16s rRNA analysis workshop

Hena R. Ramay

2020-06-15

Contents

1 Introduction **5**
  1.1 Workshop Schedule 5
  1.2 Important links 6

2 Introduction **7**
  2.1 What are we trying to achieve 7
  2.2 Basics & Background 7

3 Primer Removal **9**
  3.1 CutAdapt 9

4 DADA2 **11**
  4.1 DADA2 pipeline (v1.2) 11

5 Day Two **15**
  5.1 Main concepts to be discussed: 15
  5.2 Alpha & beta Diversity 19

6 Summary **21**

7 What Next? **23**

Chapter 1

Introduction

Our intention is to develop workshop material as we go along. For each day of the workshop, the basic material will be uploaded and more details will be added based on your questions and problems we encounter. So please ask as many questions are you can to help us in making this workshop better!!

1.1 Workshop Schedule

We will try to cover the following material in the course:

- Day1
    - 9:00 - 9:45 am Introductions and a presentation of the basic concepts
    - 9:45 - 10:30 am CutAdapt- how remove primers
    - 10:30 - 10:45 am Break
    - 10:45 - 12 pm Installations of the required packages and data download
    - 12 - 1 pm Lunch break
    - 1 - 3 pm Reading data in and inspecting read quality
- Day2
    - 9:30 -10:30 am Filter and trim + learn error rates + Sample inference + Merge samples
    - 10:30 - 10:45 am Break
    - 10:45 - 12 Generate sequence table and remove Chimeras
    - 12 - 1 pm Lunch break
    - 1- 3 pm Taxonomy explanation + assign your sequences. If time permits, make a phyloseq object
- Day3
    - 9-10:30 am Using a phyloseq object to calculate alpha and beta diversity

  - 10:30 on wards we will use a real world dataset to re-do what we have learned here

1.2 Important links

- DADA2 Tutorial : link
- Day One presentation : link
- Day One dataset : link
- Project dataset : link

Chapter 2

Introduction

We are very excited to teach this workshop and share what we know with you all !

So lets start by talking about the very basics:

2.1 What are we trying to achieve

Our goal is very similar to gathering data on a city neighbourhood to find out who lives there, how the demographic changes over time or in case of a drastic event. We can gather more information by asking about neighbours, quality of life etc. Similarly when we are looking at microbial communities our first question is who is there, how abundant and how their presence changes over time or when conditions change. We can also ask questions like how the microbiomes are interacting with each other (metabolites).

For the scope of this workshop we will stick to the simple questions: who and how much?

2.2 Basics & Background

Here is the link to the lecture we will start with today: workshop

Key points are:

- Think of a hypothesis before doing an experiment
- Spend time on experiment design.
  - Sample size, 16s region to amplify etc
  - Talk to a bioinformatician

  - Think about the depth of sequencing if you want to capture the less abundant taxa
  - Add negative control to account for contamination
- Thoughtful data analysis is critical for successful identification of microbes

“If you torture the data long enough, it will confess.”- Ronald Coase, Economist

Chapter 3

Primer Removal

3.1 CutAdapt

The first step that everyone performs before doing an analysis is data cleaning. Data cleaning can mean multiple things in this context: primer removal, quality trimming, removing very short sequences etc. Remember inconsistent and incorrect data leads to false conclusions. In short, garbage in, garbage out applies to all data.

The first step that we will do with amplicon data is check which 16s rRNA gene region was sequenced, find the primer that were used and remove them. For this purpose there are a few software available like, cutAdapt, Trimmomatic, skewer. For the purpose of this workshop we will use cutAdapt.

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from your high-throughput sequencing reads. It also alows you to perform quality trimming on your reads.

3.1.1 Primer removal for paired end reads

Here, the input reads are in reads.1.fastq and reads.2.fastq, and the result will be written to out.1.fastq and out.2.fastq.

In paired-end mode, the options -a, -b, -g and -u that also exist in single-end mode are applied to the forward reads only. To modify the reverse read, these options have uppercase versions -A, -B, -G and -U that work just like their counterparts. In the example above, ADAPTER_FWD will therefore be trimmed from the forward reads and ADAPTER_REV from the reverse reads.

Single-end/R1 option	Corresponding option for R2
–adapter, -a	-A
–front, -g	-G
–anywhere, -b	-B
–cut, -u	-U
–output, -o	–paired-output, -p

In paired-end mode, Cutadapt checks whether the input files are properly paired. An error is raised if one of the files contains more reads than the other or if the read names in the two files do not match. The read name comparison ignores a trailing /1 or /2 to allow processing some old Illumina paired-end files.

Cutadapt supports compressed input and output files. Whether an input file needs to be decompressed or an output file needs to be compressed is detected automatically by inspecting the file name: For example, if it ends in .gz, then gzip compression is assumed

3.1.2 Quality trimming

-q 15,10

3.1.3 Call cutAdapt from R

Follow this link to see how you can call cutAdapt from within R https://benjjneb.github.io/dada2/ITS_workflow.html

Chapter 4

DADA2

4.1 DADA2 pipeline (v1.2)

From now on, we will be working on the DADA2 package version 1.12. DADA2 has great documentation and an excellent tutorial online.Please go to the following link http://benjjneb.github.io/dada2/tutorial.html

All the notes from now on are my additional comments to help you follow the pipeline:

4.1.1 Data for the tutorial

The data to use for the tutorial can be downloaded from here

4.1.2 Getting ready (load packages and get file list)

Functions that we will be using here are :

- list.files()
- sort()
- strsplit()
- basename()
- sapply()

4.1.3 Inspect read quality profiles

Read in files

```r
library(dada2);
library(limma)
path <- "MiSeq_SOP/"
list.files(path)
```

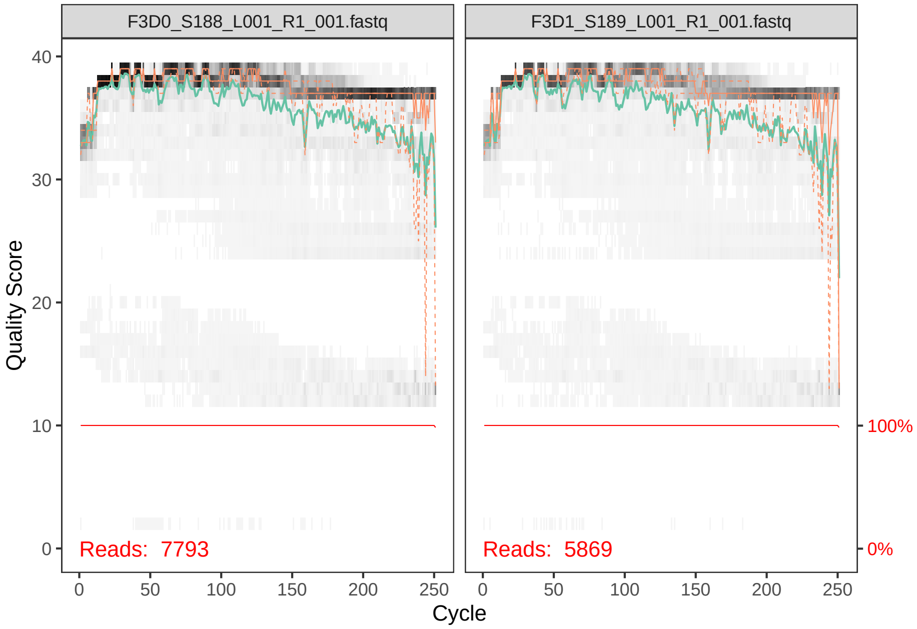
```
##  [1] "F3D0_S188_L001_R1_001.fastq"   "F3D0_S188_L001_R2_001.fastq"  
##  [3] "F3D1_S189_L001_R1_001.fastq"   "F3D1_S189_L001_R2_001.fastq"  
##  [5] "F3D141_S207_L001_R1_001.fastq" "F3D141_S207_L001_R2_001.fastq"
##  [7] "F3D142_S208_L001_R1_001.fastq" "F3D142_S208_L001_R2_001.fastq"
##  [9] "F3D143_S209_L001_R1_001.fastq" "F3D143_S209_L001_R2_001.fastq"
## [11] "F3D144_S210_L001_R1_001.fastq" "F3D144_S210_L001_R2_001.fastq"
## [13] "F3D145_S211_L001_R1_001.fastq" "F3D145_S211_L001_R2_001.fastq"
## [15] "F3D146_S212_L001_R1_001.fastq" "F3D146_S212_L001_R2_001.fastq"
## [17] "F3D147_S213_L001_R1_001.fastq" "F3D147_S213_L001_R2_001.fastq"
## [19] "F3D148_S214_L001_R1_001.fastq" "F3D148_S214_L001_R2_001.fastq"
## [21] "F3D149_S215_L001_R1_001.fastq" "F3D149_S215_L001_R2_001.fastq"
## [23] "F3D150_S216_L001_R1_001.fastq" "F3D150_S216_L001_R2_001.fastq"
## [25] "F3D2_S190_L001_R1_001.fastq"   "F3D2_S190_L001_R2_001.fastq"  
## [27] "F3D3_S191_L001_R1_001.fastq"   "F3D3_S191_L001_R2_001.fastq"  
## [29] "F3D5_S193_L001_R1_001.fastq"   "F3D5_S193_L001_R2_001.fastq"  
## [31] "F3D6_S194_L001_R1_001.fastq"   "F3D6_S194_L001_R2_001.fastq"  
## [33] "F3D7_S195_L001_R1_001.fastq"   "F3D7_S195_L001_R2_001.fastq"  
## [35] "F3D8_S196_L001_R1_001.fastq"   "F3D8_S196_L001_R2_001.fastq"  
## [37] "F3D9_S197_L001_R1_001.fastq"   "F3D9_S197_L001_R2_001.fastq"  
## [39] "HMP_MOCK.v35.fasta"            "Mock_S280_L001_R1_001.fastq"  
## [41] "Mock_S280_L001_R2_001.fastq"   "mouse.dpw.metadata"           
## [43] "mouse.time.design"             "stability.batch"              
## [45] "stability.files"
```

```r
# Forward and reverse fastq filenames have format: SAMPLENAME_R1_001.fastq and SAMPLEN
fnFs <- sort(list.files(path, pattern="_R1_001.fastq", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001.fastq", full.names = TRUE))
# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq


tmp<-strsplit(basename(fnFs), "_")


# As tmp is a list of vectors, we will have to use sapply function to get the first po

samples.names<-sapply(tmp,'[',1)
```

4.1.4 Lets make plots to look at the quality for multiple files

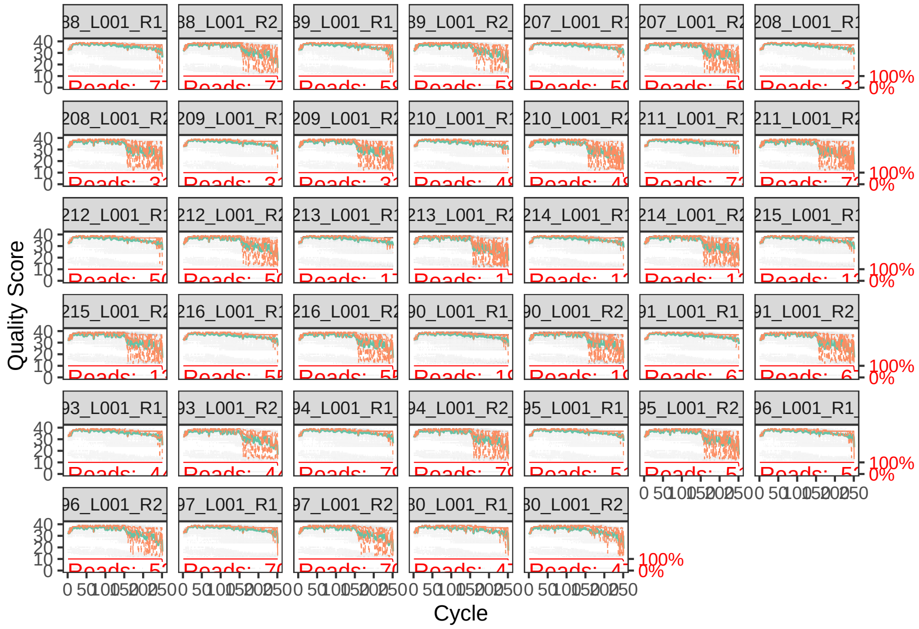
```
plotQualityProfile(fnFs[1:2])
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```



```
joint_fnames<-c(rbind(fnFs,fnRs))

plotQualityProfile(joint_fnames)
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```



```
num=ceiling(length(joint_fnames)/6)

for(i in 1: num)

{
  plotQualityProfile(joint_fnames[i:i+5])


}
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

Chapter 5

Day Two

5.1 Main concepts to be discussed:

- Finish dada2 pipeline
- Assign Taxonomy
- Intro to Phyolseq package
- Create a Phylum level bar plots
- Alpha diversity plots
- Beta diversity plots

Once we are finished using the dada2 package, we will have a sequence table and taxonomy table.

Lets look at the metadata files we have to get more information about these samples.

```
library(ggplot2)
library(dplyr)
time_info<-read.delim("~/projects/16sanalysis_workshop/workshoptutorial/MiSeq_SOP/mouse.time.desi
dpw_info<-read.delim("~/projects/16sanalysis_workshop/workshoptutorial/MiSeq_SOP/mouse.dpw.metada

sample_info<-left_join(time_info,dpw_info,by="group")

rownames(sample_info)<- sample_info$group

sample_info
```

```
##         group  time dpw
## F3D0     F3D0 Early   0
## F3D1     F3D1 Early   1
## F3D141 F3D141  Late 141
## F3D142 F3D142  Late 142
## F3D143 F3D143  Late 143
## F3D144 F3D144  Late 144
## F3D145 F3D145  Late 145
## F3D146 F3D146  Late 146
## F3D147 F3D147  Late 147
## F3D148 F3D148  Late 148
## F3D149 F3D149  Late 149
## F3D150 F3D150  Late 150
## F3D2     F3D2 Early   2
## F3D3     F3D3 Early   3
## F3D5     F3D5 Early   5
## F3D6     F3D6 Early   6
## F3D7     F3D7 Early   7
## F3D8     F3D8 Early   8
## F3D9     F3D9 Early   9
```

Lets make a phyloseq object

Now that we have sample_info lets try to make a phyloseq object out of this

```r
library(phyloseq)
taxa<-readRDS("~/projects/16sanalysis_workshop/workshoptutorial/output/taxa.rds")
seq<-readRDS("~/projects/16sanalysis_workshop/workshoptutorial/output/seq.rds")

ps <- phyloseq(otu_table(seq,taxa_are_rows = F),
               sample_data(sample_info),
               tax_table(taxa))

ps
```

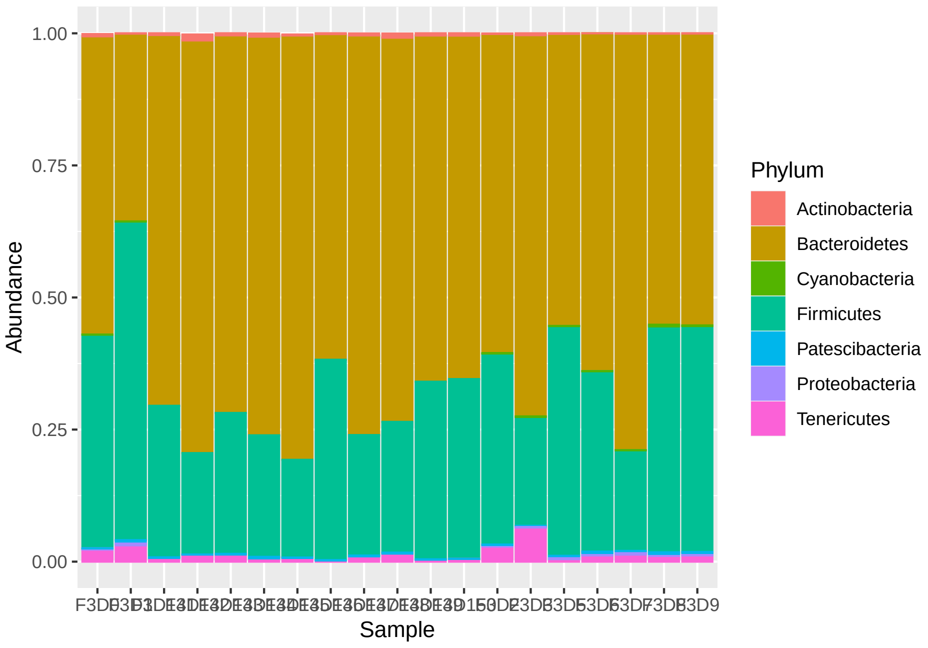
```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 234 taxa and 19 samples ]
## sample_data() Sample Data:       [ 19 samples by 3 sample variables ]
## tax_table()   Taxonomy Table:    [ 234 taxa by 7 taxonomic ranks ]
```

You can see that this object has an OTU table(ASV table), sample data and tax_table. You can use functions tax_table(), sample_data() and otu_table() to access the data.

Take a look at:

- subset_samples()
- subset_taxa()
- tax_glom()
- sample_sums()
- prune_samples()
- transform_sample_counts()
- psmelt()

```
ps2<-tax_glom(ps,taxrank = "Phylum")
ps2 = transform_sample_counts(ps2, function(x) x/sum(x))
pmelt<-psmelt(ps2) %>% arrange(desc(Abundance))

cutoff<-0.005
pmelt_filt<-pmelt %>% group_by(Phylum,time) %>% filter(sum(Abundance) >cutoff)

ggplot(pmelt_filt,aes(x=Sample,y=Abundance,color=Phylum,fill=Phylum)) +geom_bar(stat = "identity'
```



```
pmelt_filt %>% group_by(time,Phylum) %>% summarise(mean_Abundance=mean(Abundance)) %>% ggplot(.,a
```

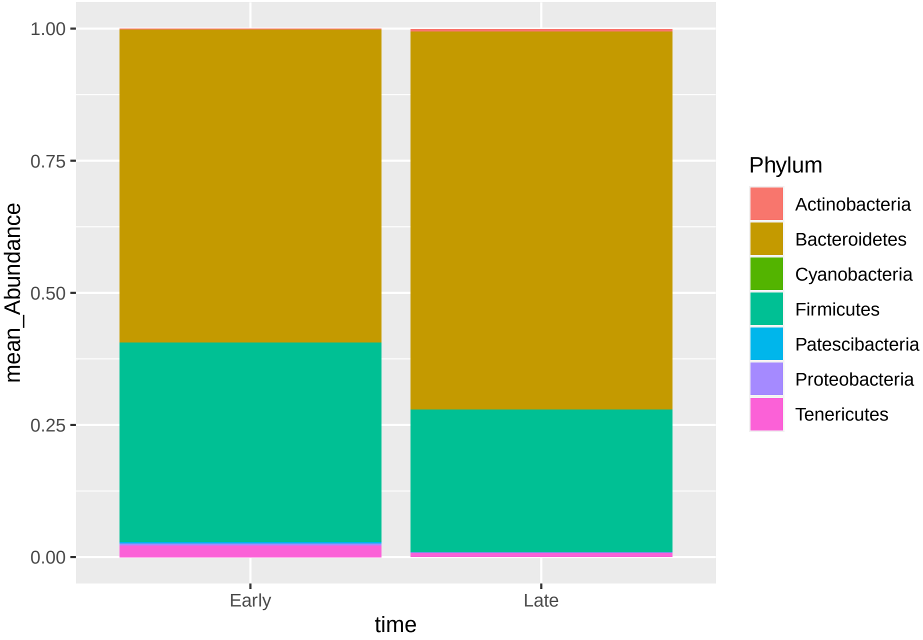
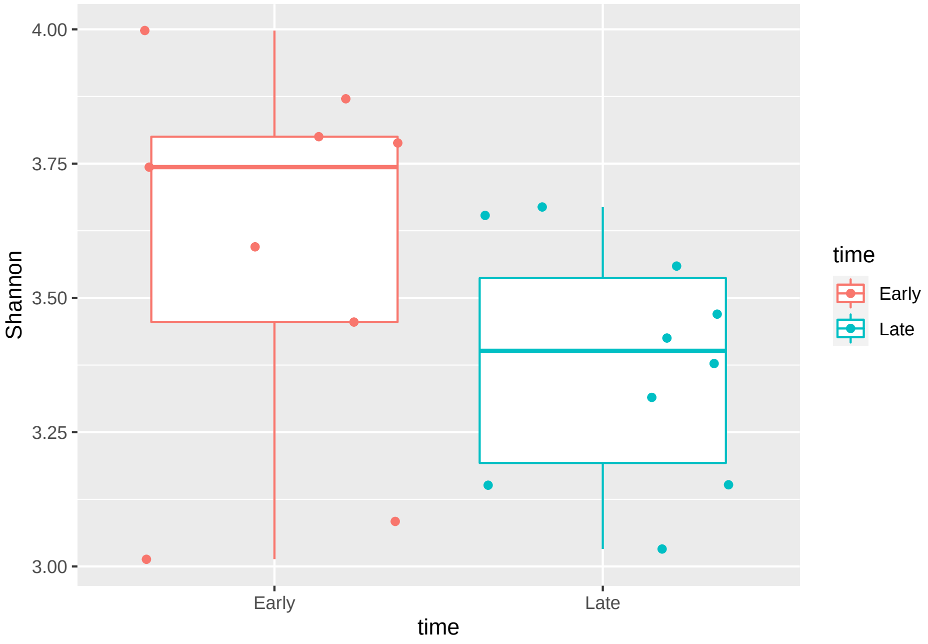


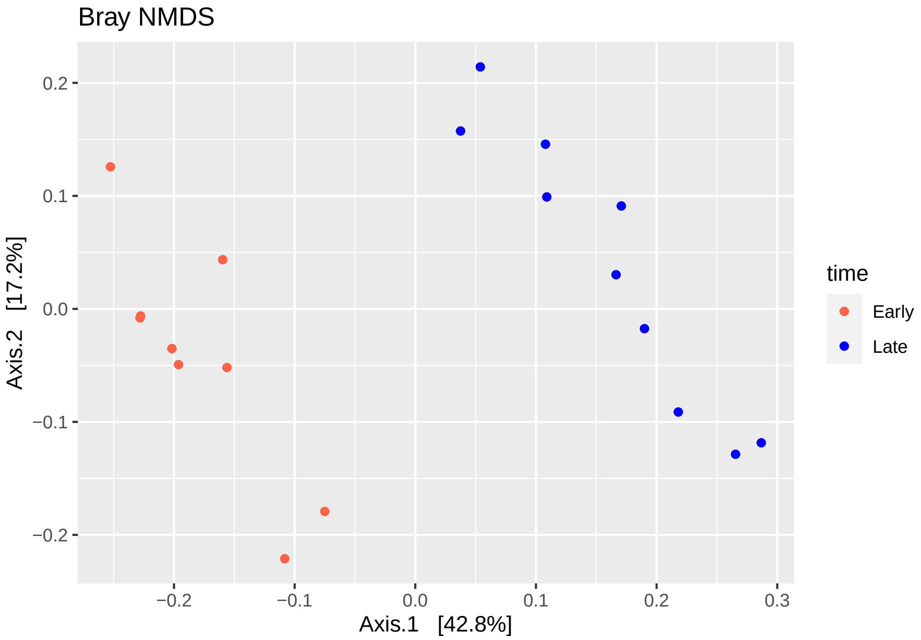
Figure out how to change the colors to custom colors!!

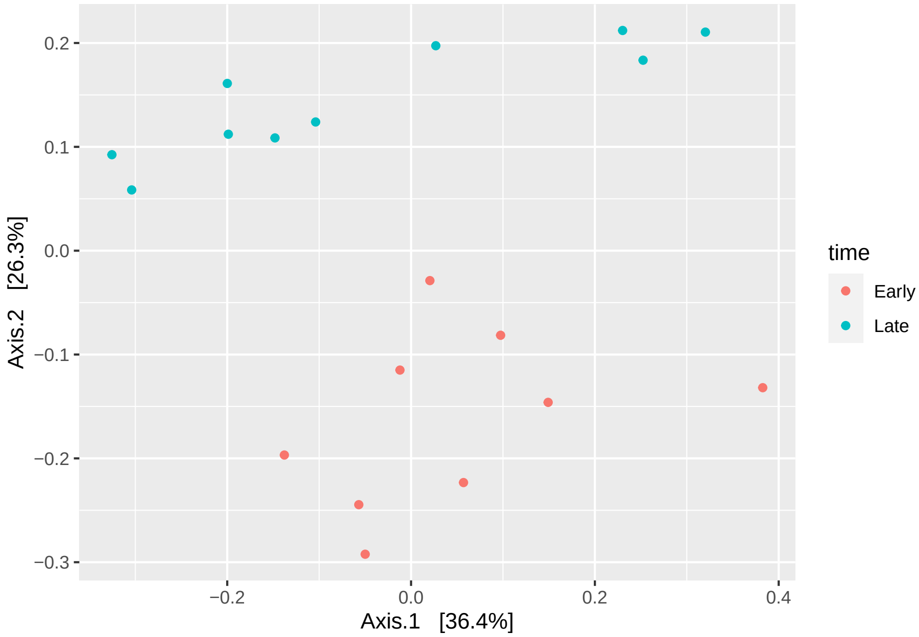
```
pmelt_filt %>% group_by(time,Phylum) %>% summarise(mean=mean(Abundance))
```

```
## # A tibble: 12 x 3
## # Groups:   time [2]
##    time  Phylum              mean
##    <fct> <fct>              <dbl>
##  1 Early Actinobacteria  0.00133
##  2 Early Bacteroidetes   0.592
##  3 Early Cyanobacteria   0.000594
##  4 Early Firmicutes      0.378
##  5 Early Patescibacteria 0.00285
##  6 Early Proteobacteria  0.00331
##  7 Early Tenericutes     0.0218
##  8 Late  Actinobacteria  0.00472
##  9 Late  Bacteroidetes   0.716
## 10 Late  Firmicutes      0.269
## 11 Late  Patescibacteria 0.000993
## 12 Late  Tenericutes     0.00845
```

5.2 Alpha & beta Diversity






0.2
0.1
0.0
−0.1
−0.2
−0.3
Axis.2 [26.3%]
−0.2
0.0
0.2
0.4
Axis.1 [36.4%]
time
Early
Late

Chapter 6

Summary

Chapter 7

What Next?