Analysis of community composition data using `phyloseq`

Mahendra Mariadassou - INRAE, France

January 2020
GDC, Zurich

Outline

Goals

phyloseq

Become familiar with `phyloseq` `R` package 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

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

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 = "dev")
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 = "dev")
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

```r
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.

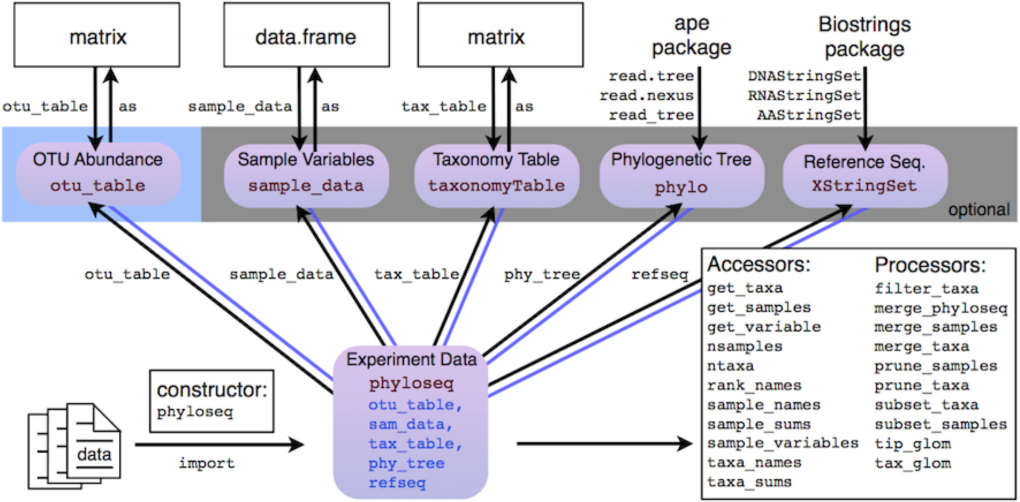
Data structure

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 mach) 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 `mach`) and guess what they do.

Dimensions

```r
ntaxa(mach)

## [1] 7857

nsamples(mach)

## [1] 543
```

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

Dimensions

```r
ntaxa(mach)

## [1] 7857

nsamples(mach)

## [1] 543
```

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

sample_names, taxa_names

```r
head(sample_names(mach))
```

```
## [1] "sample_SF.0092" "sample_SF.0104" "sample_SF.0109" "sample_SF.0131"
## [5] "sample_SF.0132" "sample_SF.0133"
```

```r
head(taxa_names(mach))
```

```
## [1] "otu_692"  "otu_1686" "otu_2192" "otu_3292" "otu_4395" "otu_2267"
```

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

sample_names, taxa_names

```
head(sample_names(mach))

## [1] "sample_SF.0092" "sample_SF.0104" "sample_SF.0109" "sample_SF.0131"
## [5] "sample_SF.0132" "sample_SF.0133"

head(taxa_names(mach))

## [1] "otu_692"  "otu_1686" "otu_2192" "otu_3292" "otu_4395" "otu_2267"
```

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

sample_names, taxa_names

```
head(sample_names(mach))

## [1] "sample_SF.0092" "sample_SF.0104" "sample_SF.0109" "sample_SF.0131"
## [5] "sample_SF.0132" "sample_SF.0133"

head(taxa_names(mach))

## [1] "otu_692"  "otu_1686" "otu_2192" "otu_3292" "otu_4395" "otu_2267"
```

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

sample_sums, taxa_sums

```
head(sample_sums(mach))

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131 sample_SF.0132
##            924            951            986           1104           1231
## sample_SF.0133
##           1224

head(taxa_sums(mach))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##       27        6        2        3        3        5
```

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(mach))

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131 sample_SF.0132
##            924            951            986           1104           1231
## sample_SF.0133
##           1224

head(taxa_sums(mach))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##       27        6        2        3        3        5
```

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(mach))

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131 sample_SF.0132
##            924            951            986           1104           1231
## sample_SF.0133
##           1224

head(taxa_sums(mach))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##       27        6        2        3        3        5
```

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(mach)

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

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

rank_names

```
rank_names(mach)

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

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

sample_variables

```
head(sample_variables(mach))
```

```
## [1] "SampleID" "Run"      "Project"  "Time"     "Bande"    "sex"
```

Names of the contextual data recorded on the samples.

sample_variables

```r
head(sample_variables(mach))
```

```
## [1] "SampleID" "Run"       "Project"  "Time"      "Bande"     "sex"
```

Names of the contextual data recorded on the samples.

Quick practice

Find the

- library size of samples sample_SF.0140, sample_SF.0142, sample_SF.0144
- overall abudance of otus otu_692, otu_4395, otu_2584

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

```
## sample library sizes
sample_sums(mach)[c("sample_SF.0140", "sample_SF.0142", "sample_SF.0144")]

## sample_SF.0140 sample_SF.0142 sample_SF.0144
##           1367           1246           1029

## Otu overall abundances
taxa_sums(mach)[c("otu_16089", "otu_15374", "otu_12332")]

## otu_16089 otu_15374 otu_12332
##        16        32         5
```

Quick practice

Find the

- library size of samples sample_SF.0140, sample_SF.0142, sample_SF.0144
- overall abudance of otus otu_692, otu_4395, otu_2584

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

```
## sample library sizes
sample_sums(mach)[c("sample_SF.0140", "sample_SF.0142", "sample_SF.0144")]

## sample_SF.0140 sample_SF.0142 sample_SF.0144
##           1367           1246           1029

## Otu overall abundances
taxa_sums(mach)[c("otu_16089", "otu_15374", "otu_12332")]

## otu_16089 otu_15374 otu_12332
##        16        32         5
```

get_variable, get_sample, get_taxa

```
head(get_variable(mach, varName = "Time"))

## [1] D60 D60 D60 D60 D60 D60
## Levels: D14 D36 D48 D60 D70 Sow

head(get_sample(mach, i = "otu_12332"), n = 4)

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131
##              0              0              0              0

head(get_taxa(mach, i = "sample_SF.0131"))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##        0        0        0        0        0        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(mach, varName = "Time"))

## [1] D60 D60 D60 D60 D60 D60
## Levels: D14 D36 D48 D60 D70 Sow

head(get_sample(mach, i = "otu_12332"), n = 4)

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131
##              0              0              0              0

head(get_taxa(mach, i = "sample_SF.0131"))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##        0        0        0        0        0        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

```r
head(get_variable(mach, varName = "Time"))

## [1] D60 D60 D60 D60 D60 D60
## Levels: D14 D36 D48 D60 D70 Sow

head(get_sample(mach, i = "otu_12332"), n = 4)

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131
##              0              0              0              0

head(get_taxa(mach, i = "sample_SF.0131"))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##        0        0        0        0        0        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(mach, varName = "Time"))

## [1] D60 D60 D60 D60 D60 D60
## Levels: D14 D36 D48 D60 D70 Sow

head(get_sample(mach, i = "otu_12332"), n = 4)

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131
##              0              0              0              0

head(get_taxa(mach, i = "sample_SF.0131"))

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##        0        0        0        0        0        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. For example, to add a sample variable Weaned to our metadata, we must use sample_data:

```
sample_data(mach)$Weaned <- ifelse(sample_data(mach)$Time == "D14", FALSE, TRUE)
sample_variables(mach) ## Weaned successfully added

## [1] "SampleID" "Run"      "Project"  "Time"     "Bande"    "sex"      "mere"
## [8] "Weaned"
```

You can also change Bande to a factor

```
sample_data(mach)$Bande <- as.factor(sample_data(mach)$Bande)
```

Modifying some values (II)

Another use is to transform EnvType to a factor with meaningful level ordering (meat products first and seafood second):

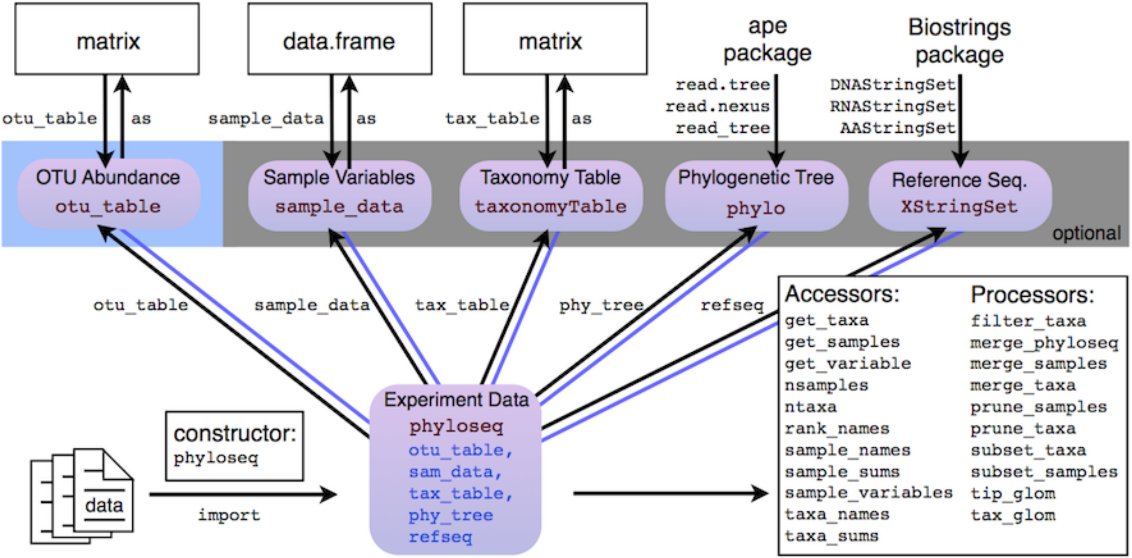
```
correct.order <- c("BoeufHache", "VeauHache", "DesLardons",
                   "MerguezVolaille", "SaumonFume", "FiletSaumon",
                   "FiletCabillaud", "Crevette")
correct.labels <- c("Ground beef", "Ground veal", "Bacon", "Poultry sausage",
                    "Smoked salmon", "Salmon fillet", "Cod fillet", "Shrimp")
sample_data(food)$EnvType <- factor(sample_data(food)$EnvType,
                                    levels = correct.order,
                                    labels = correct.labels)
levels(get_variable(food, "EnvType"))

## [1] "Ground beef"     "Ground veal"     "Bacon"           "Poultry sausage"
## [5] "Smoked salmon"   "Salmon fillet"   "Cod fillet"      "Shrimp"
```

Finally, you can also correct taxonomic affiliation and otu count in a given sample as follows:

```
otu_table(mach)["otu_16089", "SFM.46"] <- 1
tax_table(mach)["otu_16089", "Order"] <- "Clostridiales"
```

If you start from BIOM files, corrections are more easily made in R than by hand in the source

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

```
prune_samples(sample_names(mach)[1:10], mach)

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7857 taxa and 10 samples ]
## sample_data() Sample Data:       [ 10 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 7857 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 7857 tips and 7856 internal nodes ]
```

Subset

```
subset_samples(mach, Time %in% c("D60", "D70"))

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7857 taxa and 361 samples ]
## sample_data() Sample Data:       [ 361 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 7857 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 7857 tips and 7856 internal nodes ]
```

Prune and subset

Prune

```
prune_samples(sample_names(mach)[1:10], mach)

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7857 taxa and 10 samples ]
## sample_data() Sample Data:       [ 10 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 7857 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 7857 tips and 7856 internal nodes ]
```

Subset

```
subset_samples(mach, Time %in% c("D60", "D70"))

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7857 taxa and 361 samples ]
## sample_data() Sample Data:       [ 361 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 7857 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 7857 tips and 7856 internal nodes ]
```

A bit more about subset (II)

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

```
small.mach <- subset_taxa(mach, Phylum == "Firmicutes" & Class == "Bacilli")
head(tax_table(small.mach)[ , c("Phylum", "Class", "Order")], n = 4)

## Taxonomy Table:     [4 taxa by 3 taxonomic ranks]:
##            Phylum       Class     Order
## otu_1571  "Firmicutes" "Bacilli" "Lactobacillales"
## otu_16950 "Firmicutes" "Bacilli" "Lactobacillales"
## otu_6692  "Firmicutes" "Bacilli" "Lactobacillales"
## otu_1547  "Firmicutes" "Bacilli" "Lactobacillales"

## Unique combinations (Phylum, Class, Order)
unique(tax_table(small.mach)[ , c("Phylum", "Class", "Order")])

## Taxonomy Table:     [4 taxa by 3 taxonomic ranks]:
##            Phylum       Class     Order
## otu_1571  "Firmicutes" "Bacilli" "Lactobacillales"
## otu_11894 "Firmicutes" "Bacilli" "Turicibacterales"
## otu_1064  "Firmicutes" "Bacilli" "Gemellales"
## otu_1150  "Firmicutes" "Bacilli" "Bacillales"
```

Advanced filters

You can also filter out OTUs satisfying a condition (*e.g.* abundance higher than 2) in a minimum (*e.g.* 10) number of samples by combining `genefilter_sample` and `prune_taxa`.

```
## Keep only taxa with abundance at least 2 in at least 10 samples
test_function <- function(x) { x >= 2 }
taxa.to.keep <- genefilter_sample(mach, test_function, A = 10)
head(taxa.to.keep) ## logical mask of taxa passing the filter

##  otu_692 otu_1686 otu_2192 otu_3292 otu_4395 otu_2267
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE

prune_taxa(taxa.to.keep, mach) ## 1197 taxa pass the filter

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1197 taxa and 543 samples ]
## sample_data() Sample Data:       [ 543 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 1197 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1197 tips and 1196 internal nodes ]
```

Practicing

Create a `kinetic` object (resp. a `kinetic.rare`) with

- only the samples belonging to Project "Kinetic"
- only OTUs with overall abundance higher than 0.005 % of the total smpling depth (use `taxa_sums`)

```
kinetic <- subset_samples(mach, Project %in% c("Kinetic"))
total.depth <- sum(otu_table(kinetic))
threshold <- 5e-5 * total.depth
kinetic.rare <- prune_taxa(taxa_sums(kinetic) > threshold, kinetic)
kinetic.rare
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1029 taxa and 155 samples ]
## sample_data() Sample Data:       [ 155 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 1029 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1029 tips and 1028 internal nodes ]
```

Practicing

Create a kinetic object (resp. a kinetic.rare) with

- only the samples belonging to Project "Kinetic"
- only OTUs with overall abundance higher than 0.005 % of the total smpling depth (use taxa_sums)

```
kinetic <- subset_samples(mach, Project %in% c("Kinetic"))
total.depth <- sum(otu_table(kinetic))
threshold <- 5e-5 * total.depth
kinetic.rare <- prune_taxa(taxa_sums(kinetic) > threshold, kinetic)
kinetic.rare

## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1029 taxa and 155 samples ]
## sample_data() Sample Data:       [ 155 samples by 8 sample variables ]
## tax_table()   Taxonomy Table:    [ 1029 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1029 tips and 1028 internal nodes ]
```

Outline

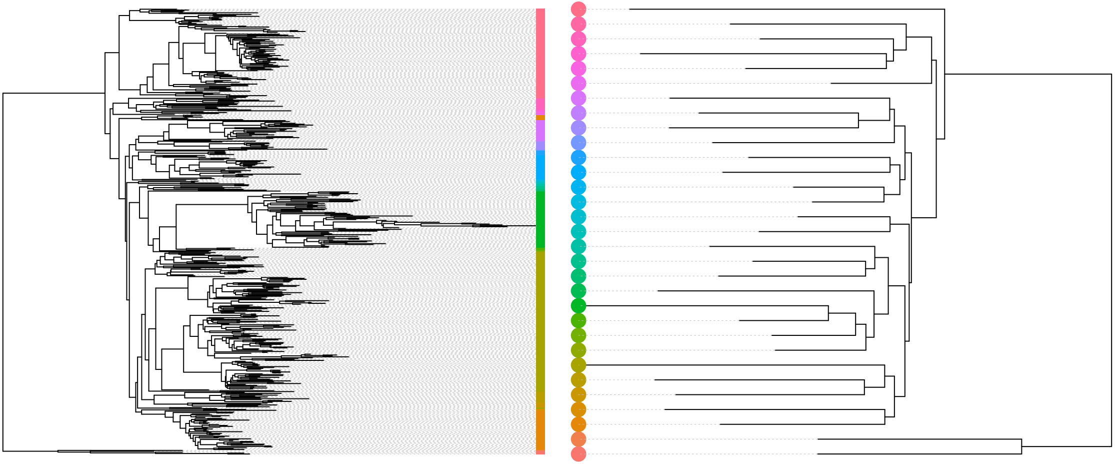
Smoothing with tax_glom (I)

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

```
coarse.mach <- tax_glom(mach, "Phylum")
ntaxa(coarse.mach) ## number of different phyla

## [1] 17

tax_table(coarse.mach)[1:2, c("Phylum", "Order", "Class")]

## Taxonomy Table:     [2 taxa by 3 taxonomic ranks]:
##           Phylum          Order Class
## otu_25  "Bacteroidetes" NA    NA
## otu_525 "Fibrobacteres" NA    NA
```

Smoothing with tax_glom (II)

Effect best understood on the phylogenetic tree (otus colored by phylum).



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.

```
mach.rare <- rarefy_even_depth(mach, rngseed = 20200123)

## `set.seed(20200123)` was used to initialize repeatable random subsampling.
## Please record this for your records so others can reproduce.
## Try `set.seed(20200123); .Random.seed` for the full vector
## ...
## 994OTUs were removed because they are no longer
## present in any sample after random subsampling
## ...

sample_sums(mach.rare)[1:5]

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131 sample_SF.0132
##            924            924            924            924            924
```

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.

```
mach.trans <- transform_sample_counts(mach, count_to_prop)
sample_sums(mach.trans)[1:5] ## should be 1

## sample_SF.0092 sample_SF.0104 sample_SF.0109 sample_SF.0131 sample_SF.0132
##              1              1              1              1              1
```

Practicing

Create a `kinetic.rare` by selecting only samples from the Kinetic project and rarefy them.

```
kinetic <- subset_samples(mach, Project %in% c("Kinetic"))
kinetic.rare <- rarefy_even_depth(kinetic, rngseed = 20200120)

## `set.seed(20200120)` was used to initialize repeatable random subsampling.
## Please record this for your records so others can reproduce.
## Try `set.seed(20200120); .Random.seed` for the full vector
## ...
## 4841OTUs were removed because they are no longer
## present in any sample after random subsampling
## ...
```

Practicing

Create a `kinetic.rare` by selecting only samples from the Kinetic project and rarefy them.

```
kinetic <- subset_samples(mach, Project %in% c("Kinetic"))
kinetic.rare <- rarefy_even_depth(kinetic, rngseed = 20200120)

## `set.seed(20200120)` was used to initialize repeatable random subsampling.
## Please record this for your records so others can reproduce.
## Try `set.seed(20200120); .Random.seed` for the full vector
## ...
## 4841OTUs were removed because they are no longer
## present in any sample after random subsampling
## ...
```

Saving and loading

You can do preprocessing once only by saving your filtered/smoothed/rarefied object into an `.RData` file using `save` and loading it into your session using `load`.

```
save(kinetic, kinetic.rare, file = "kinetic.RData")
load("kinetic.RData")
```

Saving and loading

You can do preprocessing once only by saving your filtered/smoothed/rarefied object into an `.RData` file using `save` and loading it into your session using `load`.

```
save(kinetic, kinetic.rare, file = "kinetic.RData")
load("kinetic.RData")
```

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 .

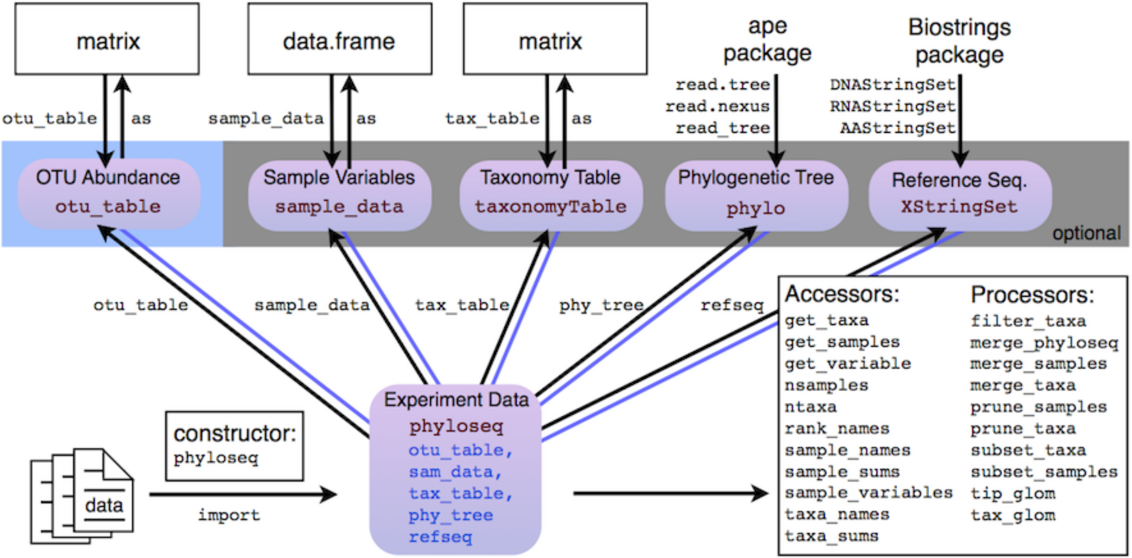
- Functions to import data from biom files, qiime output files or plain tabular files.
- Accessors to access different component of your dataset
- Samples and taxa names are coherent between the different components.
- Filters to keep only part of the dataset.
- Smoothers to aggregate parts of the dataset.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

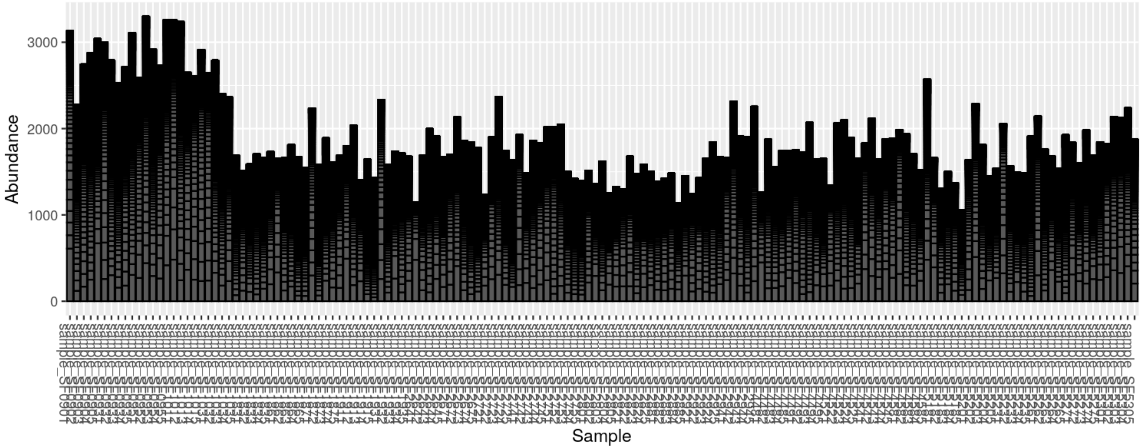
phyloseq recap (II)



Outline

Looking at your samples (plot_bar)

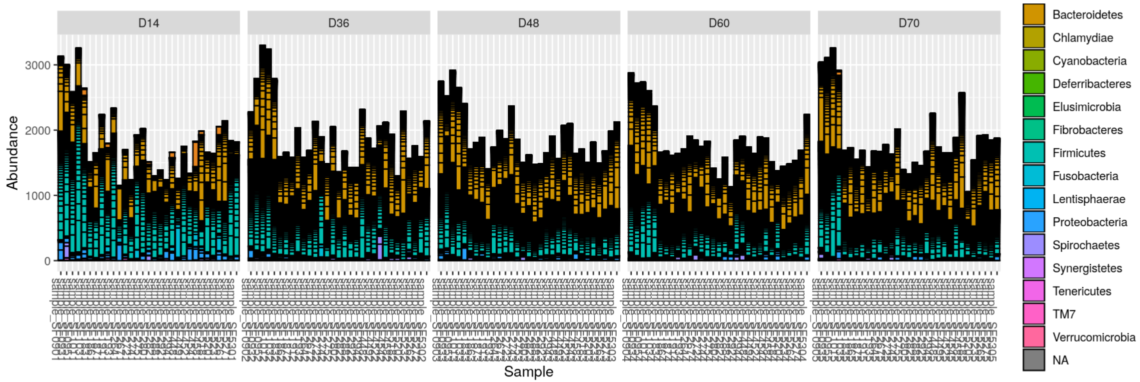
```
p <- plot_bar(kinetic)
plot(p) ## Base graphic, ugly
```



Looking at your samples (`plot_bar`)

Organize samples by sampling time and color otu by Phylum

```
p <- plot_bar(kinetic, fill = "Phylum") ## aes, fill bar according to phylum
p <- p + facet_wrap(~Time, 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)

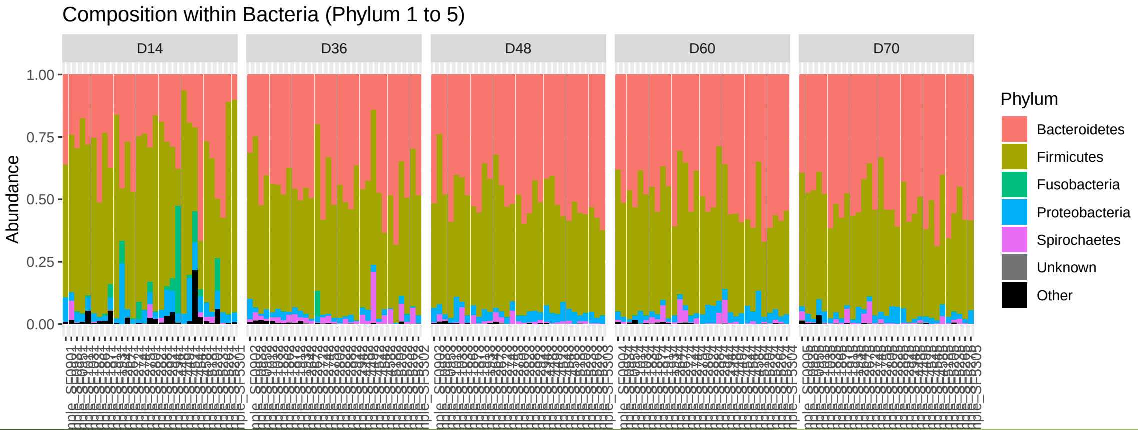
Select Bacteria (at Kingdom level) and aggregate by Phylum.

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



Looking at your samples (plot_composition) (II)

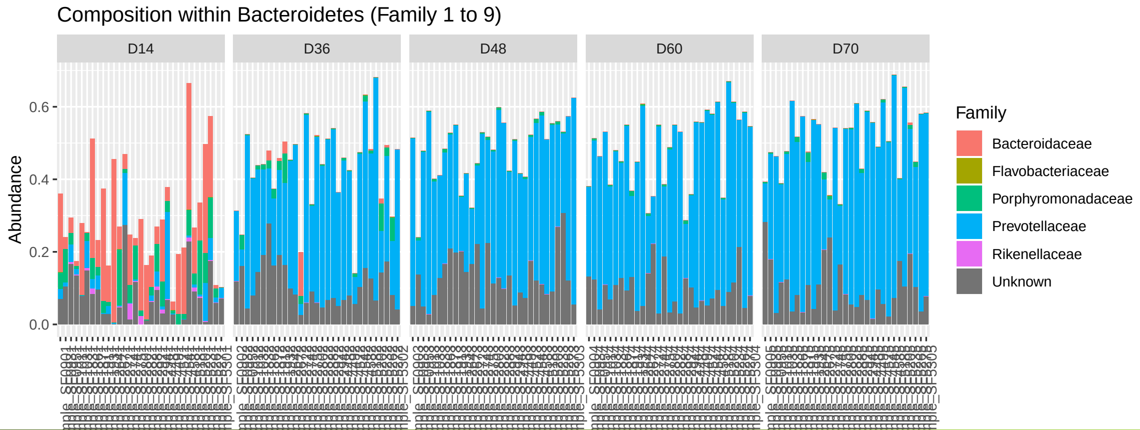
Select Bacteroidetes (at Phylum level) and aggregate by Family.

```
p <- plot_composition(kinetic, "Phylum", "Bacteroidetes", "Family", numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```

Looking at your samples (plot_composition) (II)

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

```
p <- plot_composition(kinetic, "Phylum", "Bacteroidetes", "Family", numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```



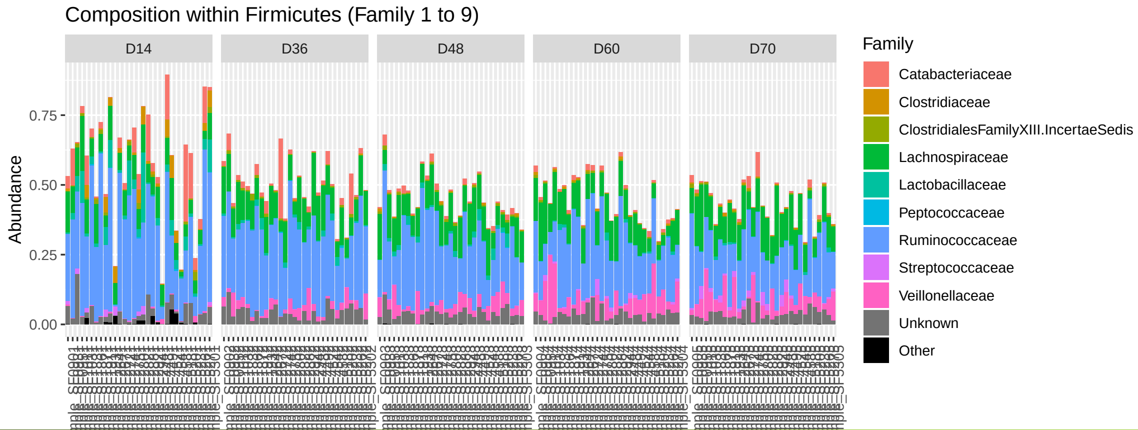
Looking at your samples (plot_composition) (III)

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

```
p <- plot_composition(kinetic, "Phylum", "Firmicutes", "Family", numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```

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

```
p <- plot_composition(kinetic, "Phylum", "Firmicutes", "Family", numberOfTaxa = 9, fill = "Family")
p <- p + facet_wrap(~Time, 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 meaning 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 meaning 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 meaning 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 meaning for bacterial communities);

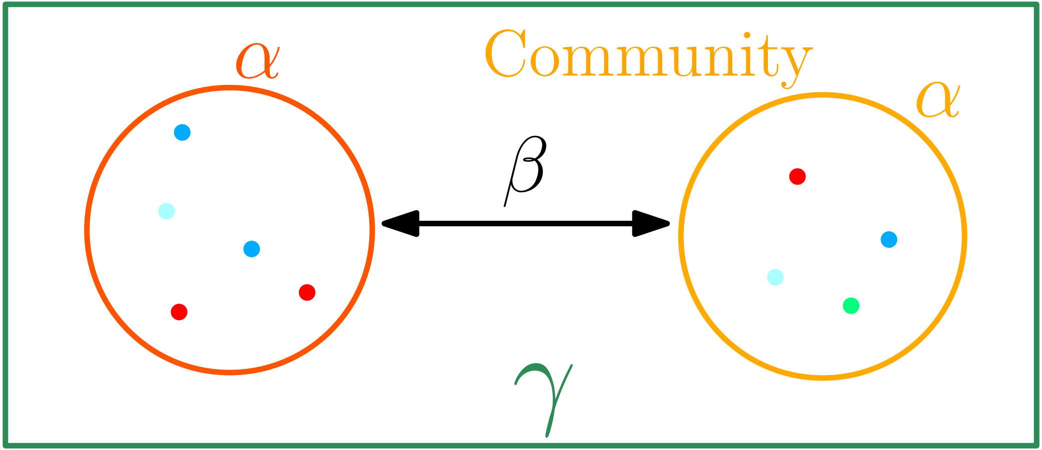
Diversity decomposition

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

β-dissimilarities/distances

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

A schematic view of diversity



Landscape

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.

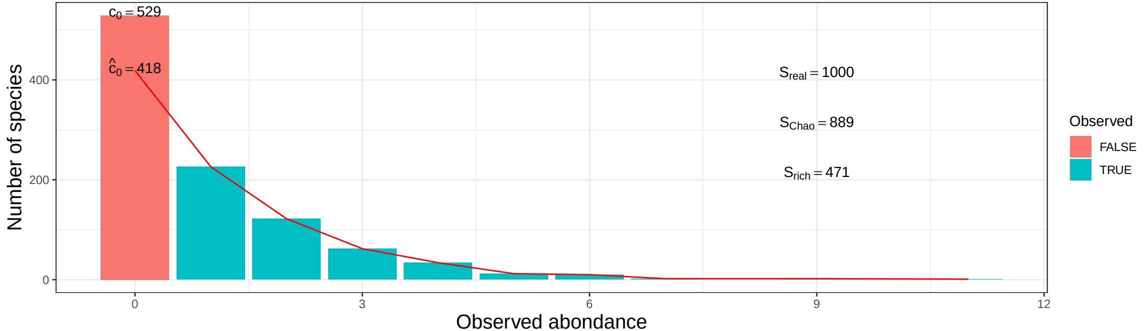
Compositional vs. Phylogenetic

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

Outline

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
$S_{\text{Shan}} = -\sum_s p_s \log(p_s) \leq \log(S)$	$S_{\text{Inv-Simp}} = \dfrac{1}{p_1^2 + \cdots + p_S^2} \leq S$

```
## Error in ddply(df, .(even), summarize, prop = abondance/sum(abondance)):  impossible de
trouver la fonction "ddply"
## Error in ddply(annotation.df, .(even), summarize, richness = sum(prop > :  impossible de
trouver la fonction "ddply"
## Error:  id variables not found in data:  even
## Error in `$<-.data.frame`(`*tmp*`, height, value = structure(numeric(0), .Names =
character(0))):  replacement has 0 rows, data has 2
## Error in paste0("S[", variable, "] ==", ifelse(variable == "shan", paste0(" log(", :
objet 'variable' introuvable
## Error in FUN(X[[i]], ...):  objet 'height' introuvable
```

α diversities

Available in `phyloseq`

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

Inverse-Simpson and exponential of Shannon can be understood as effective number of species

α diversities

Available in `phyloseq`

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

Inverse-Simpson and exponential of Shannon can be understood as effective number of species

Available in `phyloseq`

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

Inverse-Simpson and exponential of Shannon can be understood as effective number of species

α diversities

Available in `phyloseq`

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

Inverse-Simpson and exponential of Shannon can be understood as effective number of species

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 = "Time"` passed on to the `aes` mapping of a `ggplot`.

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

α diversity and filtering (I)

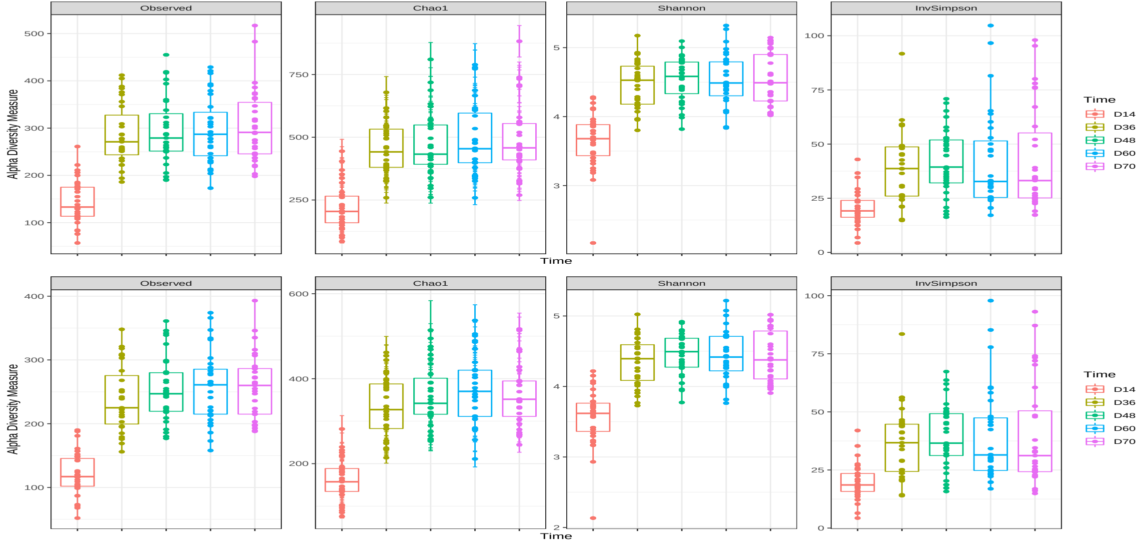
Many α diversities (richness, Chao) depend a lot on rare otus. Do not **trim** rare otus before computing them as it can drastically alter the result (see next slide).

Richness

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

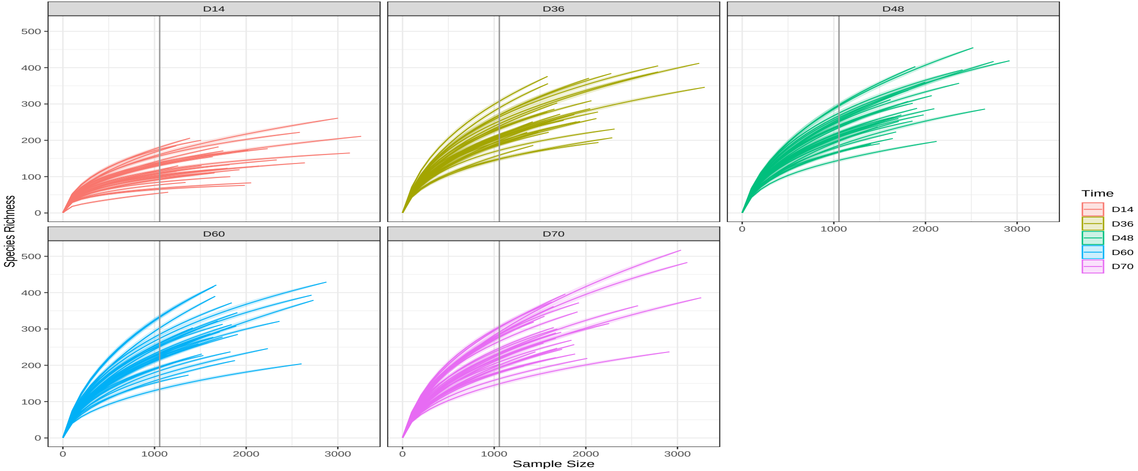
```
p <- plot_richness(kinetic, color = "Time", x = "Time",
              measures = c("Observed", "Chao1", "Shannon", "InvSimpson"))
p <- p + geom_boxplot()
plot(p)
```

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



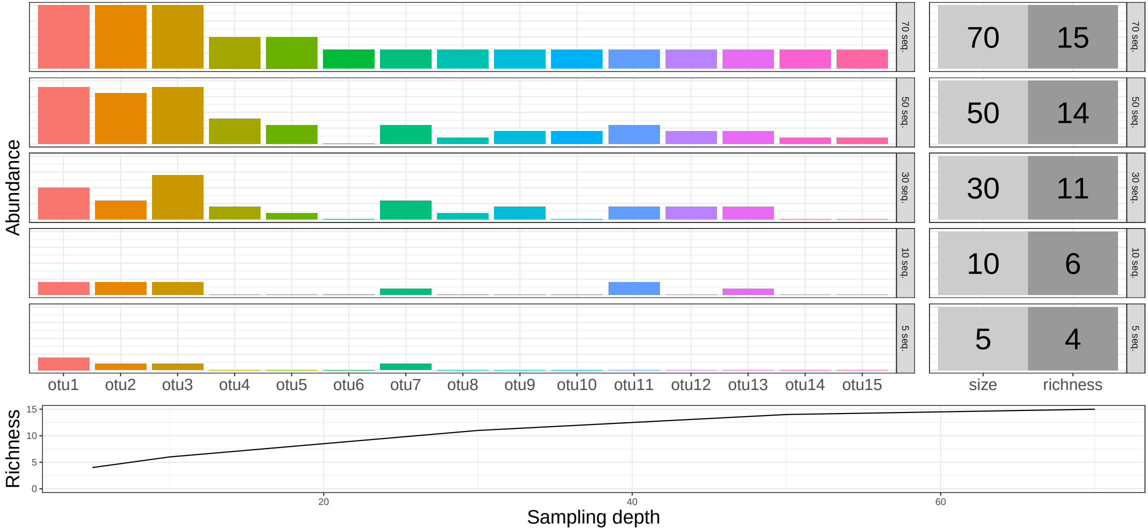
α diversity and sampling effort

Many α diversities (richness, Chao) depend a lot on rare otus and sampling efforts: use rarefaction/correct for depth before comparing them.



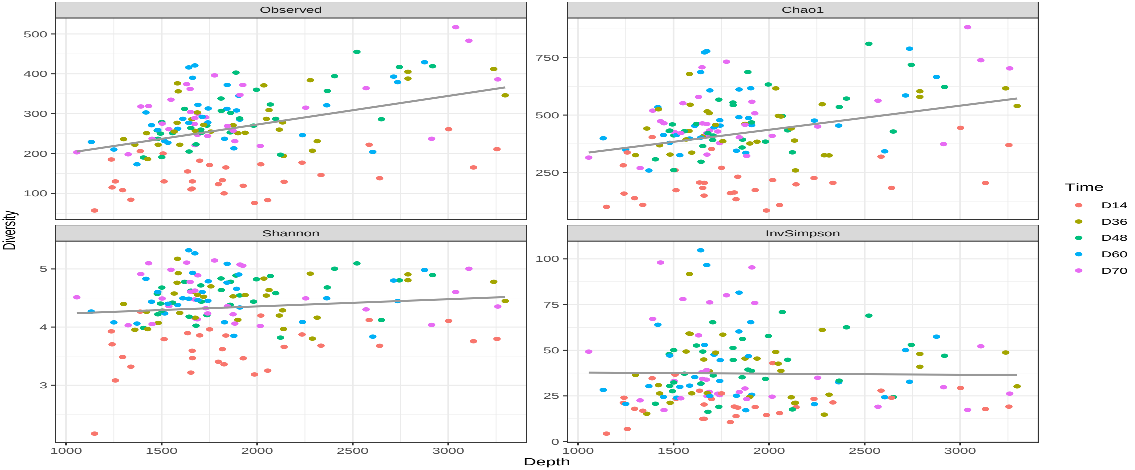
Outline

Rarefaction curve (I)



α diversity and sampling effort (II)

Quantitative α-diversities (Shannon, InvSimpson) are more robust to uneven sampling depths.



α diversity: numeric values

Numeric values of α-diversities are given by `estimate_richness` (used internally by `plot_richness`)

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

```
##                Observed    Chao1 se.chao1  Shannon InvSimpson
## sample_SF0901       165 204.4167 15.68345 3.755973   17.78229
## sample_SF0902       384 547.0769 34.18593 4.919017   61.11511
## sample_SF0903       417 718.5147 58.47191 4.802921   52.91534
## sample_SF0904       429 665.9231 48.68488 4.981387   57.42655
## sample_SF0905       517 882.5111 62.61703 4.602740   17.33970
## sample_SF0931       261 444.4054 47.19955 4.106061   29.32526
```

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

α diversity: A quick ANOVA (I)

```
data <- cbind(sample_data(kinetic), alpha.diversity)
data$Depth <- sample_sums(kinetic)
kinetic.richness.anova <- aov(Observed ~ Depth + Time*sex, data)
summary(kinetic.richness.anova) ## Depth is very significant

##               Df Sum Sq Mean Sq F value   Pr(>F)
## Depth          1 191170  191170  61.319 9.66e-13 ***
## Time           4 567509  141877  45.508  < 2e-16 ***
## sex            1  13815   13815   4.431    0.037 *
## Time:sex       4   3094     773   0.248    0.910
## Residuals    144 448935    3118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

α diversity: A quick ANOVA (II)

```
kinetic.simpson.anova <- aov(InvSimpson ~ Depth + Time*sex, data)
summary(kinetic.simpson.anova) ## as expected, Depth is barely significant

##              Df Sum Sq Mean Sq F value   Pr(>F)
## Depth         1     14    13.8   0.041    0.840
## Time          4  10681  2670.2   7.964 7.91e-06 ***
## sex           1    584   583.9   1.741    0.189
## Time:sex      4    389    97.2   0.290    0.884
## Residuals   144  48280   335.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation

Interpretation

- Diversity increases with time (with strong housing effect)
- Low shannon/InvSimpson diversities (compared to Observed, Chao1)
- $\Rightarrow$ communities dominated by a moderate number of abundant taxa

Comments

- Effective diversities more robust to depth bias
- Either correct for depth or perform rarefaction before comparing diversities

Interpretation

- Diversity increases with time (with strong housing effect)
- Low shannon/InvSimpson diversities (compared to Observed, Chao1)
- ⇒ communities dominated by a moderate number of abundant taxa

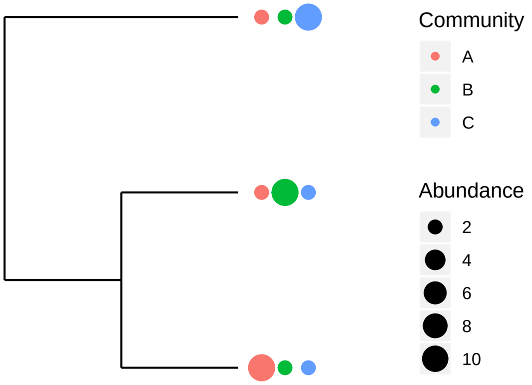
Comments

- Effective diversities more robust to depth bias
- Either correct for depth or perform rarefaction before comparing diversities

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 ($s = 1, \dots, S$) 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.

```
## Error in pivot_wider(df, id_cols = c("otu", "exp"), names_from = comm, :  impossible de
trouver la fonction "pivot_wider"
## Error in pivot_wider(df, id_cols = c("otu", "exp"), names_from = comm, :  impossible de
trouver la fonction "pivot_wider"
## Error in pivot_longer(., cols = one_of(c("Community_1", "Community_2", :  impossible de
trouver la fonction "pivot_longer"
## Error in pivot_longer(., cols = Community_1:Community_2, values_to = "abondance", :
impossible de trouver la fonction "pivot_longer"
## Error in do.call(rbind, list(df.jac[, cols], df.bc[, cols])):  objet 'df.bc' introuvable
## Error in factor(df.dist$class, levels = c("Jaccard", "Bray-Curtis")):  objet 'df.dist'
introuvable
## Error in factor(df.dist$nature, levels = c("Specific to 1", "Specific to 2", :  objet
'df.dist' introuvable
## Error in eval(expr, envir, enclos):  objet 'df.dist' introuvable
## Error in ggplot(df.dist, aes(x = otu, y = abondance)):  objet 'df.dist' introuvable
## Error in arrangeGrob(...):  objet 'p.dist' introuvable
```

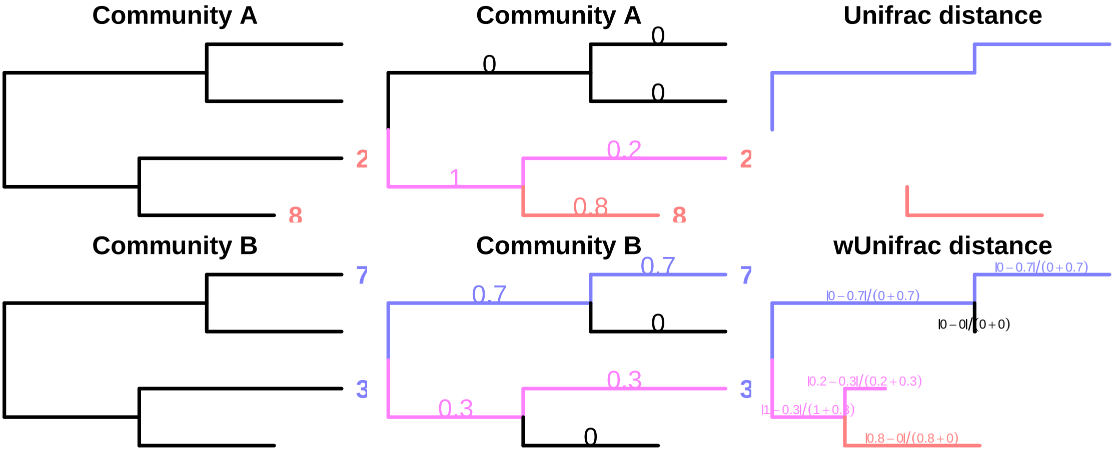
β-diversity: phylogenetic

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.

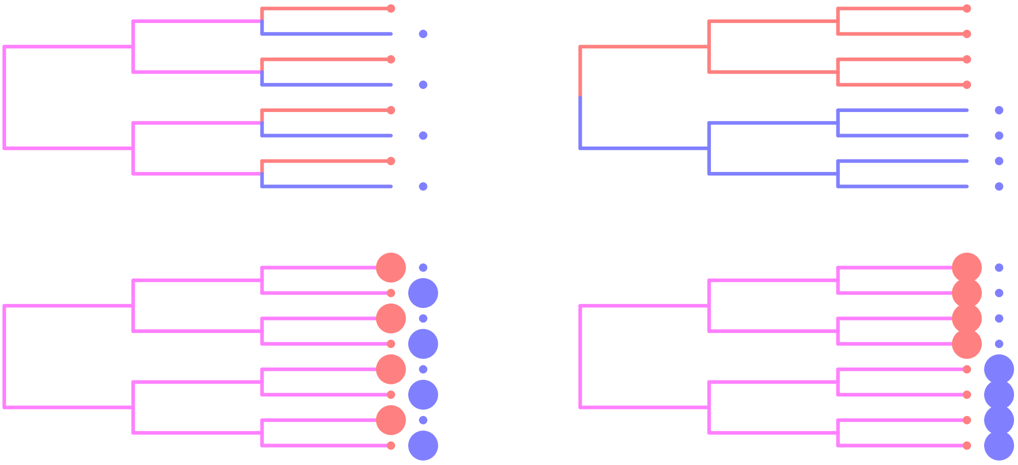
Unifrac	Weighted Unifrac
Fraction of the tree specific to either 1 or 2	Fraction of the diversity specific to 1 or to 2
$d_{\text{UF}} = \frac{\sum_e l_e \left[1_{\{p_e>0, q_e=0\}} + 1_{\{q_e>0, p_e=0\}}\right]}{\sum_e l_e \times 1_{\{p_e+q_e>0\}}}$	$d_{\text{UF}} = \frac{\sum_e l_e \lvert p_e - q_e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

β-diversity: phylogenetic

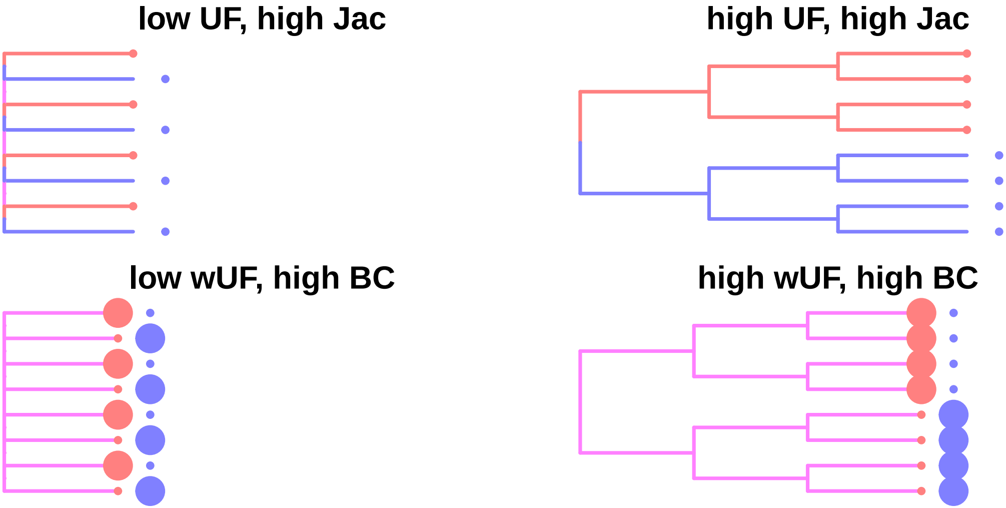
For each branch e, note l_e its length and p_e (resp. q_e) the fraction of community 1 (resp. community 2) below branch e. We focus on shared features.



Differences between the β-dissimilarities



Differences between the β-dissimilarities



β-dissimilarities/distances in phyloseq

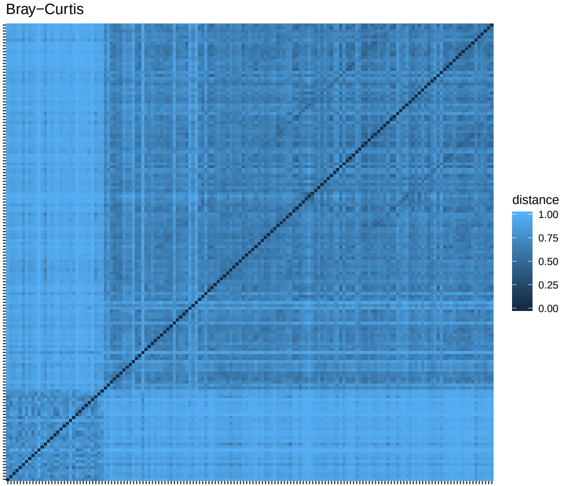
β dissimilarities are computed with `distance`.

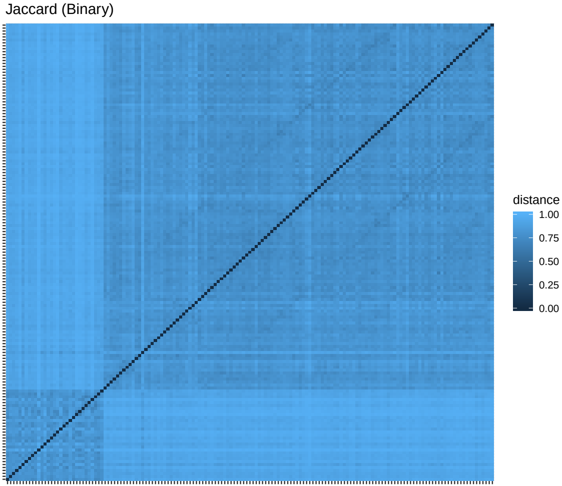
```
dist.bc <- distance(kinetic.rare, method = "bray") ## Bray-Curtis
```

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"
##
```

β-dissimilarities/distances in phyloseq (II)



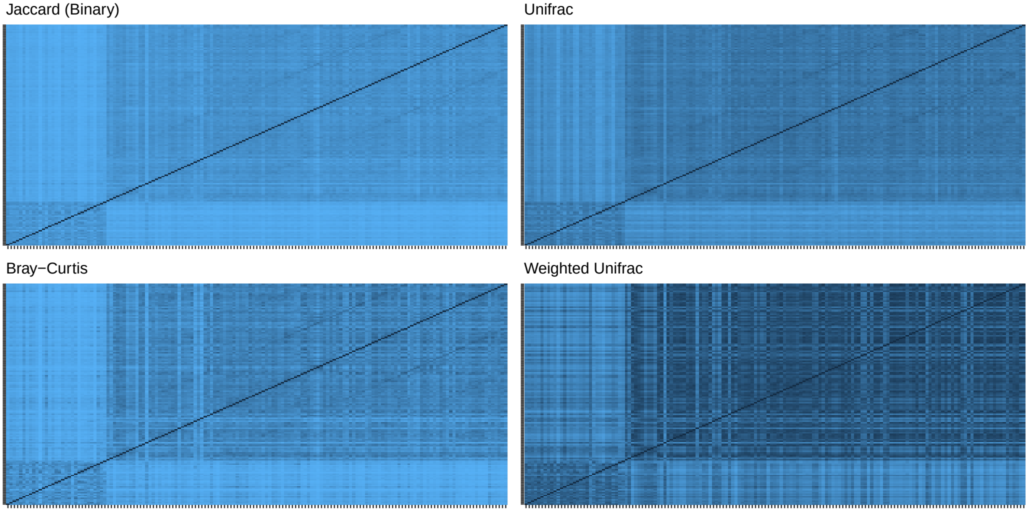


Phylogenetic β-dissimilarities/distances in phyloseq (II)

Phylogenetic distances require rooted tree

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

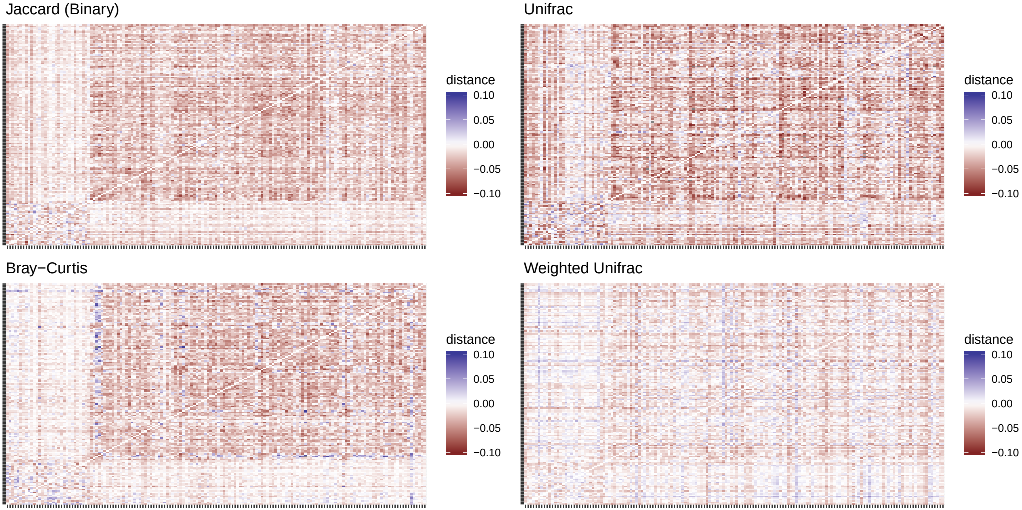
Compositional vs Qualitative



Compositional vs Qualitative (II)

- Jaccard higher than Bray-Curtis $\Rightarrow$ abondant taxa are shared
- Jaccard higher than Unifrac $\Rightarrow$ communities' taxa are distinct but phylogenetically related
- Unifrac higher than weighted Unifrac $\Rightarrow$ abondant taxa in communities are phylogenetically close.

Raw counts vs rarefied counts



Raw counts vs rarefied counts (II)

- Different sampling efforts lead to biased distances
- Bias higher for qualitative (Jaccard/UniFrac) than quantitative (Bray-Curtis/wUniFrac)distances.
- wUniFrac most robust to different sampling depths (unaffected in principle, works on relative abundances)

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

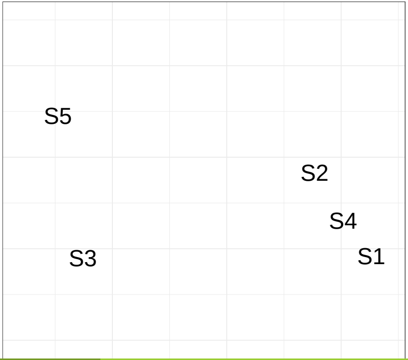
But variance is not a very good measure of β-diversity.

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 $\|X_i - X_j\| \simeq d_{ij}$

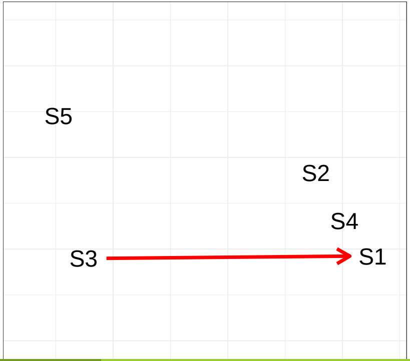
	S1	S2	S3	S4	S5
S1	0.00	2.21	6.31	0.99	7.50
S2	2.21	0.00	5.40	1.22	5.74
S3	6.31	5.40	0.00	5.75	3.16
S4	0.99	1.22	5.75	0.00	6.64
S5	7.50	5.74	3.16	6.64	0.00



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

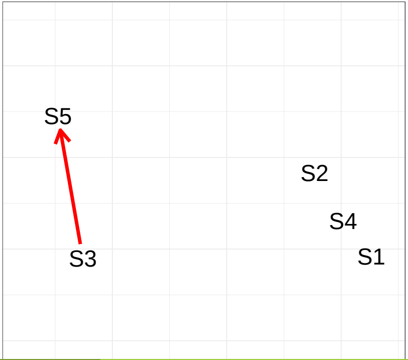


MultiDimensional Scaling (MDS/PCoA)

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(kinetic.rare, method = "MDS", distance = "bray")
```

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

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

The graphic is then produced with plot_ordination

```
p <- plot_ordination(kinetic.rare, ord, color = "Time", shape = "Bande")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
p <- p + stat_ellipse(aes(group = Time)) ## add ellipses around each time level
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(kinetic.rare, method = "MDS", distance = "bray")
```

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

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

The graphic is then produced with plot_ordination

```
p <- plot_ordination(kinetic.rare, ord, color = "Time", shape = "Bande")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
p <- p + stat_ellipse(aes(group = Time)) ## add ellipses around each time level
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(kinetic.rare, method = "MDS", distance = "bray")
```

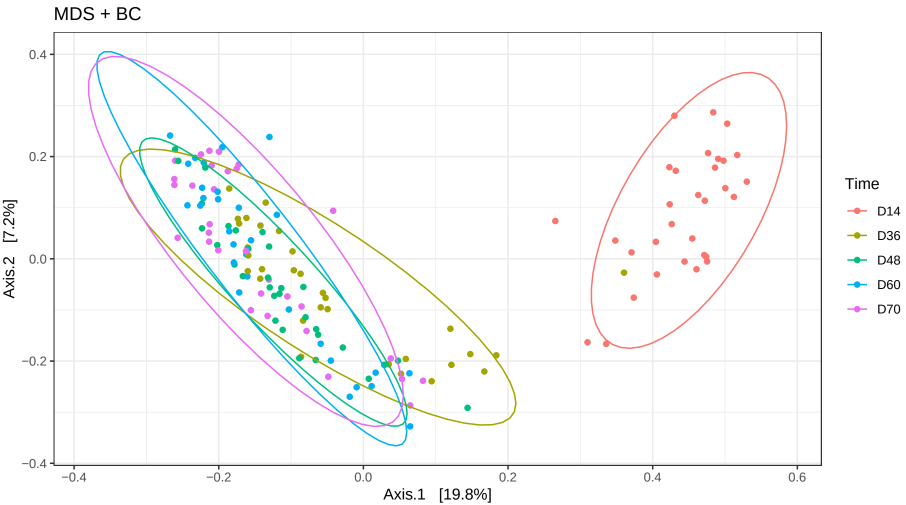
or by passing a distance matrix directly (useful if you already computed it)

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

The graphic is then produced with plot_ordination

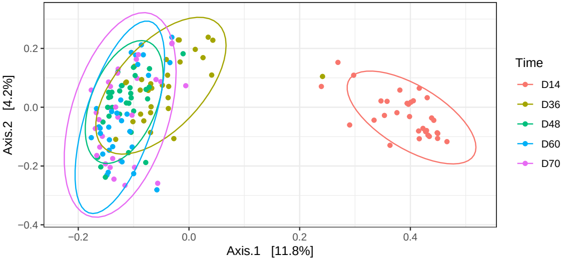
```
p <- plot_ordination(kinetic.rare, ord, color = "Time", shape = "Bande")
p <- p + theme_bw() + ggtitle("MDS + BC") ## add title and plain background
p <- p + stat_ellipse(aes(group = Time)) ## add ellipses around each time level
plot(p)
```

Ordination in phyloseq : plot_ordination

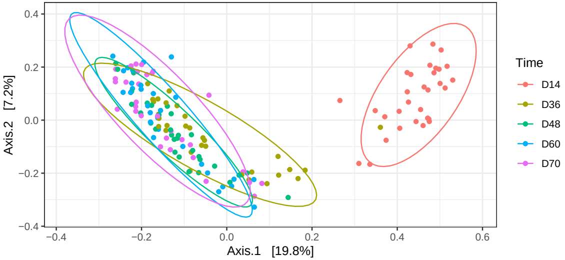


Impact of distance

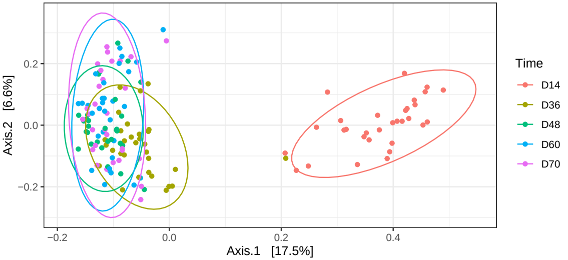
MDS + BC



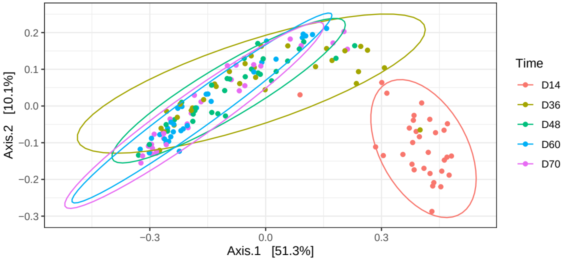
MDS + Jaccard



MDS + UF



MDS + wUF



Interpretation

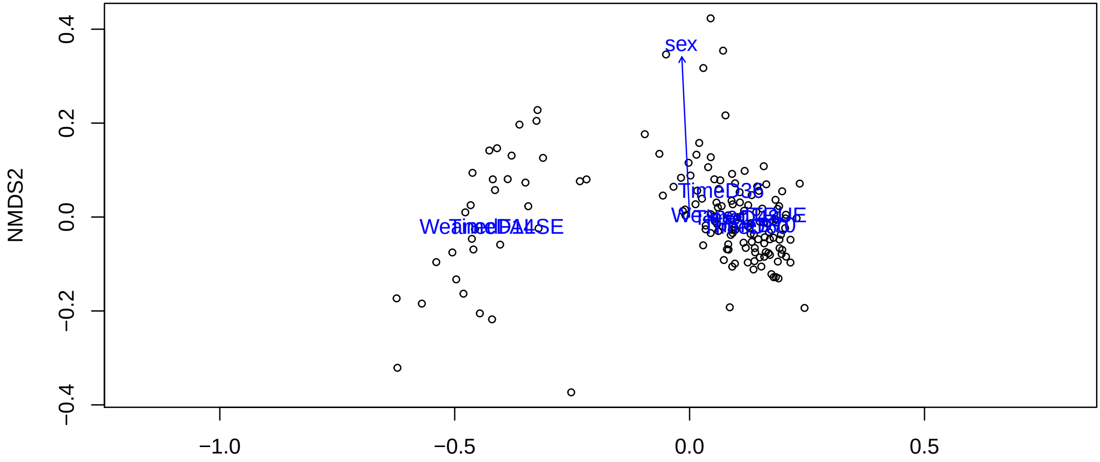
- Qualitative distances (Unifrac, Jaccard) separate D14 and the rest.
- wUF mixes up some sample: the taxa séparating D14 from the rest may be replaced by (phylogenetically) close siblings.
- All distances (wUnifrac) exhibit a high gradient corresponding to high heterogeneity of samples on axis 2.
- Large overlap between groups in terms of both relative composition and species composition (a side effect of undersampling?)
- Warning The 2-D representation captures only part of the original distances.

Ordination using vegan's ordiplot

"Vector fitting" overlays metadata on the ordination plot.

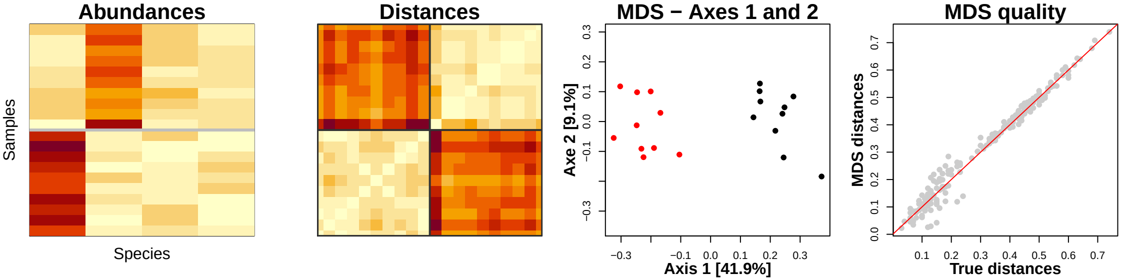
```
## Ordination
dist.bc <- distance(kinetic.rare, "bray")
kin.mds <- metaMDS(dist.bc, trace = 0)
## Vector fitting
ef <- envfit(kin.mds, sample_data(kinetic.rare))
## Plot only most significant variables
plot(kin.mds)
plot(ef, p.max = 0.5)
```

Ordination using vegan's `ordiplot` (II)

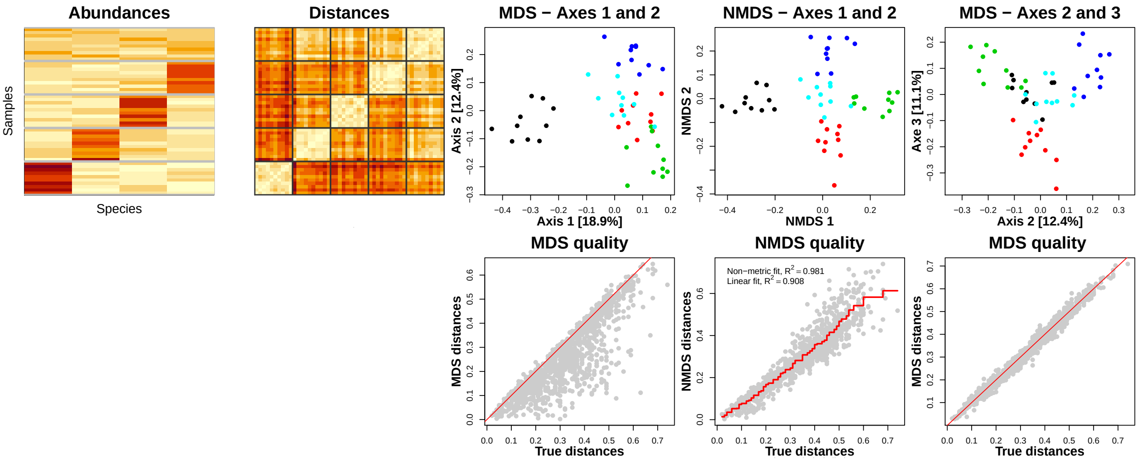


What about Nonmetric MDS (NMDS)?

NMDS does not preserve distance values but rather their **relative ordering**.

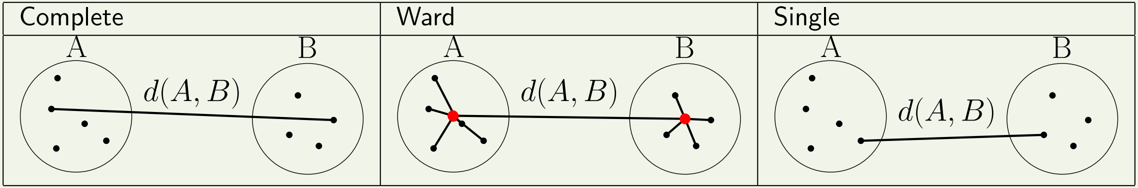


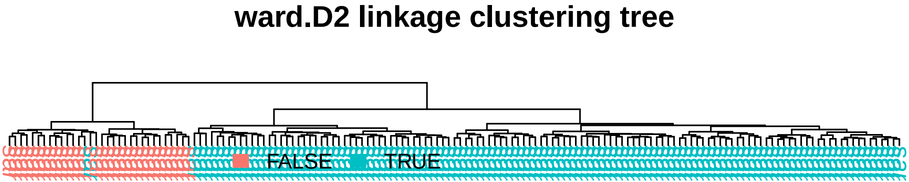
When MDS fails



Outline

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





Clustering with `hclust`

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

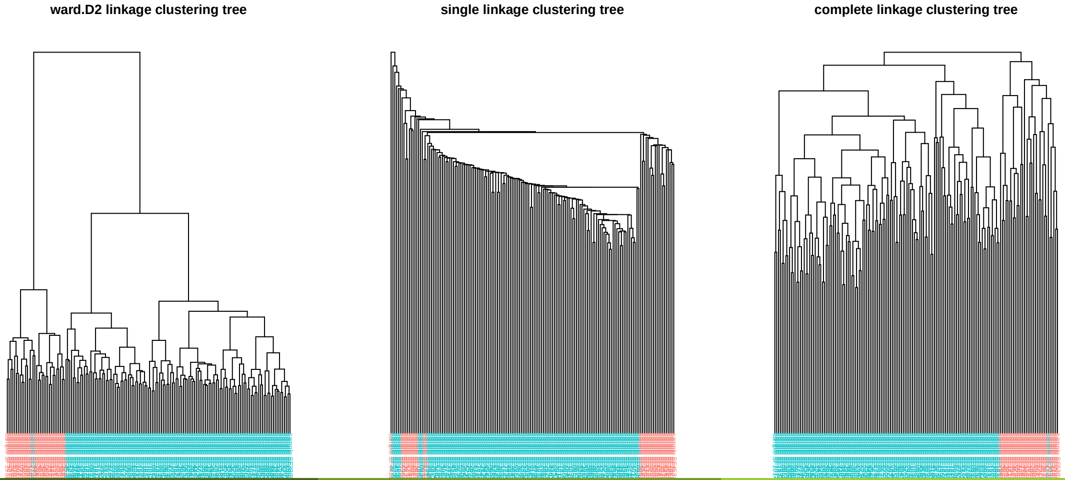
Feed to `hclust` and plot

```
plot_clust(kinetic.rare, dist = "bray", method = "linkage.function", color = "Weaned")
## Or if you already computed the distance matrix
plot_clust(kinetic.rare, dist = dist.bc, color = "Weaned")
```

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_clust(kinetic.rare, "bray", method = "ward.D2", color = "Weaned")
plot_clust(kinetic.rare, "bray", method = "single", color = "Weaned")
plot_clust(kinetic.rare, "bray", method = "complete", color = "Weaned")
```



Remarks

- Consistent with the ordination plots, clustering shows a good structure (D14 vs. rest) for the Bray-Curtis distance for the Ward linkage
- Differents distances would result (in this case) in similar results.
- 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

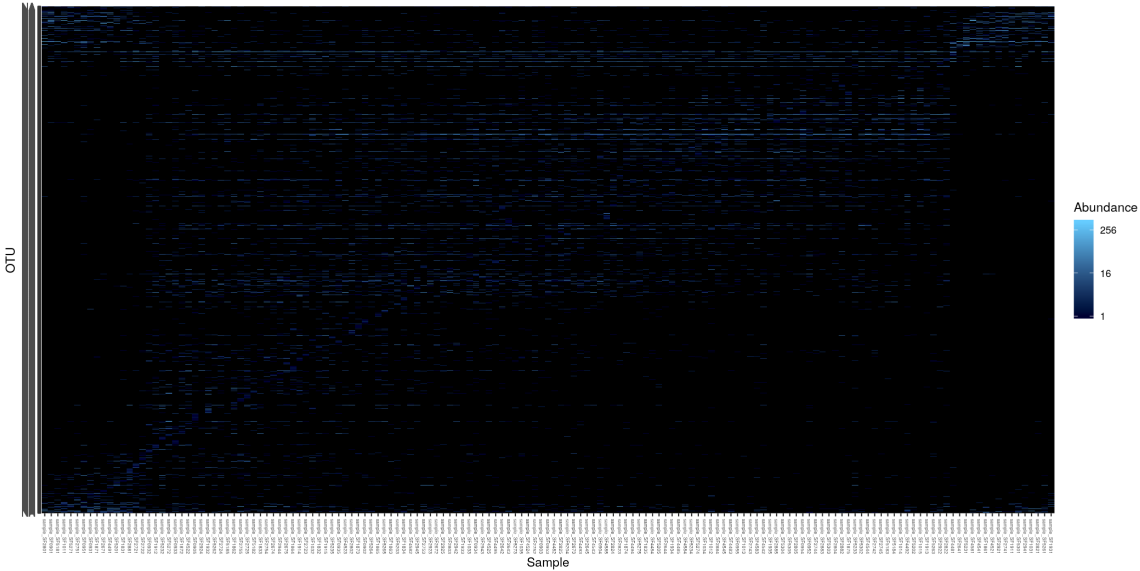
```
p <- plot_heatmap(kinetic.rare, low = "yellow", high = "red", na.value = "white",
              sample.order = mySampleOrder, taxa.order = myTaxaOrder)
## add facetting
p <- p + facet_grid(~Time, scales = "free_x")
plot(p)
```

Heatmap with plot_heatmap

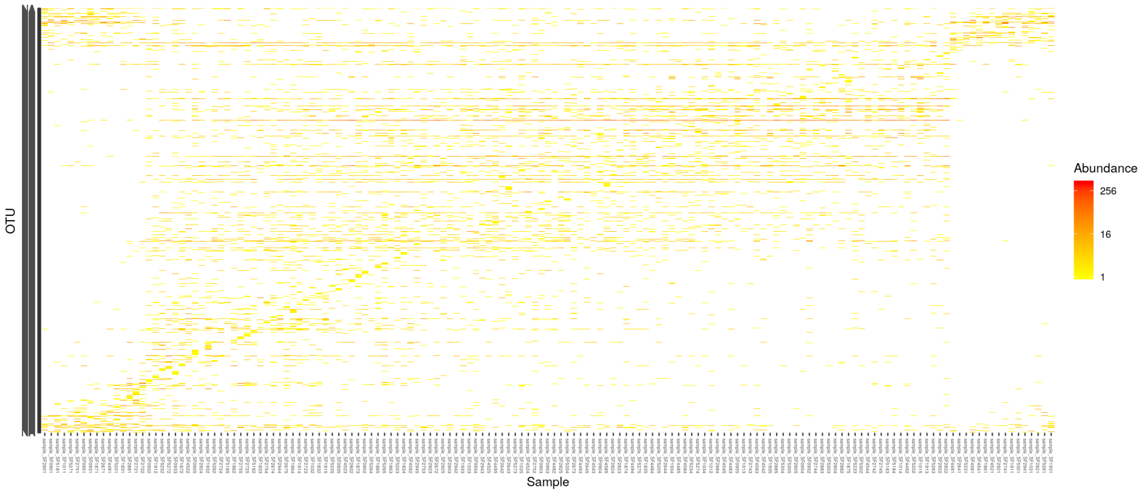
plot_heatmap is a versatile function to visualize the count table.

- 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

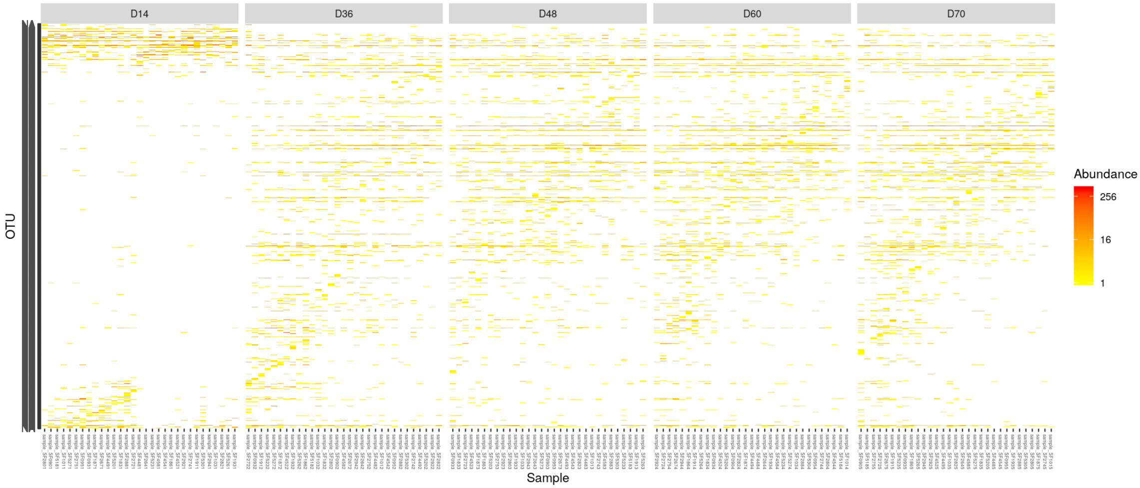
```
p <- plot_heatmap(kinetic.rare, low = "yellow", high = "red", na.value = "white",
                  sample.order = mySampleOrder, taxa.order = myTaxaOrder)
## add facetting
p <- p + facet_grid(~Time, scales = "free_x")
plot(p)
```


OTU
Sample
Abundance
256
16
1

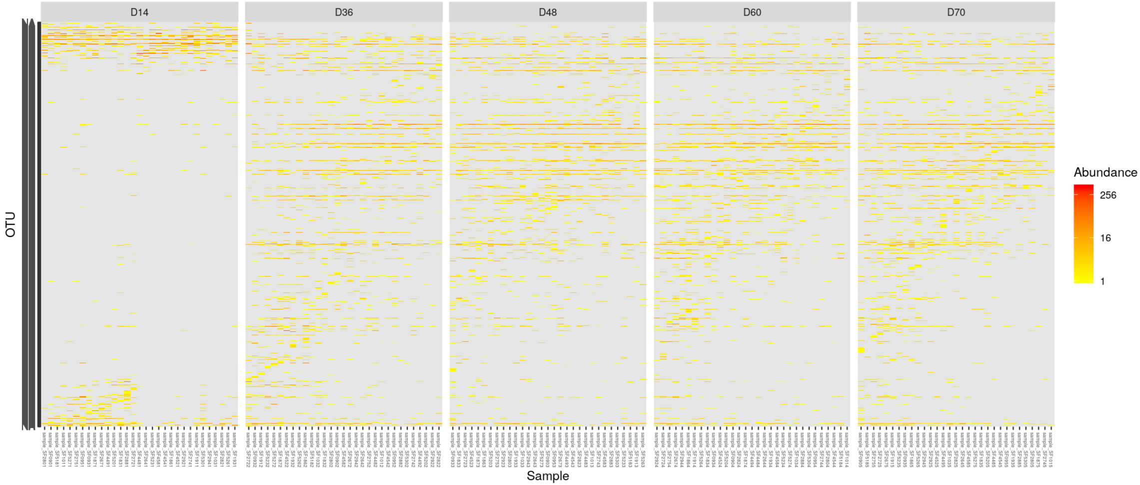
```r
plot_heatmap(kinetic.rare, low = "yellow", high = "red", na.value = "white")
```



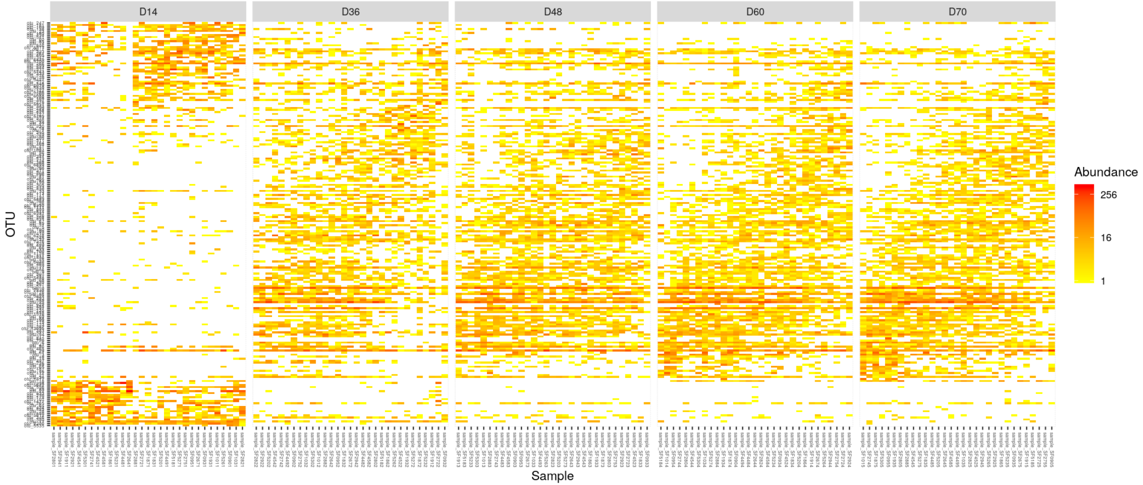
```r
plot_heatmap(kinetic.rare, low = "yellow", high = "red", na.value = "white") +
    facet_grid(~Time, scales = "free_x")
```



```r
plot_heatmap(kinetic.rare, low = "yellow", high = "red", na.value = "grey90") +
    facet_grid(~Time, scales = "free_x")
```

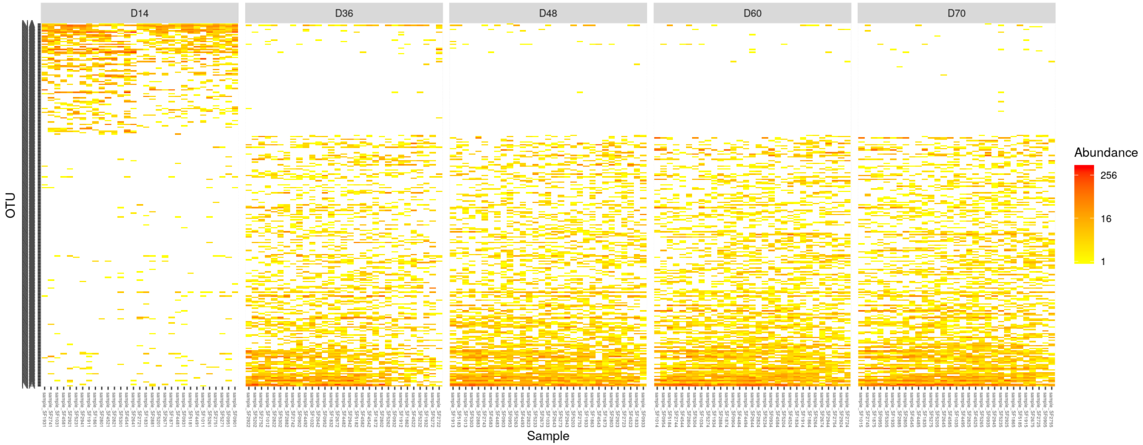


Heatmap of the 200 most abundant taxa only



If you have differentially abundant taxa sorted by effect size in da.otus.

```
plot_heatmap(prune_taxa(da.otus, kinetic.rare), taxa.order = da.otus,
             low = "yellow", high = "red", na.value = "white") +
    facet_grid(~Time, scales = "free_x")
```



Interpretation

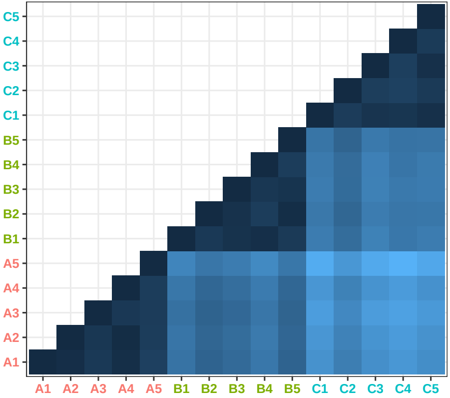
- Block-like structure of the abundance table
- Interaction between (groups of) taxa and (groups of) samples
- Core and condition-specific microbiota
- $\Rightarrow$ Classification of taxa and use of custom taxa order to highlight structure

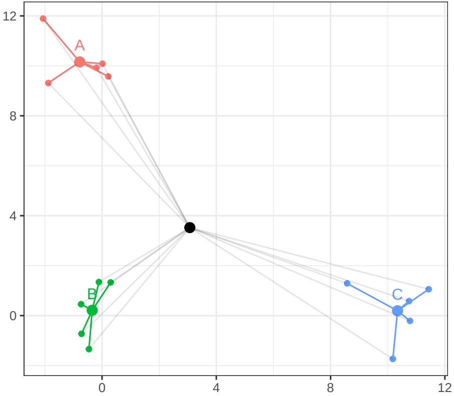
Outline

Rationale

Idea

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





Outline

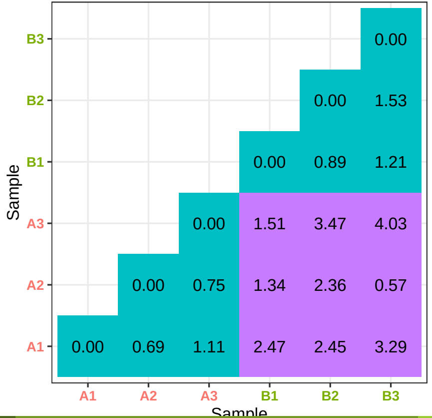
Multivariate ANOVA

Idea

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

Idea

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



Multivariate ANOVA with adonis

The covariates explains roughly 22% of the total variation.

```
metadata <- as(sample_data(kinetic.rare), "data.frame")
adonis(dist.bc ~ Time + sex, data = metadata, perm = 9999)

##
## Call:
## adonis(formula = dist.bc ~ Time + sex, data = metadata, permutations = 9999)
##
## Permutation: free
## Number of permutations: 9999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Time        4     9.587 2.39681  9.6645 0.20450  1e-04 ***
## sex         1     0.341 0.34118  1.3757 0.00728  1e-01 .
## Residuals 149    36.952 0.24800         0.78822
## Total     154    46.881                 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)
- PERMANOVA assumes linear responses to the covariate

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)
- PERMANOVA assumes linear responses to the covariate

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

Constrained Analysis of Principal Coordinates (CAP)

Idea

Find associations between community composition and environmental variables

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

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(kinetic.rare), "data.frame") ## convert sample_data to data.frame
cap <- capscale(dist.bc ~ Time + sex, data = metadata)
```

CAP with capscale (II)

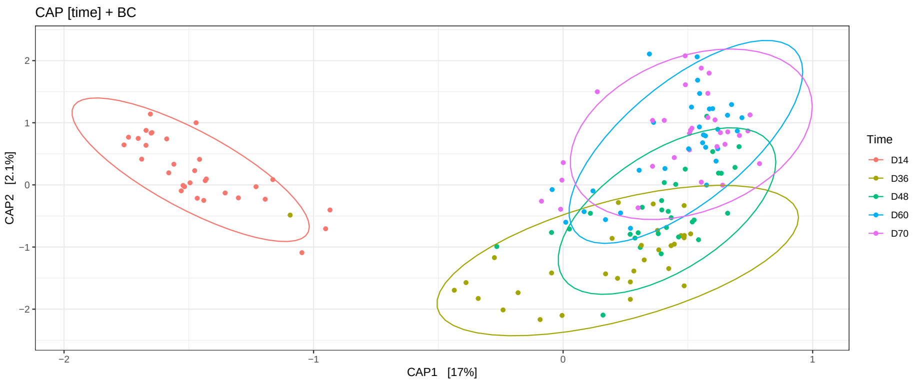
Sample type explains roughly 22% of the total variation between samples (as measured by wUnifrac)

```
cap

## Call: capscale(formula = dist.bc ~ Time + sex, data = metadata)
##
##                 Inertia Proportion Rank
## Total         46.880766   1.000000
## Constrained    9.937185   0.211967    5
## Unconstrained 37.354201   0.796792  135
## Imaginary     -0.410621  -0.008759   19
## Inertia is squared Bray distance
##
## Eigenvalues for constrained axes:
##  CAP1  CAP2  CAP3  CAP4  CAP5
## 8.079 1.011 0.387 0.320 0.140
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8
## 3.809 1.817 1.657 1.357 1.234 1.066 0.954 0.884
## (Showing 8 of 135 unconstrained eigenvalues)
```

CAP with capscale (III)

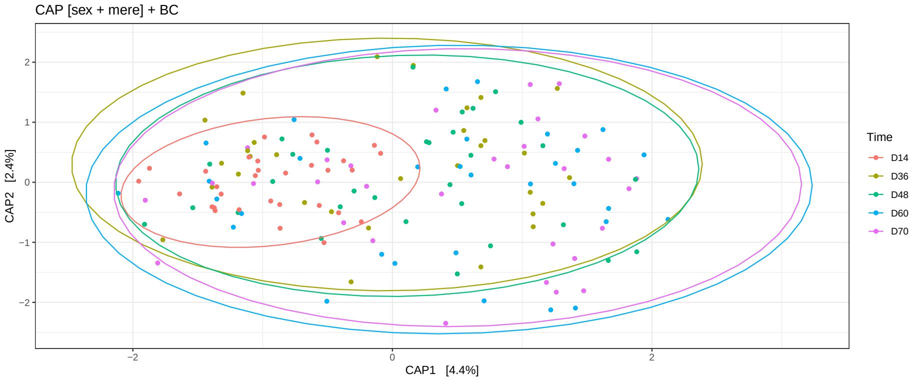
```
anova <- anova(cap, permutations = 999)
print(anova)

## Permutation test for capscale under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: capscale(formula = dist.bc ~ Time + sex, data = metadata)
##           Df SumOfSqs      F Pr(>F)
## Model      5    9.937 7.9276  0.001 ***
## Residual 149   37.354
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
p <- plot_ordination(kinetic.rare, ordinate(kinetic.rare, "CAP", "bray", ~ Time), color = "Time")
p <- p + theme_bw() + ggtitle("CAP [time] + BC") + stat_ellipse(aes(group = Time))
plot(p)
```



CAP as an ordination method

```
p <- plot_ordination(kinetic.rare, ordinate(kinetic.rare, "CAP", "bray", ~ sex + mere),
                     color = "Time")
p <- p + theme_bw() + ggtitle("CAP [sex + mere] + BC") + stat_ellipse(aes(group = Time))
plot(p)
```



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

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:

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

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(kinetic, ~ Weaned)

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

and then fit the model

```
dds <- DESeq2::DESeq(cds)

## estimating size factors
## Error in estimateSizeFactorsForMatrix(counts(object), locfunc = locfunc, :  every gene
contains at least one zero, cannot compute log geometric means
```

DESeq2 with phyloseq (II)

In our case, fitting failed because the dataset is way too sparse and not really adapted to DA analysis using sophisticated model. We'll be be smarter and tell DESeq to use only positive counts when computing the size factors.

```
cds <- phyloseq_to_deseq2(kinetic, ~ Weaned)

## converting counts to integer mode
```

and then fit the model (this can take some time and still throws some warnings)

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

DESeq2 with phyloseq (III)

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

```
options(digits = 3)
results <- DESeq2::results(dds, name = "WeanedTRUE", tidy = TRUE)
## results <- DESeq2::results(dds, contrast = c("Time", "D14", "D36")) for testing Time D36 against D14
da.otus <- results %>% rename(OTU = row)
head(da.otus, 2)

##         OTU baseMean log2FoldChange lfcSE  stat pvalue padj
## 1  otu_692  0.04243          0.788  3.63 0.217  0.828   NA
## 2 otu_1686  0.00595          0.672  3.63 0.185  0.853   NA

da.otus <- subset(da.otus, padj < 0.01) ## significant otus
dim(da.otus)

## [1] 321   7
```

DESeq2 with phyloseq (IV)

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

```
tax_df <- tax_table(kinetic) %>%
  as("matrix") %>% as.data.frame() %>%
  mutate(OTU = taxa_names(kinetic))
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_123     8.85           2.33 0.328 7.11 1.15e-12 8.15e-12 Bacteria
## 2  otu_55     1.59           3.84 0.618 6.22 4.92e-10 2.61e-09 Bacteria
##           Phylum       Class         Order          Family      Genus
## 1 Bacteroidetes Bacteroidia Bacteroidales Prevotellaceae Prevotella
## 2 Bacteroidetes Bacteroidia Bacteroidales Prevotellaceae Prevotella
```

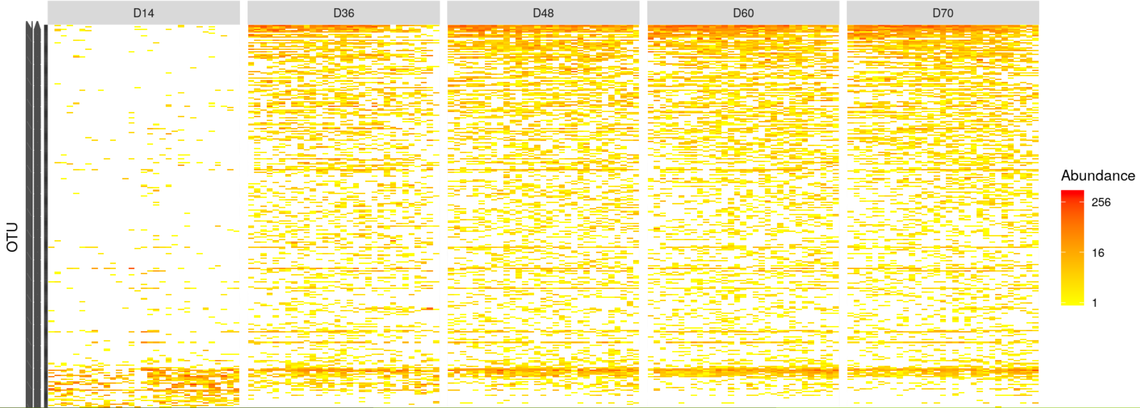
Sort taxa by $\log_2$ fold change

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

```
##         OTU baseMean log2FoldChange lfcSE  stat    pvalue      padj  Kingdom
## 1   otu_22    17.30          -7.96 0.788 -10.1 5.09e-24 9.79e-23 Bacteria
## 2 otu_1477     6.14          -7.08 0.646 -11.0 6.54e-28 1.68e-26 Bacteria
```

DESeq2 with phyloseq (VI)

```
plot_heatmap(prune_taxa(da.otus$OTU, kinetic.rare),
             taxa.order = da.otus$OTU,
             low = "yellow", high = "red", na.value = "white") +
    facet_grid(~Time, scales = "free_x")
```



DA taxa

We will now add a "DA" column to the taxonomy to say which OTUs are (significantly) more abundant after weaning, before weaning or neither.

```
## create OTU status vector
da.class <- rep("None", ntaxa(kinetic))
names(da.class) <- taxa_names(kinetic)
weaned.otus <- subset(da.otus, log2FoldChange < 0)$OTU
not.weaned.otus <- subset(da.otus, log2FoldChange > 0)$OTU
da.class[weaned.otus] <- "Before Weaning"
da.class[not.weaned.otus] <- "After Weaning"
## Add new vector to taxonomy
tax_table(kinetic) <- cbind(tax_table(kinetic)[, 1:6], da.class)
```

DA taxa (II)

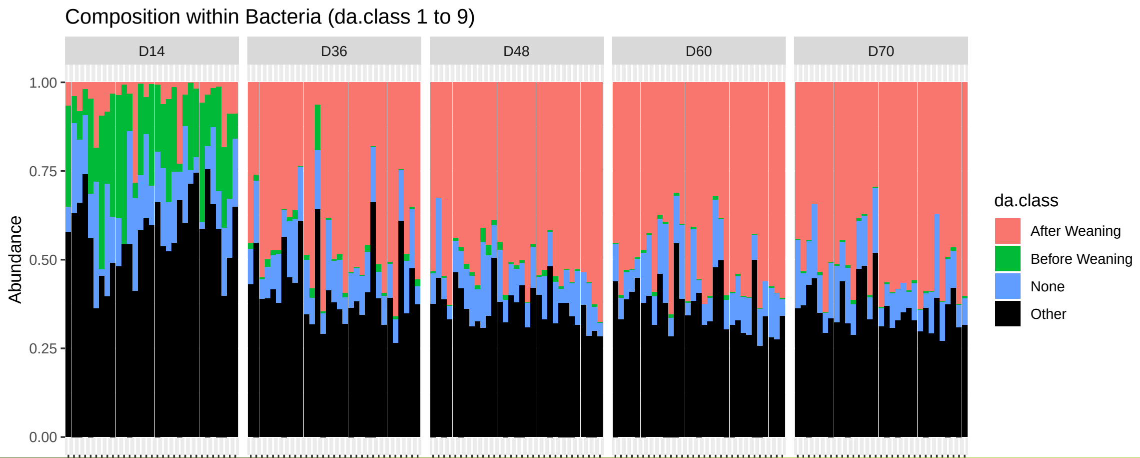
We will now add a "DA" column to the taxonomy to say which OTUs are (significantly) more abundant after weaning, before weaning or neither.

```
head(tax_table(kinetic))
```

```
## Taxonomy Table:     [6 taxa by 7 taxonomic ranks]:
##          Kingdom    Phylum          Class         Order
## otu_692  "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
## otu_1686 "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
## otu_2192 "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
## otu_3292 "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
## otu_4395 "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
## otu_2267 "Bacteria" "Bacteroidetes" "Bacteroidia" "Bacteroidales"
##          Family            Genus        da.class
## otu_692  "Prevotellaceae" "Prevotella" "None"
## otu_1686 "Prevotellaceae" "Prevotella" "None"
## otu_2192 "Prevotellaceae" "Prevotella" "None"
## otu_3292 "Prevotellaceae" "Prevotella" "None"
## otu_4395 "Prevotellaceae" "Prevotella" "None"
## otu_2267 "Prevotellaceae" "Prevotella" "None"
```

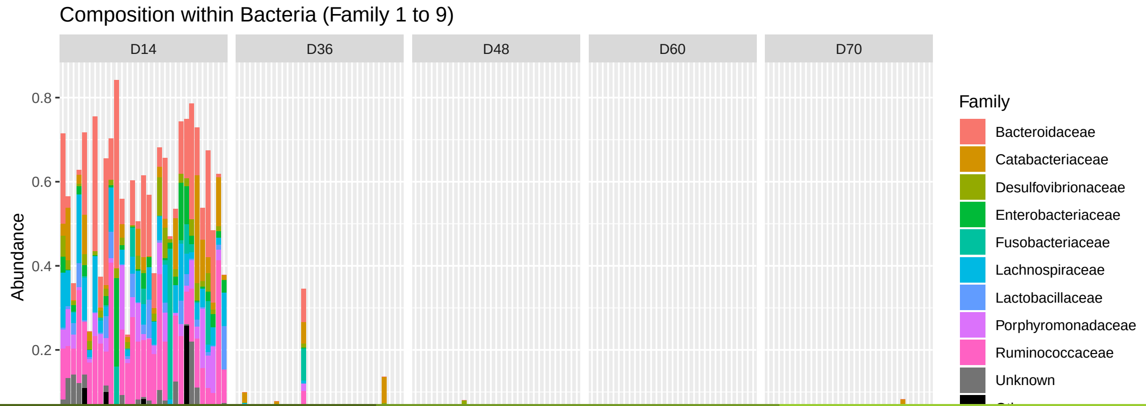
DA taxa (III)

```r
p <- plot_composition(kinetic, "Kingdom", "Bacteria", "da.class", fill = "da.class")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```



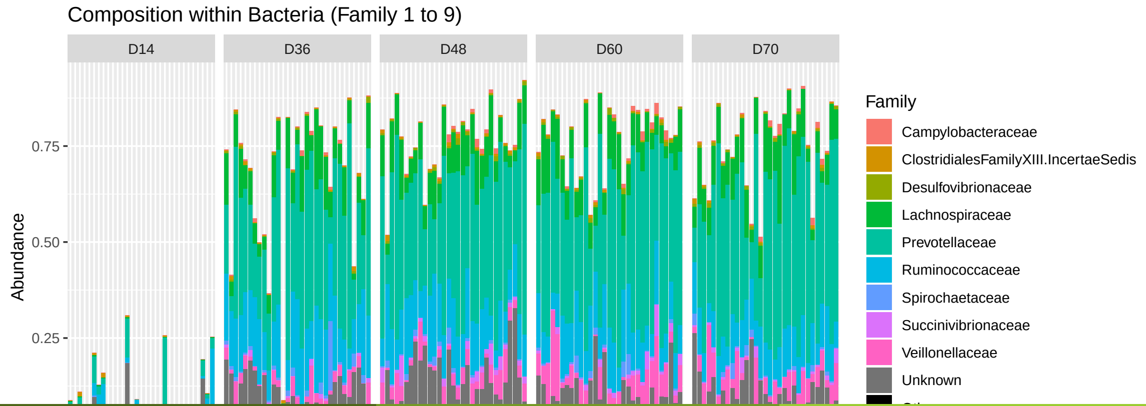
DA taxa (IV)

```
weaned_fraction <- kinetic %>% transform_sample_counts(fun = count_to_prop) %>%
  subset_taxa(da.class == "Before Weaning")
p <- plot_composition(weaned_fraction, "Kingdom", "Bacteria", "Family", fill = "Family")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```



DA taxa (V)

```
not_weaned_fraction <- kinetic %>% transform_sample_counts(fun = count_to_prop) %>%
  subset_taxa(da.class == "After Weaning")
p <- plot_composition(not_weaned_fraction, "Kingdom", "Bacteria", "Family", fill = "Family")
p <- p + facet_wrap(~Time, scales = "free_x", nrow = 1)
plot(p)
```



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 (∼93% 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 (∼93% 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 ($\sim$93% for `kinetic`)

- Erroneous OTUs
- Group/Environment-specific OTUs.

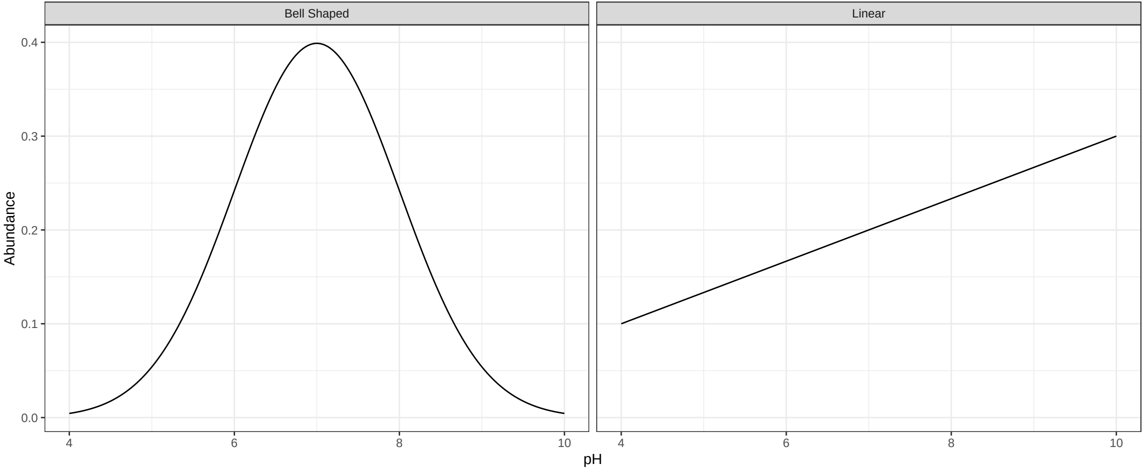
Not clear how negative binomial models cope with this sparsity

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

Outline

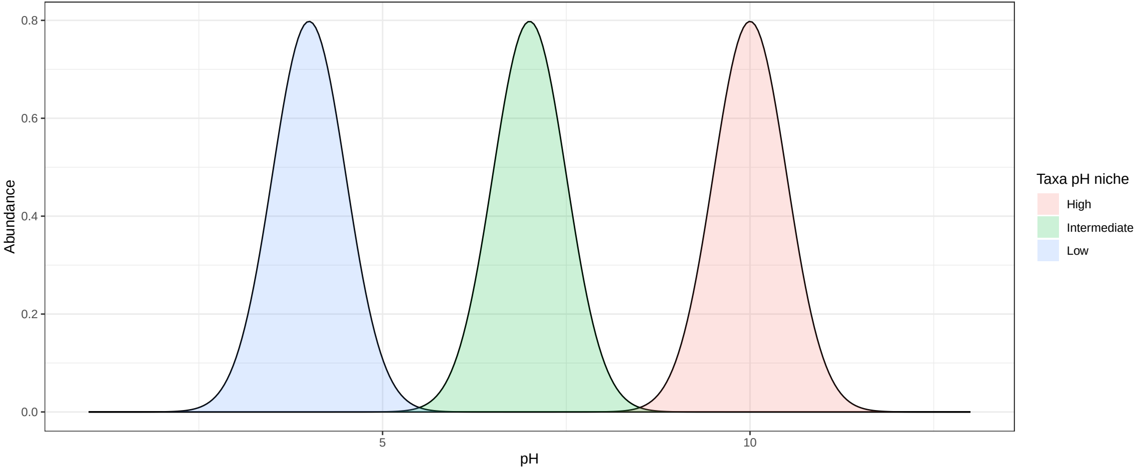
A few words about linear responses

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)



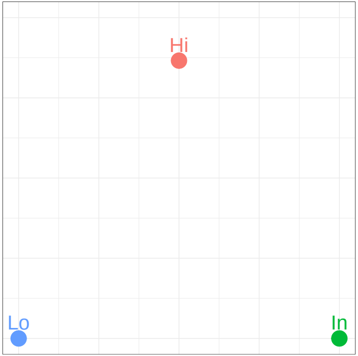
A word about linear responses (II)

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



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

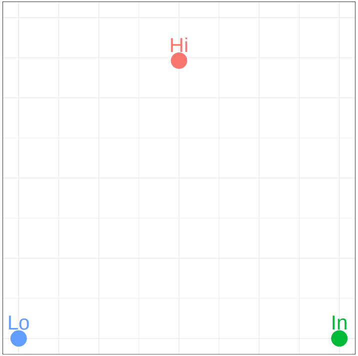


But troubles start when you add more communities...

A word about linear responses (III)

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

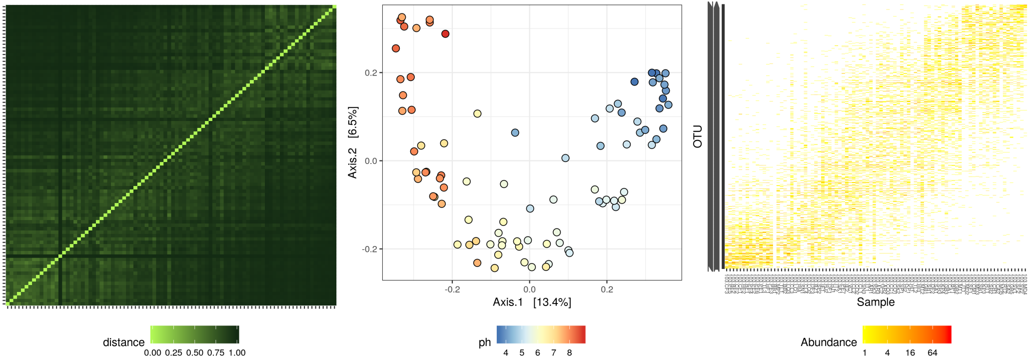
	Lo	In	Hi
Lo	0.00	1.00	1.00
In	1.00	0.00	1.00
Hi	1.00	1.00	0.00



But troubles start when you add more communities...

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

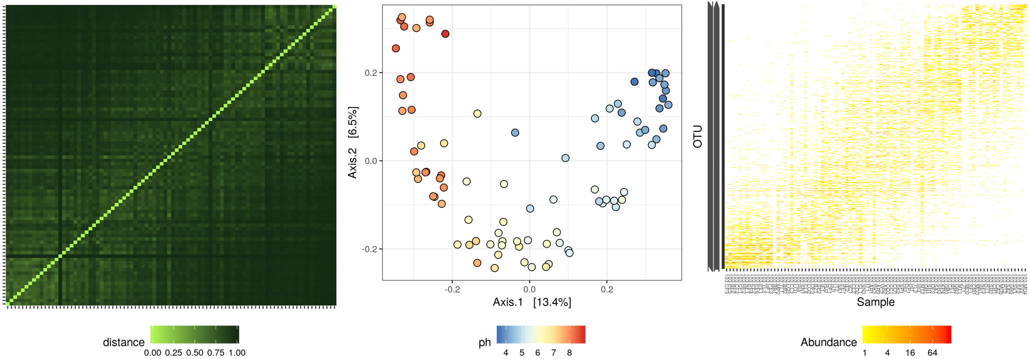
Distances saturate → 2D plot doesn't capture *linear gradient* shown in heatmap.



- 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`
- Explore graphics with `plotly`

Final word about graphics

Most of the graphics were produced using `ggplot`. If you have installed `plotly` on your computer, you can navigate them by replacing `plot` with `ggplotly` (requires RStudio version ≥ 1.0).

```
install.package(plotly)
ord <- ordinate(kinetic.rare, method = "MDS", distance = "bray")
p <- plot_ordination(kinetic.rare, ord, color = "Time", shape = "Bande")
ggplotly(p) ## replaces plot(p)
```

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.

- 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: Chaillou

Dataset from Chaillou et al. (2015) used to study bacterial communities from 8 different food products (`EnvType`), distributed as 4 meat products and 4 seafoods. Used to find core microbiota of food products.

- Compare α-diversities across environments (`EnvType`). Which environments are more/less diverse? Is it consistent with your intuition?
- Are the difference between food products reflected in the communities?
- What happens to ordination plots when you move from one distance to another (among the four seen previously)? What does it tell you?
- `DesLardons` (sliced bacon) use sea salt. Is it coherent with the results observed using Jaccard and Unifrac distance?
- Are some taxa differentially abundant between meat and seafood?

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)?