Analysis of community composition data using `phyloseq` and `easy16S`

Mahendra Mariadassou, Cédric Midoux, Olivier Rué

April 2021
INRAE MaIAGE - Jouy-en-Josas

Outline

Goals

`phyloseq` and `Easy16S`

Become familiar with `phyloseq` and `Easy16S` for the analysis of microbial census data.

Exploratory Data Analysis

- α-diversity: how diverse is my community?
- β-diversity: how different are two communities?
- Use a distance matrix to study structures:
  - Hierarchical clustering: how do the communities cluster?
  - Permutational ANOVA: Communities structured by some *known* environmental factor?
- 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?
- Differential Abundances: which taxa are differentially abundant?

Outline

About phyloseq and Easy16S

1. R package (McMurdie and Holmes, 2013) to analyze community composition data in a phylogenetic framework
2. Community ecology functions from vegan, ade4, picante
3. Tree manipulation from ape
4. Graphics from ggplot2
5. Differential analysis from DESeq2
6. Easy16S is shiny web app to ease analyses

Accessing Easy16S

https://shiny.migale.inrae.fr/app/easy16S

Installing phyloseq

From bioconductor

```
## install.packages("BiocManager")
BiocManager::install("phyloseq")
```

From developer's website

```
## install.packages("remotes") ## If not installed previously
remotes::install_github("joey711/phyloseq")
```

Basic 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).

Outline

Let's get started

We first load the `phyloseq` package and some additional functions:

```
## remotes::install_github("mahendra-mariadassou/phyloseq-extended", ref =
library(phyloseq)
library(phyloseq.extended)
```

And start by loading some data, `GlobalPatterns` (Caporaso *et al.*, 2011) distributed with the `phyloseq` package

```
data(GlobalPatterns); gp <- GlobalPatterns;print(gp)

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 19216 taxa and 26 samples ]
## sample_data() Sample Data:       [ 26 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 19216 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 19216 tips and 19215 internal nodes ]
```

What's inside the phyloseq object? What does it remind you of?

Let's get started

We first load the `phyloseq` package and some additional functions:

```
## remotes::install_github("mahendra-mariadassou/phyloseq-extended", ref =
library(phyloseq)
library(phyloseq.extended)
```

And start by loading some data, `GlobalPatterns` (Caporaso *et al.*, 2011) distributed with the `phyloseq` package

```
data(GlobalPatterns); gp <- GlobalPatterns;print(gp)

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 19216 taxa and 26 samples ]
## sample_data() Sample Data:       [ 26 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 19216 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 19216 tips and 19215 internal nodes ]
```

What's inside the phyloseq object? What does it remind you of?

Let's get started (II)

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

- OTU Table
- Sample Data
- Taxomony Table
- Phylogenetic Tree

Let's have a quick look at each using the hinted at functions `otu_table`, `sample_data`, `tax_table` and `phy_tree`.

otu_table: matrix-like object

```
head(otu_table(gp), n = 4)

## OTU Table:          [4 taxa and 26 samples]
##                      taxa are rows
##        CL3 CC1 SV1 M31Fcsw M11Fcsw M31Plmr M11Plmr F21Plmr M31Tong M11Tong
## 549322   0   0   0       0       0       0       0       0       0       0
## 522457   0   0   0       0       0       0       0       0       0       0
## 951      0   0   0       0       0       0       1       0       0       0
## 244423   0   0   0       0       0       0       0       0       0       0
##        LMEpi24M SLEpi20M AQC1cm AQC4cm AQC7cm NP2 NP3 NP5 TRRsed1 TRRsed2
## 549322        0        1     27    100    130   1   0   0       0       0
## 522457        0        0      0      2      6   0   0   0       0       0
## 951           0        0      0      0      0   0   0   0       0       0
## 244423        0        0      0     22     29   0   0   0       0       0
##        TRRsed3 TS28 TS29 Even1 Even2 Even3
## 549322       0    0    0     0     0     0
## 522457       0    0    0     0     0     0
## 951          0    0    0     0     0     0
## 244423       0    0    0     0     0     0
```

tax_table: matrix-like object

```
head(tax_table(gp))
```

```
## Taxonomy Table:     [6 taxa by 7 taxonomic ranks]:
##        Kingdom   Phylum           Class           Order           Family
## 549322 "Archaea" "Crenarchaeota" "Thermoprotei" NA              NA
## 522457 "Archaea" "Crenarchaeota" "Thermoprotei" NA              NA
## 951    "Archaea" "Crenarchaeota" "Thermoprotei" "Sulfolobales" "Sulfolobaceae"
## 244423 "Archaea" "Crenarchaeota" "Sd-NA"        NA              NA
## 586076 "Archaea" "Crenarchaeota" "Sd-NA"        NA              NA
## 246140 "Archaea" "Crenarchaeota" "Sd-NA"        NA              NA
##        Genus        Species
## 549322 NA           NA
## 522457 NA           NA
## 951    "Sulfolobus" "Sulfolobusacidocaldarius"
## 244423 NA           NA
## 586076 NA           NA
## 246140 NA           NA
```

sample_data: data.frame-like object

```
head(sample_data(gp), n = 4)
```

```
## Sample Data:        [4 samples by 7 sample variables]:
##         X.SampleID  Primer Final_Barcode Barcode_truncated_plus_T
## CL3            CL3 ILBC_01        AACGCA                   TGCGTT
## CC1            CC1 ILBC_02        AACTCG                   CGAGTT
## SV1            SV1 ILBC_03        AACTGT                   ACAGTT
## M31Fcsw    M31Fcsw ILBC_04        AAGAGA                   TCTCTT
##         Barcode_full_length SampleType
## CL3              CTAGCGTGCGT       Soil
## CC1              CATCGACGAGT       Soil
## SV1              GTACGCACAGT       Soil
## M31Fcsw          TCGACATCTCT      Feces
##                                      Description
## CL3       Calhoun South Carolina Pine soil, pH 4.9
## CC1       Cedar Creek Minnesota, grassland, pH 6.1
## SV1     Sevilleta new Mexico, desert scrub, pH 8.3
## M31Fcsw    M3, Day 1, fecal swab, whole body study
```

phy_tree

phylo-class (tree) object

```
phy_tree(gp)

##
## Phylogenetic tree with 19216 tips and 19215 internal nodes.
##
## Tip labels:
##   549322, 522457, 951, 244423, 586076, 246140, ...
## Node labels:
##   , 0.858.4, 1.000.154, 0.764.3, 0.995.2, 1.000.2, ...
##
## Rooted; includes branch lengths.
```

Data structure

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

1. `otu_table`: an otu abundance table;
2. `sample_data`: a table of sample metadata, like sequencing technology, location of sampling, etc;
3. `tax_table`: a table of taxonomic descriptors for each otu, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
4. `phy_tree`: a phylogenetic tree of the otus;
5. `refseq`: a set of reference sequences (one per otu), not present in `gp`.

Data structure

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

1. `otu_table`: an otu abundance table;
2. `sample_data`: a table of sample metadata, like sequencing technology, location of sampling, etc;
3. `tax_table`: a table of taxonomic descriptors for each otu, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
4. `phy_tree`: a phylogenetic tree of the otus;
5. `refseq`: a set of reference sequences (one per otu), not present in `gp`.

Data structure

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

1. `otu_table`: an otu abundance table;
2. `sample_data`: a table of sample metadata, like sequencing technology, location of sampling, etc;
3. `tax_table`: a table of taxonomic descriptors for each otu, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
4. `phy_tree`: a phylogenetic tree of the otus;
5. `refseq`: a set of reference sequences (one per otu), not present in `gp`.

Data structure

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

1. `otu_table`: an otu abundance table;
2. `sample_data`: a table of sample metadata, like sequencing technology, location of sampling, etc;
3. `tax_table`: a table of taxonomic descriptors for each otu, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
4. `phy_tree`: a phylogenetic tree of the otus;
5. `refseq`: a set of reference sequences (one per otu), not present in `gp`.

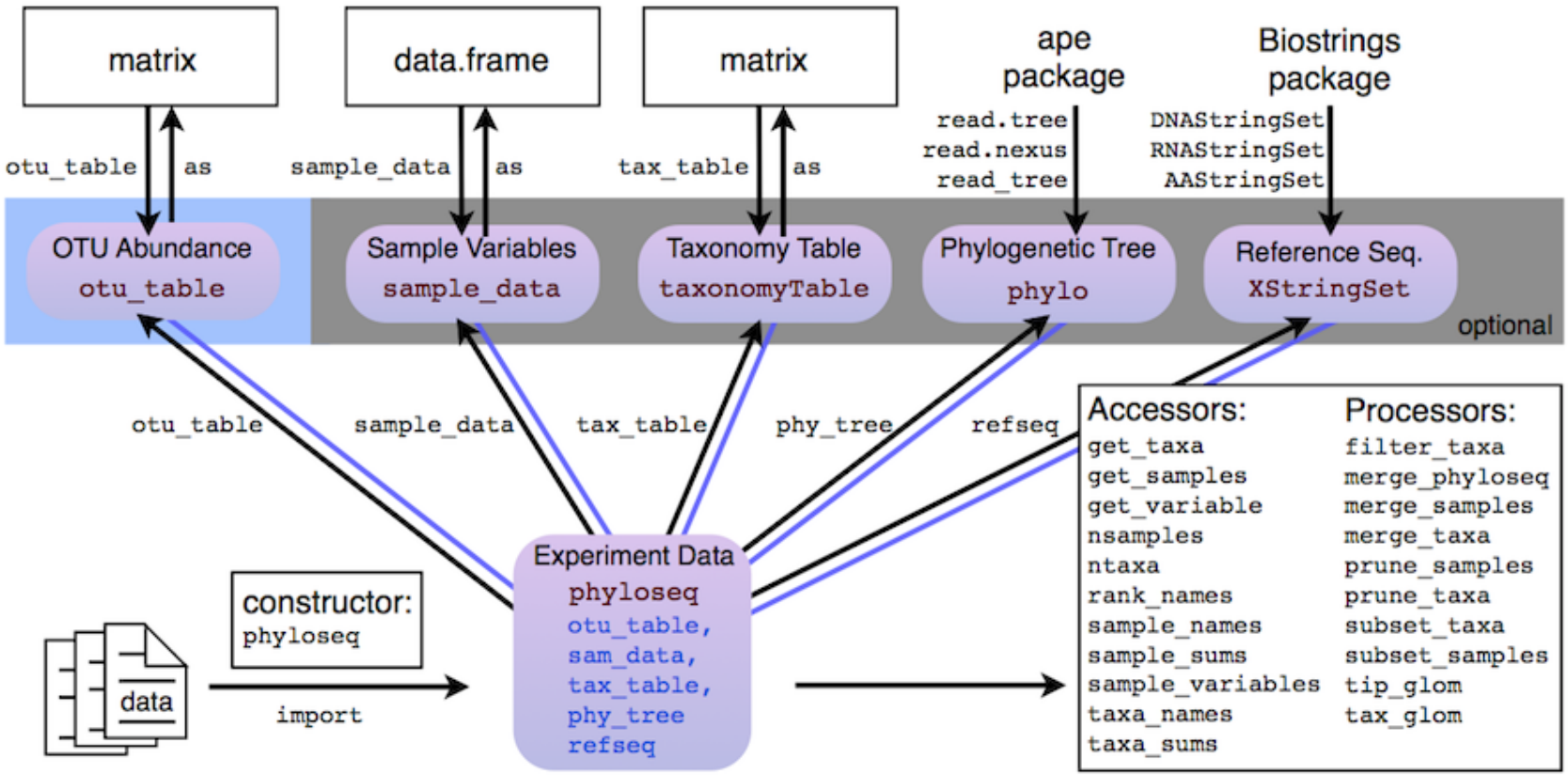
Data structure

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

1. `otu_table`: an otu abundance table;
2. `sample_data`: a table of sample metadata, like sequencing technology, location of sampling, etc;
3. `tax_table`: a table of taxonomic descriptors for each otu, typically the taxonomic assignation at different levels (phylum, order, class, etc.);
4. `phy_tree`: a phylogenetic tree of the otus;
5. `refseq`: a set of reference sequences (one per otu), not present in `gp`.

Data structure (II)

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



Outline

From a biom dataset: import_biom

The biom format natively supports

- otu count tables (the otu_table)
- otu description (the tax_table)
- sample description (the sample_data)

The other components are optional and must be stored in separate files

- phylogenetic tree in Newick format (the phy_tree)
- sequences in fasta format (the refset)

In our example, the taxonomy is in greengenes (*à la* qiime) format:
"k__Bacteria", "p__Proteobacteria", "c__Gammaproteobacteria", "o__Enterobacteriales"

From a biom dataset: import_biom

The biom format natively supports

- otu count tables (the otu_table)
- otu description (the tax_table)
- sample description (the sample_data)

The other components are optional and must be stored in separate files

- phylogenetic tree in Newick format (the phy_tree)
- sequences in fasta format (the refset)

In our example, the taxonomy is in greengenes (*à la* qiime) format:
"k__Bacteria", "p__Proteobacteria", "c__Gammaproteobacteria", "o__Enterobacteriales"

From a biom dataset: import_biom

The biom format natively supports

- otu count tables (the otu_table)
- otu description (the tax_table)
- sample description (the sample_data)

The other components are optional and must be stored in separate files

- phylogenetic tree in Newick format (the phy_tree)
- sequences in fasta format (the refset)

In our example, the taxonomy is in greengenes (*à la* qiime) format: "k__Bacteria", "p__Proteobacteria", "c__Gammaproteobacteria", "o__Enterobacteriales"

import_biom: example

Our toy dataset includes a biom, a tree and a set of references sequences.

```
biomfile <- "data/chaillou/chaillou.biom"
treefile <- "data/chaillou/tree.nwk"
```

The import is quite easy (our specific parseFunction is used for greengenes formatted taxonomy)

```
food <- import_biom(biomfile, treefile,
                            parseFunction = parse_taxonomy_greengenes)
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 ]
```

Importing data from tabular files (I)

Start by loading data in R and converting it to the proper format (matrix/data.frame)

```
otu <- as.matrix(read.table("data/mach/otu_table.tsv"))
tax <- as.matrix(read.table("data/mach/tax_table.tsv"))
tree <- read.tree("data/mach/tree.nwk")
map <- read.table("data/mach/metadata.tsv")
```

Importing data from tabular files (II)

Let's have a look at the different tables:

```
otu[1:2, 1:6]

##           sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131
## otu_16089              0              0              0              0
## otu_7290               0              0              0              1
##           sample_SF.0132 sample_SF.0133
## otu_16089              0              1
## otu_7290               0              0
```

Importing data from tabular files (III)

Let's have a look at the different tables:

```
tax[1:2, ]

##           Kingdom    Phylum       Class         Order
## otu_16089 "Bacteria" "Firmicutes" "Clostridia" "Clostridiales"
## otu_7290  "Bacteria" "Firmicutes" "Clostridia" "Clostridiales"
##           Family             Genus
## otu_16089 "Ruminococcaceae" NA
## otu_7290  "Ruminococcaceae" NA
```

Importing data from tabular files (IV)

Let's have a look at the different tables:

```
map[1:2, ]

##                SampleID Run Project Time Bande sex        mere
## sample_SF.0092  SF.0092   1     D60  D60  1105   2 17MAG101827
## sample_SF.0104  SF.0104   1     D60  D60  1105   2 17MAG102066
```

Importing data from tabular files (V)

You are now ready to build the phyloseq object

```
mach <- phyloseq(otu_table(otu, taxa_are_rows = TRUE),
                 tax_table(tax),
                 phy_tree(tree),
                 sample_data(map))
```

Import: A few words

- The import functions create consistent objects with
  - the same otus in the count table, the taxonomy table and the phylogenetic tree;
  - the same samples in the count table and the metadata table
- Samples/Taxa are matched by column names and/or rownames. Make sure that the table have them!!!
- Any otu absent from some components will be trimmed.
- Any sample absent from some components will be trimmed.
- Check number of taxa/samples and be wary of names mismatches.

About gp, food and mach

Global Patterns (Caporaso et al., 2011)

Global 16S survey of bacterial communities from very diverse environments (SampleType) using ultra deep sequencing. Used to stuy global ecological structures.

Food (Chaillou et al., 2015)

16S survey of bacterial communities from 8 different food products (EnvType), distributed as 4 meat products and 4 seafoods. Used to find core microbiota of food products.

Mach (Mach et al., 2015)

16S survey of gut microbiome from early life swines. Used (among others) to study the impact of weaning (Time and Weaned) on bacterial communities.

Outline

Other accessors

`phyloseq` also offers the following accessors:

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

to extract parts of a `phyloseq` object.

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

Other accessors

`phyloseq` also offers the following accessors:

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

to extract parts of a `phyloseq` object.

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

Dimensions

```
ntaxa(food)

## [1] 508

nsamples(food)

## [1] 64
```

- `ntaxa` returns the number of taxa;
- `nsamples` returns the number of samples;

Dimensions

```
ntaxa(food)

## [1] 508

nsamples(food)

## [1] 64
```

- `ntaxa` returns the number of taxa;
- `nsamples` returns the number of samples;

sample_names, taxa_names

```
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_005
```

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

sample_names, taxa_names

```
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_005
```

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

sample_names, taxa_names

```
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_005
```

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

sample_sums, taxa_sums

```
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)

sample_sums, taxa_sums

```
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)

sample_sums, taxa_sums

```
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)

rank_names

```
rank_names(food)

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

Names of the taxonomic levels available in the `tax_table` slot.

rank_names

```
rank_names(food)

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

Names of the taxonomic levels available in the `tax_table` slot.

sample_variables

```
head(sample_variables(food))

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

Names of the contextual data recorded on the samples.

sample_variables

```
head(sample_variables(food))

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

Names of the contextual data recorded on the samples.

A little exercice

Find the

- library size of samples `MVT0.LOT01`, `MVT0.LOT07`, `MVT0.LOT09`
- overall abudance 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
```

A little exercice

Find the

- library size of samples `MVTO.LOT01`, `MVTO.LOT07`, `MVTO.LOT09`
- overall abudance 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("MVTO.LOT01", "MVTO.LOT07", "MVTO.LOT09")]

## MVTO.LOT01 MVTO.LOT07 MVTO.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
```

get_variable, get_sample, get_taxa

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

## [1] "DesLardons" "DesLardons" "DesLardons" "DesLardons" "DesLardons"
## [6] "DesLardons"

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

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
```

- values for variable `varName` in sample data
- abundance values of otu `i` in all samples (row of OTU table).
- abundance values of all otus in sample `i` (column of OTU table)

get_variable, get_sample, get_taxa

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

## [1] "DesLardons" "DesLardons" "DesLardons" "DesLardons" "DesLardons"
## [6] "DesLardons"

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

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
```

- values for variable `varName` in sample data
- abundance values of otu `i` in all samples (row of OTU table).
- abundance values of all otus in sample `i` (column of OTU table)

get_variable, get_sample, get_taxa

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

## [1] "DesLardons" "DesLardons" "DesLardons" "DesLardons" "DesLardons"
## [6] "DesLardons"

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

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
```

- values for variable `varName` in sample data
- abundance values of otu `i` in all samples (row of OTU table).
- abundance values of all otus in sample `i` (column of OTU table)

get_variable, get_sample, get_taxa

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

## [1] "DesLardons" "DesLardons" "DesLardons" "DesLardons" "DesLardons"
## [6] "DesLardons"

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

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
```

- values for variable `varName` in sample data
- abundance values of otu `i` in all samples (row of OTU table).
- abundance values of all otus in sample `i` (column of OTU table)

Modifying some values

To modify parts of a `phyloseq` object, we must use (high-levels) accessors such as `otu_table`.

To transform `EnvType` to a factor with meaningful level ordering (meat products first and seafood second), we must use `sample_data`:

```
correct.order <- c("BoeufHache", "VeauHache", "DesLardons",
                   "MerguezVolaille", "SaumonFume", "FiletSaumon",
                   "FiletCabillaud", "Crevette")
sample_data(food)$EnvType <- factor(sample_data(food)$EnvType,
                                    levels = correct.order)
levels(get_variable(food, "EnvType"))
```

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

Likewise, to modify the count of OTU `otu_00520` in sample `DLT0.LOT08`, or its species affiliation we would do

```
otu_table(food)["otu_00520", "DLT0.LOT08"] <- 0
tax_table(food)["otu_00520", "Species"] <- "Ornithinolytica"
```

Modifying some values

To modify parts of a `phyloseq` object, we must use (high-levels) accessors such as `otu_table`.

To transform `EnvType` to a factor with meaningful level ordering (meat products first and seafood second), we must use `sample_data`:

```
correct.order <- c("BoeufHache", "VeauHache", "DesLardons",
                   "MerguezVolaille", "SaumonFume", "FiletSaumon",
                   "FiletCabillaud", "Crevette")
sample_data(food)$EnvType <- factor(sample_data(food)$EnvType,
                                    levels = correct.order)
levels(get_variable(food, "EnvType"))

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

Likewise, to modify the count of OTU `otu_00520` in sample `DLT0.LOT08`, or its species affiliation we would do

```
otu_table(food)["otu_00520", "DLT0.LOT08"] <- 0
tax_table(food)["otu_00520", "Species"] <- "Ornithinolytica"
```

Modifying some values

To modify parts of a `phyloseq` object, we must use (high-levels) accessors such as `otu_table`.

To transform `EnvType` to a factor with meaningful level ordering (meat products first and seafood second), we must use `sample_data`:

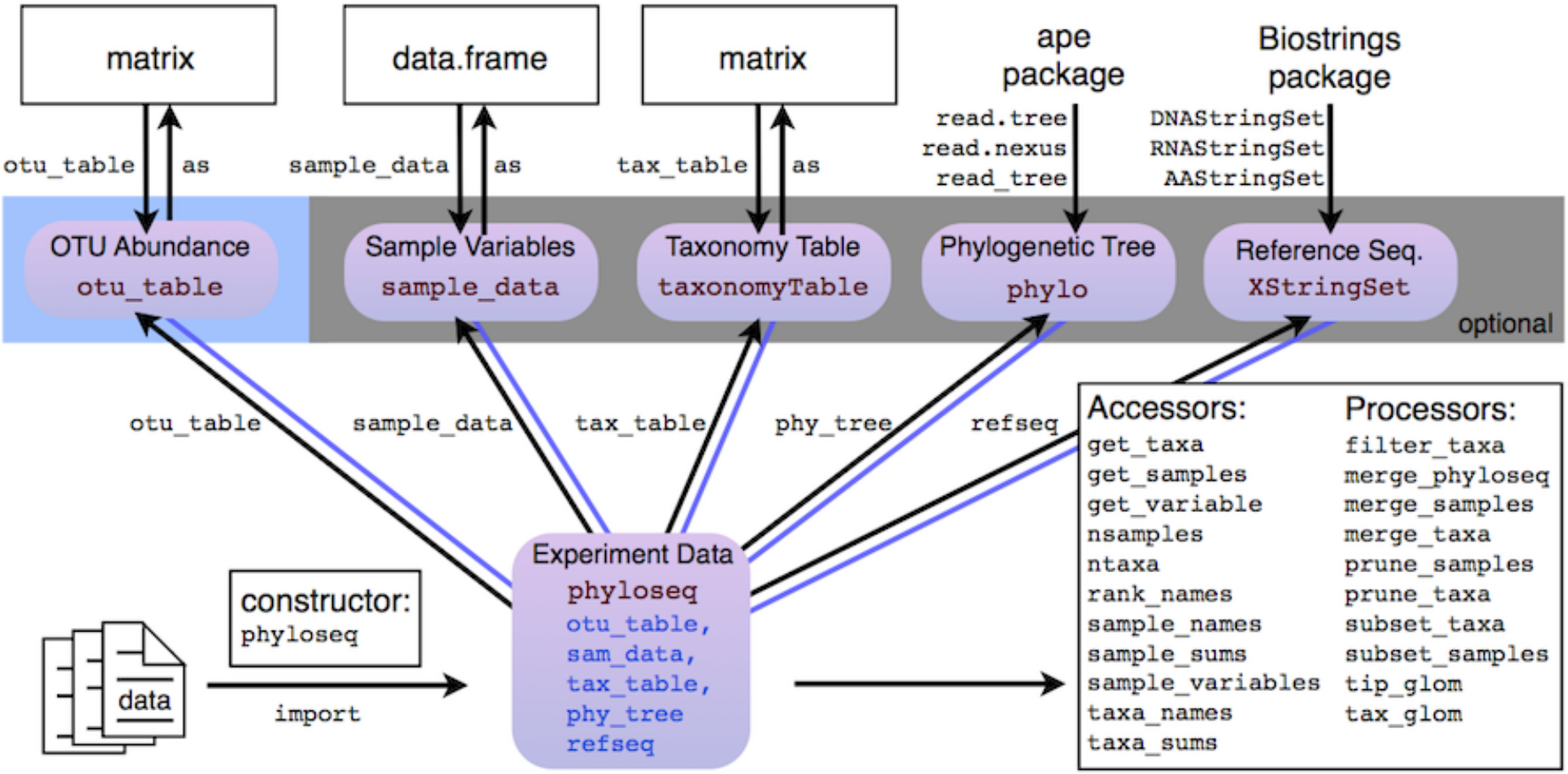
```
correct.order <- c("BoeufHache", "VeauHache", "DesLardons",
                   "MerguezVolaille", "SaumonFume", "FiletSaumon",
                   "FiletCabillaud", "Crevette")
sample_data(food)$EnvType <- factor(sample_data(food)$EnvType,
                                    levels = correct.order)
levels(get_variable(food, "EnvType"))

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

Likewise, to modify the count of OTU `otu_00520` in sample `DLT0.LOT08`, or its species affiliation we would do

```
otu_table(food)["otu_00520", "DLT0.LOT08"] <- 0
tax_table(food)["otu_00520", "Species"] <- "Ornithinolytica"
```

Data structure Recap



Outline

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Filtering via prune, subset and filter (I)

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 (otu1, otu4) or of logical (TRUE, FALSE, FALSE, TRUE)
- `prune_taxa(taxa, physeq)` would keep only otus otu1, otu4

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.

Prune and subset

Prune

```
samplesToKeep <- sample_names(food)[1:10]
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 ]
```

Subset

```
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 ]
```

Prune and subset

Prune

```
samplesToKeep <- sample_names(food)[1:10]
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 ]
```

Subset

```
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 ]
```

A bit more about subset (II)

Multiple conditions can be combined with the usual logical operator (& for AND and | for OR)

```
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"
```

Outline

Rarefaction with rarefy_even_depth

rarefy_even_depth downsamples all samples to the same depth and prunes otus that disappear from all samples as a result.

```
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
## ...

sample_sums(foodRare)[1:5]

## DLT0.LOT08 DLT0.LOT05 DLT0.LOT03 DLT0.LOT07 DLT0.LOT06
##      11718      11718      11718      11718      11718
```

Transforming abundance counts with transform_sample_counts

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

```
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 recap

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 recap

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 recap

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 recap

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 recap

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 recap

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

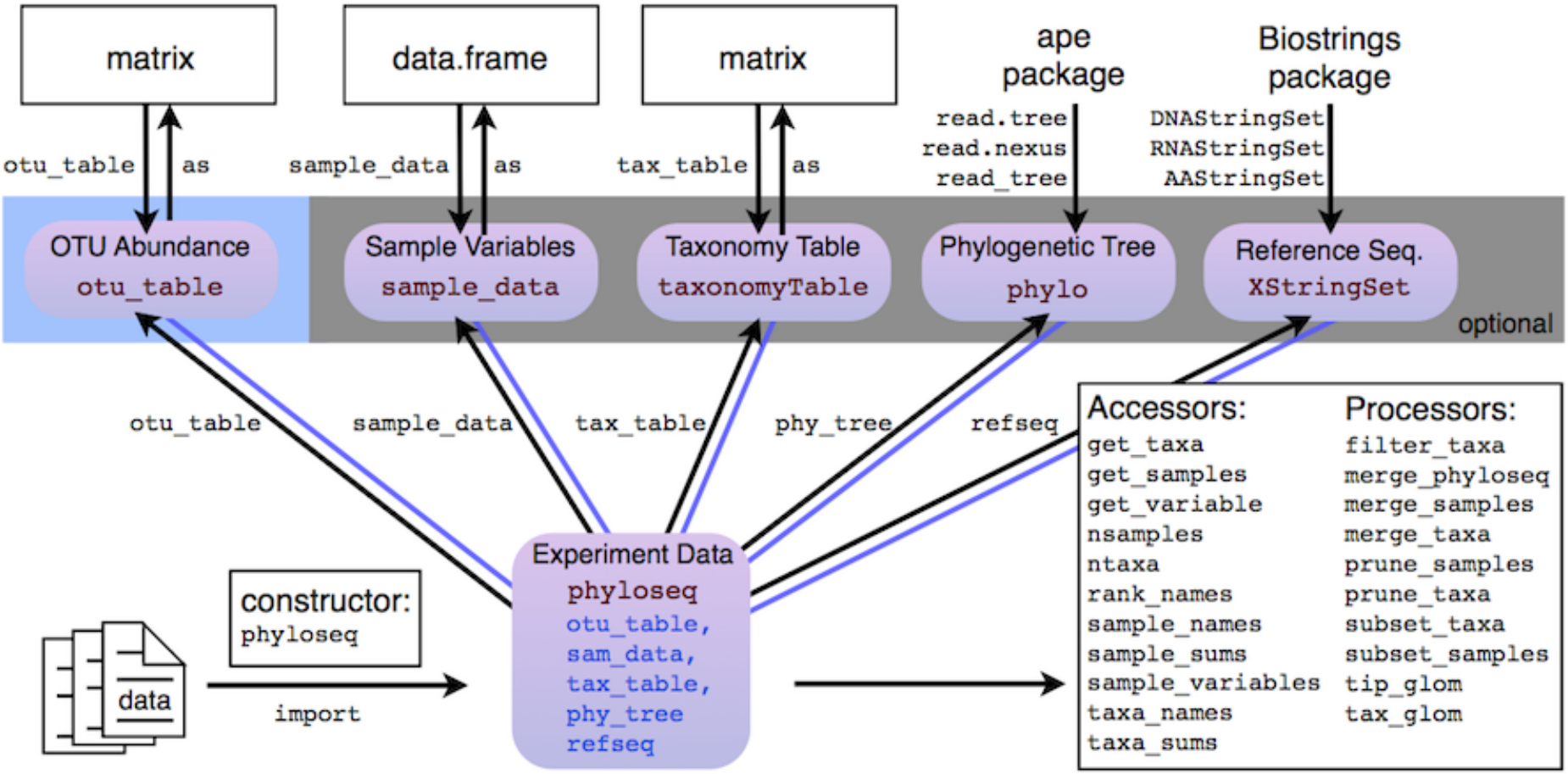
- Functions to import data from biom files, qiime output files or plain tabular files.
- Accessors to access different component of your dataset
- Samples and taxa names are coherent between the different components.
- Filters to keep only part of the dataset.
- Smoothers to aggregate parts of the dataset.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

phyloseq in recap

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

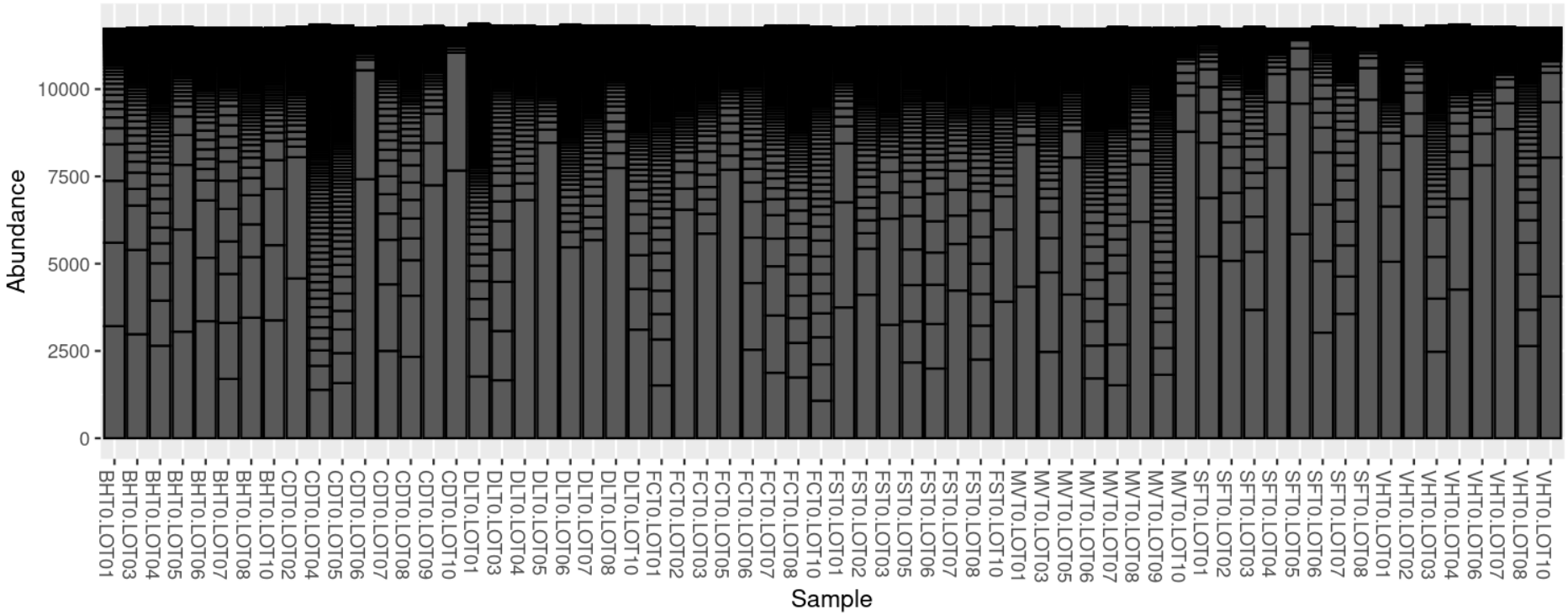
phyloseq recap (II)



Outline

Looking at your samples (plot_bar)

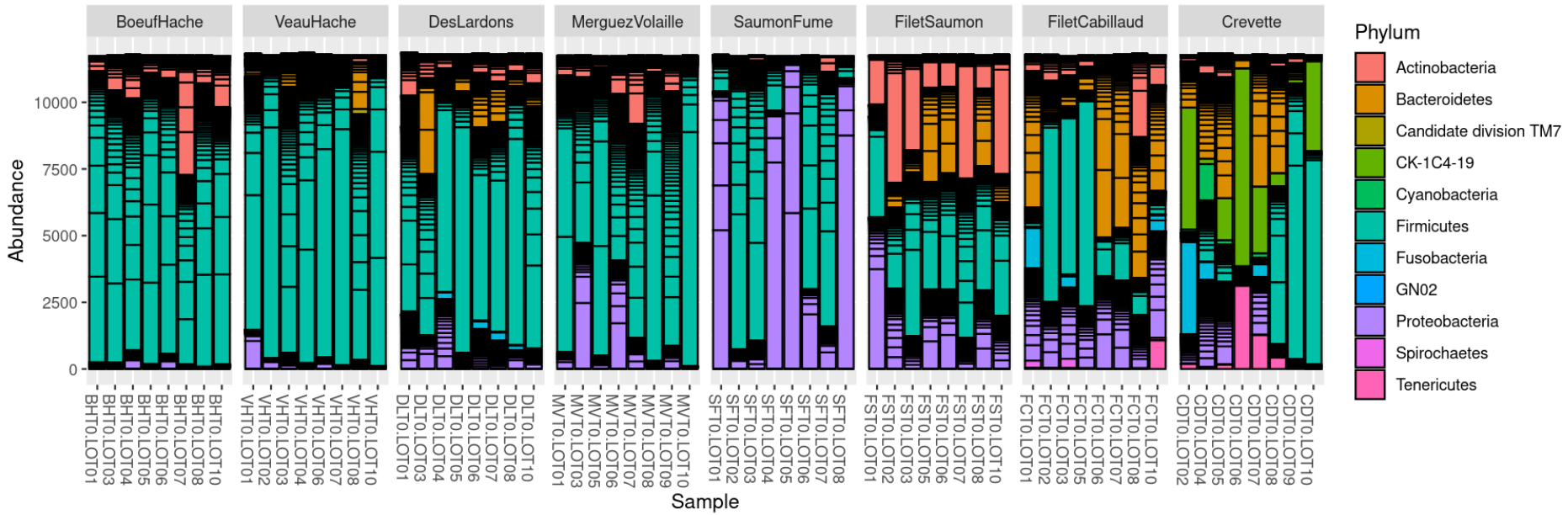
```
p <- plot_bar(food)
plot(p) ## Base graphic, ugly
```



Looking at your samples (plot_bar)

Organize samples and color otu by Phylum

```
p <- plot_bar(food, fill = "Phylum") ## aes, fill bar according to phylum
p <- p + facet_wrap(~EnvType, scales = "free_x", nrow = 1) ## add facets
plot(p)
```



Limitations of `plot_bar`

`plot_bar`

- `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)

Custom function `plot_composition`

- subset otus at a given taxonomic level
- aggregate otus at another taxonomic level
- Show only a given number of otus.
- Works with relative abundances

Limitations of `plot_bar`

`plot_bar`

- `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)

Custom function `plot_composition`

- subset otus at a given taxonomic level
- aggregate otus at another taxonomic level
- Show only a given number of otus.
- Works with relative abundances

Looking at your samples (plot_composition) (I)

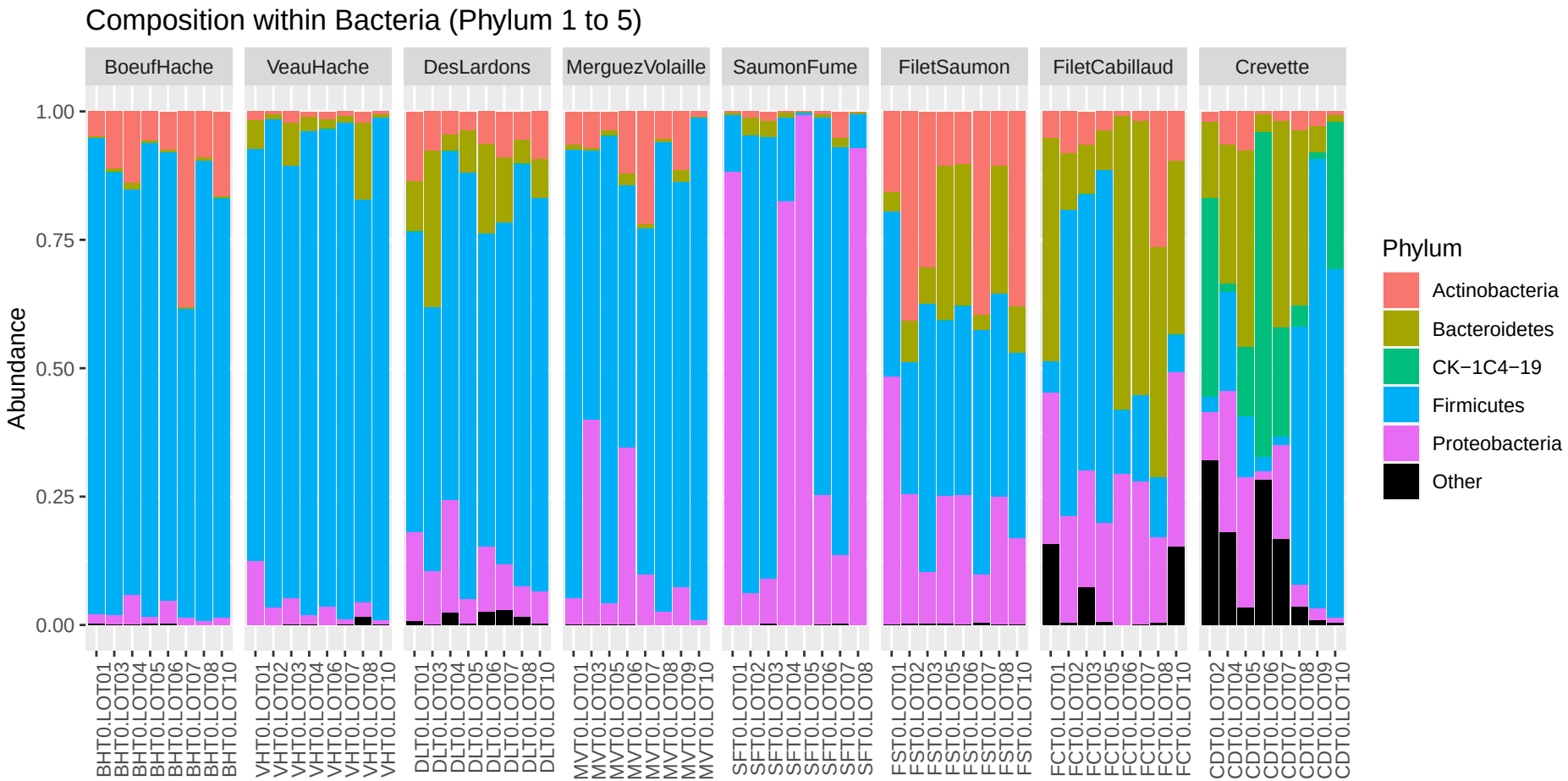
Select Bacteria (at Kingdom level) and aggregate by Phylum.

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

Looking at your samples (plot_composition) (I)

Select Bacteria (at Kingdom level) and aggregate by Phylum.

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



Looking at your samples (plot_composition) (II)

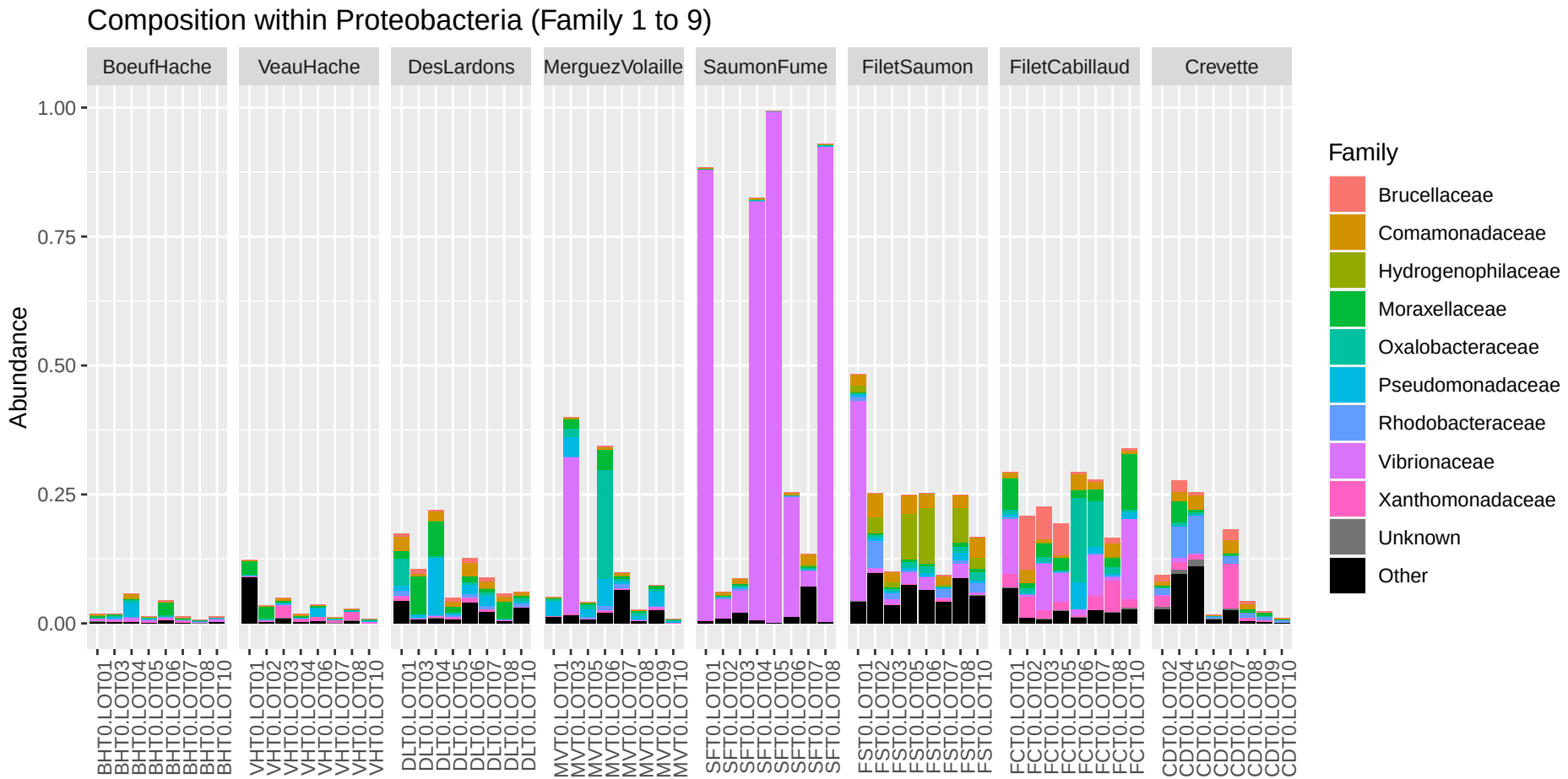
Select Proteobacteria (at Phylum level) and aggregate by Family.

```
p <- plot_composition(food, "Phylum", "Proteobacteria", "Family",
                      numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~EnvType, scales = "free_x", nrow = 1)
plot(p)
```

Looking at your samples (plot_composition) (II)

Select Proteobacteria (at Phylum level) and aggregate by Family.

```
p <- plot_composition(food, "Phylum", "Proteobacteria", "Family",
                      numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~EnvType, scales = "free_x", nrow = 1)
plot(p)
```



Outline

Different kinds of biodiversity indices...

16S surveys used to monitor the bacterial biodiversity.

Three flavors of diversity

- α-diversity: diversity within a community;
- β-diversity: diversity between communities;
- γ-diversity: diversity at the landscape scale (blurry for bacterial communities);

Diversity decomposition

$$\gamma = \alpha \ +|\times \ \beta$$

β-dissimilarities/distances

- Dissimilarities between pairs of communities
- Often used as a first step to compute β-diversity

Different kinds of biodiversity indices...

16S surveys used to monitor the bacterial biodiversity.

Three flavors of diversity

- α-diversity: diversity within a community;
- β-diversity: diversity between communities;
- γ-diversity: diversity at the landscape scale (blurry for bacterial communities);

Diversity decomposition

$$\gamma = \alpha \ + | \times \ \beta$$

β-dissimilarities/distances

- Dissimilarities between pairs of communities
- Often used as a first step to compute β-diversity

Different kinds of biodiversity indices...

16S surveys used to monitor the bacterial biodiversity.

Three flavors of diversity

- α-diversity: diversity within a community;
- β-diversity: diversity between communities;
- γ-diversity: diversity at the landscape scale (blurry for bacterial communities);

Diversity decomposition

$$\gamma = \alpha \ + | \times \ \beta$$

β-dissimilarities/distances

- Dissimilarities between pairs of communities
- Often used as a first step to compute β-diversity

Different kinds of biodiversity indices...

16S surveys used to monitor the bacterial biodiversity.

Three flavors of diversity

- α-diversity: diversity within a community;
- β-diversity: diversity between communities;
- γ-diversity: diversity at the landscape scale (blurry for bacterial communities);

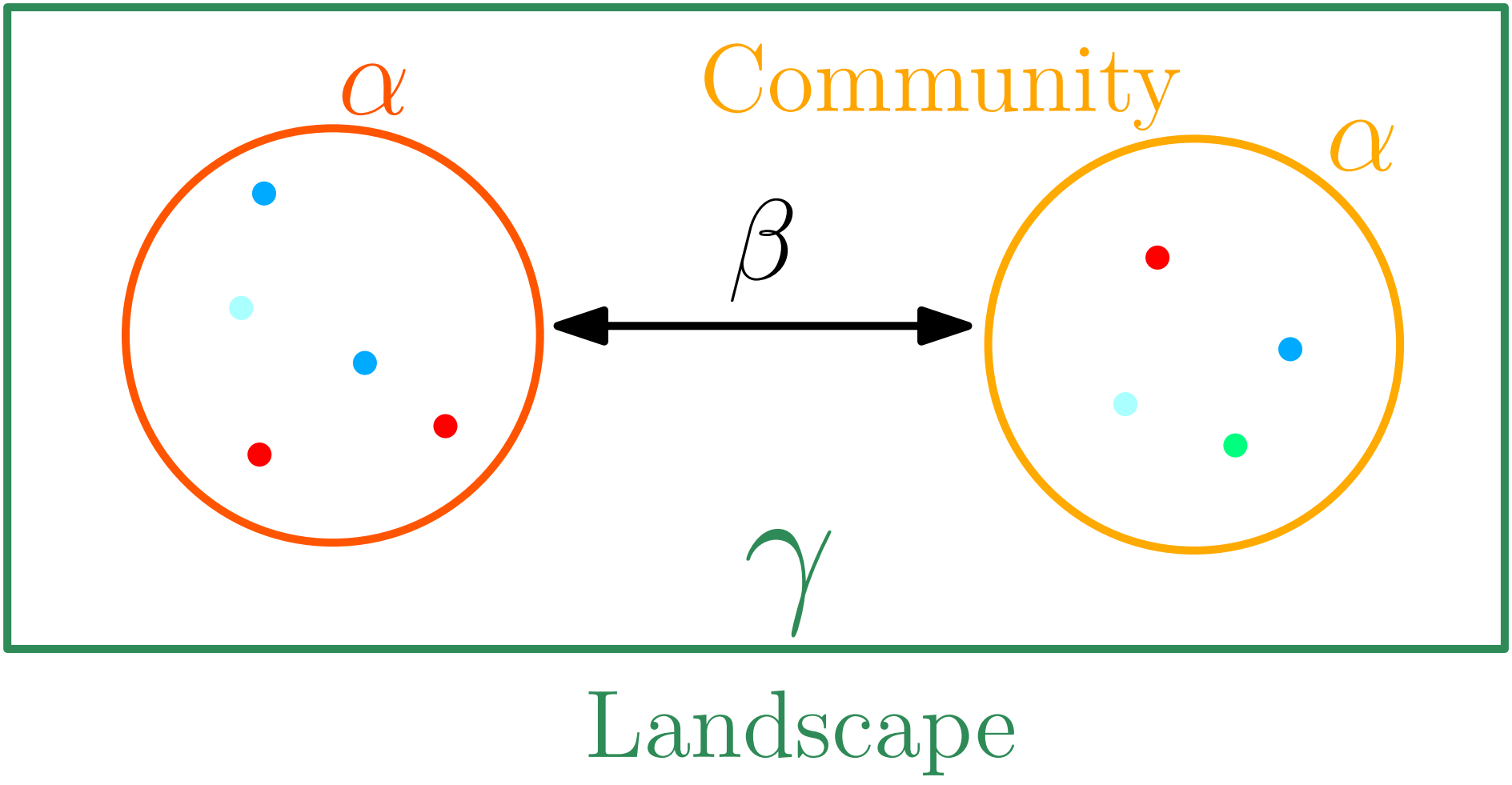
Diversity decomposition

$$\gamma = \alpha \ + | \times \ \beta$$

β-dissimilarities/distances

- Dissimilarities between pairs of communities
- Often used as a first step to compute β-diversity

A schematic view of diversity



Based on different types of data

Presence/Absence (qualitative) vs. Abundance (quantitative)

- Presence/Absence gives less weight to dominant species;
- is more sensitive to differences in sampling depths;
- 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.

Based on different types of data

Presence/Absence (qualitative) vs. Abundance (quantitative)

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

Compositional vs. Phylogenetic

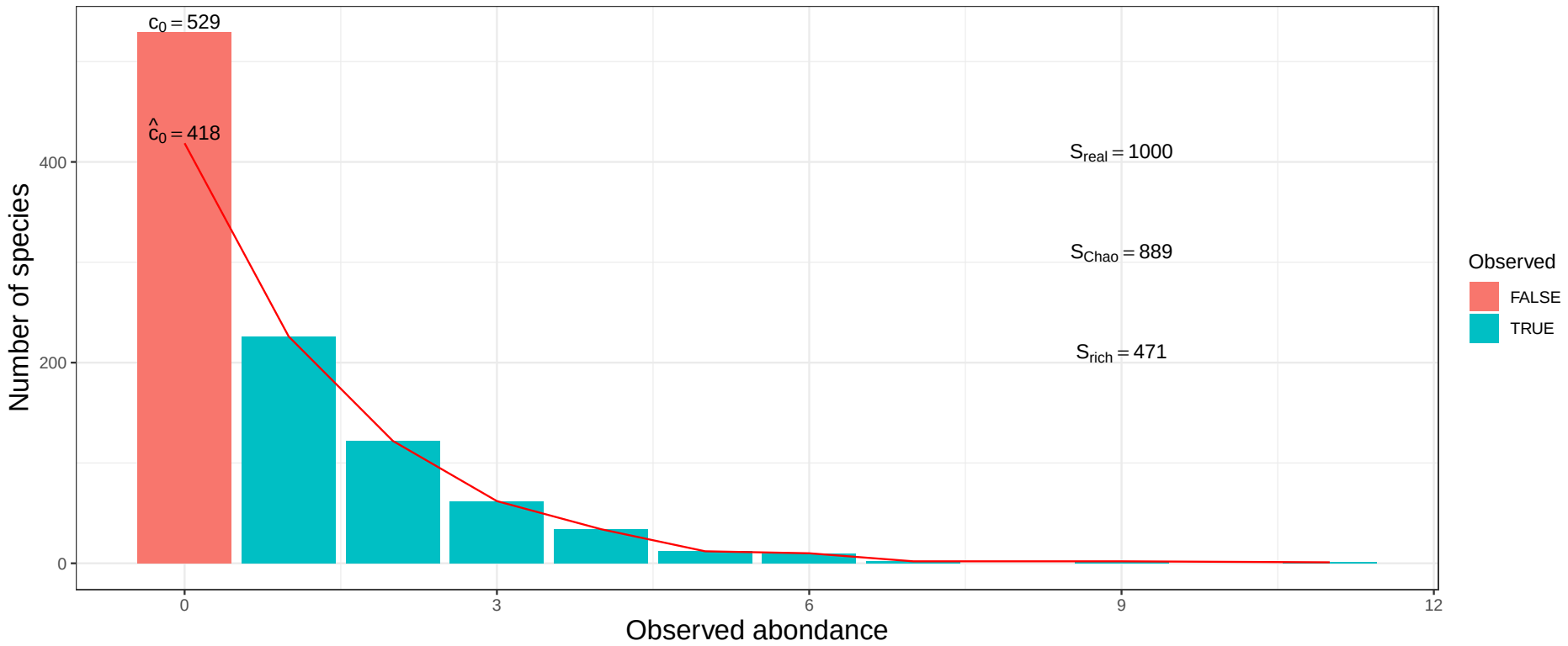
- Compositional does not require a phylogenetic tree;
- is more sensitive to erroneous otu picking;
- gives the same importance to all otus.

Outline

α-diversity: number of species (richness)

Note c_i the number of species observed i times ($i = 1, 2, \dots$) and p_s the proportion of species s ($s = 1, \dots, S$)

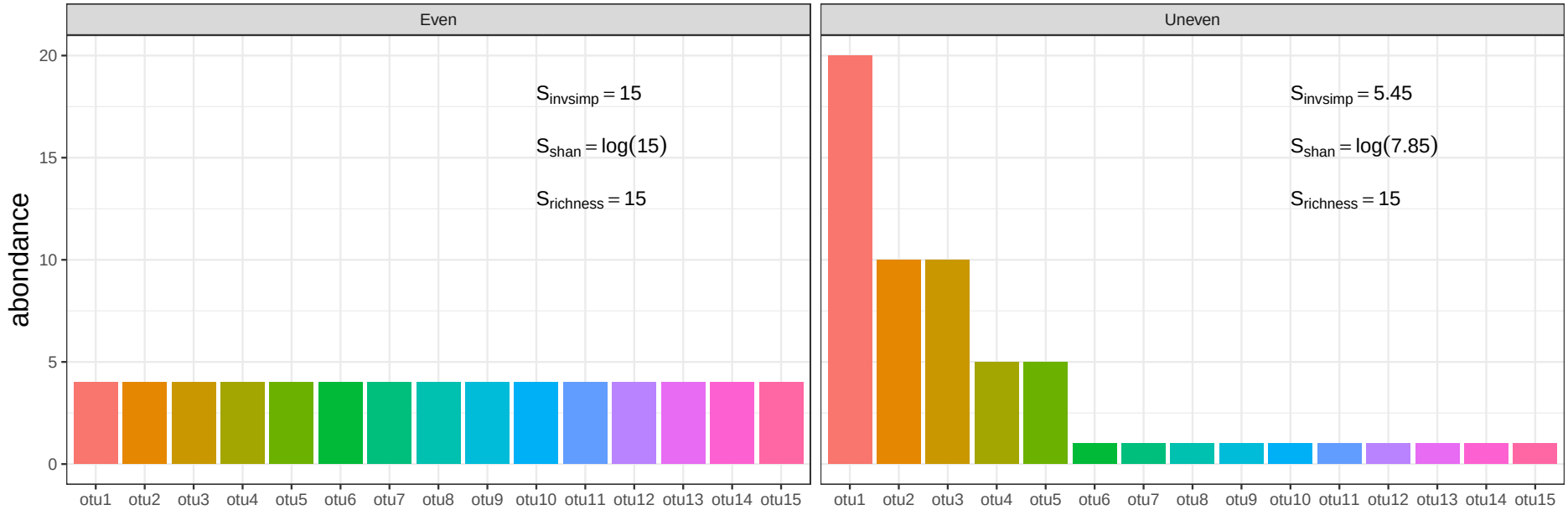
Richness	Chao1
Number of observed species	Richness + (estimated) number of unobserved species
$S_{\text{rich}} = \sum_s 1_{\{p_s>0\}} = \sum_i c_i$	$S_{\text{Chao}} = S_{\text{rich}} + \hat{c}_0$



α-diversity: evenness of the species distribution

Give more weight to abundant species

Shannon	Inv-Simpson
Evenness of the species abundance distribution	Inverse probability that two sequences sampled at random come from the same species
$S_{\text{Shan}} = -\sum_s p_s \log(p_s) \leq \log(S)$	$S_{\text{Inv-Simp}} = \frac{1}{p_1^2 + \cdots + p_S^2} \leq S$



α diversities

Available in `phyloseq`

- **Species richness**: number of observed 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.
- **Chao1**: number of observed otu + estimate of the number of unobserved otus

α diversities

Available in `phyloseq`

- **Species richness**: number of observed 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.
- **Chao1**: number of observed otu + estimate of the number of unobserved otus

α diversities

Available in `phyloseq`

- **Species richness**: number of observed 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.
- **Chao1**: number of observed otu + estimate of the number of unobserved otus

α diversities

Available in `phyloseq`

- **Species richness**: number of observed 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.
- **Chao1**: number of observed otu + estimate of the number of unobserved otus

α diversities

Available in `phyloseq`

- **Species richness**: number of observed 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.
- **Chao1**: number of observed otu + estimate of the number of unobserved otus

α diversity and filtering (I)

Many α diversities (richness, Chao) depend a lot on rare otus. Do not **trim** rare otus before computing them as it can drastically alter the result (see next slide).

Richness

Richness are plotted with `plot_richness`. Note the `x = "EnvType"` passed on to the `aes` mapping of a `ggplot`.

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

α diversity and filtering (I)

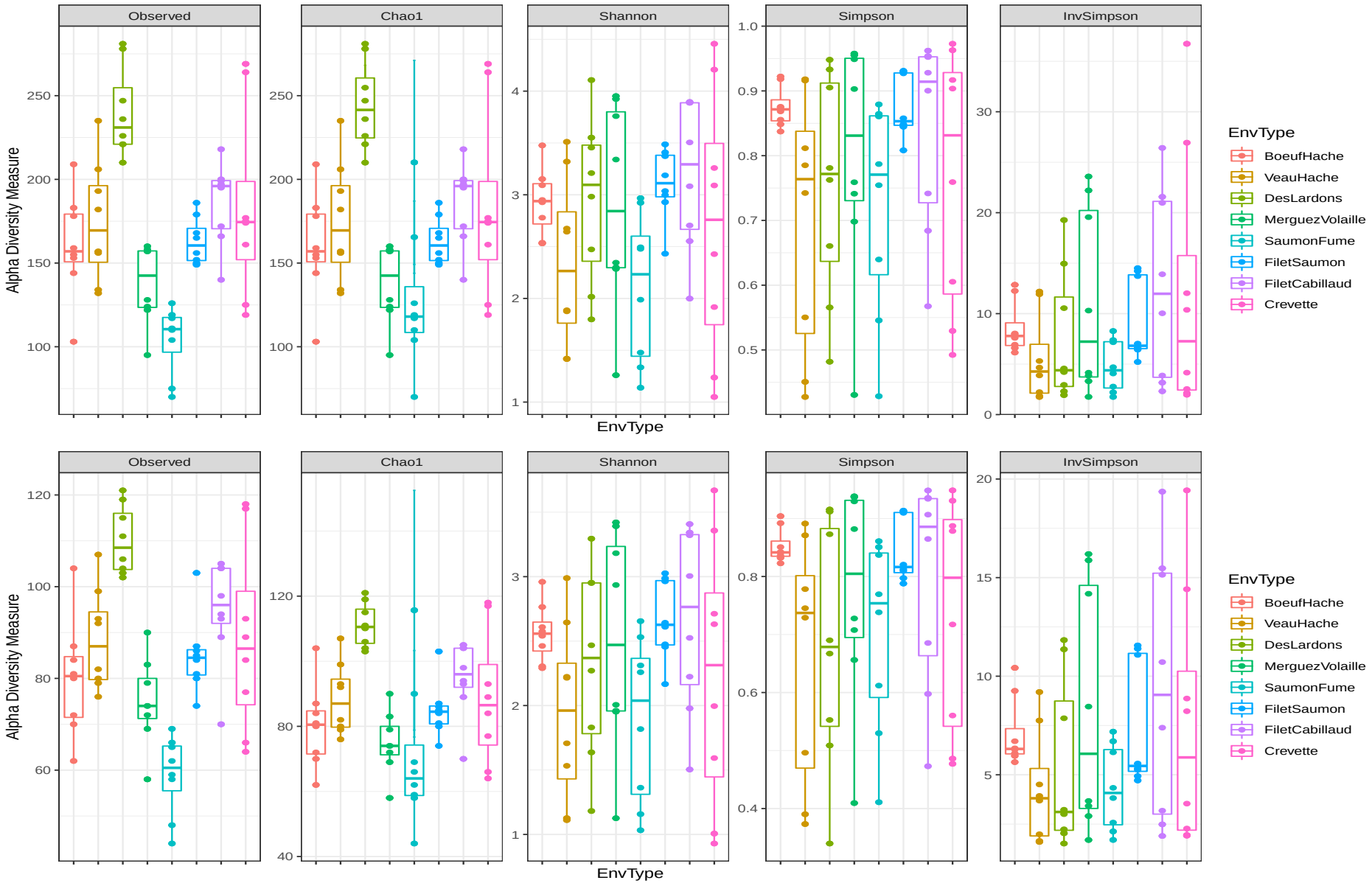
Many α diversities (richness, Chao) depend a lot on rare otus. Do not **trim** rare otus before computing them as it can drastically alter the result (see next slide).

Richness

Richness are plotted with `plot_richness`. Note the `x = "EnvType"` passed on to the `aes` mapping of a `ggplot`.

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

α diversity: without (top) and with (bottom) trimming



α diversity: numeric values

Numeric values of α-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")
```

α diversity: A quick ANOVA

```
data <- cbind(sample_data(food), alpha.diversity)
food.anova <- aov(Observed ~ EnvType, data)
summary(food.anova) ## significant effect of environment type on richness
```

```
##              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
```

```
food.anova <- aov(Shannon ~ EnvType, data)
summary(food.anova) ## effect on Shannon diversity is not significant
```

```
##              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
```

α diversity: A quick ANOVA

```
data <- cbind(sample_data(food), alpha.diversity)
food.anova <- aov(Observed ~ EnvType, data)
summary(food.anova) ## significant effect of environment type on richness
```

```
##              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
```

```
food.anova <- aov(Shannon ~ EnvType, data)
summary(food.anova) ## effect on Shannon diversity is not significant
```

```
##              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
```

Interpretation

Interpretation

- Many taxa observed in `Deslardons` (high Chao1, high Observed)...
- ...but low Shannon and Inverse-Simpson
- $\Rightarrow$ communities dominated by a few abundant taxa

Interpretation

- Environments differ a lot in terms of richness...
- ...but not so much in terms of Shannon diversity
- $\Rightarrow$ *Effective* diversities are quite similar

Interpretation

Interpretation

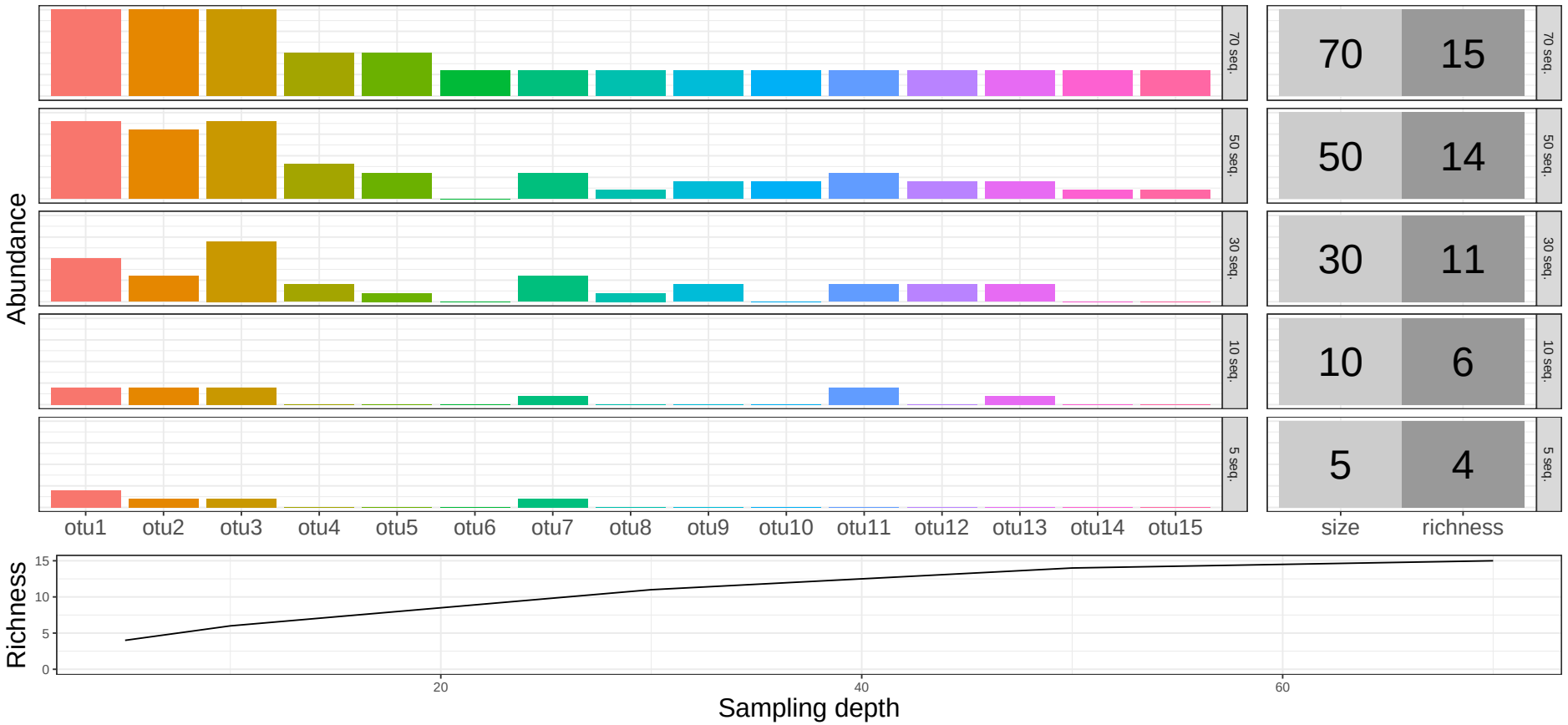
- Many taxa observed in `Deslardons` (high Chao1, high Observed)...
- ...but low Shannon and Inverse-Simpson
- $\Rightarrow$ communities dominated by a few abundant taxa

Interpretation

- Environments differ a lot in terms of richness...
- ...but not so much in terms of Shannon diversity
- $\Rightarrow$ *Effective* diversities are quite similar

Outline

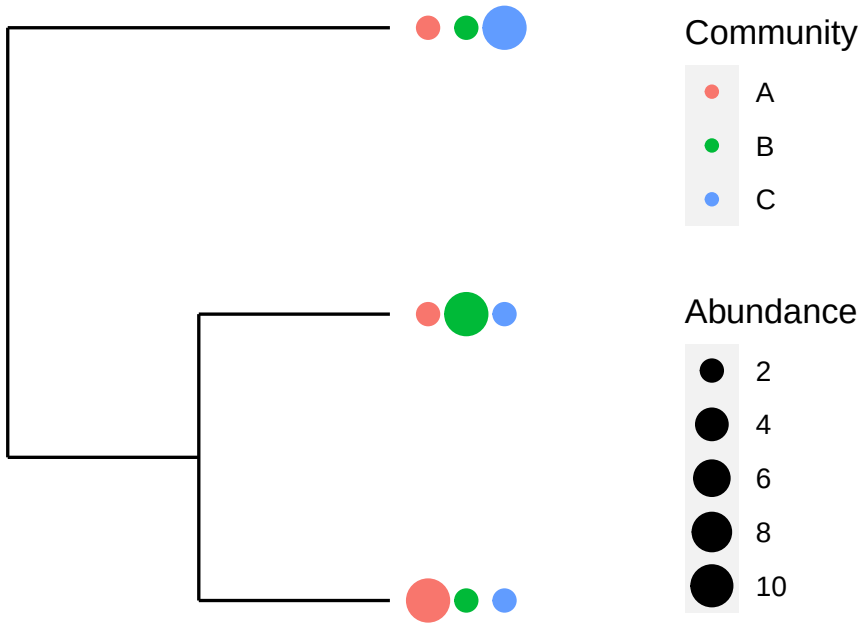
Rarefaction curve (I)



Outline

β dissimilarities

- Many β diversities (both compositional and phylogenetic) offered by `phyloseq` through the generic `distance` function.
- Different dissimilarities capture different features of the communities.



β-diversity: compositional

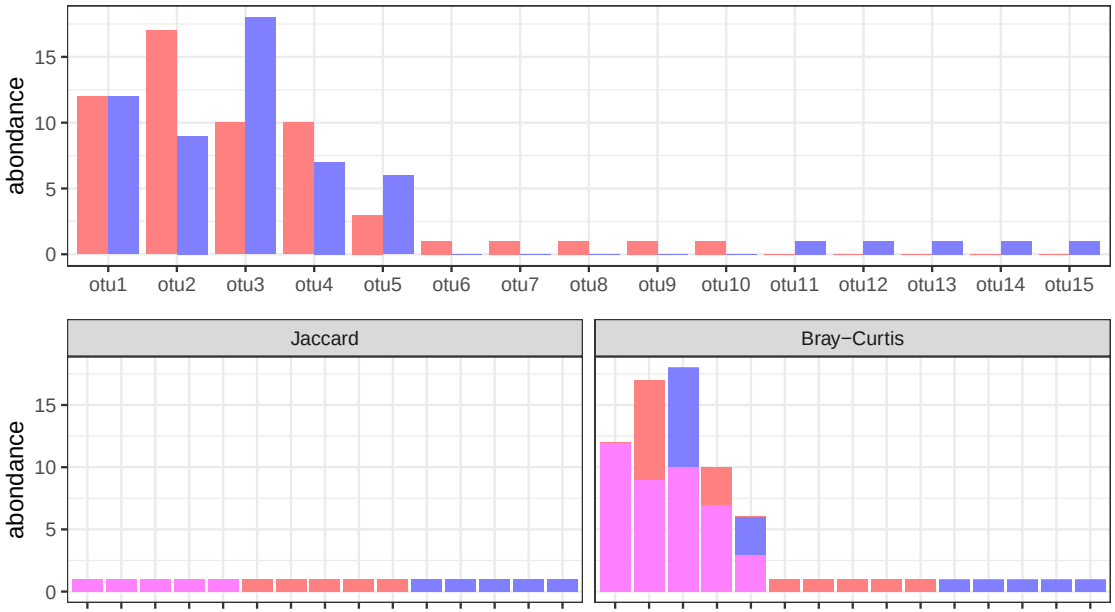
Note n_s^1 the count of species s ($s = 1, \ldots, S$) in community 1 and n_s^2 the count in community 2. We focus on shared features.

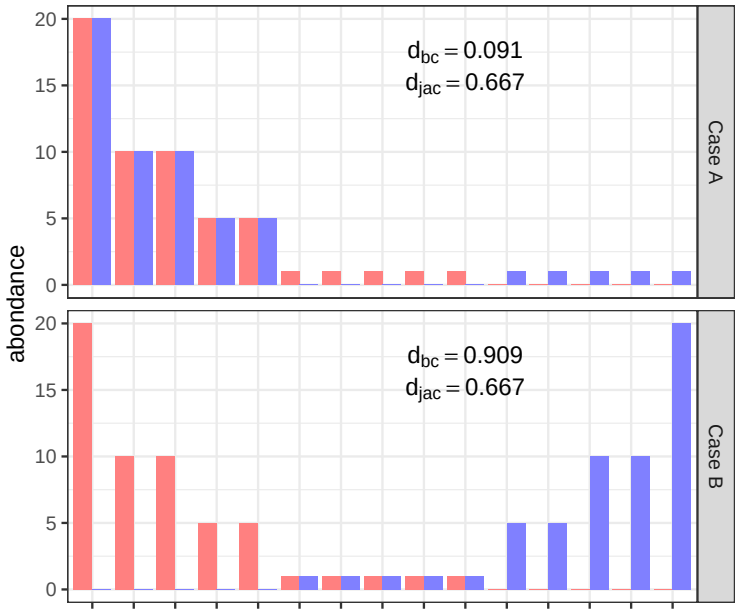
Jaccard	Bray-Curtis
Fraction of species specific to either 1 or 2	Fraction of the community specific to 1 or to 2
$d_{\text{Jac}} = \frac{\sum_s 1_{\{n_s^1>0, n_s^2=0\}} + 1_{\{n_s^2>0, n_s^1=0\}}}{\sum_s 1_{\{n_s^1+n_s^2>0\}}}$	$d_{\text{BC}} = \sum_s \lvert n_s^1 - n_s^2 \rvert / \sum_s \lvert n_s^1 + n_s^2 \rvert$

β-diversity: compositional

Note n_s^1 the count of species s ($s = 1, \ldots, S$) in community 1 and n_s^2 the count in community 2. We focus on shared features.

Jaccard	Bray-Curtis
Fraction of species specific to either 1 or 2	Fraction of the community specific to 1 or to 2
$d_{\text{Jac}} = \frac{\sum_s 1_{\{n_s^1>0, n_s^2=0\}} + 1_{\{n_s^2>0, n_s^1=0\}}}{\sum_s 1_{\{n_s^1+n_s^2>0\}}}$	$d_{\text{BC}} = \sum_s \lvert n_s^1 - n_s^2 \rvert / \sum_s \lvert n_s^1 + n_s^2 \rvert$





β-diversity: phylogenetic

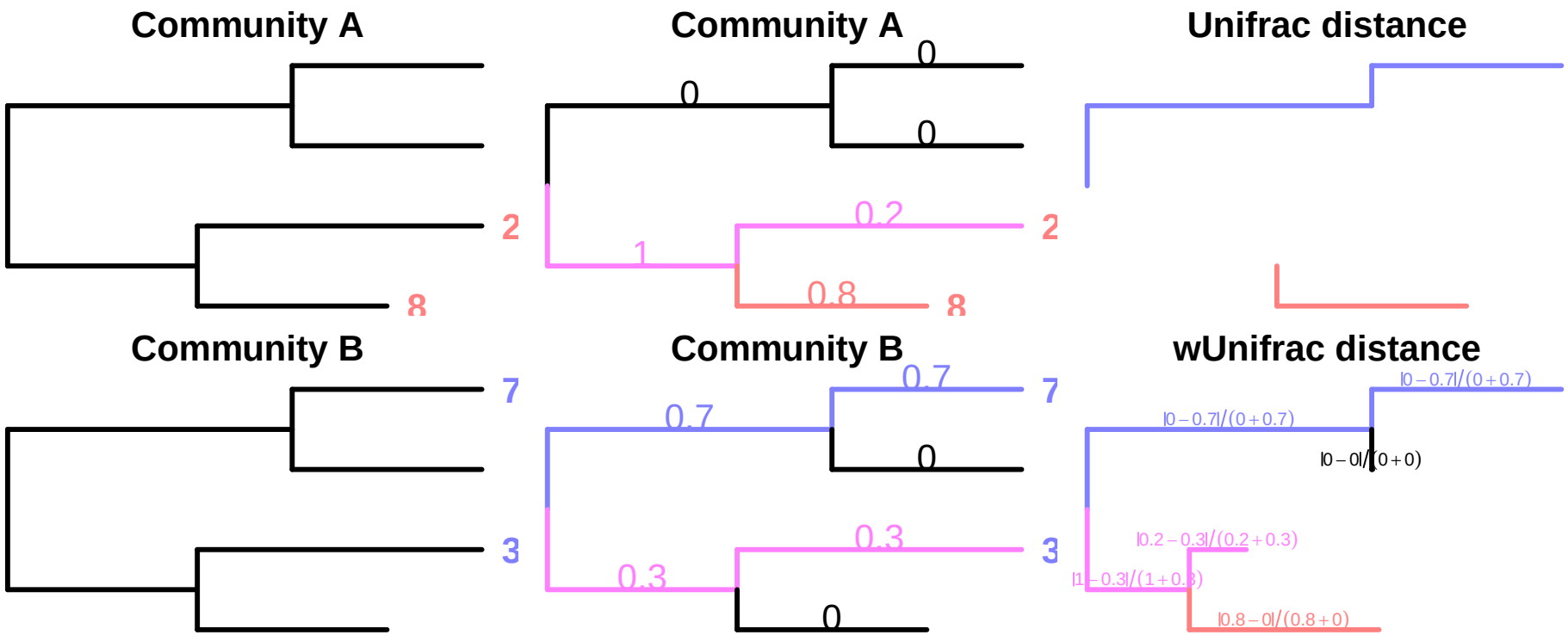
For each branch e, note l_e its length and p_e (resp. q_e) the fraction of community 1 (resp. community 2) below branch e. We focus on shared features.

Unifrac	Weighted Unifrac
Fraction of the tree specific to either 1 or 2	Fraction of the diversity specific to 1 or to 2
$d_{\mathsf{UF}} = \frac{\sum_e l_e \left[1_{\{p_e>0, q_e=0\}} + 1_{\{q_e>0, p_e=0\}}\right]}{\sum_e l_e \times 1_{\{p_e+q_e>0\}}}$	$d_{\mathsf{wUF}} = \frac{\sum_e l_e \lvert p_e - q_e \rvert}{\sum_e l_e (p_e+q_e)}$

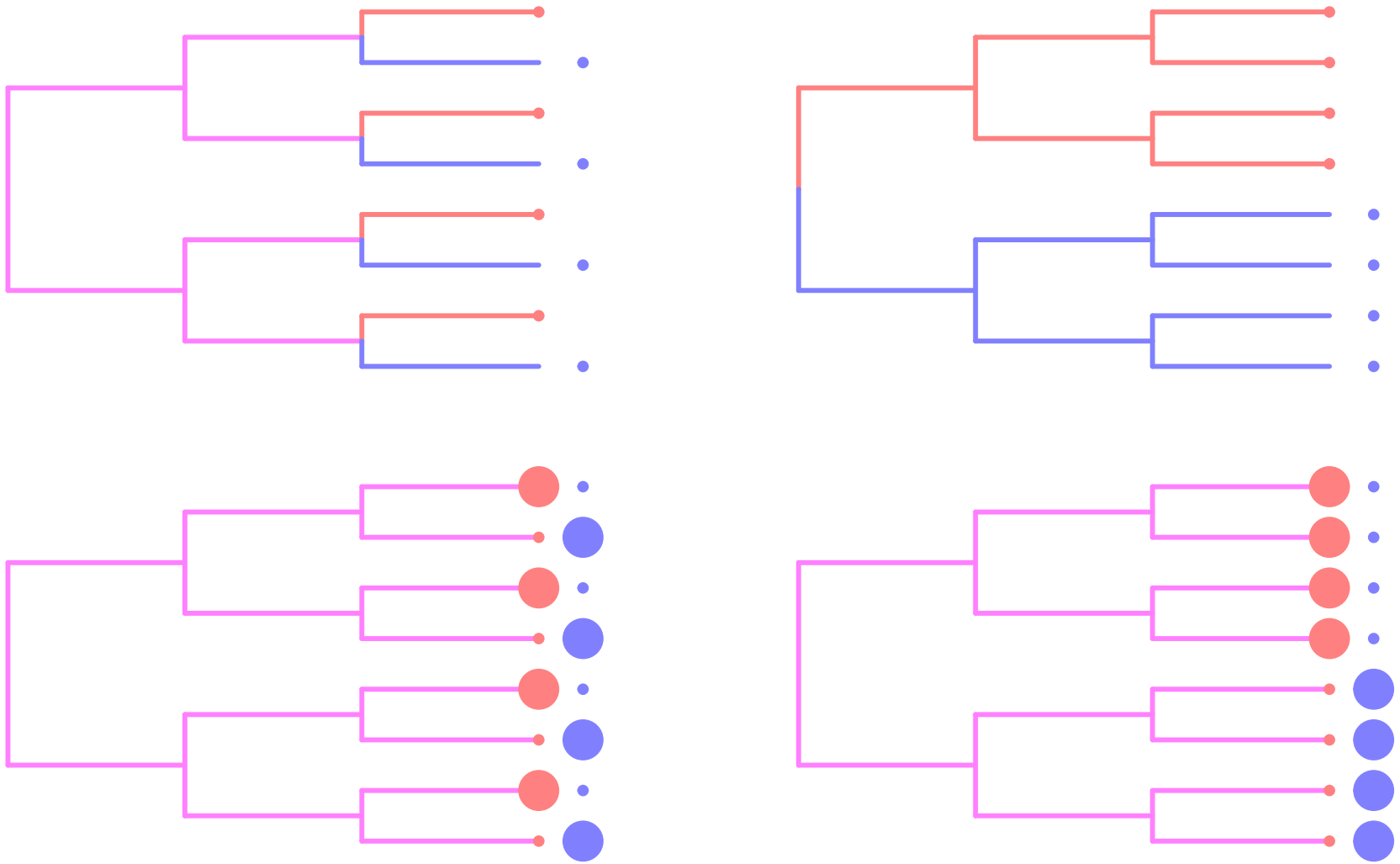
β-diversity: phylogenetic

For each branch e, note l_e its length and p_e (resp. q_e) the fraction of community 1 (resp. community 2) below branch e. We focus on shared features.

Unifrac	Weighted Unifrac
Fraction of the tree specific to either 1 or 2	Fraction of the diversity specific to 1 or to 2
$d_{\text{UF}} = \frac{\sum_e l_e \left[1_{\{p_e>0, q_e=0\}} + 1_{\{q_e>0, p_e=0\}}\right]}{\sum_e l_e \times 1_{\{p_e+q_e>0\}}}$	$d_{\text{wUF}} = \frac{\sum_e l_e \lvert p_e - q_e \rvert}{\sum_e l_e (p_e+q_e)}$

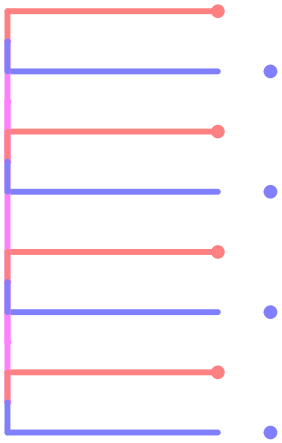


Differences between the β-dissimilarities

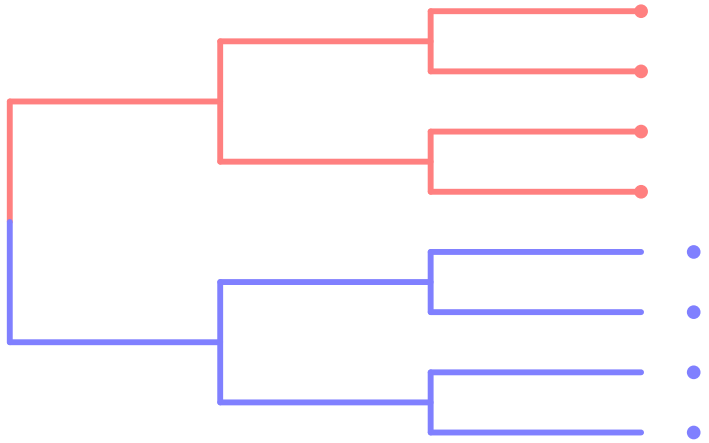


Differences between the β-dissimilarities

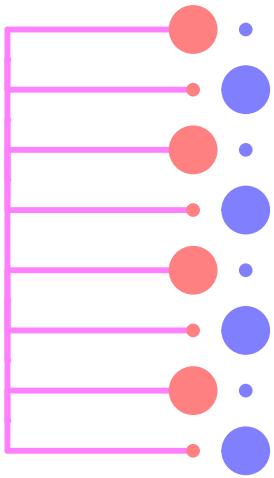
low UF, high Jac



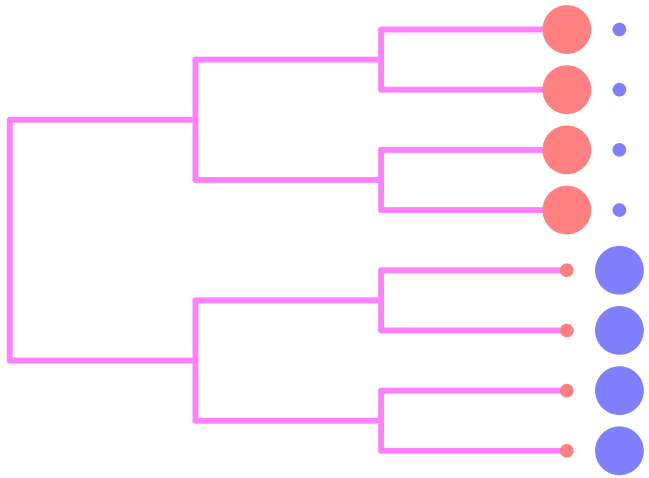
high UF, high Jac



low wUF, high BC



high wUF, high BC



β-dissimilarities/distances in phyloseq

β dissimilarities are computed with `distance`

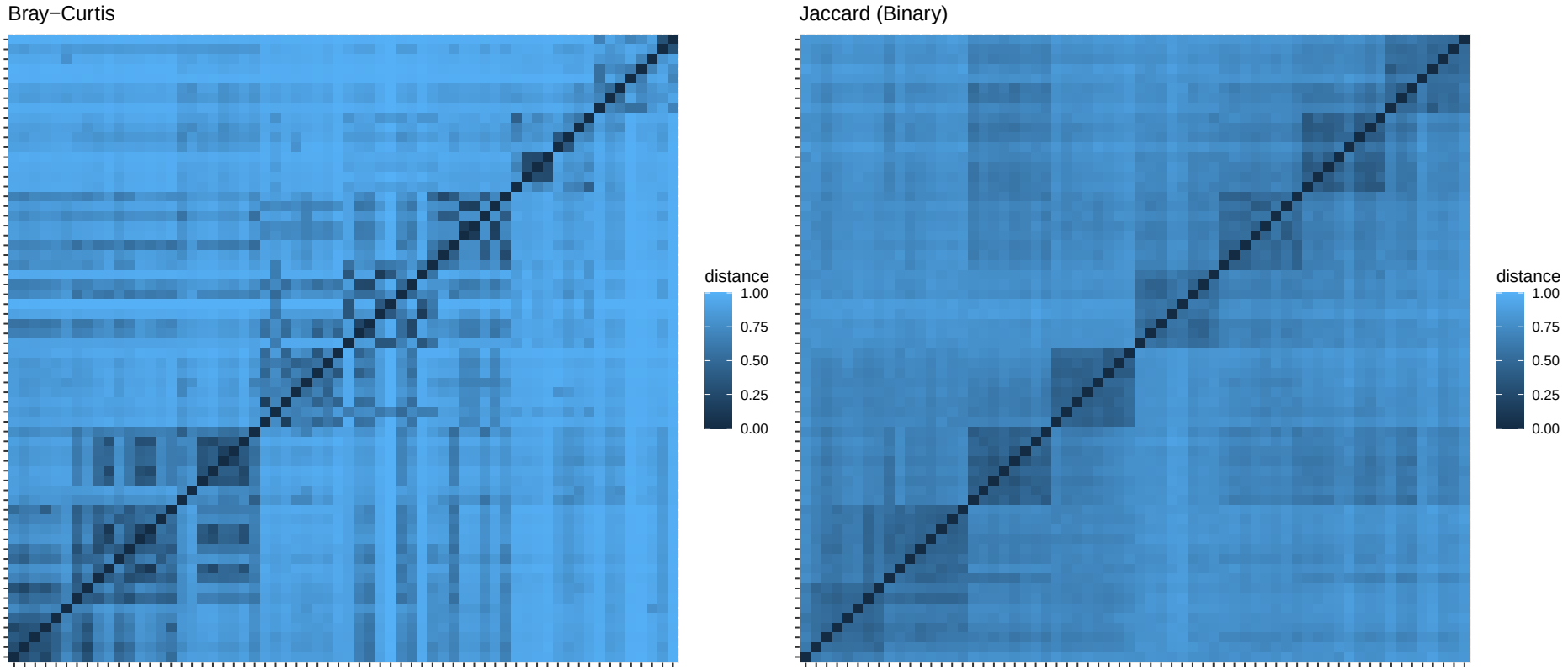
```
dist.bc <- distance(food, method = "bray") ## Bray-Curtis
```

All available distances are available with

```
distanceMethodList

## $UniFrac
## [1] "unifrac"  "wunifrac"
##
## $DPCoA
## [1] "dpcoa"
##
## $JSD
## [1] "jsd"
##
## $vegdist
##  [1] "manhattan"  "euclidean"  "canberra"   "bray"       "kulczynski"
##  [6] "jaccard"    "gower"      "altGower"   "morisita"   "horn"
## [11] "mountford"  "raup"       "binomial"   "chao"       "cao"
##
## $betadiver
##  [1] "w"   "-1"  "c"   "wb"  "r"   "I"   "e"   "t"   "me"  "j"   "sor" "m"
## [13] "-2"  "co"  "cc"  "g"   "-3"  "l"   "19"  "hk"  "rlb" "sim" "gl"  "z"
```

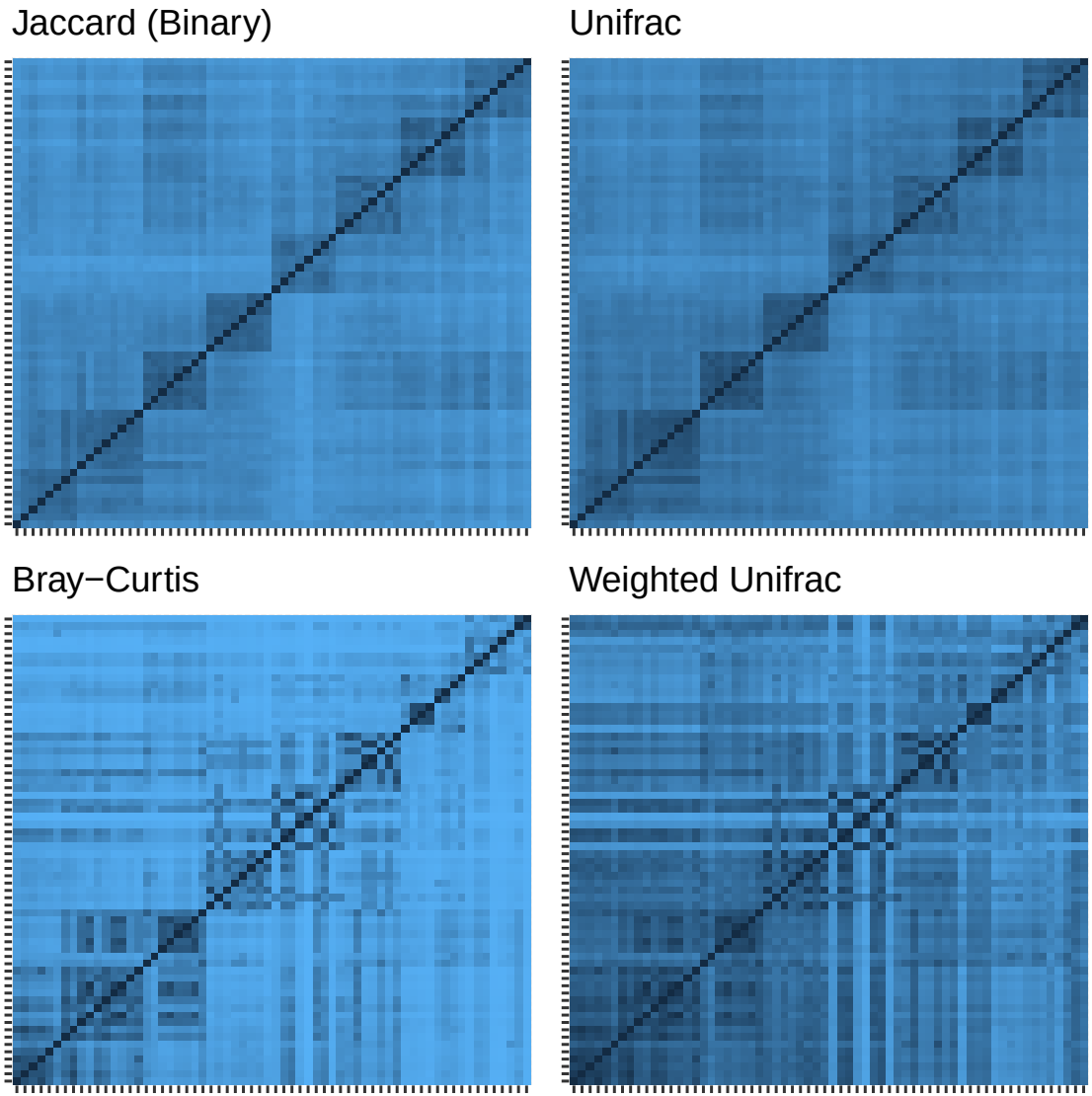
β-dissimilarities/distances in `phyloseq` (II)



Phylogenetic β-dissimilarities/distances in phyloseq (II)

```
dist.uf <- distance(food, method = "unifrac") ## Unifrac
dist.wuf <- distance(food, method = "wunifrac") ## Weighted Unifrac
```

Compositional vs Qualitative



Compositional vs Qualitative (II)

- Jaccard lower than Bray-Curtis $\Rightarrow$ abondant taxa are not shared
- Jaccard higher than Unifrac $\Rightarrow$ communities' taxa are distinct but phylogenetically related
- Unifrac higher than weighted Unifrac $\Rightarrow$ abondant taxa in both communities are phylogenetically close.

General remarks about β diversity

In general, qualitative diversities are most 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 of no flow 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"

General remarks about β diversity

In general, qualitative diversities are most 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 of no flow 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"

Outline

PCA and MDS

Principal Component Analysis (PCA)

- 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 β-diversity.

PCA and MDS

Principal Component Analysis (PCA)

- 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

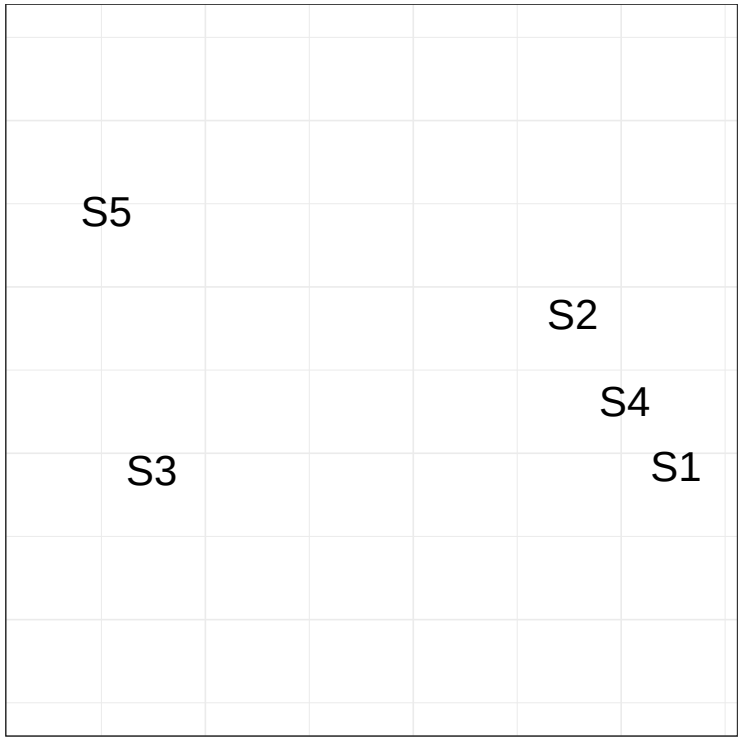
But variance is not a very good measure of β-diversity.

MultiDimensional Scaling (MDS/PCoA)

MDS/PCoA

- Start from a distance matrix $D = (d_{ij})$
- Project the communities $\text{Com}_i \mapsto X_i$ in a euclidian space such that distances are preserved $\|X_i - X_j\| \simeq d_{ij}$

	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

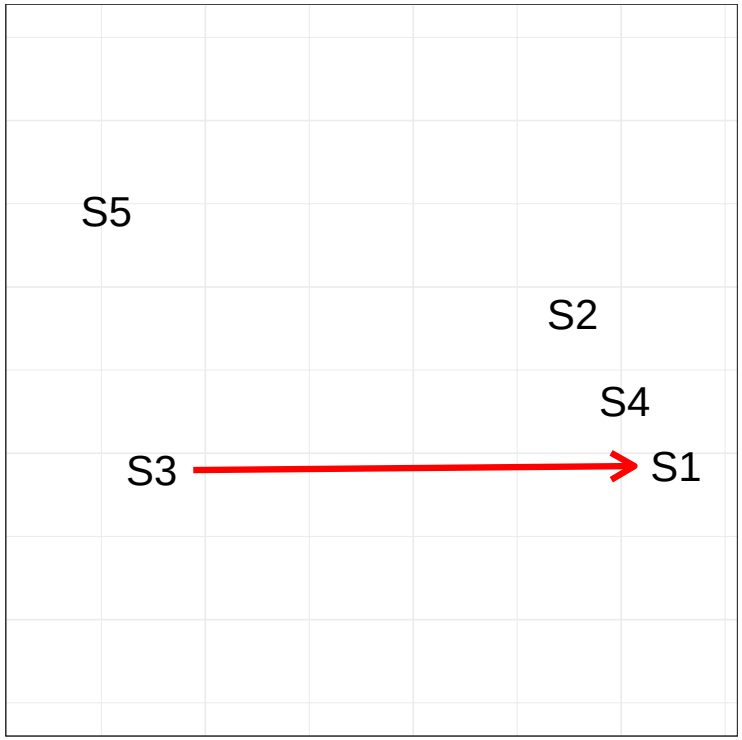


MultiDimensional Scaling (MDS/PCoA)

MDS/PCoA

- Start from a distance matrix $D = (d_{ij})$
- Project the communities $\text{Com}_i \mapsto X_i$ in a euclidian space such that distances are preserved $\|X_i - X_j\| \simeq d_{ij}$

	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

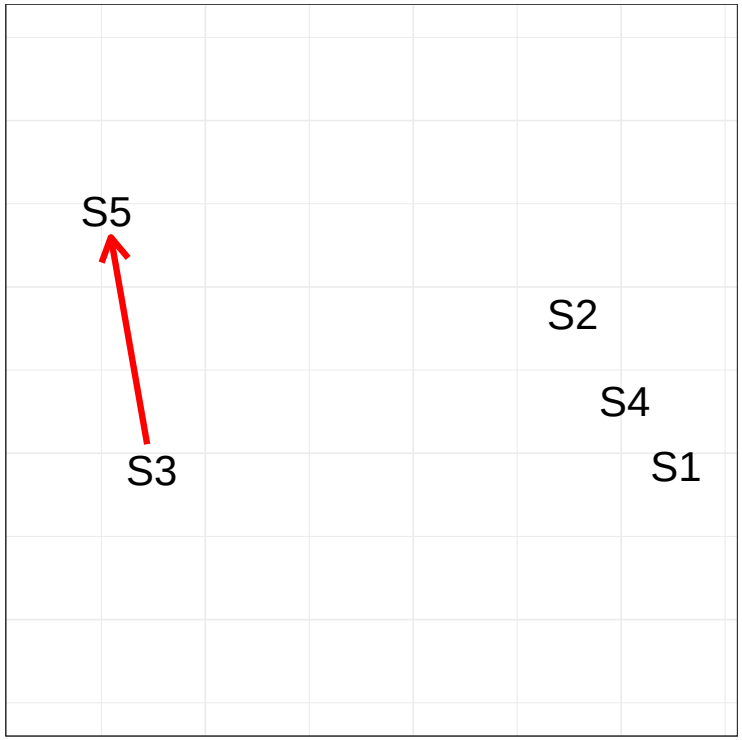


MultiDimensional Scaling (MDS/PCoA)

MDS/PCoA

- Start from a distance matrix $D = (d_{ij})$
- Project the communities $\text{Com}_i \mapsto X_i$ in a euclidian space such that distances are preserved $\|X_i - X_j\| \simeq d_{ij}$

	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



Ordination in phyloseq : ordinate

Ordination is done through the ordinate function:

Ordination

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)
```

The graphic is then produced with plot_ordination

```
p <- plot_ordination(food, ord, color = "EnvType")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
plot(p)
```

Ordination in phyloseq : ordinate

Ordination is done through the ordinate function:

Ordination

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)
```

The graphic is then produced with plot_ordination

```
p <- plot_ordination(food, ord, color = "EnvType")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
plot(p)
```

Ordination in phyloseq : ordinate

Ordination is done through the ordinate function:

Ordination

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

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

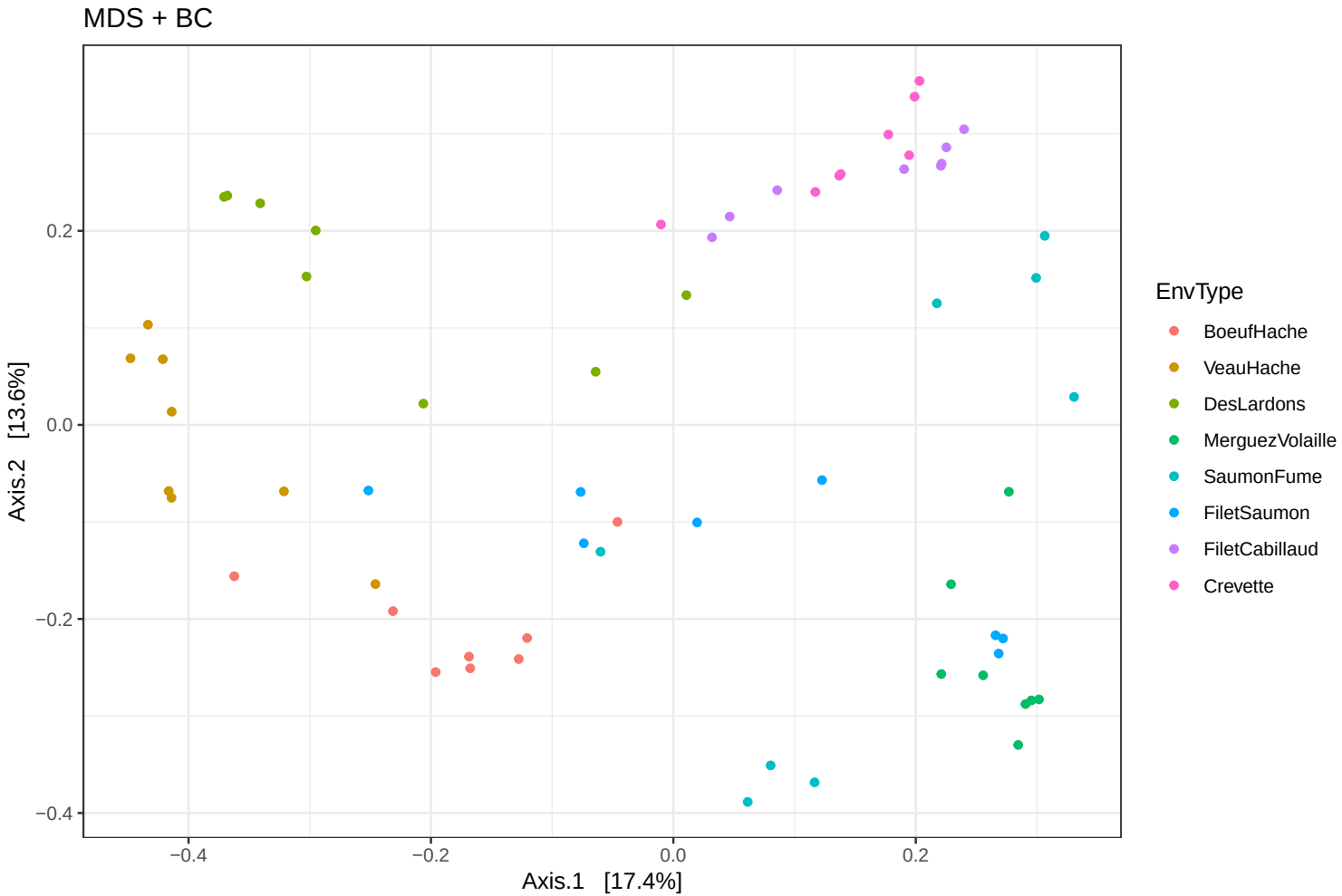
or by passing a distance matrix directly (useful if you already computed it)

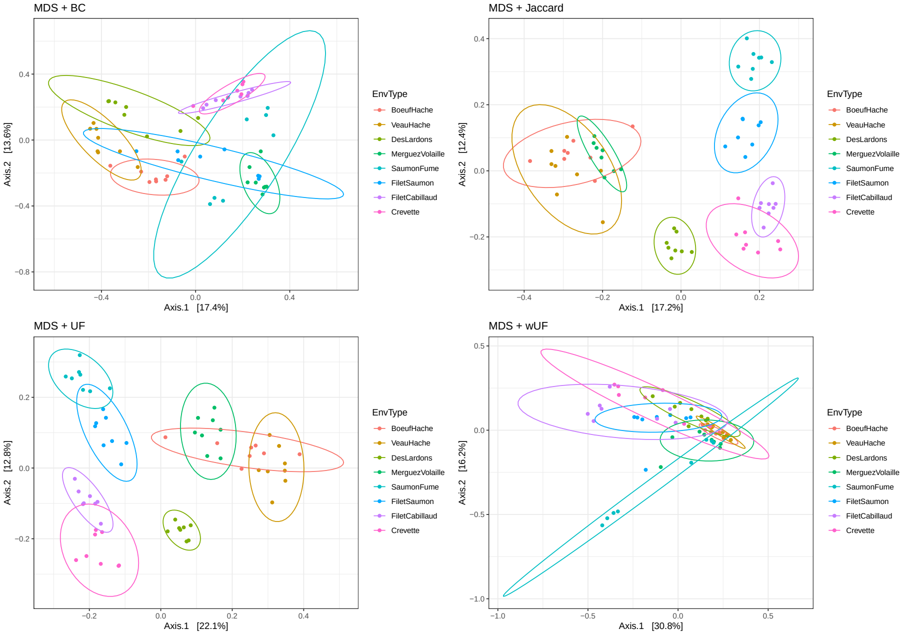
```
dist.bc <- distance(food, method = "bray")
ord <- ordinate(food, method = "MDS", distance = dist.bc)
```

The graphic is then produced with plot_ordination

```
p <- plot_ordination(food, ord, color = "EnvType")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
plot(p)
```

Ordination in phyloseq : plot_ordination




MDS + BC
MDS + Jaccard
MDS + UF
MDS + wUF
Axis.1 [17.4%]
Axis.2 [13.6%]
Axis.1 [17.2%]
Axis.2 [12.4%]
Axis.1 [22.1%]
Axis.2 [12.8%]
Axis.1 [30.8%]
Axis.2 [16.2%]
EnvType
BoeufHache
VeauHache
DesLardons
MerguezVolaille
SaumonFume
FiletSaumon
FiletCabillaud
Crevette

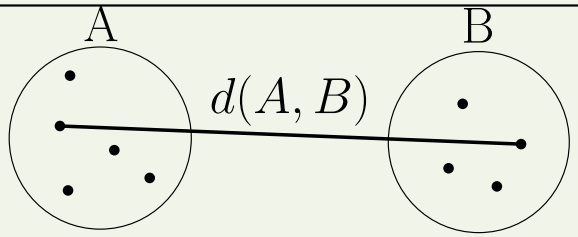
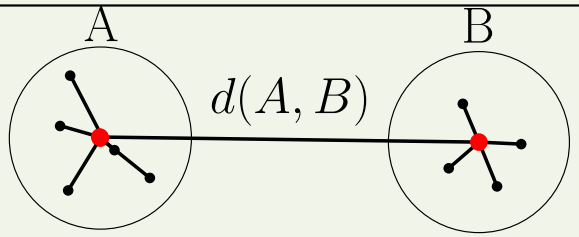
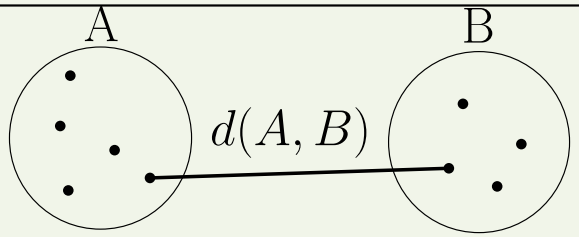
Interpretation

- Qualitative distances (Unifrac, Jaccard) separate meat products from seafood ones $\Rightarrow$ detected taxa segregate by origin
- `DesLardons` is somewhere in between $\Rightarrow$ 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.

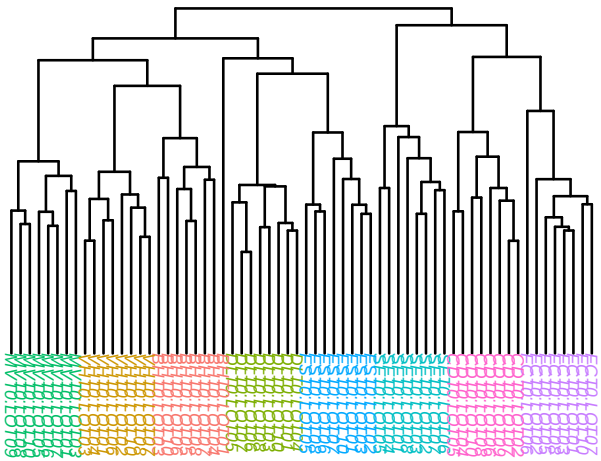
Outline

Hierarchical Clustering

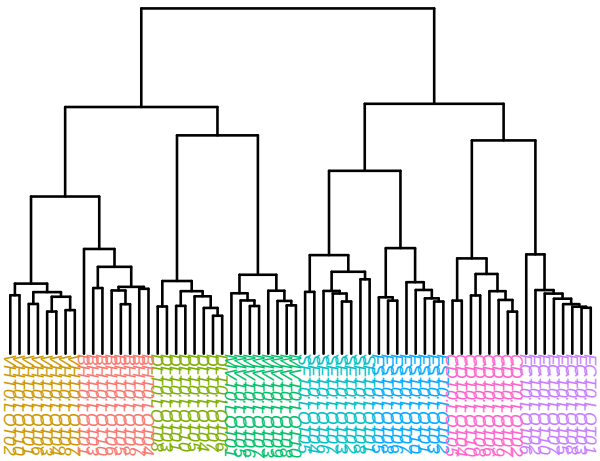
- Merge closest communities (according to some distance)
- Update distances between sets of communities using linkage function
- Repeat until all communities have been merged

Complete	Ward	Single
		

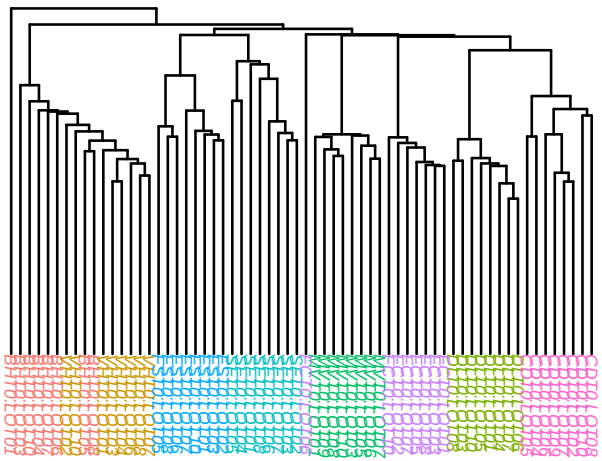
complete linkage



ward.D2 linkage



single linkage



Clustering with `hclust`

- Choose a distance (among Jaccard, Bray-Curtis, Unifrac, etc)
- Choose a linkage function

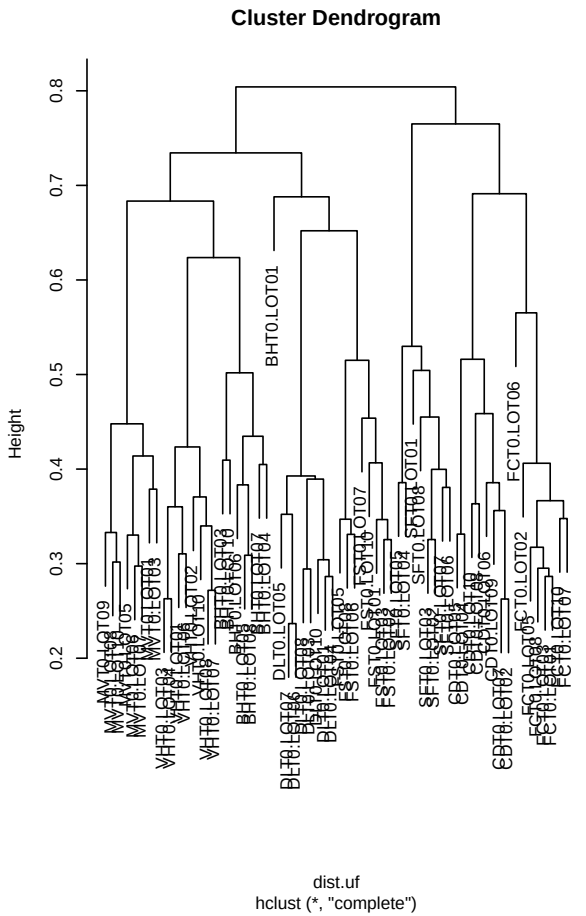
Feed to `hclust` and plot

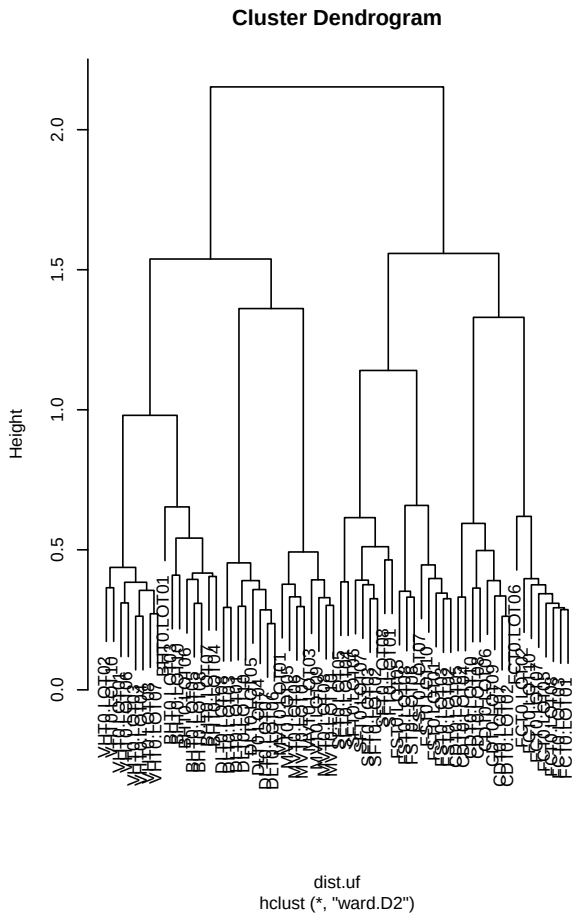
```
clustering <- hclust(distance.matrix, method = "linkage.function")
plot(clustering)
```

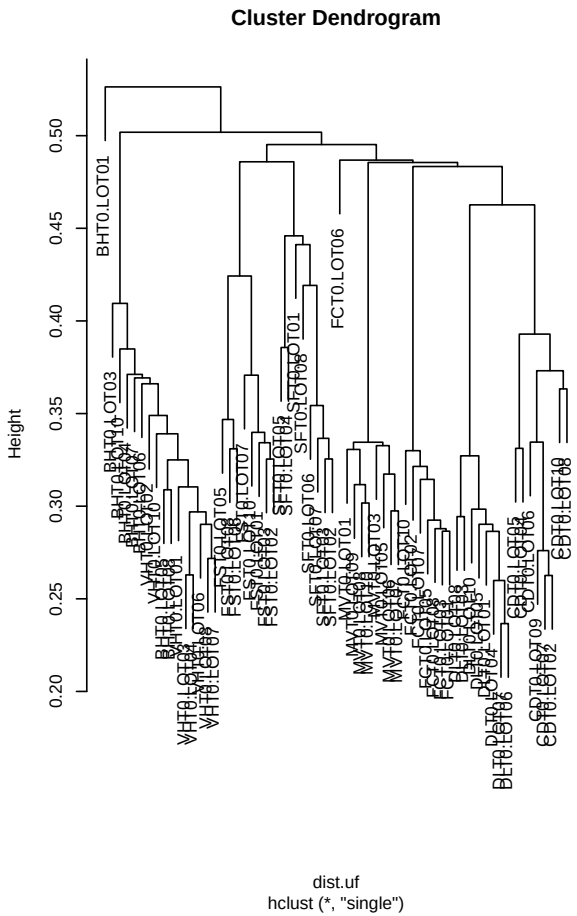
linkage function

- **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.

```r
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"))
```





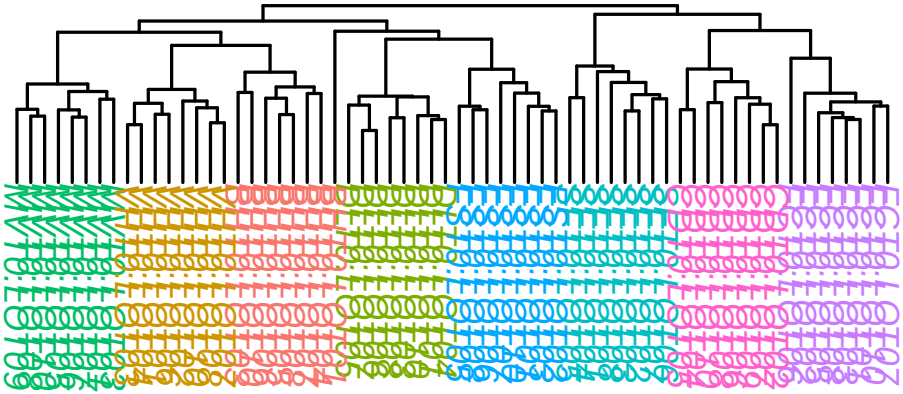


Better dendrograms

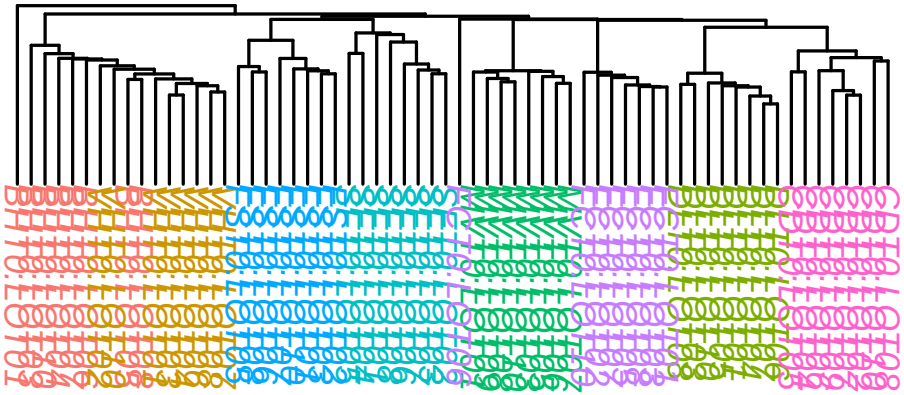
With some effort (see companion R script), we can produce better dendrograms and color sample by food type (appreciate what `ggplot` does for you behind the hook).

```
## 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))
clust.uf <- as.phylo(hclust(dist.uf, method = "complete"))
plot(clust.uf, tip.color = tipColor, direction = "downwards",
     main = paste(method, "linkage"))
```

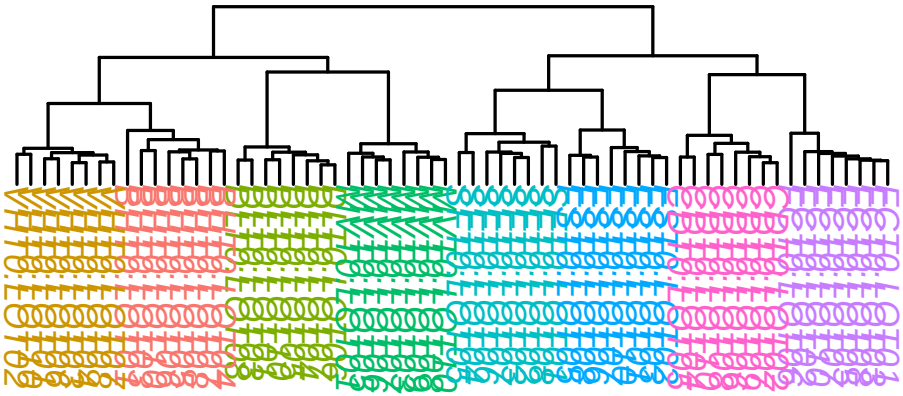
complete linkage



single linkage



ward.D2 linkage



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

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)

Outline

Heatmap with `plot_heatmap`

`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 `gpplot2` object, easy to manipulate and customize

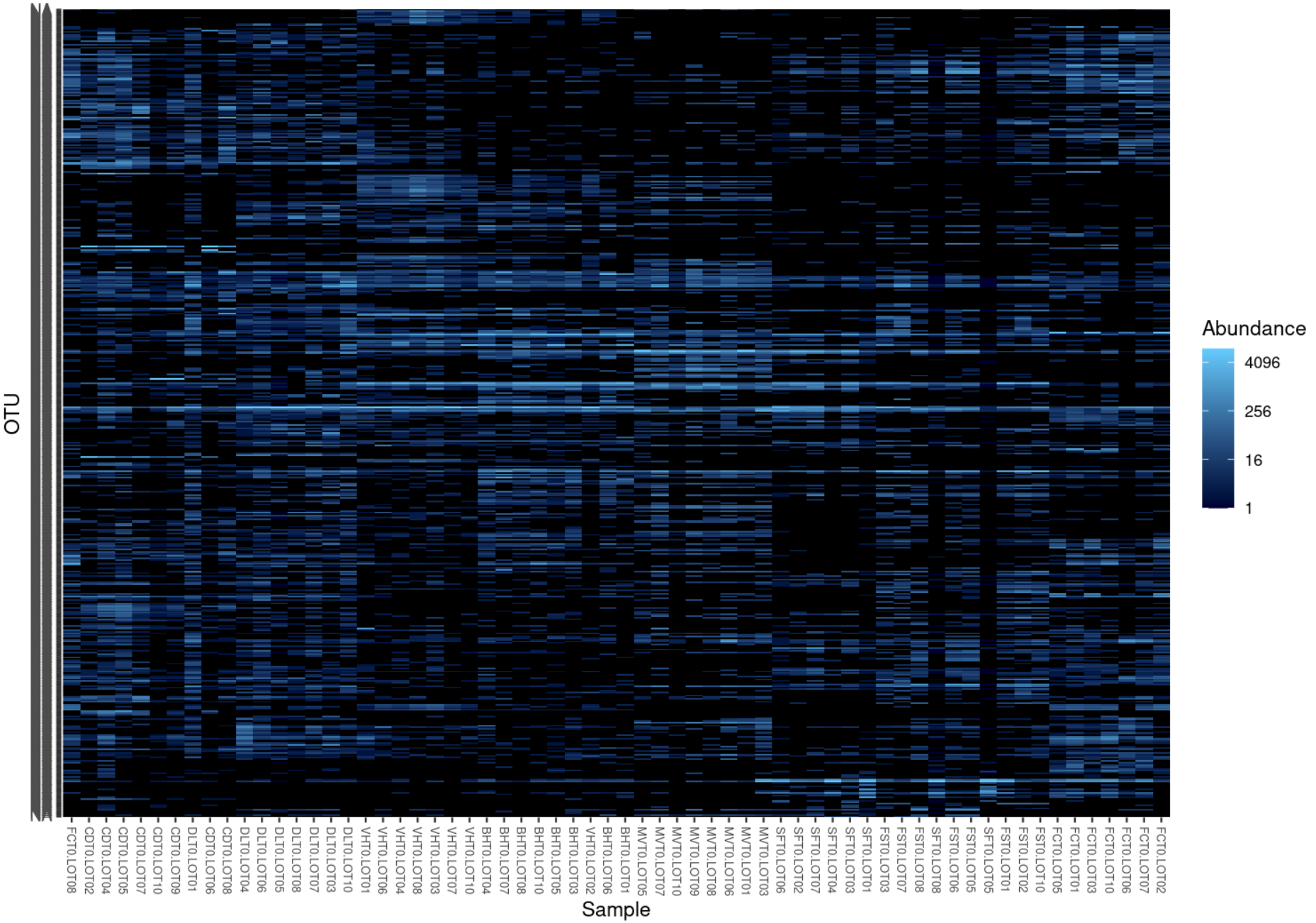
```
p <- plot_heatmap(food, low = "yellow", high = "red", na.value = "white",
                  sample.order = mySampleOrder, taxa.order = myTaxaOrder)
## add facetting
p <- p + facet_grid(~EnvType, scales = "free_x")
plot(p)
```

Heatmap with plot_heatmap

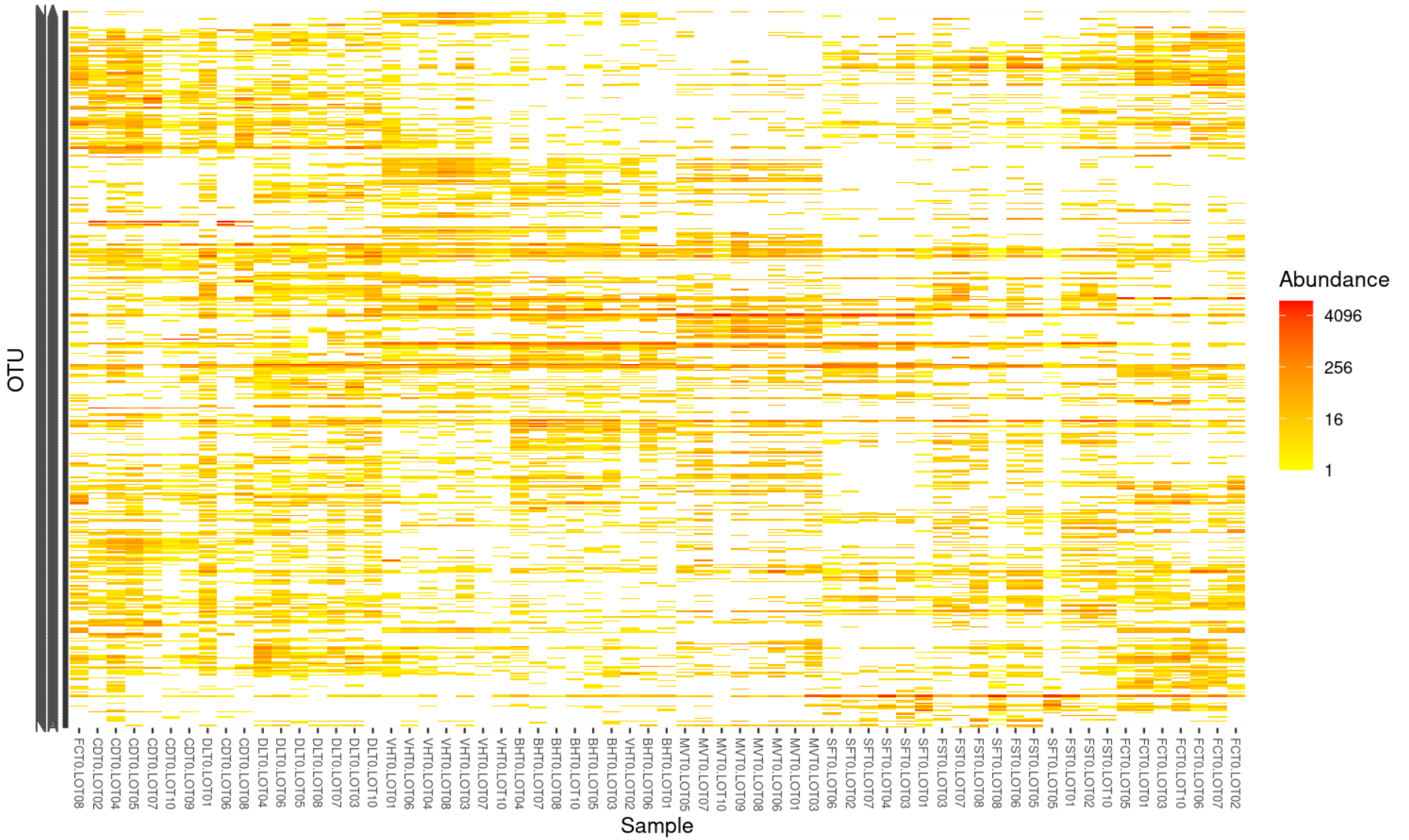
plot_heatmap is a versatile function to visualize the count table.

- 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 gpplot2 object, easy to manipulate and customize

```
p <- plot_heatmap(food, low = "yellow", high = "red", na.value = "white",
                  sample.order = mySampleOrder, taxa.order = myTaxaOrder)
## add facetting
p <- p + facet_grid(~EnvType, scales = "free_x")
plot(p)
```

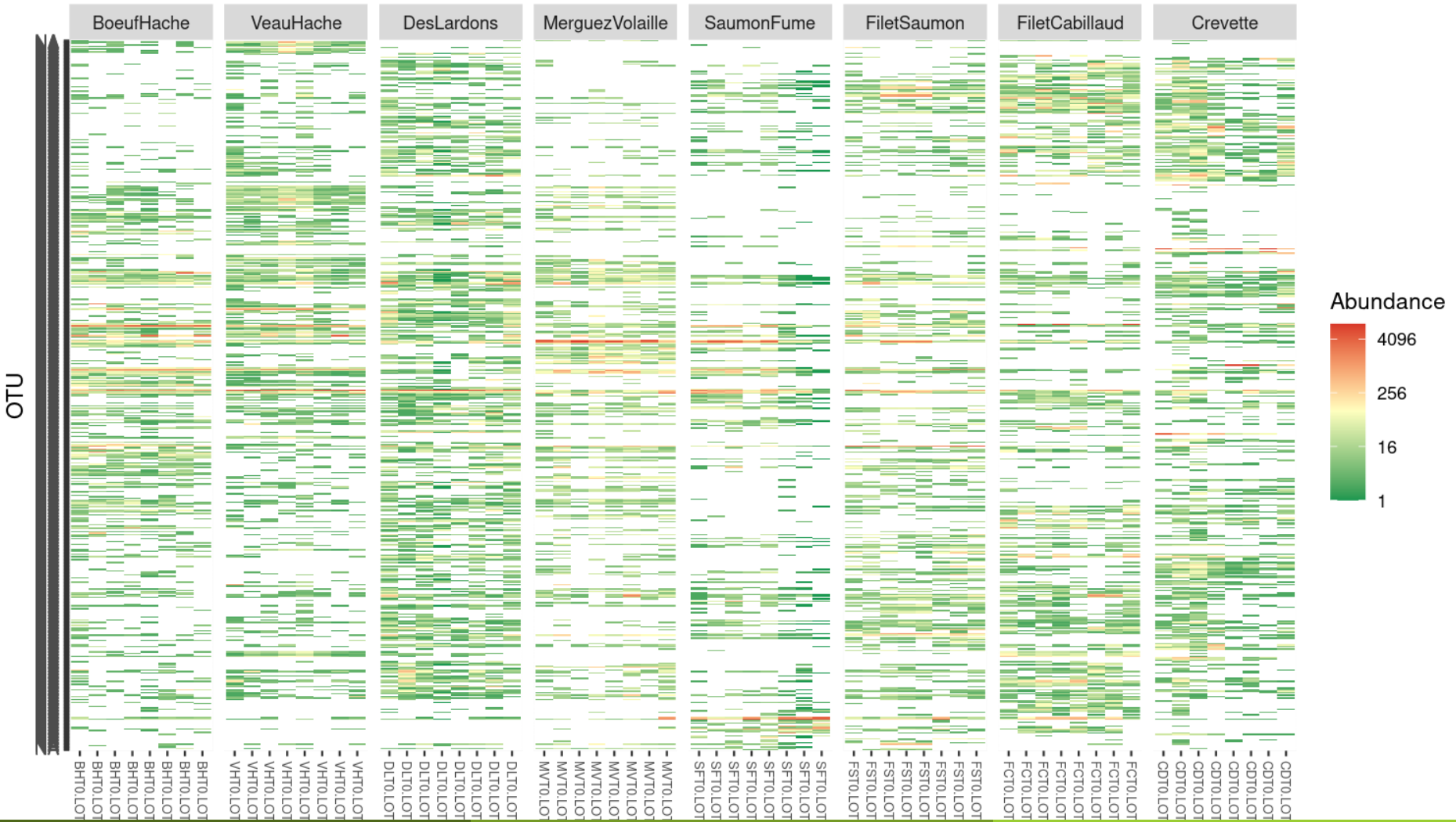

OTU
Sample
Abundance
4096
256
16
1
FCT0.LOT08
CDT0.LOT02
CDT0.LOT04
CDT0.LOT05
CDT0.LOT07
CDT0.LOT10
CDT0.LOT09
DLT0.LOT01
CDT0.LOT06
CDT0.LOT08
DLT0.LOT04
DLT0.LOT06
DLT0.LOT05
DLT0.LOT08
DLT0.LOT07
DLT0.LOT03
DLT0.LOT10
VHT0.LOT01
VHT0.LOT06
VHT0.LOT04
VHT0.LOT08
VHT0.LOT03
VHT0.LOT07
VHT0.LOT10
BHT0.LOT04
BHT0.LOT07
BHT0.LOT08
BHT0.LOT10
BHT0.LOT05
BHT0.LOT03
VHT0.LOT02
BHT0.LOT06
BHT0.LOT01
MVT0.LOT05
MVT0.LOT07
MVT0.LOT10
MVT0.LOT09
MVT0.LOT08
MVT0.LOT06
MVT0.LOT01
MVT0.LOT03
SFT0.LOT06
SFT0.LOT02
SFT0.LOT07
SFT0.LOT04
SFT0.LOT03
SFT0.LOT01
FST0.LOT03
FST0.LOT07
FST0.LOT08
SFT0.LOT08
FST0.LOT06
FST0.LOT05
SFT0.LOT05
SFT0.LOT01
FST0.LOT02
FST0.LOT10
FCT0.LOT05
FCT0.LOT01
FCT0.LOT03
FCT0.LOT10
FCT0.LOT06
FCT0.LOT07
FCT0.LOT02

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



```
plot_heatmap(food, low = "yellow", high = "red", na.value = "white") +
    facet_grid(~EnvType, scales = "free_x")
```

```
plot_heatmap(food) +
    scale_fill_gradient2(low = "#1a9850", mid = "#ffffbf", high = "#d73027"
                         na.value = "white", trans = log_trans(4),
                         midpoint = log(100, base = 4)) +
    facet_grid(~EnvType, scales = "free_x")
```



Interpretation

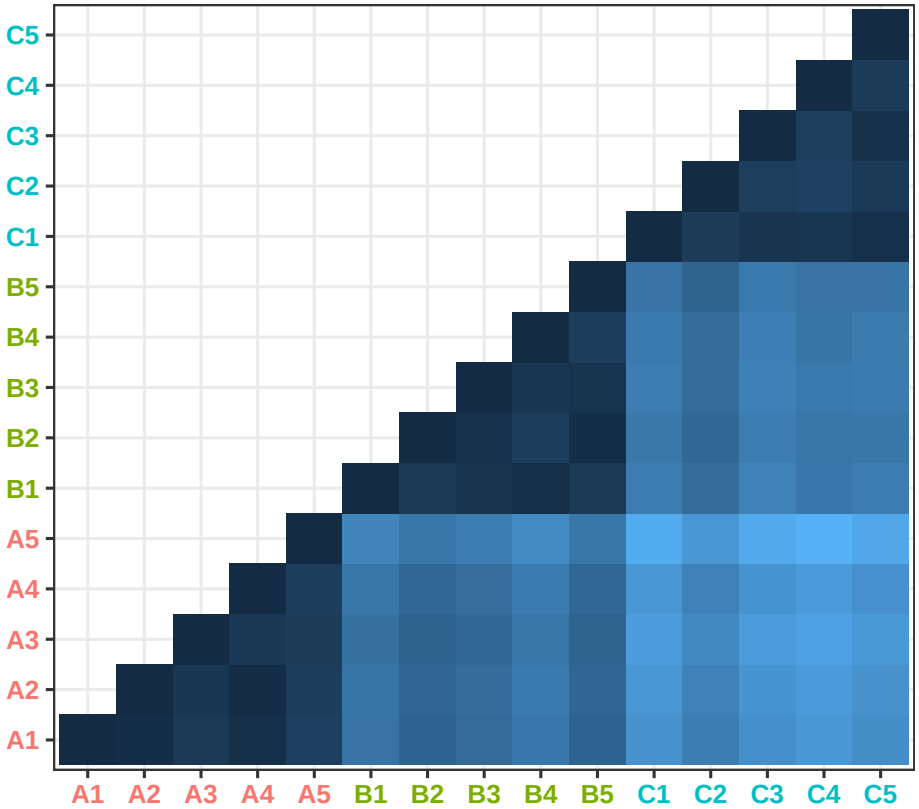
- Block-like structure of the abundance table
- Interaction between (groups of) taxa and (groups of) samples
- Core and condition-specific microbiota
- $\Rightarrow$ Classification of taxa and use of custom taxa order to highlight structure

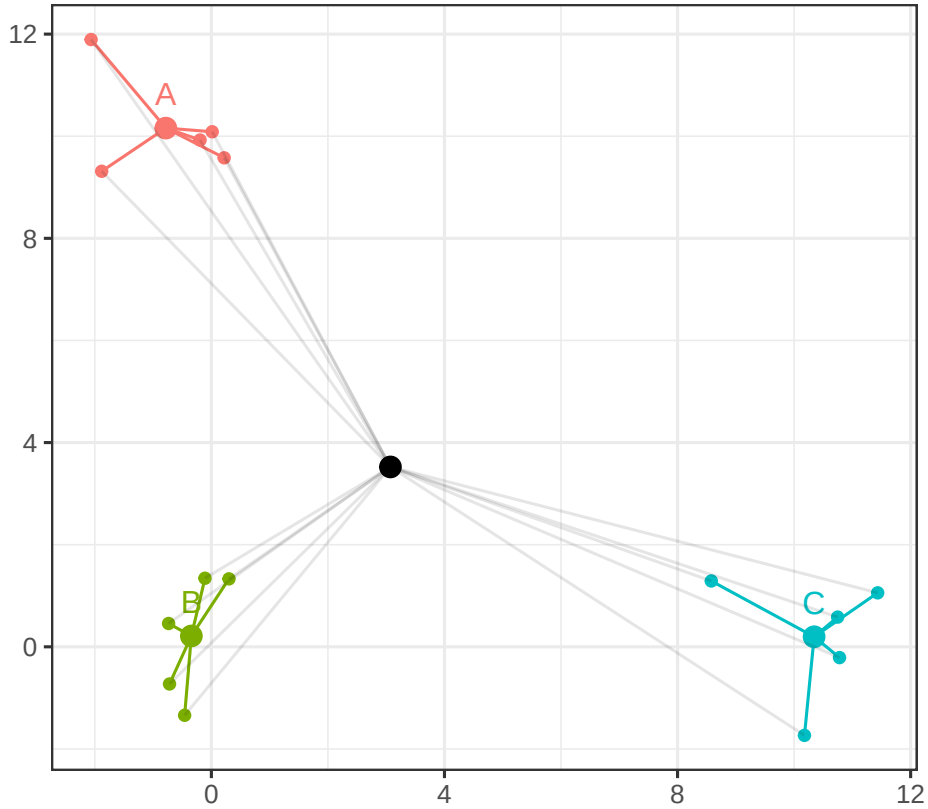
Outline

Rationale

Idea

- Test composition differences of communities from different groups using a distance matrix
- Compare within group to between group distances





Outline

Constrained Analysis of Principal Coordinates (CAP)

Idea

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

Method	Input	Steps	Axis	Variation explained
PCA	X (sample $\times$ var.)	$X \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Variance of samples (rows of X)
RDA	X (sample $\times$ var.) Y (sample $\times$ otus)	$(Y, X) \xrightarrow{Proj.} \hat{Y}(X)$ $\hat{Y}(X) \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Variance of projected samples (rows of $\hat{Y}(X)$)
CAP	X (sample $\times$ var.) D (samp. $\times$ samp.)	$D \xrightarrow{PCoA/MDS} Y$ $(Y, X) \xrightarrow{Proj.} \hat{Y}(X)$ $\hat{Y}(X) \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Distance between samples

Constrained Analysis of Principal Coordinates (CAP)

Idea

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

Method	Input	Steps	Axis	Variation explained
PCA	X (sample $\times$ var.)	$X \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Variance of samples (rows of X)
RDA	X (sample $\times$ var.) Y (sample $\times$ otus)	$(Y, X) \xrightarrow{Proj.} \hat{Y}(X)$ $\hat{Y}(X) \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Variance of projected samples (rows of $\hat{Y}(X)$)
CAP	X (sample $\times$ var.) D (samp. $\times$ samp.)	$D \xrightarrow{PCoA/MDS} Y$ $(Y, X) \xrightarrow{Proj.} \hat{Y}(X)$ $\hat{Y}(X) \xrightarrow{PCA}$ Axis	Lin. comb. of var. (columns of X)	Distance between samples

CAP with capscale (I)

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

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

CAP with capscale (II)

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

```
cap

## Call: capscale(formula = dist.uf ~ EnvType, data = metadata)
##
##                Inertia Proportion Rank
## Total         12.127840   1.000000
## Constrained    7.657073   0.631363    7
## Unconstrained  4.503170   0.371308   56
## Imaginary     -0.032403  -0.002672    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
## (Showing 8 of 56 unconstrained eigenvalues)
```

CAP with capscale (III)

```
cap <- capscale(dist.uf ~ EnvType, data = metadata)
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 SumOfSqs      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
```

Assumptions and caveats

Assumptions

- Community composition responds linearly to environmental changes
- Permutation test can accommodate complex designs

Caveats

- Inadequate for non-linear responses
- Permutation should preserve the design (nestedness)

Assumptions and caveats

Assumptions

- Community composition responds linearly to environmental changes
- Permutation test can accommodate complex designs

Caveats

- Inadequate for non-linear responses
- Permutation should preserve the design (nestedness)

Outline

Multivariate ANOVA

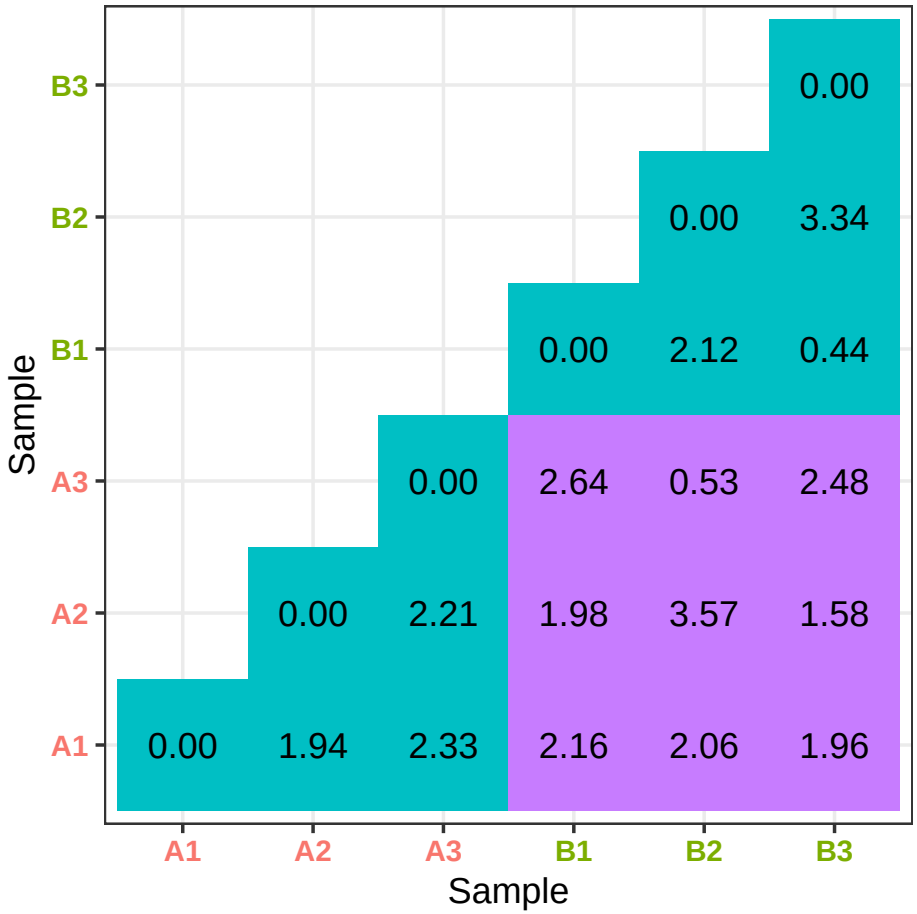
Idea

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

Multivariate ANOVA

Idea

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



Multivariate ANOVA with adonis

Sample type explains again roughly 63% of the total variation.

```
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
```

Assumptions behind Multivariate ANOVA

Assumptions

- PERMANOVA tests location effect ($\simeq$ mean)
- PERMANOVA assumes equal dispersions ($\simeq$ variance)

Limitations

- If groups have different dispersions, p-value are not adequate.
- (Not a problem if differences in dispersion matter as much as differences in location)
- p-values computed using permutations, permutations must respect the design.

Assumptions behind Multivariate ANOVA

Assumptions

- PERMANOVA tests location effect ($\simeq$ mean)
- PERMANOVA assumes equal dispersions ($\simeq$ variance)

Limitations

- If groups have different dispersions, p-value are not adequate.
- (Not a problem if differences in dispersion matter as much as differences in location)
- p-values computed using permutations, permutations must respect the design.

Outline

Why differential analyses?

Exploratory Data Analysis

- Comparisons at the global level: is there structure in the data?
- With PERMANOVA: Does weaning affect community composition?
- Are groups A and B different?

Differential Analysis

- We know that groups A and B are different.
- How do they differ (in terms of taxa)?

Why differential analyses?

Exploratory Data Analysis

- Comparisons at the global level: is there structure in the data?
- With PERMANOVA: Does weaning affect community composition?
- Are groups A and B different?

Differential Analysis

- We know that groups A and B are different.
- How do they differ (in terms of taxa)?

Differential analyses of count data

Differential analyses of count data based on negative binomial generalized linear model are widely popular in transcriptomics.

The model is defined as follows:

$$K_{ij} \sim \mathsf{NB}(\mu_{ij}; \alpha_i)$$
$$\mu_{ij} = s_j q_{ij}$$
$$\log_2(q_{ij}) = x_j \beta_i$$

where

- K_{ij} is the count for otu i in sample j
- μ_{ij} is the otu $\times$ sample mean
- α_i is the otu-specific dispersion
- s_j is the sample-specific size-factor (*e.g.* sequencing depth)
- q_{ij} expected true abundance of otu i in sample j.
- The coefficients β_i give the $\log_2$ fold-changes for each variable in the model matrix X.

Example model matrix

```
##    [,1] [,2]
## A1    1    0
## A2    1    0
## B1    1    1
## B2    1    1
```

- β_{i1}: the base (logarithmic) abundance of otu i. If group A is the reference group, this is the expected log-abundance of the otu in samples from group A (up to the sample-specific scaling factor) s_j.
- β_{i2}: the $\log_2$ fold change between groups A and B.

A few important points

`DESeq2` implementation has differences with standard linear model:

- The sample-specific size-factor s_j controls for sequencing depths, there is no need to rarefy to even depths;
- The effect are additive in the $\log$-scale (*i.e.* multiplicative in the natural scale), unlike linear model where they are additive in the natural scale;
- The dispersions α_i are estimated through partial pooling of the otus and not independently for each otu;
- The estimates of β_i are maximum *a posteriori* estimates using a zero-mean normal prior: the estimates are *moderated* by the use of this prior.

Typical Analysis

A typical `DESeq2` analysis consists in

1. formatting the count data and sample metadata appropriately
2. estimating the size factors s_j with `estimateSizeFactors`
3. estimating the dispersions α_i with `estimateDispersions`
4. fitting the negative binomial models, testing the significance of the β_i with Wald test (`nbinomWaldTest` or Likelihood Ratio Tests (LRT, `nbinomLRT`)
5. extracting significant OTUs for a given comparison using `results`

The estimation steps (2 to 4) are done all at once using the `DESeq` function.

DESeq2 with phyloseq (I)

phyloseq takes care of the formatting, you just need to specify the model:

```
cds <- phyloseq_to_deseq2(food, ~ EnvType)

## Loading required namespace:  DESeq2
## converting counts to integer mode
```

and then fit the model

```
dds <- DESeq2::DESeq(cds, sfType = "poscounts")

## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## -- replacing outliers and refitting for 19 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
## estimating dispersions
```

DESeq2 with phyloseq (III)

Select otus that differ BoeufHache and VeauHache at $p < 0.01$ (after correction for multiple testing)

```
options(digits = 3)
results <- DESeq2::results(dds, contrast = c("EnvType", "BoeufHache", "VeauHache"),
  rename(OTU = row) %>% filter(padj < 0.01)
da.otus <- results
head(da.otus, 2)

##         OTU baseMean log2FoldChange lfcSE  stat   pvalue    padj
## 1 otu_01680     31.4          -4.51 1.175 -3.84 0.000124 0.00333
## 2 otu_01408     22.3          -3.37 0.959 -3.52 0.000436 0.00803

dim(da.otus)

## [1] 20  7
```

DESeq2 with phyloseq (IV)

Enrich results with taxonomic information and add OTU number in a column

```
tax_df <- tax_table(food) %>%
  as("matrix") %>% as.data.frame() %>%
  mutate(OTU = taxa_names(food))
da.otus <- inner_join(da.otus, tax_df, by = c("OTU"))
head(da.otus, n = 2)

##         OTU baseMean log2FoldChange lfcSE  stat   pvalue    padj  Kingdom
## 1 otu_01680     31.4          -4.51 1.175 -3.84 0.000124 0.00333 Bacteria
## 2 otu_01408     22.3          -3.37 0.959 -3.52 0.000436 0.00803 Bacteria
##           Phylum               Class           Order           Family
## 1 Proteobacteria Gammaproteobacteria Pseudomonadales    Moraxellaceae
## 2 Proteobacteria Gammaproteobacteria Xanthomonadales Xanthomonadaceae
##           Genus Species
## 1 Psychrobacter   Fozii
## 2    Fulvimonas    Soli
```

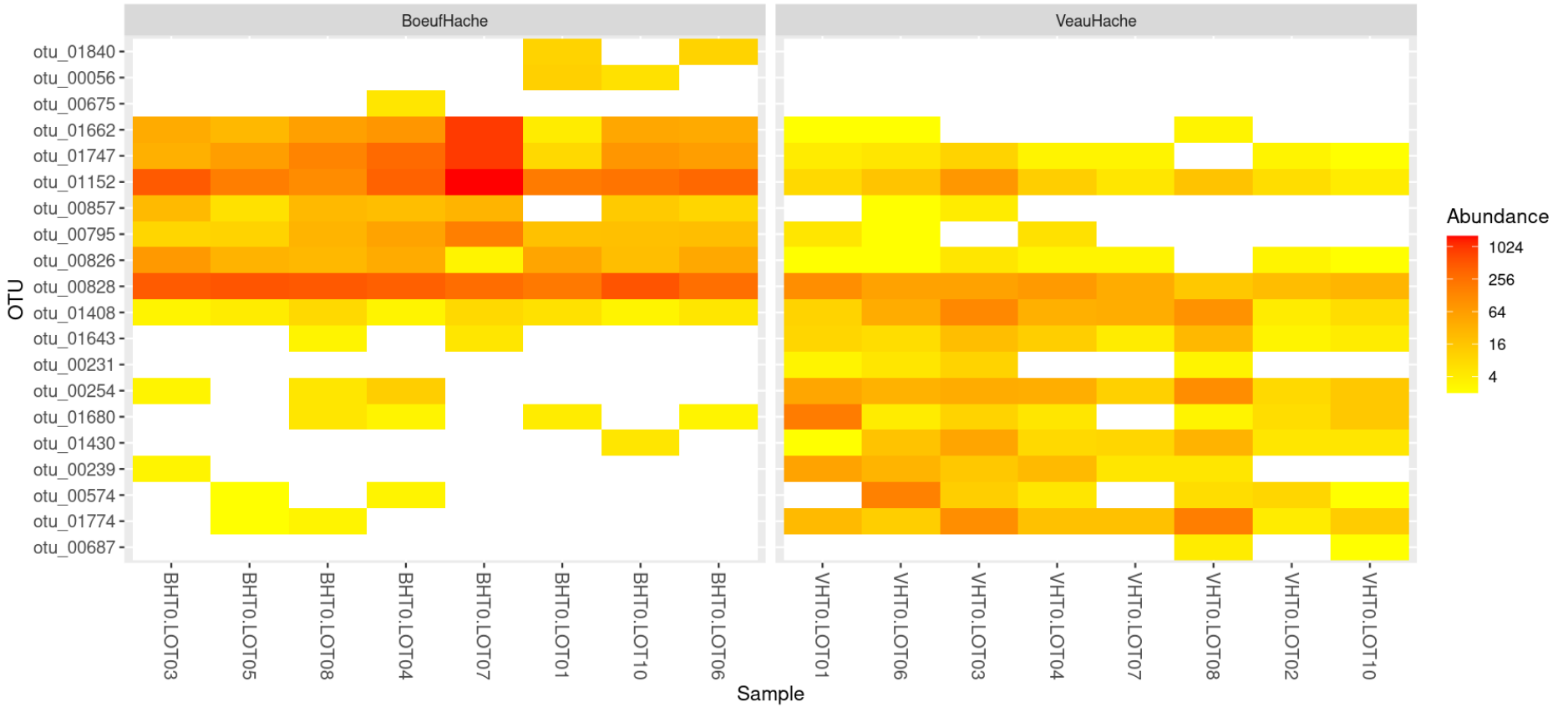
Sort taxa by $\log_2$ fold change

```
da.otus <- arrange(da.otus, log2FoldChange)
head(da.otus, n = 2)

##         OTU baseMean log2FoldChange lfcSE  stat   pvalue     padj  Kingdom
```

DESeq2 with phyloseq (VI)

```
plot_heatmap(prune_taxa(da.otus$OTU, food) %>%
               subset_samples(EnvType %in% c("BoeufHache", "VeauHache")),
             taxa.order = da.otus$OTU,
             low = "yellow", high = "red", na.value = "white") +
  facet_grid(~EnvType, scales = "free_x")
```



Points to keep in mind

- Negative binomial models were developed for transcriptomics data
- Normalization assumes that most transcripts are not DA
- Reasonable for comparison before/after antibiotic intervention
- Not so when comparing Soil against Seawater

Amplicon metagenomics data are typically very sparse (∼66% for `kinetic`)

- Erroneous OTUs
- Group/Environment-specific OTUs.

Not clear how negative binomial models cope with this sparsity

- Transcripts compete for the same limiting resource (ribosomes)
- Translates to ecological equivalence for OTUs

Points to keep in mind

- Negative binomial models were developed for transcriptomics data
- Normalization assumes that most transcripts are not DA
- Reasonable for comparison before/after antibiotic intervention
- Not so when comparing Soil against Seawater

Amplicon metagenomics data are typically very sparse (∼66% for `kinetic`)

- Erroneous OTUs
- Group/Environment-specific OTUs.

Not clear how negative binomial models cope with this sparsity

- Transcripts compete for the same limiting resource (ribosomes)
- Translates to ecological equivalence for OTUs

Points to keep in mind

- Negative binomial models were developed for transcriptomics data
- Normalization assumes that most transcripts are not DA
- Reasonable for comparison before/after antibiotic intervention
- Not so when comparing Soil against Seawater

Amplicon metagenomics data are typically very sparse (∼66% for `kinetic`)

- Erroneous OTUs
- Group/Environment-specific OTUs.

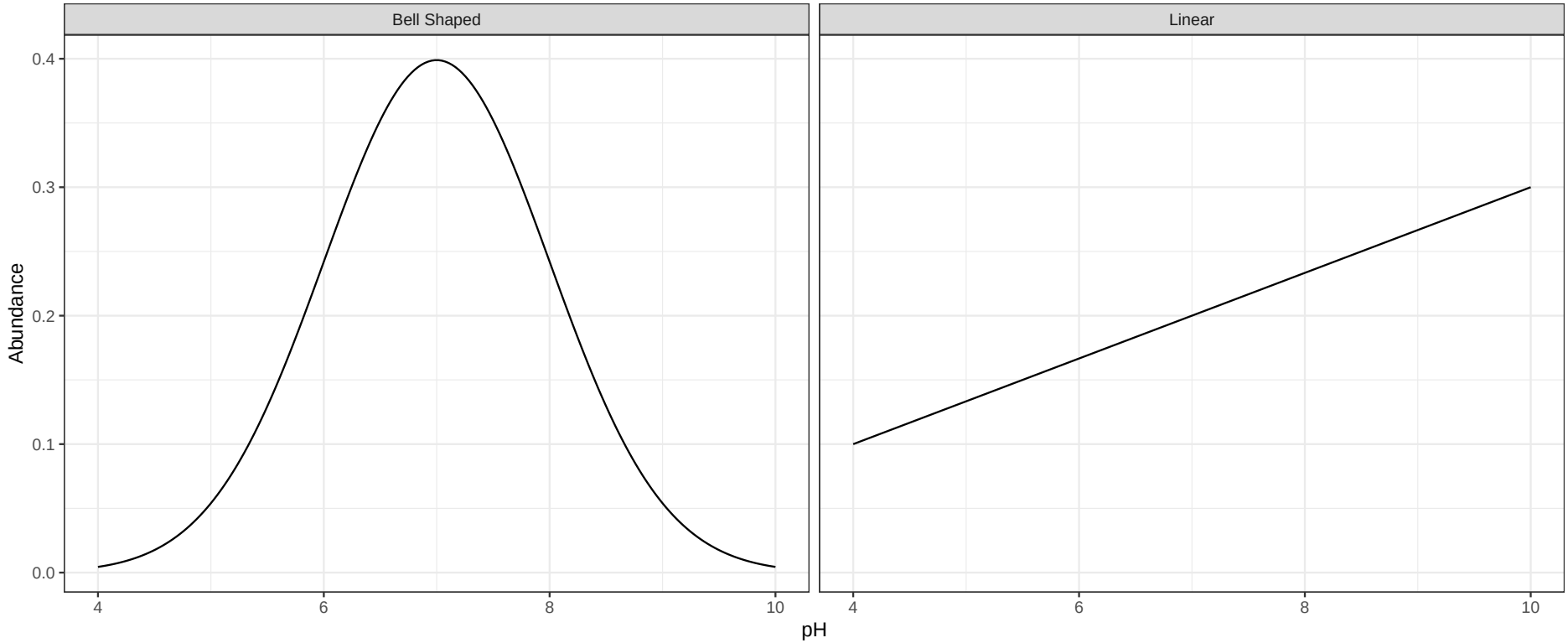
Not clear how negative binomial models cope with this sparsity

- Transcripts compete for the same limiting resource (ribosomes)
- Translates to ecological equivalence for OTUs

Outline

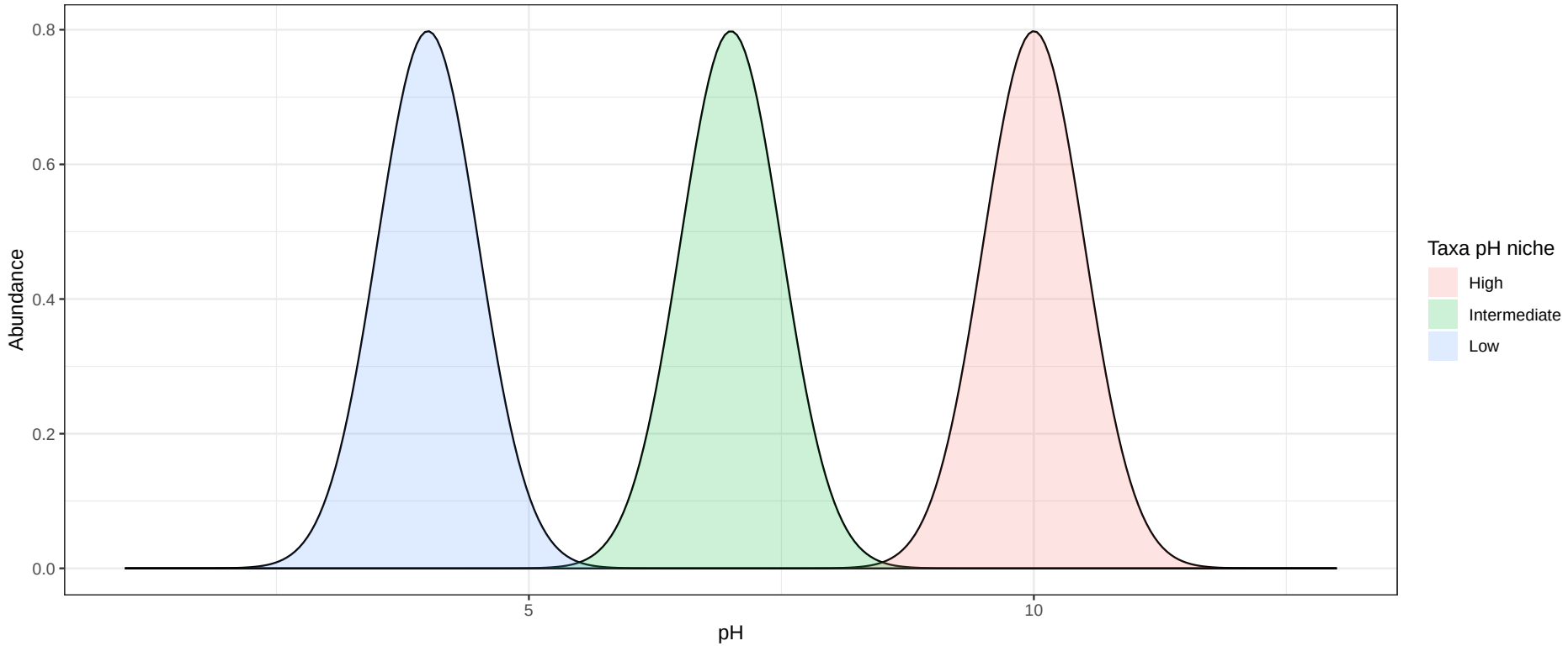
A few words about linear responses

PERMANOVA (resp. `DESEq2`) is based on the idea of linear (resp. multiplicative) responses but ecological responses are usually bell-shaped (*e.g.* optimal pH range for a taxa)



A word about linear responses (II)

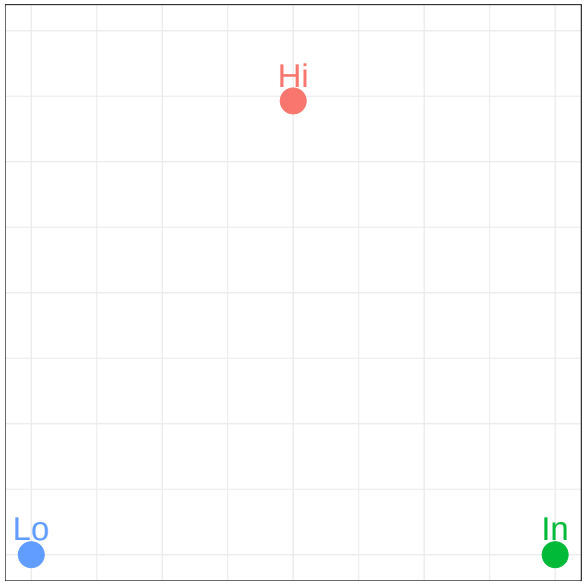
In particular, if you get too far away along a linear gradient (*e.g.* pH), communities don't share any species



A word about linear responses (III)

And communities "High", "Intermediate" and "Low" are all at distance 1 of each other. 2D-plots are perfect!

	Lo	In	Hi
Lo	0.00	1.00	1.00
In	1.00	0.00	1.00
Hi	1.00	1.00	0.00

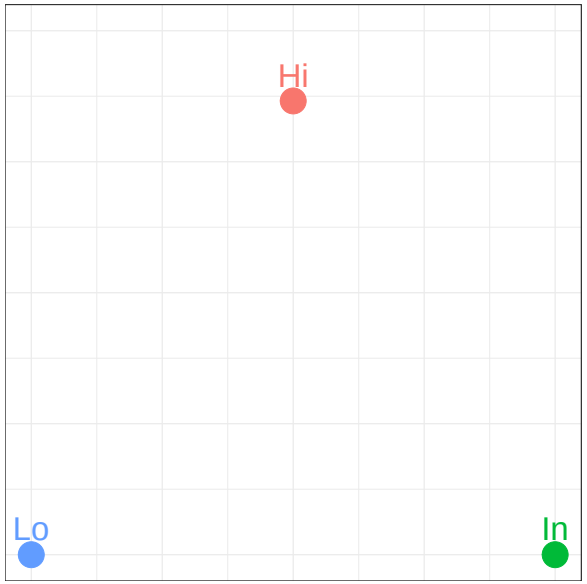


But troubles start when you add more communities...

A word about linear responses (III)

And communities "High", "Intermediate" and "Low" are all at distance 1 of each other. 2D-plots are perfect!

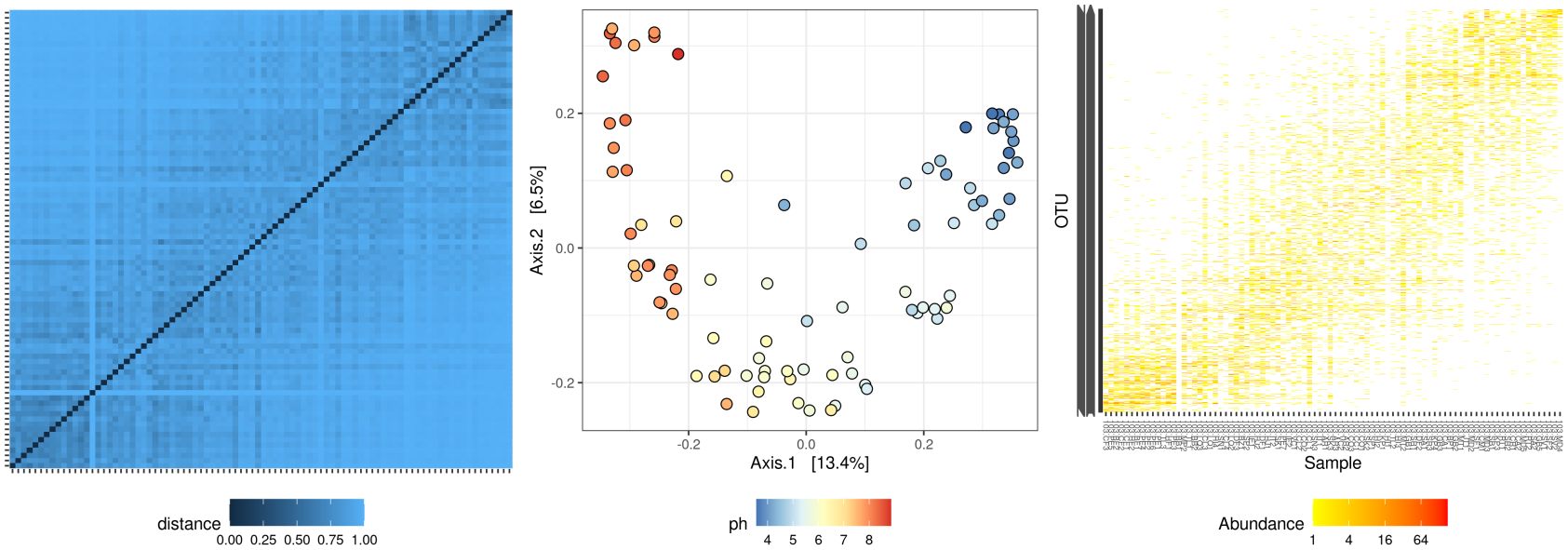
	Lo	In	Hi
Lo	0.00	1.00	1.00
In	1.00	0.00	1.00
Hi	1.00	1.00	0.00



But troubles start when you add more communities...

88 soils from Morton et al. (2017) ordered by pH

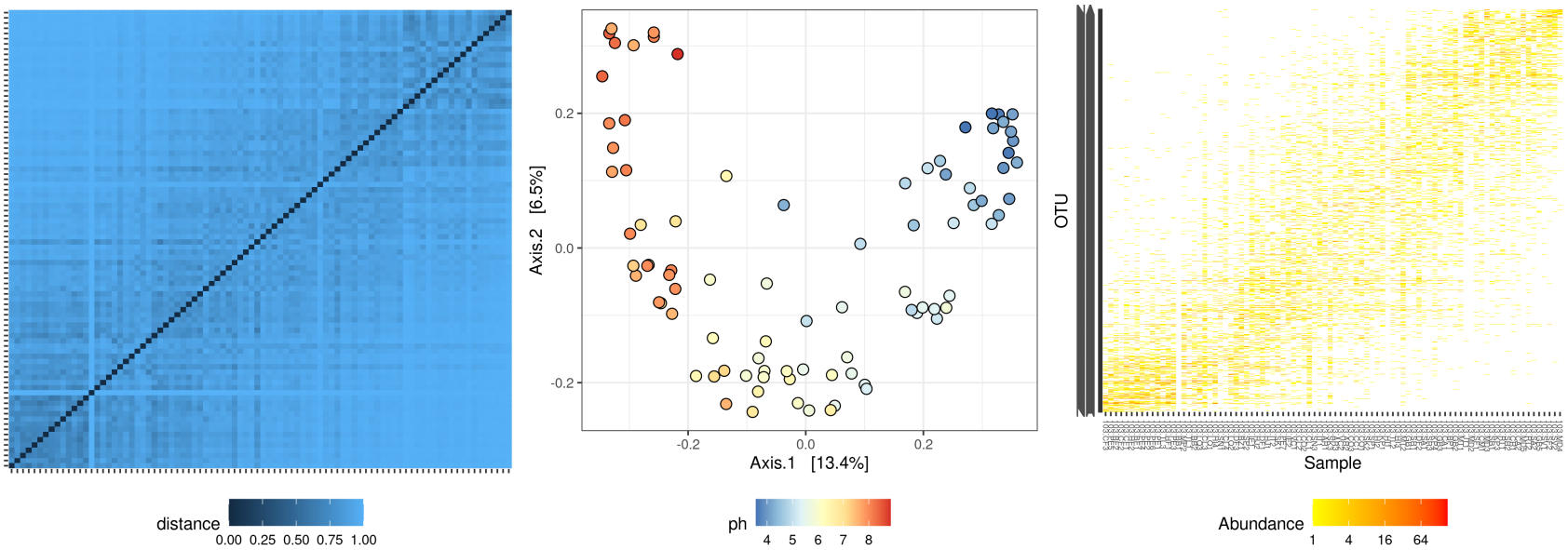
Distances saturate → 2D plot doesn't capture *linear gradient* shown in heatmap.



- Taxonomic distances are (i) bounded/saturated and (ii) may not capture large functional differences.
- Taxa do not respond linearly nor multiplicatively

88 soils from Morton et al. (2017) ordered by pH

Distances saturate → 2D plot doesn't capture *linear gradient* shown in heatmap.



- Taxonomic distances are (i) bounded/saturated and (ii) may not capture large functional differences.
- Taxa do not respond linearly nor multiplicatively

Conclusion

- Import your data into phyloseq using `import_qiime` or `import_biom`
- Filter OTUs, select part of the data with `prune_taxa`, `subset_taxa` and their counterpart for samples.
- Rarefy counts (when needed) using `rarefy_even_depth`
- Compute α-diversities using `estimate_richness`
- Compute β-diversities using `distance`
- Vizualise samples using `plot_ordination`
- Overlay environmental variables using `envfit`
- Vizualise count table using `plot_heatmap` (useful to emphasize block structure)
- Test effect of covariates using PERMANOVA with `adonis`
- Find differentially abundant taxa with `DESEq2`

Caporaso, J. G., Lauber, C. L., Walters, W. A., Berg-Lyons, D., Lozupone, C. A., Turnbaugh, P. J., Fierer, N., and Knight, R. (2011). Global patterns of 16s rrna diversity at a depth of millions of sequences per sample. *Proc Natl Acad Sci U S A*, 108 Suppl 1:4516–4522.

Chaillou, S., Chaulot-Talmon, A., Caekebeke, H., Cardinal, M., Christieans, S., Denis, C., Desmonts, M. H., Dousset, X., Feurer, C., Hamon, E., Joffraud, J.-J., La Carbona, S., Leroi, F., Leroy, S., Lorre, S., Macé, S., Pilet, M.-F., Prévost, H., Rivollier, M., Roux, D., Talon, R., Zagorec, M., and Champomier-Vergès, 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.

Mach, N., Berri, M., Estellé, J., Levenez, F., Lemonnier, G., Denis, C., Leplat, J.-J., Chevaleyre, C., Billon, Y., Doré, J., and et al. (2015). Early-life establishment of the swine gut microbiome and impact on host phenotypes. *Environmental Microbiology Reports*, 7(3):554–569.

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.

Morton, J. T., Toran, L., Edlund, A., Metcalf, J. L., Lauber, C., and Knight, R. (2017). Uncovering the horseshoe effect in microbial analyses. *mSystems*, 2(1).

Ravel, J., Gajer, P., Abdo, Z., Schneider, G. M., Koenig, S. S. K., McCulle, S. L., Karlebach, S., Gorle, R., Russell, J., Tacket, C. O., and et al. (2011). Vaginal microbiome of reproductive-age women. *Proceedings of the National Academy of Sciences*, 108(Suppl. 1):4680–4687.

Homeworks: Global Patterns

Dataset from Caporaso et al. (2011) used to study microbial diversity in very diverse environments with ultra-deep sequencing.

- Rarefy the data as they are highly uneven.
- Compare α-diversities across environments (`SampleType`). Which environments are more/less diverse? Is it consistent with your intuition?
- Using β-diversities, what could you say about the different environments?

Homeworks: Mach

Dataset from Mach et al. (2015) used to study gut microbiota of 31 early life swine, in particular the impact of Weaning and Time. Interesting covariates include `Time` (sample time, with 5 values D14, D36, D48, D60, D70), `Weaned` (weaned (D14) or not (all other times)), `sex` (1 for male, 2 for female), `mere` (swine's mother) `Bande` (*feeding place*).

- Look at the composition of the communities, zoom in on the dominant phyla to find classes / order / genera that separate weaned and unweaned samples.
- Have a look at the rarefaction curves. Should you rarefy the samples? Why?
- Between which consecutives time points do you observe differences in terms of microbiota ?

Homeworks: Bacterial Vaginosis

Dataset from Ravel et al. (2011) used to study the vaginal microbiome of reproductive-age women. They looked at Ethnic Group (`Ethnic_Group`), pH (`pH`), Nugent score and category (`Nugent_Score` and `Nugent_Cat`, a score used to predict bacterial vaginosis - BV, with higher scores corresponding to higher likelihood of disease - and a discrete traduction as low, intermediate and high values) and created 5 groups (`CST`).

- Is there a correlation between pH, Nugent score, group, Ethnic group and the α-diversity?
- Do these covariates have an impact on community composition?
- How do groups compare in terms of community composition?
- Try to find how the group were made. What's special about group IV (hint: look at the count data)
- If you knew the group (CST) of a patient, how could you guess its status (BV or not)?