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")
```

Table 1: 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")
```

Table 2: 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")
```

Table 3: 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")
```

Table 4: 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

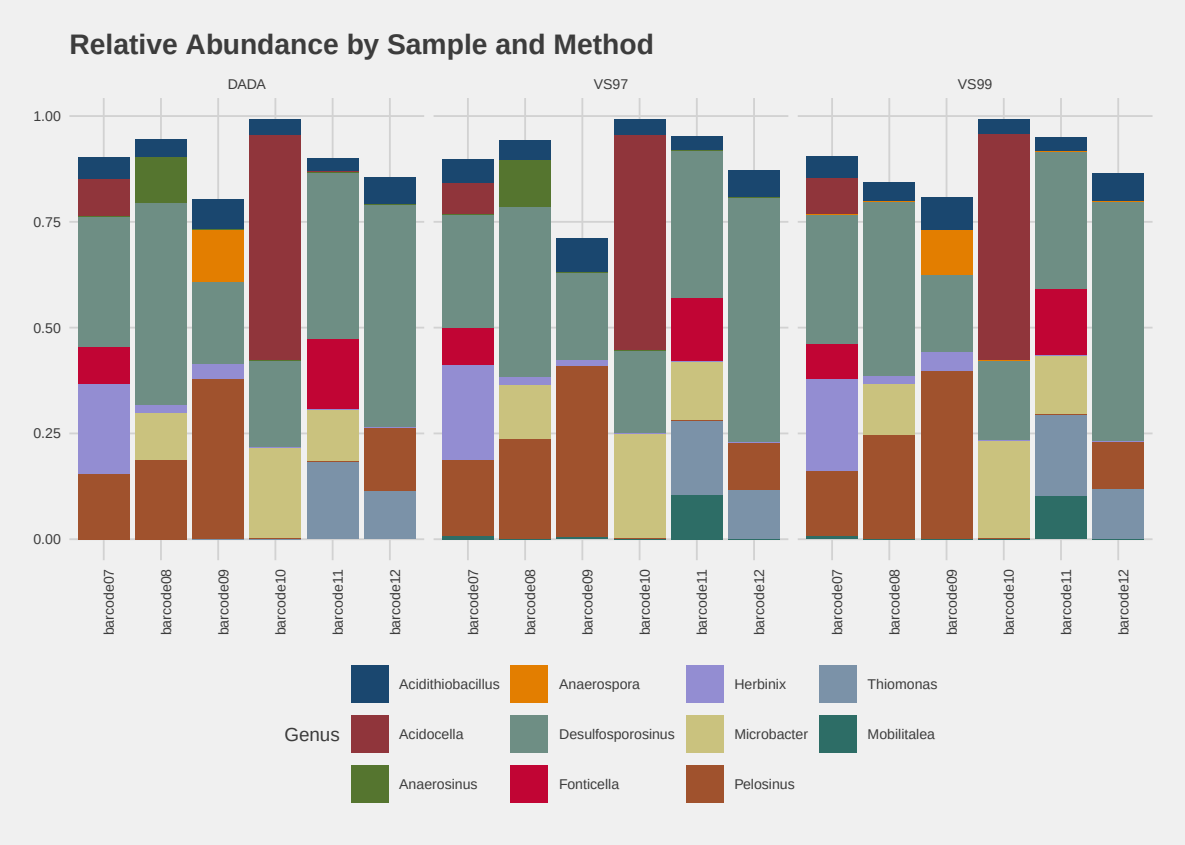
For barplot, we need to melt assay data

```
#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")
```

Table 5: 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

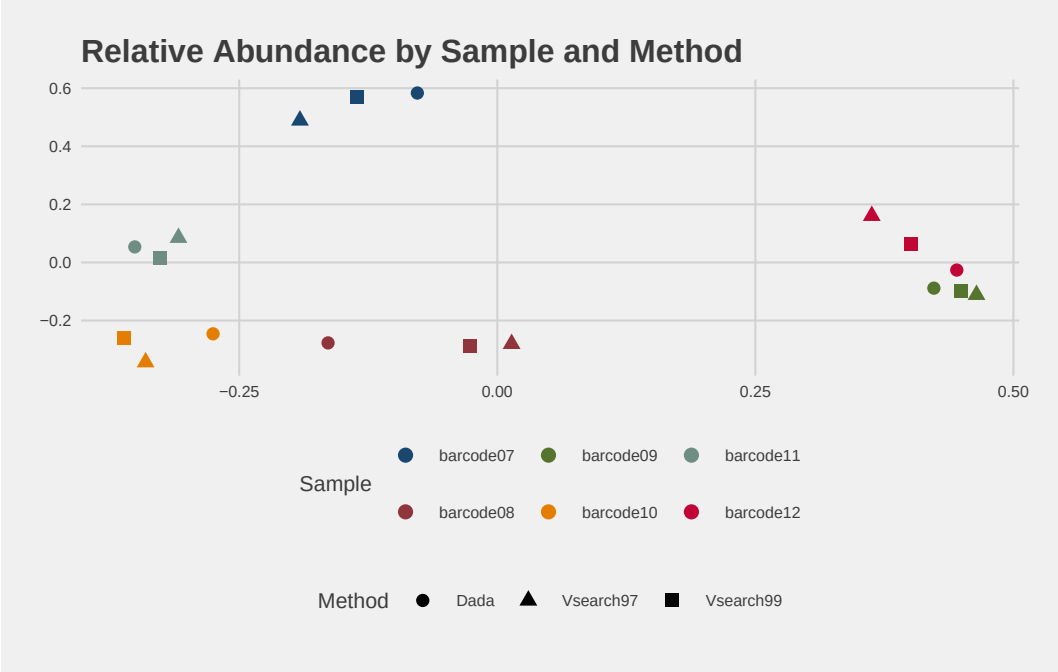
Beta diversity is compared using bray-curtis dissimilarity

```
#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 %.