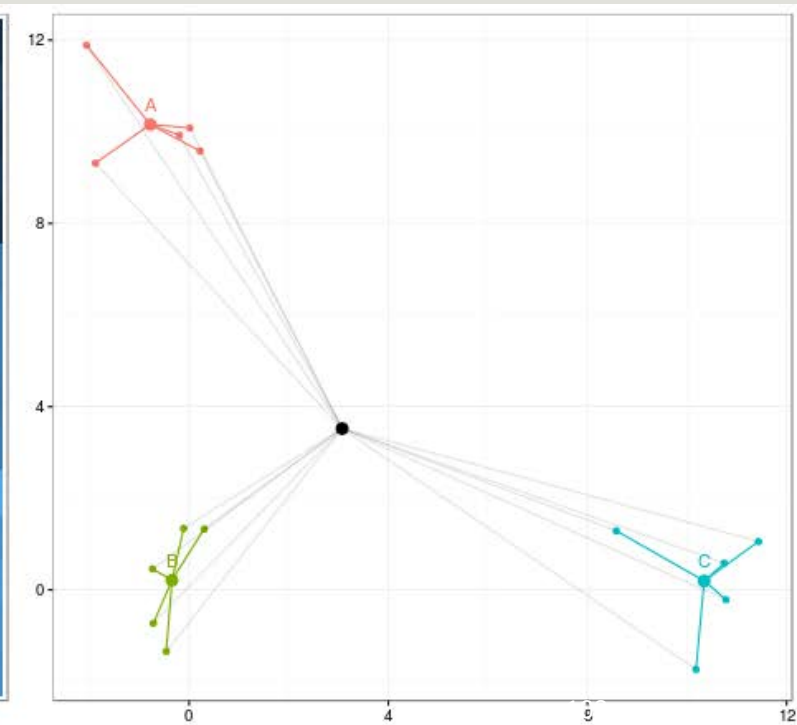
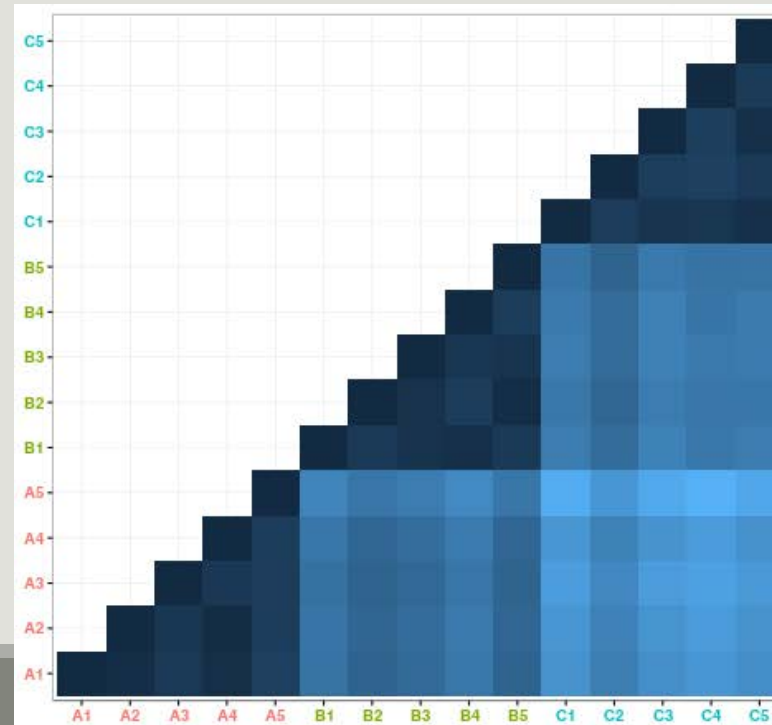
---

# Diversity partitioning

Are the structures seen linked to metadata ? Do the metadata have an effect on our communities composition ?

To answer these questions, **multivariate analyses** that :

- tests **composition differences** of communities from different groups **using a distance matrix**
- compares **within group** to **between group** distances



# Diversity partitioning : CAP

---

Constrained Analysis of Principal Coordinates (CAP) tries to :

- Find associations between community composition and environmental variables (pH, group)
- Quantify differences between groups of samples

How it works : Regress a distance matrix against some covariates using the standard R syntax for linear models.

- Project distance matrix on metadata variables → communities are constrained to depend on metadata
- Look if constrained distance fit to non constrained distance.

```
## convert sample_data to data.frame
metadata <- as(sample_data(food), "data.frame")
cap <- capscale(dist.uf ~ EnvType, data = metadata)
```

# Diversity partitioning : CAP

---

cap

```
## Call: capscale(formula = dist.uf ~ EnvType, data = metadata)
##
##              Inertia  Proportion  Rank
## Total            12.1280
## Real Total       12.1600          1.0000
## Constrained       7.6570          0.6297    7
## Unconstrained     4.5030          0.3703   56
## Imaginary        -0.0320              6
## Inertia is squared Unknown distance
##
## Eigenvalues for constrained axes:
##   CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7
## 2.5546 1.4630 1.1087 0.8954 0.7159 0.4940 0.4255
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8
## 0.4161 0.2908 0.2540 0.2111 0.2066 0.2011 0.1675 0.1562
## (Showed only 8 of all 56 unconstrained eigenvalues)
```

Environment type explains roughly **63%** of the total variation between samples (as measured by Unifrac)

# Diversity partitioning : CAP

---

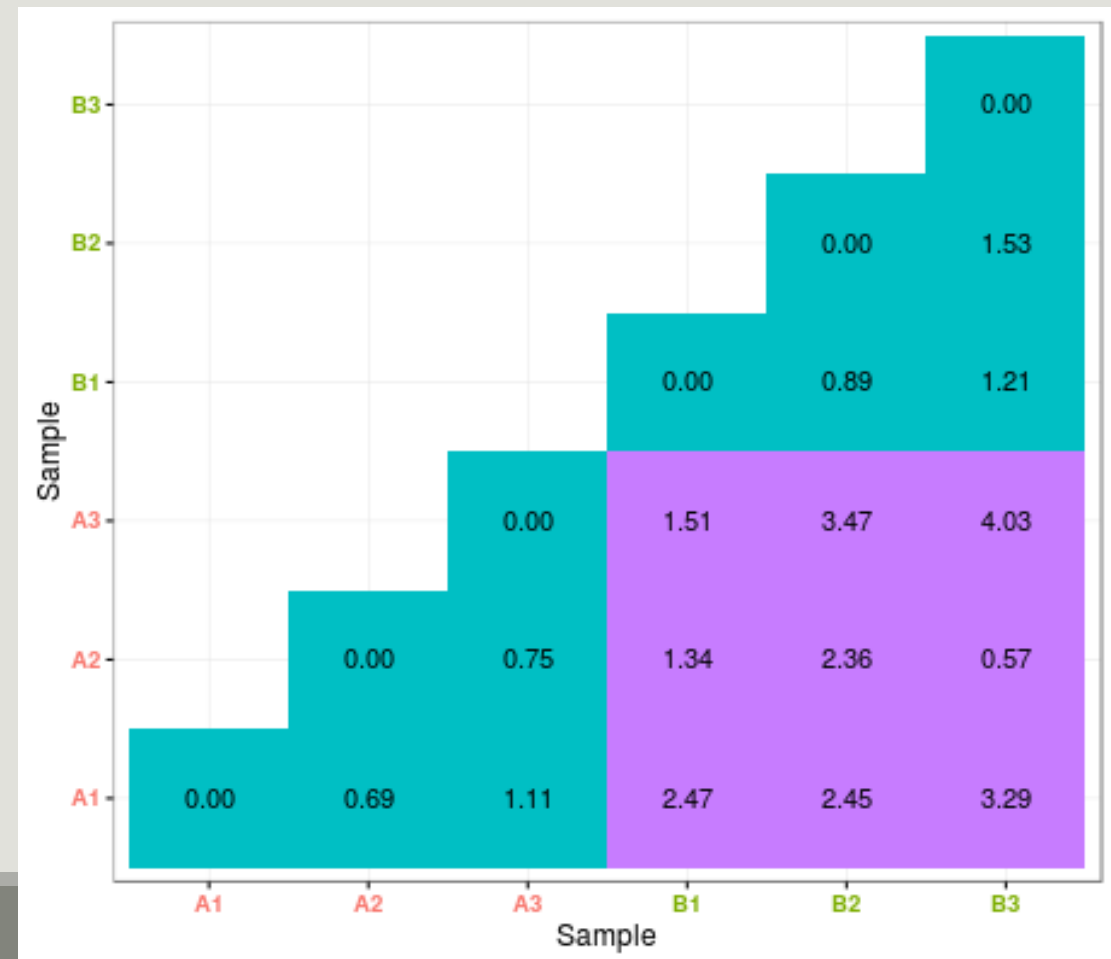
```
## test the confidence of CAP
anova <- anova(cap, permutations = 999)
## Permutation test for capscale under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: capscale(formula = dist.uf ~ EnvType, data = metadata)
## Df Variance F Pr(>F)
## Model 7 7.6571 13.603 0.001 ***
## Residual 56 4.5032
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Diversity partitioning : Multivariate ANOVA

Idea : Test **differences** in the community composition of communities **from different groups** using a **distance matrix**.

How it works ?

- No projection of distance matrix, but compute sum of square distance
- Variance analysis





# Diversity partitioning : Multivariate ANOVA

---

```
metadata <- as(sample_data(food), "data.frame")
adonis(dist.uf ~ EnvType, data = metadata, perm = 9999)
##
## Call:
## adonis(formula = dist.uf ~ EnvType, data = metadata, permutations = 9999)
##
## Permutation: free
## Number of permutations: 9999
##
## Terms added sequentially (first to last)
##
##              Df  SumsOfSqs  MeanSqs  F.Model      R2  Pr(>F)
## EnvType       7      7.6565  1.09379   13.699  0.63132 1e-04 ***
## Residuals    56      4.4713  0.07984      0.36868
## Total        63     12.1278      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 1
```

Environment type explains roughly **63%** of the total variation.

# Annexes

---

# References

---

- Chaillou, S., Chaulot-Talmon, A., Caekebeke, H., Cardinal, M., Christieans, S., Denis, C., Desmonts, M. H., Dousset, X., Feurer, C., Hamon, E., Joraud, J.-J., La Carbona, S., Leroi, F., Leroy, S., Lorre, S., Mace, S., Pilet, M.-F., Prevost, H., Rivollier, M., Roux, D., Talon, R., Zagorec, M., and Champomier-Verges, M.-C. (2015). Origin and ecological selection of core and food-specific bacterial communities associated with meat and seafood spoilage. *ISME J*, 9(5):1105{1118.
- McMurdie, P. J. and Holmes, S. (2013). phyloseq: An r package for reproducible interactive analysis and graphics of microbiome census data. *PLoS ONE*, 8(4):e61217.
- Shade, A., Jones, S. E., Caporaso, J. G., Handelsman, J., Knight, R., Fierer, N., and Gilbert, J. A. (2014). Conditionally rare taxa disproportionately contribute to temporal changes in microbial diversity. *MBio*, 5(4):e01371{e01314.

# Installing Phyloseq

---

- From bioconductor

```
## try http if https is not available
```

```
source("https://bioconductor.org/biocLite.R")
```

```
biocLite("phyloseq")
```

- From developer's website

```
install.packages("devtools") ## If not installed previously
```

```
library("devtools")
```

```
install_github("phyloseq", "joey711")
```

# Go further : phyloseq import

---

- It is possible to build a phyloseq object from plain tabular files.
- Since OTUs/sample names are not always consistent (unlike in a biom), some care must be taken.
- Otherwise the phyloseq objects consists only of OTUs and samples with **consistent names** and may end up empty.
  
- Import each component separately;
- Convert to correct base R data type (**matrix** for **otu\_table** and **tax\_table**, **data.frame** for **sample\_data**)
- Convert to phyloseq data type (**otu\_table**, **tax\_table**, **sample\_data**)
- Check name consistency
- Build phyloseq object

# Go further : phyloseq import

---

- Import each component

```
sampledata <- read.csv("data/manual/sampledata.tsv", sep = "nt",  
row.names = 1)
```

```
taxtable <- read.csv("data/manual/taxtable.tsv", sep = "nt",  
row.names = 1)
```

```
otutable <- read.csv("data/manual/otutable.tsv", sep = "nt",  
row.names = 1)
```

```
tree <- read.tree("data/manual/tree.phy")
```

- Convert to base R type

```
taxtable <- as.matrix(taxtable)
```

```
otutable <- as.matrix(otutable)
```

- Convert to phyloseq base type

```
sampledata <- sample_data(sampledata)
```

```
taxtable <- tax_table(taxtable)
```

```
otutable <- otu_table(otutable, taxa_are_rows = TRUE)
```

# Go further : phyloseq import

---

- Check name consistency

- Abundance table and sample data (sample names)

```
all(colnames(otutable) %in% rownames(sampledata)) ## sample names
## [1] TRUE
```

- Abundance table and taxonomy table (taxa names)

```
all(rownames(otutable) %in% rownames(taxtable)) ## taxa names
## [1] TRUE
```

- Abundance table and tree leaves (taxa names)

```
all(rownames(otutable) %in% tree$tip.label) ## taxa names
## [1] TRUE
```

# Go further : phyloseq import

---

- Build object

```
manualData <- phyloseq(sampledData, otutable, taxtable, tree)
```

```
manualData
```

```
## phyloseq-class experiment-level object
```

```
## otu_table() OTU Table: [ 500 taxa and 26 samples ]
```

```
## sample_data() Sample Data: [ 26 samples by 6 sample variables ]
```

```
## tax_table() Taxonomy Table: [ 500 taxa by 7 taxonomic ranks ]
```

```
## phy_tree() Phylogenetic Tree: [ 500 tips and 499 internal nodes ]
```



# Go further : phyloseq filters

---

- Specify a **sample wise, OTU wide condition** (e.g. abundance greater than 2, in the top ten OTUs, etc)
- Specify a **number** of samples A
- Select only OTUs satisfying condition in at least A samples (usually to prune them).

Examples (what do they do?)

```
condition <- function(x) { x > 0 }  
taxaToKeep <- genefilter_sample(food, condition, 5)  
prune_taxa(taxaToKeep, food)
```

```
condition <- function(x) {order(x, decreasing = TRUE) <= 250 }  
taxaToKeep <- genefilter_sample(food, condition, 3)  
prune_taxa(taxaToKeep, food)
```

# Go further : phyloseq filters

---

## First example

- condition is TRUE if a taxa is present in a sample;
- condition must be met 5 times;
- ➔ Select taxa that appear in at least 5 samples.

## Second example

- condition is TRUE if a taxa is among the 250 most abundant ones in the sample;
- condition must be met 3 times;
- ➔ Select taxa that are very abundant (top 250) in at least 3 samples.

# Go further : phyloseq filters

---

- Specify an **OTU wise, sample wide** on the OTU abundance vector condition (e.g. overall abundance greater than 3, etc);
- Selects only OTUs satisfying condition (usually to prune them)
- Works on an **OTU-by-OTU** basis. Beware of bad normalizations.

Examples (what do they do?)

```
condition <- function(x) { sum(x > 0) >= 5 }  
taxaToKeep <- filter_taxa(food, condition)  
prune_taxa(taxaToKeep, food)
```

```
condition <- function(x) { sum(x) >= 100 }  
taxaToKeep <- filter_taxa(food, condition)  
prune_taxa(taxaToKeep, food)
```

# Go further : phyloseq filters

---

## First example

- condition is TRUE if a taxa has at least 5 positive counts (across samples);
- ➔ Select taxa that appear in at least 5 samples;

Probably better done with `genefilter_sample`

## Second example

- condition is TRUE if a taxa has global abundance at least 100;
- ➔ Select taxa with overall abundance at least 100 (beware of unequal sample sizes).

Probably done better with `sample_sums`

# Go further : phyloseq smoothing

---

`merge_samples` merges samples according to a factor by summing their abundances (beware of different group sizes and library sizes)

```
mergedData <- merge_samples(food, "EnvType")
## Warning in asMethod(object): NAs introduits lors de la conversion automatique
## Warning in asMethod(object): NAs introduits lors de la conversion automatique
mergedData
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 508 taxa and 8 samples ]
## sample_data() Sample Data: [ 8 samples by 3 sample variables ]
## tax_table() Taxonomy Table: [ 508 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 508 tips and 507 internal nodes ]
sample_names(mergedData)
## [1] "BoeufHache" "VeauHache" "DesLardons" "MerguezVolaille"
## [5] "SaumonFume" "FiletSaumon" "FiletCabillaud" "Crevette"
```

# Go further : phyloseq smoothing

---

Unfortunately, merging the contextual data is hard to do automatically in a meaningful way and information is lost in the process...

```
sample_data(mergedData)[1:2, ]
```

```
## Sample Data: [2 samples by 3 sample variables]:
```

```
## EnvType FoodType Description
```

```
## BoeufHache 1 NA NA
```

```
## VeauHache 2 NA NA
```

```
sample_data(food)[1:2, ]
```

```
## Sample Data: [2 samples by 3 sample variables]:
```

```
## EnvType FoodType Description
```

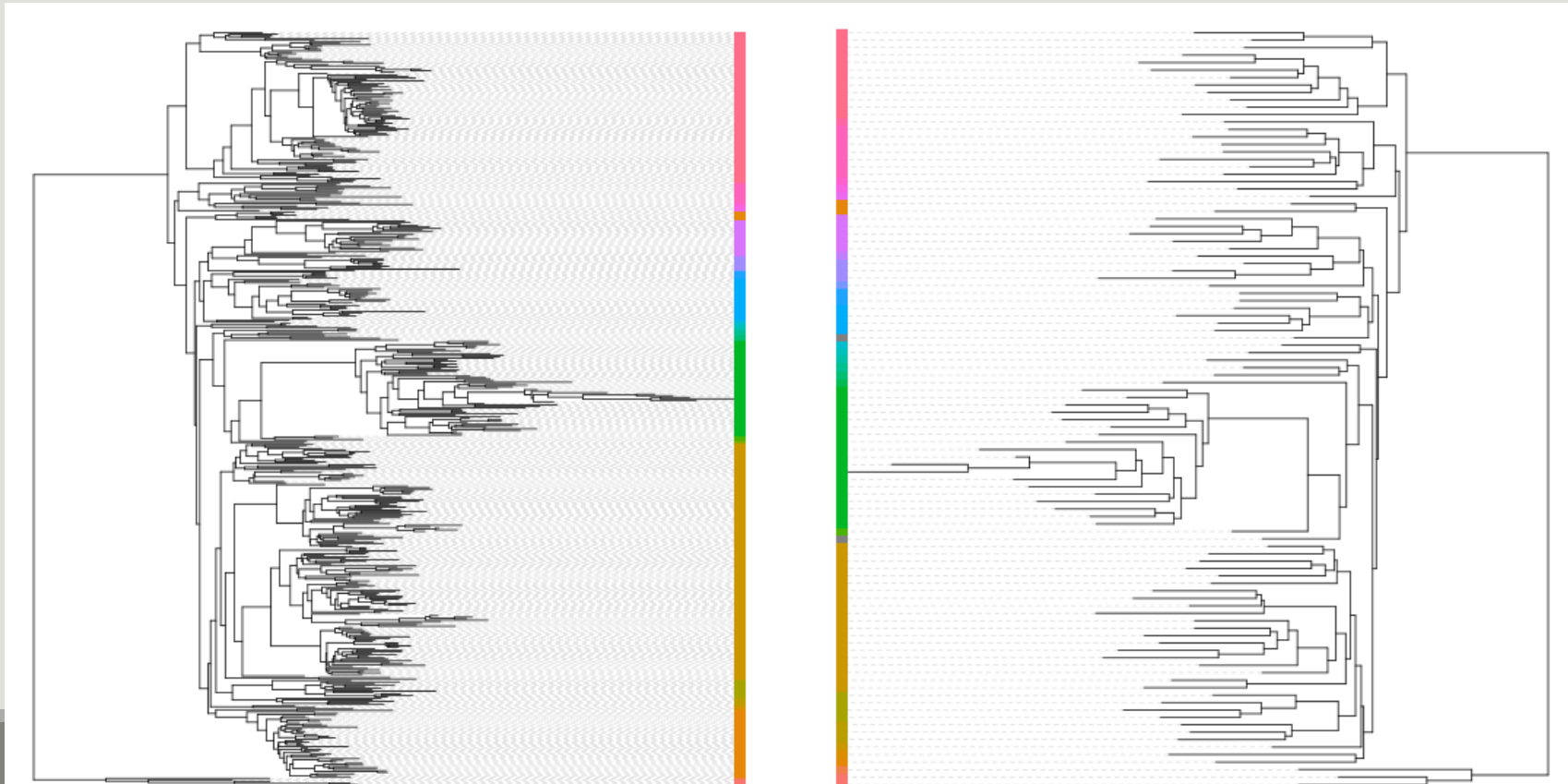
```
## DLT0.LOT08 DesLardons Meat LOT8
```

```
## DLT0.LOT05 DesLardons Meat LOT5
```

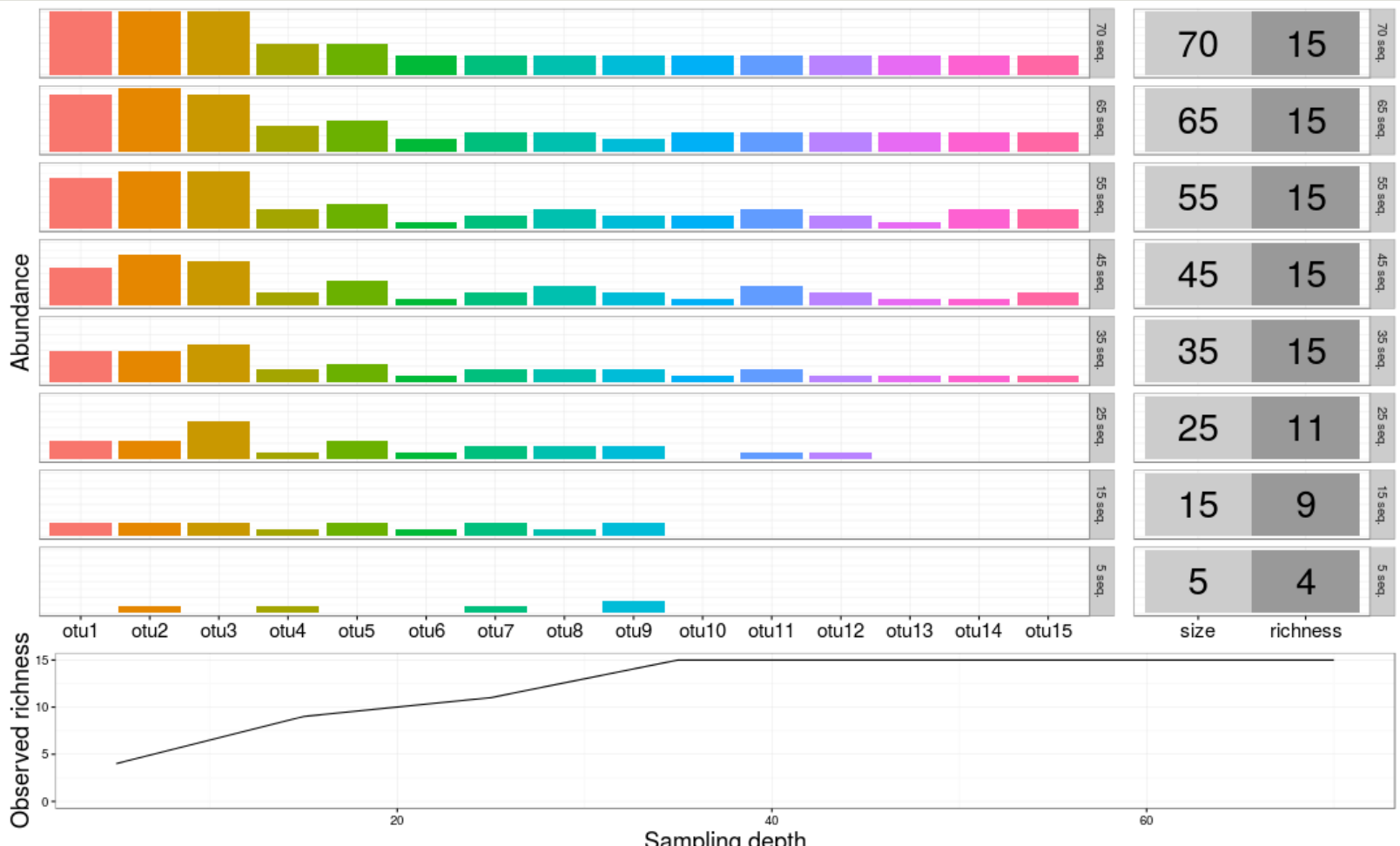
# Go further : phyloseq smoothing

tip glom agglomerates OTUs at a given height in the tree.

```
mergedData <- tip_glom(food, h = 0.3)
```



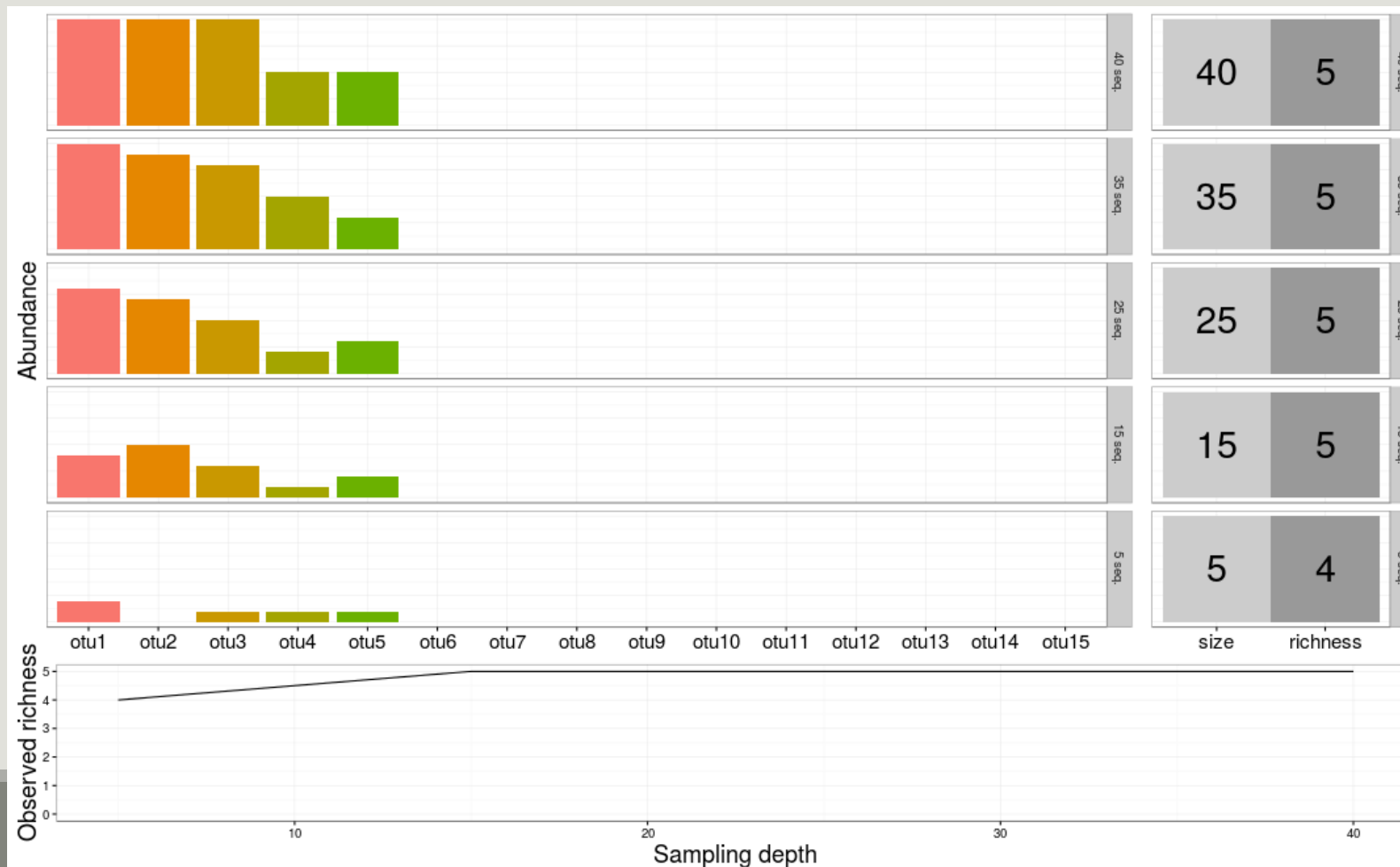
# Go further : rarefaction curves





# Go further : rarefaction curves

.. After filtering on rare OTUs



# Go further : rarefaction curves

---

- Why?
  - diversity indices are heavily influenced by sampling depths.
  - Rarefaction curves assess if sampling has exhausted the diversity.

- How?
  - Rarefy all samples to the same depth (optional, for speed here);
  - Use custom function `ggrare` and specify a step size.

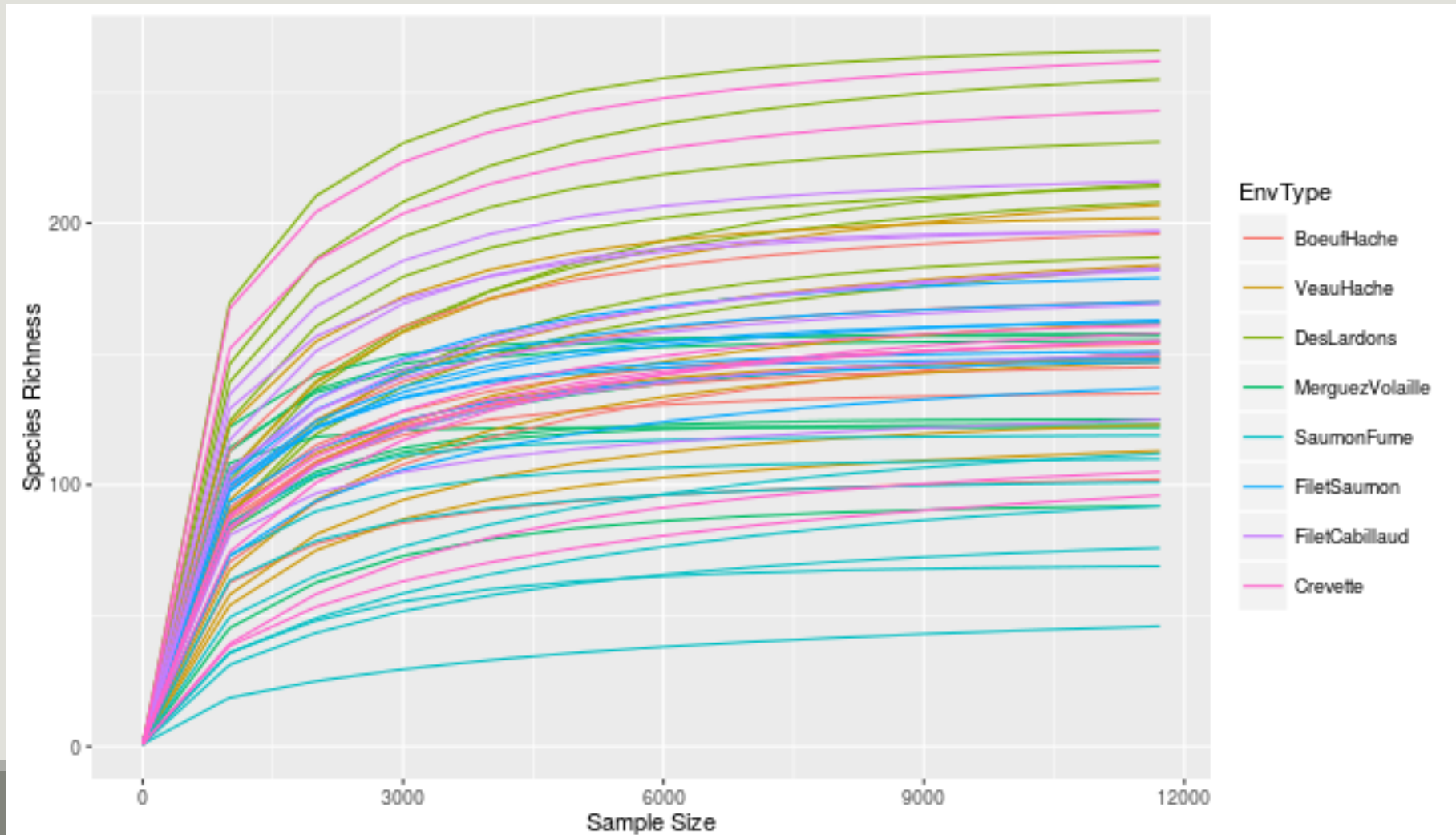
```
food <- rarefy_even_depth(food, rngseed = 1121983)
```

```
p <- ggrare(food, step = 1000, color = "SampleType", se = FALSE)
```

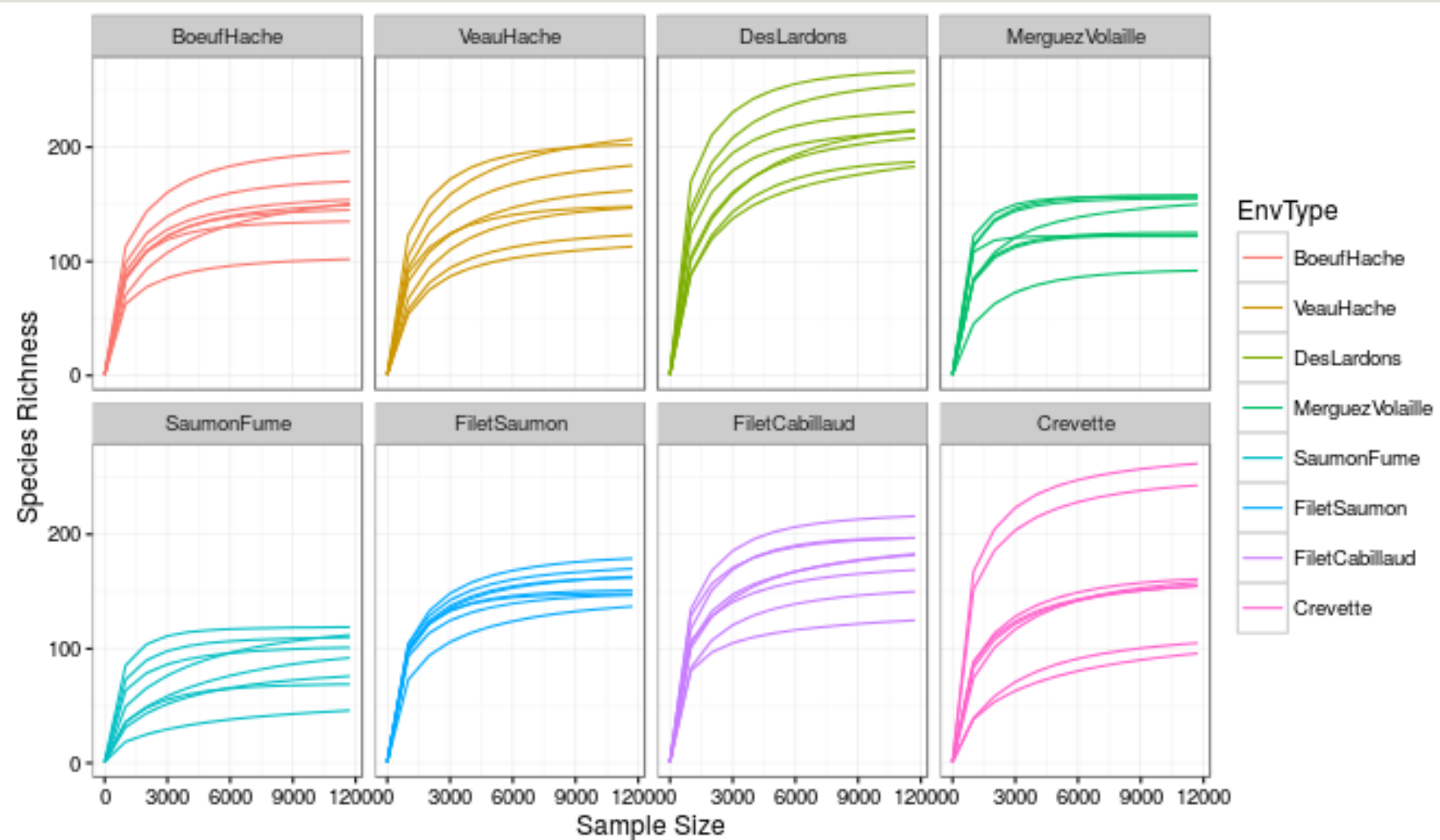
- To distinguish different environments easily, use faceting (and plain background)

```
plot(p + facet_wrap(~EnvType) + theme_bw())
```

# Go further : rarefaction curves



# Go further : rarefaction curves



# Go further : use your own color palette

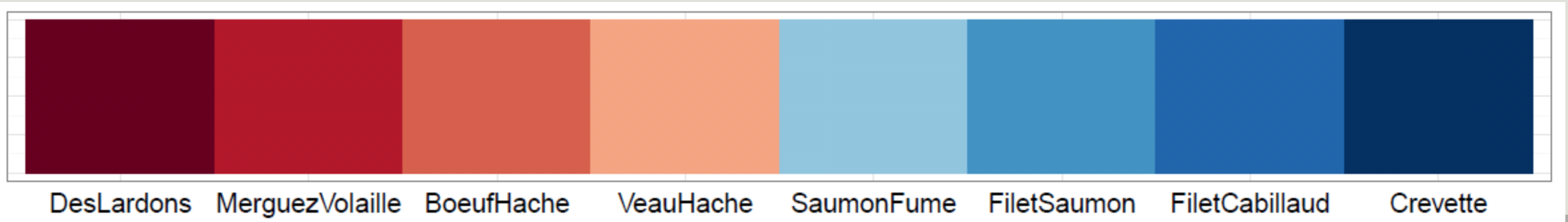
---

We already order the food types to make graphs easy to read:

```
## [1] "BoeufHache" "VeauHache" "DesLardons" "MerguezVolaille"  
## [5] "SaumonFume" "FiletSaumon" "FiletCabillaud" "Crevette"
```

Likewise, we're going to use a custom color palette

```
foodPalette <- c("#67001f", "#b2182b", "#d6604d", "#f4a582",  
"#92c5de", "#4393c3", "#2166ac", "#053061")  
  
names(foodPalette) <- levels(sample_data(food)$EnvType)
```



# Go further : use your own color palette

Try this palette on clustering plot

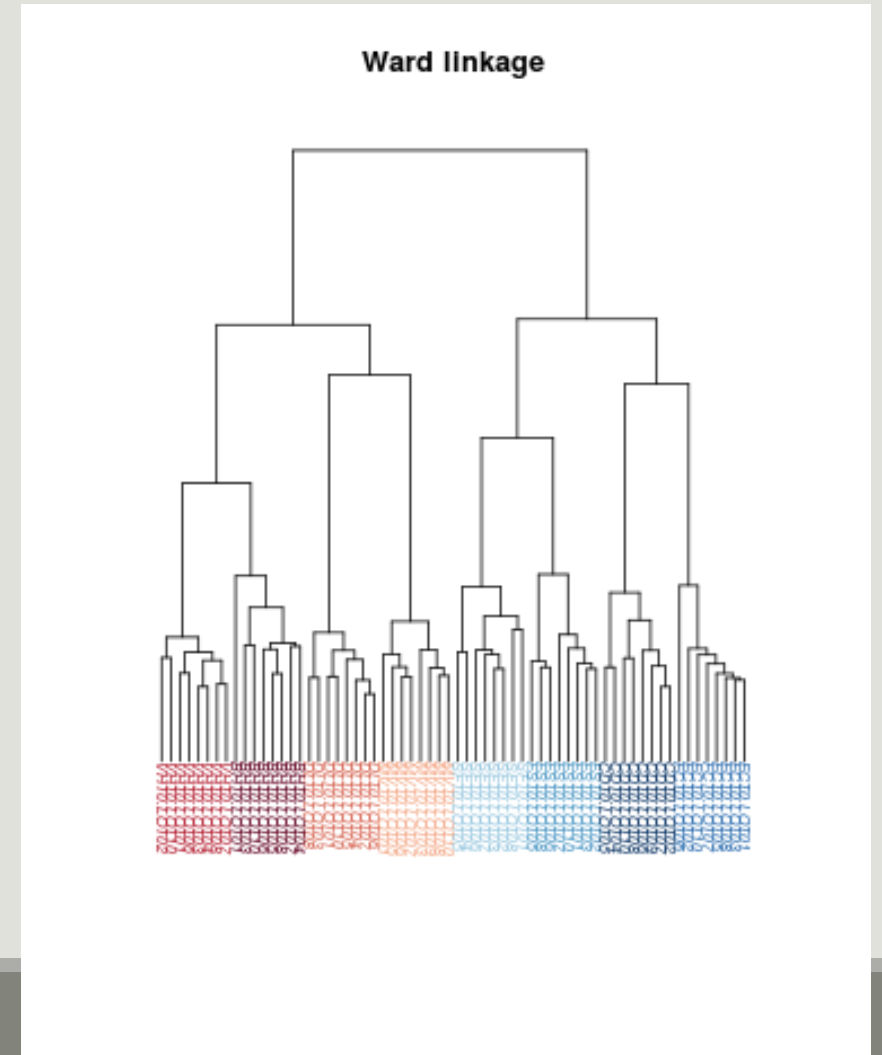
```
## Map sample type to color
```

```
tipColor <- col_factor(foodPalette, levels =  
  levels(envtype))(envtype)
```

```
## Change hclust object to phylo object and  
plot
```

```
clust.uf <- as.phylo(hclust(dist.uf, method =  
  "ward.D2"))
```

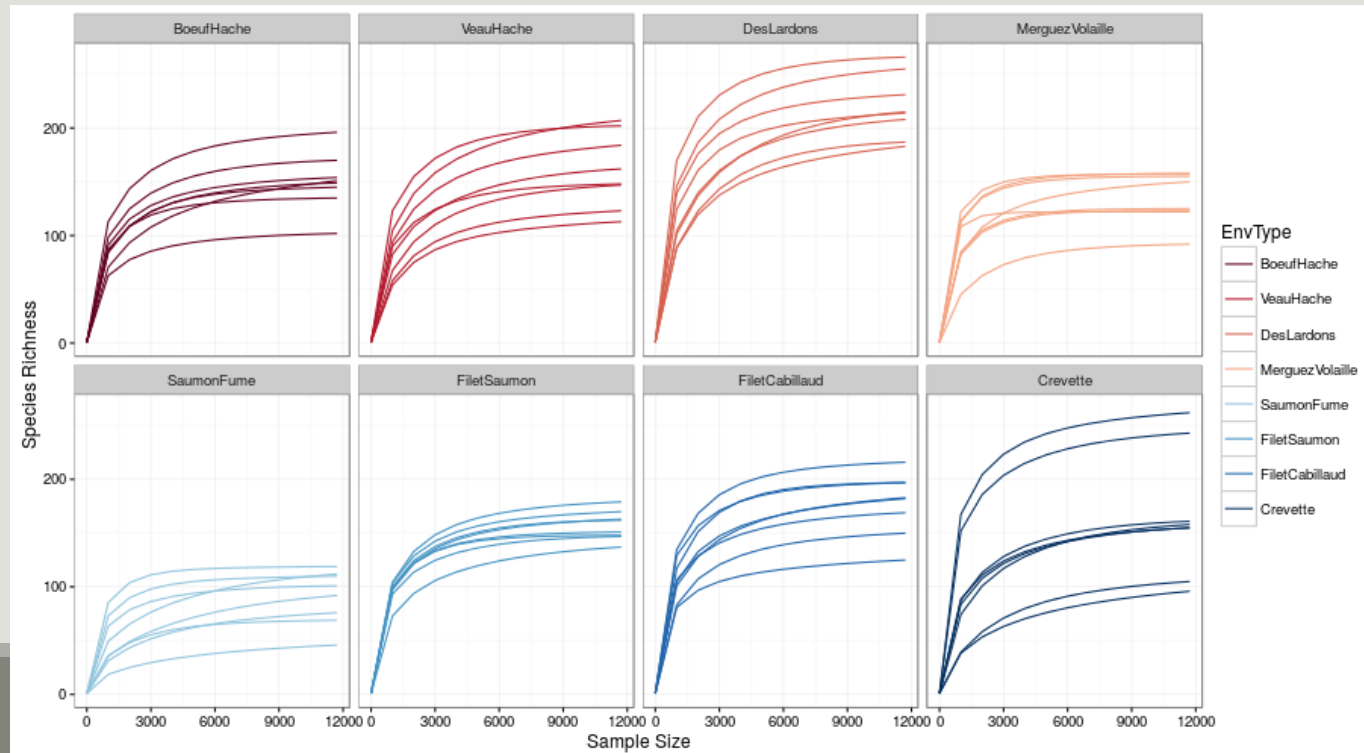
```
plot(clust.uf, tip.color = tipColor, direction  
  = "downwards", main = "Ward linkage")
```



# Go further : use your own color palette

Try this on any ggplot

```
food <- rarefy_even_depth(food, rngseed = 1121983)
p <- ggrare(food, step = 1000, color = "EnvType", se = FALSE)
p <- p + facet_wrap(~EnvType, ncol = 4) + theme_bw()
p <- p + scale_color_manual(values = foodPalette)
plot(p)
```



# Go further : heatmap order

---

Custom OTU order in function of prevalence

```
prevalence <- estimate_prevalence(food, "EnvType")
```

get in a different format to estimate correlations

```
correlationData <- estimate_prevalence(food, "EnvType", format = "wide")
```

```
correlationData <- t(correlationData)
```

```
correlation <- cor(correlationData, method="pearson")
```

clustering sample and order OTU according to the tree

```
otu.clust <- hclust(as.dist(1-c), method = "complete")
```

```
otuOrder <- otu.clust$labels[otu.clust$order]
```

plot

```
p <- plot_heatmap(food, taxa.order = otuOrder)
```

```
p <- p + facet_grid(~EnvType, scales = "free", space = "free")
```

```
p <- p + scale_fill_gradient2(low="#1a9850", mid = "#ffffbf", high="#d73027",  
na.value = "white", trans = log_trans(4), midpoint = log(100,base= 4 ))
```

```
plot(p)
```



# Go further : heatmap order

