Microbial community analysis in R

Marko Suokas

#### Libraries

library(phyloseq);packageVersion("phyloseq")

[1] '1.48.0'

library(tidyverse);packageVersion("tidyverse")

[1] '2.0.0'

library(kableExtra);packageVersion("kableExtra")

[1] '1.4.0'

library(patchwork);packageVersion("patchwork")

[1] '1.2.0'

library(mia);packageVersion("mia")

[1] '1.12.0'

library(ggplot2);packageVersion("ggplot2")

[1] '3.5.1'

library(ggthemes);packageVersion("ggthemes")

[1] '5.1.0'

library(vegan);packageVersion("vegan")

[1] '2.6.6.1'

library(scater);packageVersion("scater")

[1] '1.32.1'

#### Import

Import tables as a tse object named dada

# Path variables  
asvfile <- "results\_set2/asvs.tsv"  
metafile <- "data/set2\_meta.tsv"  
taxafile <- "results\_set2/taxonomy.tsv"  
#Abundance data is imported from tabular txt file, rownames stored and emptied  
counts <- read\_tsv(asvfile, show\_col\_types = FALSE)  
ASV\_names <- counts$ASV\_names  
counts$ASV\_names <- NULL  
#Metadata is imported from tabular txt file, rownames stored and emptied  
samples <- read\_tsv(metafile, show\_col\_types = FALSE)  
sampleid <- samples$sampleid  
samples$sampleid <- NULL  
#Taxonomy table is imported tabular txt file, rownames stored and emptied  
taxonomy <-read\_tsv(taxafile, show\_col\_types = FALSE)  
taxanames <- taxonomy$ASV\_names  
taxonomy$ASV\_names <- NULL  
#Abundance values should be in numeric matrix format  
counts <- as.matrix(counts)  
#And should be added to a SimpleList  
assays <- SimpleList(counts = counts)  
#colData and rowData should be in DataFrame format  
colData <- DataFrame(colData)  
rowData <- DataFrame(rowData)  
#Create a TreeSummarized Experiment object  
dada <- TreeSummarizedExperiment(assays = assays,  
 colData = samples,  
 rowData = taxonomy)  
#Add amplicon variant names as rownames  
rownames(dada) <- ASV\_names

Import tables from qiime pipelines

# vsearch97  
asvfile <- "results\_vsearch97/asvs\_set2.tsv"  
taxafile <- "results\_vsearch97/taxonomy\_set2.tsv"  
#Abundance data is imported from tabular txt file, rownames stored and emptied  
counts <- read\_tsv(asvfile, show\_col\_types = FALSE)  
ASV\_names <- counts$ASV\_names  
counts$ASV\_names <- NULL  
#Metadata is imported from tabular txt file, rownames stored and emptied  
samples <- read\_tsv(metafile, show\_col\_types = FALSE)  
sampleid <- samples$sampleid  
samples$sampleid <- NULL  
#Taxonomy table is imported tabular txt file, rownames stored and emptied  
taxonomy <-read\_tsv(taxafile, show\_col\_types = FALSE)  
taxanames <- taxonomy$ASV\_names  
taxonomy$ASV\_names <- NULL  
#Abundance values should be in numeric matrix format  
counts <- as.matrix(counts)  
#And should be added to a SimpleList  
assays <- SimpleList(counts = counts)  
#colData and rowData should be in DataFrame format  
colData <- DataFrame(colData)  
rowData <- DataFrame(rowData)  
#Create a TreeSummarized Experiment object  
vsearch97 <- TreeSummarizedExperiment(assays = assays,  
 colData = samples,  
 rowData = taxonomy)  
#Add amplicon variant names as rownames  
rownames(vsearch97) <- ASV\_names  
#vsearch99  
asvfile <- "results\_vsearch99/asvs\_set2.tsv"  
taxafile <- "results\_vsearch99/taxonomy\_set2.tsv"  
#Abundance data is imported from tabular txt file, rownames stored and emptied  
counts <- read\_tsv(asvfile, show\_col\_types = FALSE)  
ASV\_names <- counts$ASV\_names  
counts$ASV\_names <- NULL  
#Metadata is imported from tabular txt file, rownames stored and emptied  
samples <- read\_tsv(metafile, show\_col\_types = FALSE)  
sampleid <- samples$sampleid  
samples$sampleid <- NULL  
#Taxonomy table is imported tabular txt file, rownames stored and emptied  
taxonomy <-read\_tsv(taxafile, show\_col\_types = FALSE)  
taxanames <- taxonomy$ASV\_names  
taxonomy$ASV\_names <- NULL  
#Abundance values should be in numeric matrix format  
counts <- as.matrix(counts)  
#And should be added to a SimpleList  
assays <- SimpleList(counts = counts)  
#colData and rowData should be in DataFrame format  
colData <- DataFrame(colData)  
rowData <- DataFrame(rowData)  
#Create a TreeSummarized Experiment object  
vsearch99 <- TreeSummarizedExperiment(assays = assays,  
 colData = samples,  
 rowData = taxonomy)  
#Add amplicon variant names as rownames  
rownames(vsearch99) <- ASV\_names

Number of variants in each object. In data set, we treated sequences little bit differently and truncated reads prior dereplication to equal length of 1400 bp. Equal length might decrease number of variants clustering produces.

#create new dataframe  
variants <- data.frame(Dada = nrow(dada), Vsearch97 = nrow(vsearch97),  
 Vsearch99 = nrow(vsearch99))  
#table  
kable(variants, caption = "Number of variants" ) %>%  
 kable\_styling(latex\_options = c("HOLD\_position", "striped")) %>%  
 row\_spec(0, background = "teal", color = "ivory")

Number of variants

| Dada | Vsearch97 | Vsearch99 |
| --- | --- | --- |
| 933 | 2519 | 18314 |

Species level taxonomic classification is not very reliable. Thus, we will agglomerate all objects to genus level-

#agglomeration to genus level  
DADA<- mergeFeaturesByRank(dada, rank = "Genus", onRankOnly = FALSE,  
 na.rm = TRUE)  
VS97 <- agglomerateByRank(vsearch97, rank = "Genus", onRankOnly = FALSE,  
 na.rm = TRUE)  
VS99 <- agglomerateByRank(vsearch99, rank = "Genus", onRankOnly = FALSE,  
 na.rm = TRUE)  
#check number of variants  
nrow(DADA)

[1] 34

nrow(VS97)

[1] 49

nrow(VS99)

[1] 49

Following code will transform counts to relative abundances

#relabundance  
DADA <- transformAssay(DADA, assay.type = "counts",  
 method = "relabundance")  
VS97 <- transformAssay(VS97, assay.type = "counts",  
 method = "relabundance")  
VS99 <- transformAssay(VS99, assay.type = "counts",  
 method = "relabundance")

We will list 10 most abundant variants

#get top10 features  
top10\_DADA <- getTopFeatures(DADA, top = 10, method = "mean",  
 assay.type = "relabundance")  
top10\_VS97 <- getTopFeatures(VS97, top = 10, method = "mean",  
 assay.type = "relabundance")  
top10\_VS99 <- getTopFeatures(VS99, top = 10, method = "mean",  
 assay.type = "relabundance")  
#subset features based on top10 list  
DADA <- subsetFeatures(DADA, rowData(DADA)$Genus %in% top10\_DADA)  
VS97 <- subsetFeatures(VS97, rowData(VS97)$Genus %in% top10\_VS97)  
VS99 <- subsetFeatures(VS99, rowData(VS99)$Genus %in% top10\_VS99)  
#create dataframes  
df\_DADA <- data.frame(assays(DADA)$relabundance)  
df\_DADA <- df\_DADA %>% mutate(Genus = rownames(df\_DADA)) %>%  
 filter(Genus %in% top10\_DADA) %>% arrange(Genus)  
rownames(df\_DADA) <- NULL  
df\_DADA <- df\_DADA[,c(7,1,2,3,4,5,6)]  
df\_VS97 <- data.frame(assays(VS97)$relabundance)  
df\_VS97 <- df\_VS97 %>% mutate(Genus = rownames(df\_VS97)) %>%  
 filter(Genus %in% top10\_VS97) %>% arrange(Genus) %>%  
 mutate\_at("Genus", str\_replace, "Genus:", "")  
rownames(df\_VS97) <- NULL  
df\_VS97 <- df\_VS97[,c(7,1,2,3,4,5,6)]  
df\_VS99 <- data.frame(assays(VS99)$relabundance)  
df\_VS99 <- df\_VS99 %>% mutate(Genus = rownames(df\_VS99)) %>%  
 filter(Genus %in% top10\_VS99) %>% arrange(Genus) %>%  
 mutate\_at("Genus", str\_replace, "Genus:", "")  
rownames(df\_VS99) <- NULL  
df\_VS99 <- df\_VS99[,c(7,1,2,3,4,5,6)]

Table 2.

kable(df\_DADA, digits=2, caption = "Top10 microbes using Dada2") %>%  
 kable\_styling(latex\_options = c("HOLD\_position", "striped")) %>%  
 row\_spec(0, background = "teal", color = "ivory")

Top10 microbes using Dada2

| Genus | barcode07 | barcode08 | barcode09 | barcode10 | barcode11 | barcode12 |
| --- | --- | --- | --- | --- | --- | --- |
| Acidithiobacillus | 0.05 | 0.04 | 0.07 | 0.04 | 0.03 | 0.07 |
| Acidocella | 0.09 | 0.00 | 0.00 | 0.53 | 0.00 | 0.00 |
| Anaerosinus | 0.00 | 0.11 | 0.00 | 0.00 | 0.00 | 0.00 |
| Anaerospora | 0.00 | 0.00 | 0.12 | 0.00 | 0.00 | 0.00 |
| Desulfosporosinus | 0.31 | 0.48 | 0.20 | 0.21 | 0.39 | 0.53 |
| Fonticella | 0.09 | 0.00 | 0.00 | 0.00 | 0.17 | 0.00 |
| Herbinix | 0.21 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 |
| Microbacter | 0.00 | 0.11 | 0.00 | 0.21 | 0.12 | 0.00 |
| Pelosinus | 0.16 | 0.19 | 0.38 | 0.00 | 0.00 | 0.15 |
| Thiomonas | 0.00 | 0.00 | 0.00 | 0.00 | 0.18 | 0.12 |

Table 3.

kable(df\_VS97, digits=2, caption = "Top10 microbes using Vsearch 97") %>%  
 kable\_styling(latex\_options = c("HOLD\_position", "striped")) %>%  
 row\_spec(0, background = "teal", color = "ivory")

Top10 microbes using Vsearch 97

| Genus | barcode07 | barcode08 | barcode09 | barcode10 | barcode11 | barcode12 |
| --- | --- | --- | --- | --- | --- | --- |
| Acidithiobacillus | 0.06 | 0.05 | 0.08 | 0.04 | 0.03 | 0.07 |
| Acidocella | 0.07 | 0.00 | 0.00 | 0.51 | 0.00 | 0.00 |
| Anaerosinus | 0.00 | 0.11 | 0.00 | 0.00 | 0.00 | 0.00 |
| Desulfosporosinus | 0.27 | 0.40 | 0.21 | 0.20 | 0.35 | 0.58 |
| Fonticella | 0.09 | 0.00 | 0.00 | 0.00 | 0.15 | 0.00 |
| Herbinix | 0.22 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 |
| Microbacter | 0.00 | 0.13 | 0.00 | 0.25 | 0.14 | 0.00 |
| Mobilitalea | 0.01 | 0.00 | 0.00 | 0.00 | 0.11 | 0.00 |
| Pelosinus | 0.18 | 0.24 | 0.41 | 0.00 | 0.00 | 0.11 |
| Thiomonas | 0.00 | 0.00 | 0.00 | 0.00 | 0.18 | 0.12 |

Table 4.

kable(df\_VS99, digits=2, caption = "Top10 microbes using Vsearch 99") %>%  
 kable\_styling(latex\_options = c("HOLD\_position", "striped")) %>%  
 row\_spec(0, background = "teal", color = "ivory")

Top10 microbes using Vsearch 99

| Genus | barcode07 | barcode08 | barcode09 | barcode10 | barcode11 | barcode12 |
| --- | --- | --- | --- | --- | --- | --- |
| Acidithiobacillus | 0.05 | 0.04 | 0.08 | 0.04 | 0.03 | 0.07 |
| Acidocella | 0.09 | 0.00 | 0.00 | 0.54 | 0.00 | 0.00 |
| Anaerospora | 0.00 | 0.00 | 0.11 | 0.00 | 0.00 | 0.00 |
| Desulfosporosinus | 0.31 | 0.41 | 0.18 | 0.19 | 0.33 | 0.57 |
| Fonticella | 0.08 | 0.00 | 0.00 | 0.00 | 0.16 | 0.00 |
| Herbinix | 0.22 | 0.02 | 0.04 | 0.00 | 0.00 | 0.00 |
| Microbacter | 0.00 | 0.12 | 0.00 | 0.23 | 0.14 | 0.00 |
| Mobilitalea | 0.01 | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 |
| Pelosinus | 0.16 | 0.25 | 0.40 | 0.00 | 0.00 | 0.11 |
| Thiomonas | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.12 |

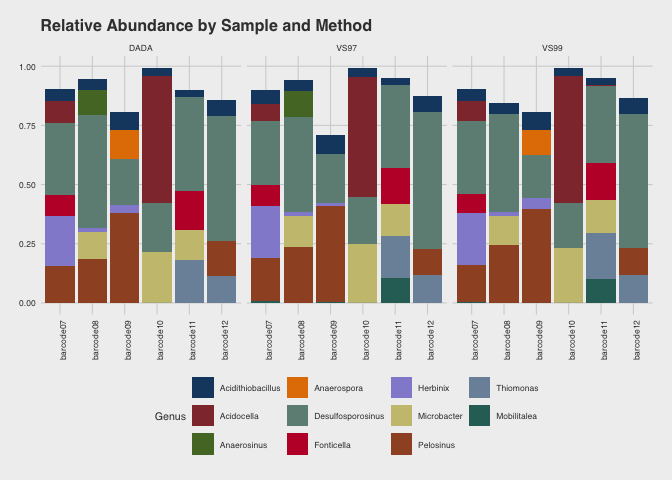
For barplot, we need to melt assay data

#transform data  
assay\_dada <- meltAssay(DADA, assay.type="relabundance")  
names(assay\_dada) <- c("Genus","Sample","Abundance")  
assay\_vs97 <- meltAssay(VS97, assay.type="relabundance")  
names(assay\_vs97) <- c("Genus", "Sample", "Abundance")  
assay\_vs99 <- meltAssay(VS99, assay.type="relabundance")  
names(assay\_vs99) <- c("Genus", "Sample", "Abundance")

Plot object

#indlude used method in each assay table  
assay\_dada$Method <- "DADA"  
assay\_vs97$Method <- "VS97"  
assay\_vs99$Method <- "VS99"  
#use bind\_rows to combine all  
combined\_data <- bind\_rows(assay\_dada, assay\_vs97, assay\_vs99)  
#create plot\_object  
abund <- ggplot(combined\_data, aes(x = Sample, y = Abundance, fill = Genus)) +  
 geom\_bar(stat = "identity", position = "stack") + facet\_wrap(~ Method) +  
 labs(title = "Relative Abundance by Sample and Method",   
 x = "Sample", y = "Relative Abundance") +  
 theme\_fivethirtyeight(base\_size=8) + scale\_fill\_stata() + theme(axis.text.x = element\_text(angle = 90))

abund



Alpha diversity is compared by calculating Shannon index for each sample and method

#dada  
dada <- transformAssay(dada, assay.type = "counts", method = "relabundance",  
 name = "relabundance")  
dada <- estimateDiversity(dada, assay.type="relabundance", index="shannon")  
#vsearch97  
vsearch97 <- transformAssay(vsearch97, assay.type = "counts", method = "relabundance",  
 name = "relabundance")  
vsearch97 <- estimateDiversity(vsearch97, assay.type="counts", index="shannon")  
#vsearch99  
vsearch99 <- transformAssay(vsearch99, assay.type = "counts", method = "relabundance",  
 name = "relabundance")  
vsearch99 <- estimateDiversity(vsearch99, assay.type="counts",index="shannon")  
#combine  
alpha <- data.frame(dada = colData(dada)$shannon, vsearch97 = colData(vsearch97)$shannon, vsearch99 = colData(vsearch99)$shannon)

Table 5.

#table  
kable(alpha, digits=2, caption = "Shannon index using different pipelines") %>%  
 kable\_styling(latex\_options = c("HOLD\_position", "striped")) %>%  
 row\_spec(0, background = "teal", color = "ivory")

Shannon index using different pipelines

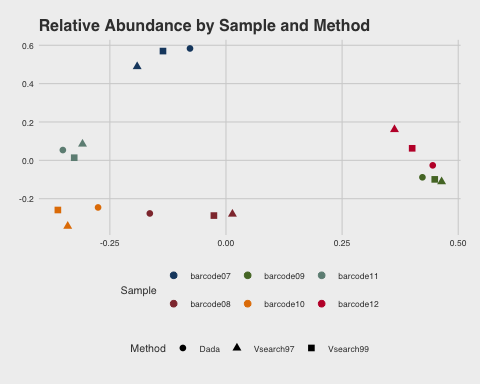
|  | dada | vsearch97 | vsearch99 |
| --- | --- | --- | --- |
| barcode07 | 3.91 | 3.19 | 4.12 |
| barcode08 | 4.31 | 3.31 | 4.04 |
| barcode09 | 4.05 | 2.45 | 3.72 |
| barcode10 | 2.58 | 1.87 | 2.17 |
| barcode11 | 3.76 | 2.70 | 3.00 |
| barcode12 | 4.05 | 2.47 | 4.57 |

Beta diversity is compared using bray-curtis dissimilarity

#create bray-curtis distance matrix  
dada <- runMDS(dada, FUN = vegan::vegdist, method = "bray",  
 name="PCoA\_BC", exprs\_values = "relabundance")  
dada\_bray <- plotReducedDim(dada, "PCoA\_BC")  
#create dataframe for plot  
bray\_dada\_df <- data.frame(pcoa1 = dada\_bray$data[,1],  
 pcoa2 = dada\_bray$data[,2],  
 Sample = colData(dada)$Sampleid)  
vsearch99 <- runMDS(vsearch99, FUN = vegan::vegdist, method = "bray",  
 name="PCoA\_BC", exprs\_values = "relabundance")  
vsearch99\_bray <- plotReducedDim(vsearch99, "PCoA\_BC")  
#create dataframe for plot  
bray\_vsearch99\_df <- data.frame(pcoa1 = vsearch99\_bray$data[,1],  
 pcoa2 = vsearch99\_bray$data[,2],  
 Sample = colData(vsearch99)$Sampleid)  
vsearch97 <- runMDS(vsearch97, FUN = vegan::vegdist, method = "bray",  
 name="PCoA\_BC", exprs\_values = "relabundance")  
vsearch97\_bray <- plotReducedDim(vsearch97, "PCoA\_BC")  
#create dataframe for plot  
bray\_vsearch97\_df <- data.frame(pcoa1 = vsearch97\_bray$data[,1],  
 pcoa2 = vsearch97\_bray$data[,2],  
 Sample = colData(vsearch97)$Sampleid)  
#add method for each dataframe and combine  
bray\_dada\_df$Method <- "Dada"  
bray\_vsearch99\_df$Method <- "Vsearch99"  
bray\_vsearch97\_df$Method <- "Vsearch97"  
combined\_pcoa <- bind\_rows(bray\_dada\_df, bray\_vsearch99\_df, bray\_vsearch97\_df)  
#create plot object  
plot\_pcoa <- ggplot(data = combined\_pcoa, aes(x=pcoa1, y=pcoa2,  
 color = Sample, shape = Method)) +  
 labs(title = "Relative Abundance by Sample and Method",   
 x = "pcoa1", y = "pcoa2") + geom\_point(size=2) +  
 theme\_fivethirtyeight(base\_size=8) + scale\_color\_stata()

Pcoa plot

plot\_pcoa



#### Observations

Overall results of the samples are highly similar. However, it should be noted that diversity in analysed samples is lower than in many environmental or clinical samples.

Clustering methods produce high number of variants compared to denoising, but findings show that core microbiome remains unchanged. Ten most common variants are almost identical. Surprisingly, denoising has slightly higher shannon index value in 4 out of 6 samples. In bray-curtis dissimilarity analysis, only one out of six samples is somewhat different when using denoising.

Currently, it is probably safest to to use vsearch clustering for long amplicons until we have more data on how denoiser performs on larger data sets. Recommended identity level is 99 %.