Microbiome Data Analysis

Kaiden Liu

26/05/2021

Phyloseq is good for storing complex phylogenetic sequencing data

Operational Taxonomic Unit (OTU): groups of closely related individuals

1) OTU table

- row: taxa/OTU
  - groups of closely related individuals
- column: sample
- value: the number of reads

```
set.seed(526)
otumat = matrix(sample(0:100, 100, replace = TRUE,prob=c(0.5,rep(0.5/100,100))), nrow = 10, ncol = 10)
rownames(otumat) <- paste0("OTU", 1:nrow(otumat))
colnames(otumat) <- paste0("Sample", 1:ncol(otumat))
otumat
```

```
##       Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 Sample7 Sample8 Sample9
## OTU1        0       0      54      18       0       0       0       0      91
## OTU2        0      69      71      65      33       0       0      98      17
## OTU3        0      90      34      80      47       0      34       0      73
## OTU4        4      25       3       0      85      79       0      26       0
## OTU5        1       4      92       0      35       0      79       0      37
## OTU6       51       0       0       0      29      73      11       0      46
## OTU7        0      63       0      64       0       0      25       0       8
## OTU8        0       9       0      97      37       0      65       0      59
## OTU9       75       0       0       0      81      76      95       0      56
## OTU10      36      25       0       2       0       0      76      98      80
##       Sample10
## OTU1        71
## OTU2        57
## OTU3        82
## OTU4        36
## OTU5        67
## OTU6         9
## OTU7         0
## OTU8        74
## OTU9        85
## OTU10       85
```

2) taxonomy table

Table of names of the taxonomic rank of the data

- row: OTU/taxonomy
- column: taxonomy rank(levels)
- value: names of the taxonomy (family)

```
taxmat = matrix(sample(letters, 70, replace = TRUE),
                nrow = nrow(otumat), ncol = 7)
rownames(taxmat) <- rownames(otumat)
colnames(taxmat) <- c("Domain", "Phylum", "Class",
                      "Order", "Family", "Genus", "Species")
taxmat
```

```
##       Domain Phylum Class Order Family Genus Species
## OTU1  "f"    "z"    "t"   "b"   "j"    "l"   "u"
## OTU2  "s"    "r"    "v"   "w"   "h"    "e"   "q"
## OTU3  "u"    "j"    "n"   "h"   "z"    "r"   "f"
## OTU4  "e"    "r"    "p"   "s"   "s"    "n"   "k"
## OTU5  "s"    "d"    "t"   "o"   "o"    "y"   "w"
## OTU6  "r"    "g"    "m"   "z"   "f"    "b"   "z"
## OTU7  "i"    "o"    "r"   "r"   "x"    "s"   "e"
## OTU8  "u"    "b"    "r"   "h"   "w"    "m"   "y"
## OTU9  "c"    "x"    "l"   "h"   "o"    "s"   "v"
```

2.5) Creating a phyloseq object

```
library("phyloseq")
OTU = otu_table(otumat, taxa_are_rows = TRUE)
TAX = tax_table(taxmat)

physeq = phyloseq(OTU, TAX)
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 10 taxa and 10 samples ]
## tax_table()   Taxonomy Table:    [ 10 taxa by 7 taxonomic ranks ]
```

3) sample variables

- Location: Location of where the sample is collected(eg: Feces, Blood, Skin)
- Depth: Number of sample sequenced

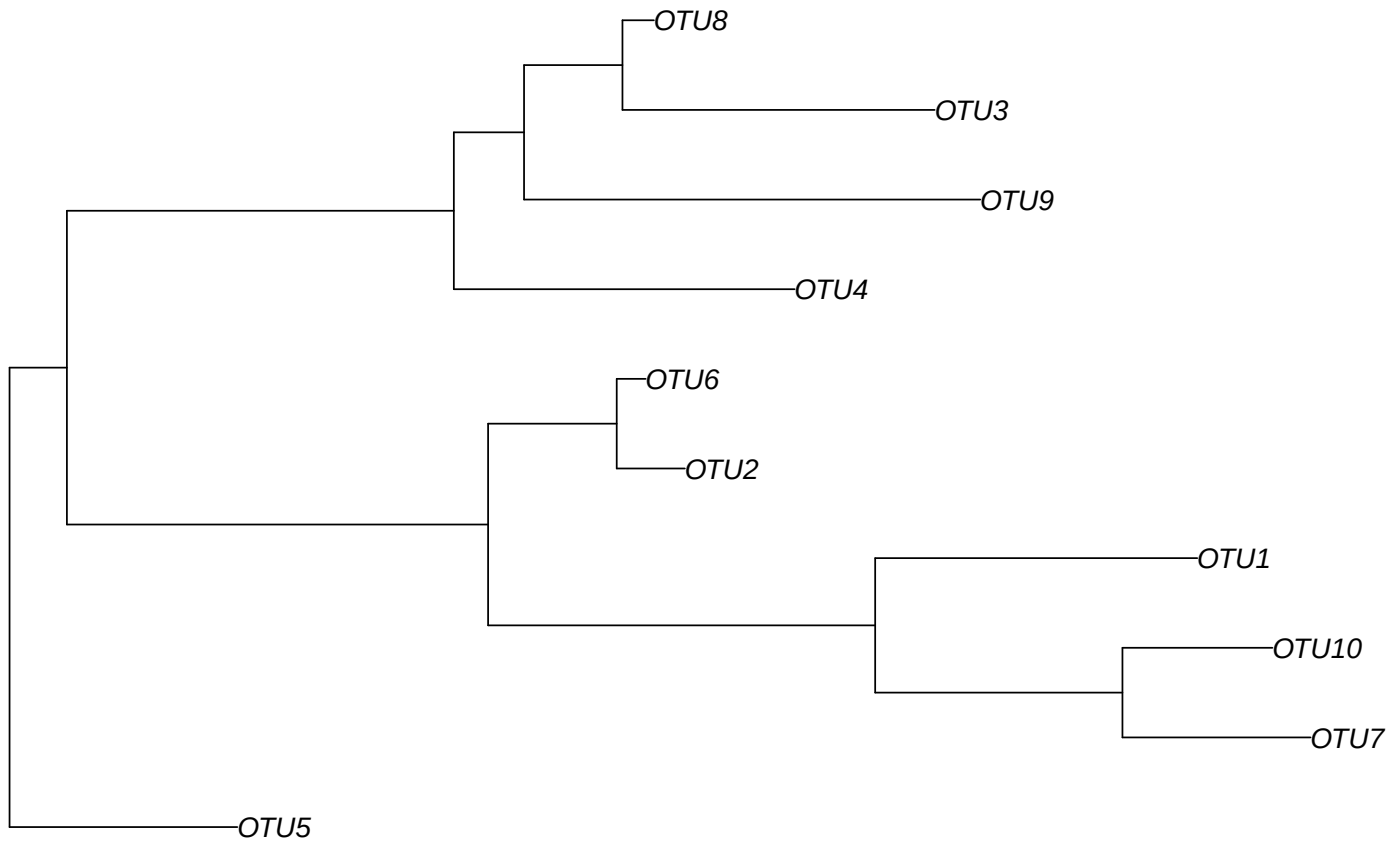
```
set.seed(999)
sampledata = sample_data(data.frame(
  Location = sample(LETTERS[1:4], size=nsamples(physeq), replace=TRUE),
  Depth = sample(50:1000, size=nsamples(physeq), replace=TRUE),
  row.names=sample_names(physeq),
  stringsAsFactors=FALSE
))
sampledata
```

```
##          Location Depth
## Sample1         C   340
## Sample2         D   328
## Sample3         A   256
## Sample4         C   630
## Sample5         A   440
## Sample6         B   727
## Sample7         A   623
## Sample8         B   452
## Sample9         B   540
## Sample10        B   131
```

4) phylogenetic Tree

Shows how the different taxa are related.

```
library("ape")
random_tree = rtree(ntaxa(physeq), rooted=TRUE,
                    tip.label=taxa_names(physeq))
plot(random_tree)
```

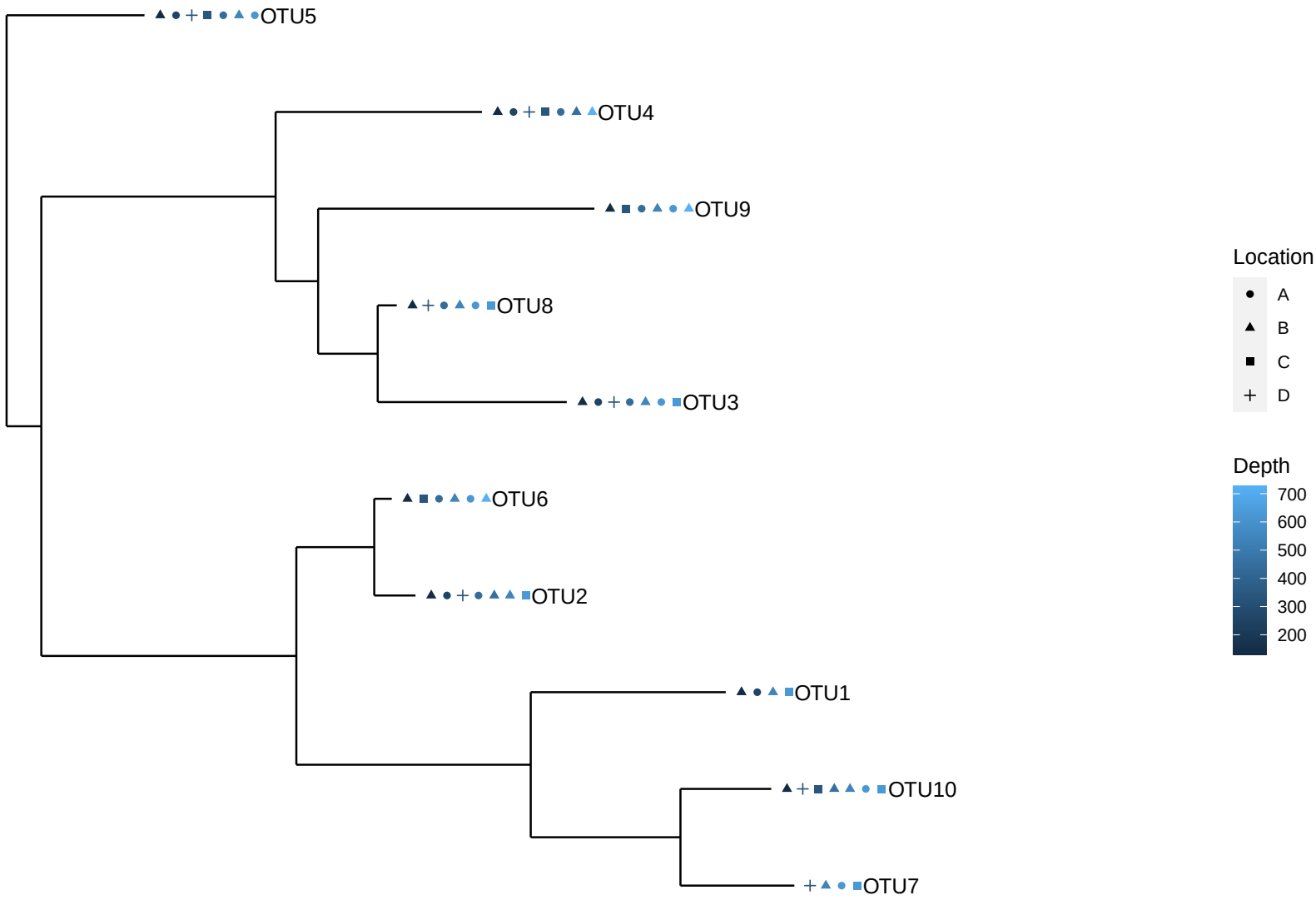


4.5) Complete the phyloseq object by merging the two new “tables”

```
physeq1 = merge_phyloseq(physeq, sampledata, random_tree)
physeq1
```

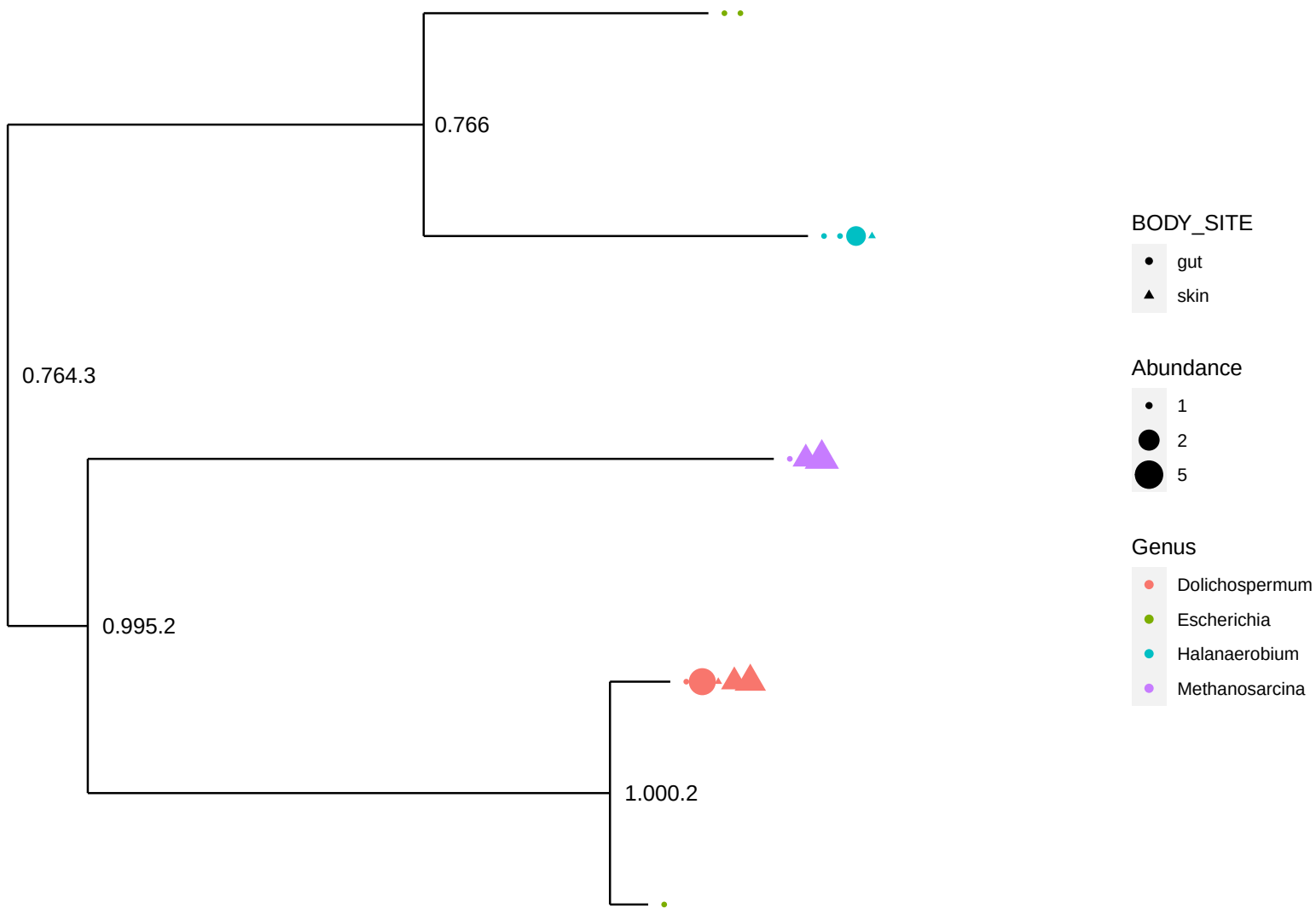
```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 10 taxa and 10 samples ]
## sample_data() Sample Data:       [ 10 samples by 2 sample variables ]
## tax_table()   Taxonomy Table:    [ 10 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 10 tips and 9 internal nodes ]
```

We can also display the new phylogenetic tree with our new phyloseq object

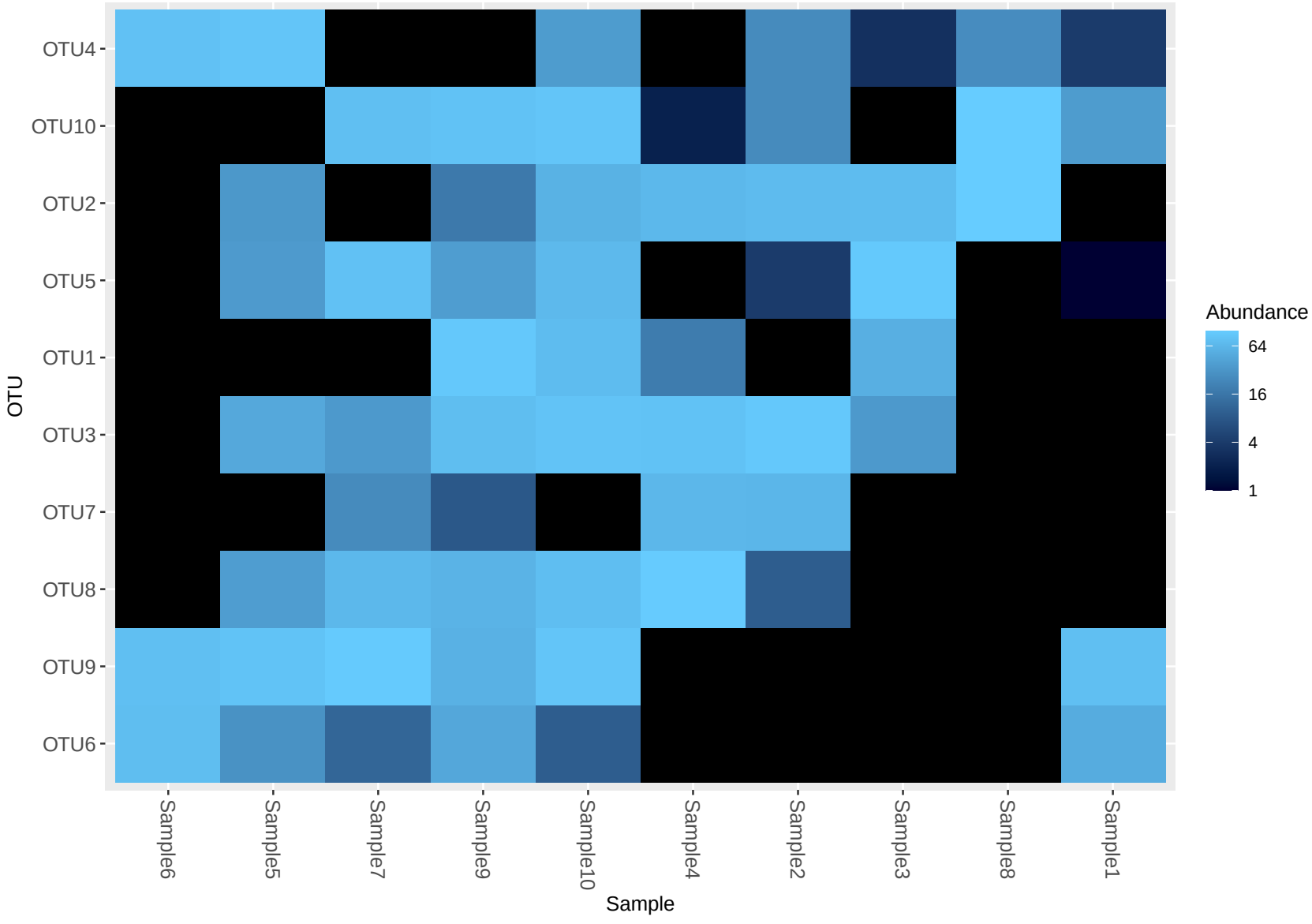


Value is the measure of support of the node, calculated by bootstrapping

1) We draw samples from the data with replacement for a specified size
2) we train a model with the samples, and fit the model to the data again.
3) calculate the “accuracy” of the result



```
plot_heatmap(physeq1)
```



5) Reference Seq

This table would give us more details about our data

```
refseq(myData)
```

```
## DNAStringSet object of length 5:
##     width seq
## [1]   334 AACGTAGGTCACAAGCGTTGTCC...CCCTTCCGTGCCGGAGTTA
## [2]   465 TACGTAGGGAGCAAGCGTTATCC...GAACCTTACCAGGGCTTGA
## [3]   249 TACGTAGGGGGCAAGCGTTATCC...TGAGGCTCGAAAGCGTGGG
## [4]   453 TACGTATGGTGCAAGCGTTATCC...TCGAAGCAACGCGAAGAAC
## [5]   178 AACGTAGGGTGCAAGCGTTGTCC...GGTGGAATGCGTAGATATC
```

ex) DNAStringSet, RNAStringSet, and AAStringSet from Biostrings package

Workflow for Microbiome Data Analysis

FASTQ format file contains biological sequence and the corresponding quality score.

With DADA2, we want to convert this FASTQ file to an OTU table.

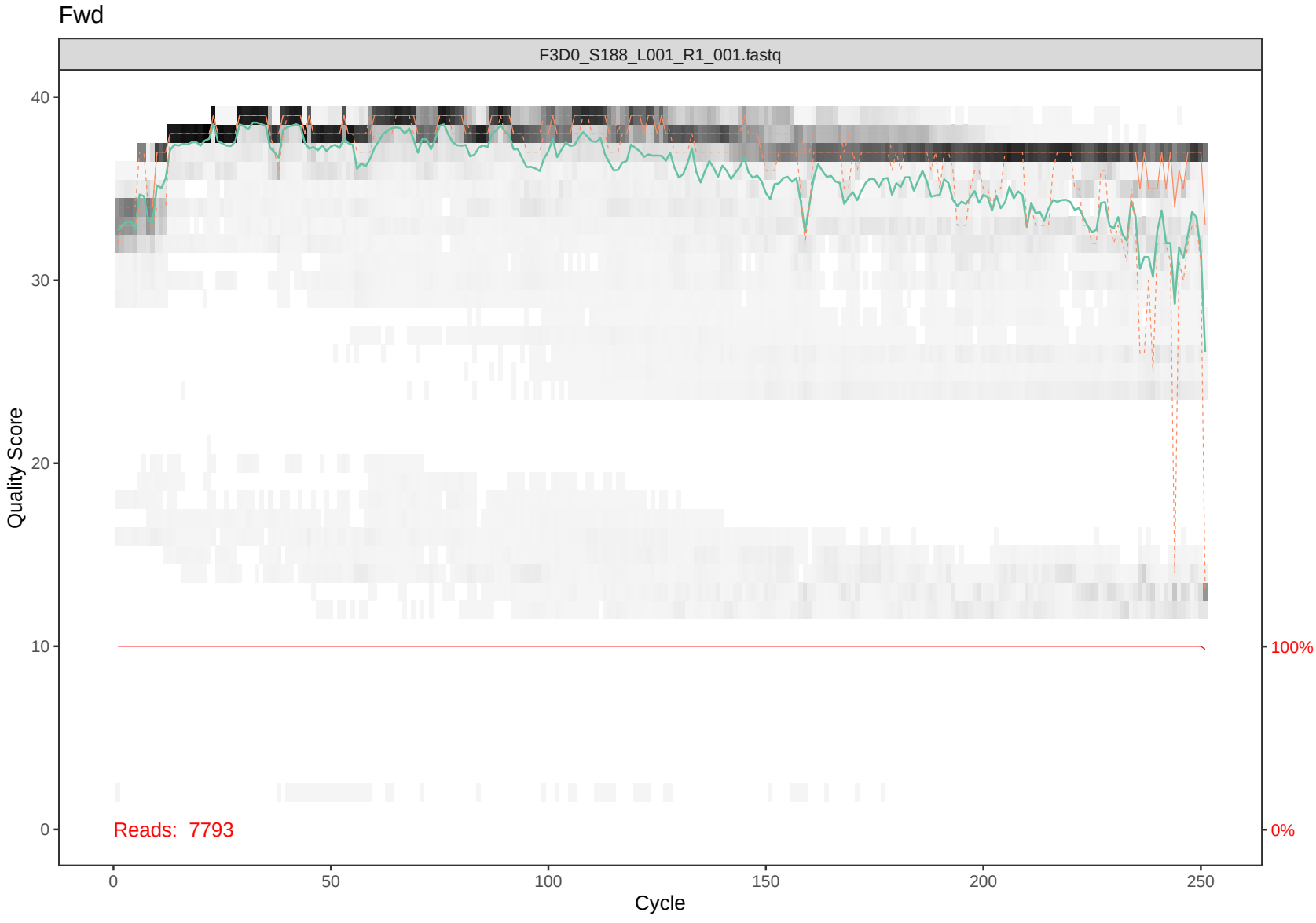
- We want to replace OTUs with ASVs(Amplicon sequence variant)
  - Higher resolution
  - Higher Accuracy
  - Linear Computation Time

But before we convert...

Trim

plotQualityProfile:

- inspect fastq file quality
- underlying heatmap shows frequency of each score at each position
- green –> mean
- orange –> quantile(dash is 25th quantile and 75th quantile)



Idea: We want to truncate the read based on this plot so that the quality scores stay near the top

Data Manipulation

Taxonomic filtering

- We want to remove the data that are rare in our taxonomy table, because they are not likely to be true in nature

Prevalence Filtering

prevalence: fraction of total samples in which a taxa is observed

- Identify and filter outlier

Agglomerating taxa

When the species are categorized too deep and starting to be redundant, we want to group the data back together by how closely related they are in terms of taxa.

- Figure 4

Abundance value transformation

The challenge of different library sizes among the samples can be accounted by transforming the count data to proportions or relative abundances.

- Figure 5

Ordination plots

principal coordinates analysis (PCoA)

We want to map our data from a high dimension to a low dimension, so we can visualize the similarities of data(by how close they are).

- First axis $>$ Second Axis (variability)

Figure 10

Distance matrix

We can summarize the relationship between points with distance

1) Bray-Curtis dissimilarity
   - based on counts
   - ranges from 0 to 1
     - 0 means the two samples are from the same group
     - 1 means the two samples are different
   - not a distance
2) Weighted UniFrac Distance
   - based on the phylogenetic distance
   - edges of the phylogenetic tree are weighted proportional to the abundance of the taxa
   - is a distance

PCA on ranks

- represent abundances by ranks
  - taxa with smallest in sample maps to 1, second smallest sample maps to 2
- Good for data with heavy-tailed
- Threshold for small abundance (absent data) –> large difference in rank

Figure 15

Network analysis

Create a network by thresholding a distance matrix

Minimum spanning tree

We assign weights to all the edges of the network.
We want to find a way to connect all dots together without any cycles, and with the smallest sum of edge weights.

- Nearest neighbors

Picture: https://en.wikipedia.org/wiki/Minimum_spanning_tree#/media/File:Minimum_spanning_tree.svg

Figure 23

pure edges: edges that connects two nodes of the same level

Graph-based two-sample tests

Null hypothesis: two samples come from the same distribution

test statistics: number of pure edges

histogram: permute sample type randomly and construct a histogram

- if number of pure edges is more than the test statistics, we reject the null hypothesis.

Supervised learning

1) Partial Least Square
2) Random Forest

```
setup_example<- (c("phyloseq", "ggplot2", "plyr", "dplyr", "reshape2",
                   "ade4", "ggrepel", 'randomForest', 'testingfail'))
lapply(setup_example,require,character.only=T)
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'testingfail'
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
##
## [[7]]
## [1] TRUE
##
## [[8]]
## [1] TRUE
##
## [[9]]
## [1] FALSE
```