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'

Reload results as a tse object named dada

# Path variables  
asvfile <- "results\_set1/asvs.tsv"  
metafile <- "data/set1\_meta.tsv"  
taxafile <- "results\_set1/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

Load results from qiime pipelines

# vsearch97  
asvfile <- "results\_vsearch97/asvs\_set1.tsv"  
taxafile <- "results\_vsearch97/taxonomy\_set1.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\_set1.tsv"  
taxafile <- "results\_vsearch99/taxonomy\_set1.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

Show number of variants in each object

#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 |
| --- | --- | --- |
| 63 | 985 | 9195 |

As species level taxonomic information is fairly unreliable, we agglomerate 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] 18

nrow(VS97)

[1] 33

nrow(VS99)

[1] 32

Clear difference between methods remains, but in much smaller extent

Next, we convert counts to relative abundance values

#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")

Then, we pick five most abundant features

#get top5 features  
top5\_DADA <- getTopFeatures(DADA, top = 5, method = "mean",  
 assay.type = "relabundance")  
top5\_VS97 <- getTopFeatures(VS97, top = 5, method = "mean",  
 assay.type = "relabundance")  
top5\_VS99 <- getTopFeatures(VS99, top = 5, method = "mean",  
 assay.type = "relabundance")  
#subset top features based on top5 list  
DADA <- subsetFeatures(DADA, rowData(DADA)$Genus %in% top5\_DADA)  
VS97 <- subsetFeatures(VS97, rowData(VS97)$Genus %in% top5\_VS97)  
VS99 <- subsetFeatures(VS99, rowData(VS99)$Genus %in% top5\_VS99)  
#create dataframes  
df\_DADA <- data.frame(assays(DADA)$relabundance)  
df\_DADA <- df\_DADA %>% mutate(Genus = rownames(df\_DADA)) %>%  
 filter(Genus %in% top5\_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)$relabundance)  
df\_VS97 <- df\_VS97 %>% mutate(Genus = rownames(df\_VS97)) %>%  
 filter(Genus %in% top5\_VS97) %>% arrange(Genus) %>%  
 mutate\_at("Genus", str\_replace, "Genus:", "")  
rownames(df\_VS97) <- NULL  
df\_VS97 <- df\_VS97[,c(7,1,2,3,4,5,6)]  
df\_VS99 <- data.frame(assays(VS99)$relabundance)  
df\_VS99 <- df\_VS99 %>% mutate(Genus = rownames(df\_VS99)) %>%  
 filter(Genus %in% top5\_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)]

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

| Genus | barcode01 | barcode02 | barcode03 | barcode04 | barcode05 | barcode06 |
| --- | --- | --- | --- | --- | --- | --- |
| Aeromonas | 0 | 1 | 0 | 0 | 0 | 0 |
| Delftia | 0 | 0 | 0 | 0 | 1 | 0 |
| Providencia | 0 | 0 | 0 | 1 | 0 | 0 |
| Pseudomonas | 1 | 0 | 0 | 0 | 0 | 0 |
| Stenotrophomonas | 0 | 0 | 1 | 0 | 0 | 1 |

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

| Genus | barcode01 | barcode02 | barcode03 | barcode04 | barcode05 | barcode06 |
| --- | --- | --- | --- | --- | --- | --- |
| Aeromonas | 0 | 1 | 0 | 0 | 0 | 0 |
| Delftia | 0 | 0 | 0 | 0 | 1 | 0 |
| Providencia | 0 | 0 | 0 | 1 | 0 | 0 |
| Pseudomonas | 1 | 0 | 0 | 0 | 0 | 0 |
| Stenotrophomonas | 0 | 0 | 1 | 0 | 0 | 1 |

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

| Genus | barcode01 | barcode02 | barcode03 | barcode04 | barcode05 | barcode06 |
| --- | --- | --- | --- | --- | --- | --- |
| Aeromonas | 0 | 1 | 0 | 0 | 0 | 0 |
| Delftia | 0 | 0 | 0 | 0 | 1 | 0 |
| Providencia | 0 | 0 | 0 | 1 | 0 | 0 |
| Pseudomonas | 1 | 0 | 0 | 0 | 0 | 0 |
| Stenotrophomonas | 0 | 0 | 1 | 0 | 0 | 1 |

For stacked barplots, we melt assay data

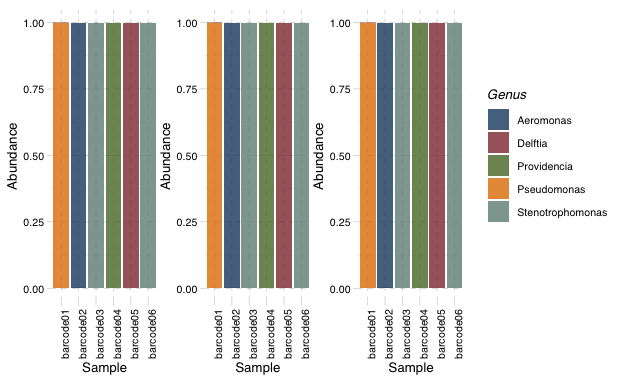
#transform data to ggplots  
assay\_dada <- meltAssay(DADA, assay.type="relabundance")  
names(assay\_dada) <- c("Genus","Sample","Abundance")  
assay\_vs97 <- meltAssay(VS97, assay.type="relabundance")  
names(assay\_vs97) <- c("Genus", "Sample", "Abundance")  
assay\_vs99 <- meltAssay(VS99, assay.type="relabundance")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

Plot objects

#Create stacked barplot  
dada\_plot <- ggplot(assay\_dada, aes(x = Sample, y = Abundance, fill = Genus)) +  
 geom\_bar(stat = "identity", show.legend = FALSE, alpha=0.8) +  
 theme\_pander(base\_size = 10) + scale\_fill\_stata() + theme(axis.text.x =  
 element\_text(angle = 90))  
  
vs97\_plot <- ggplot(assay\_vs97, aes(x = Sample, y = Abundance, fill = Genus)) +  
 geom\_bar(stat = "identity", show.legend=FALSE, alpha=0.8) +  
 theme\_pander(base\_size = 10) + scale\_fill\_stata() + theme(axis.text.x =  
 element\_text(angle = 90))  
  
vs99\_plot <- ggplot(assay\_vs99, aes(x = Sample, y = Abundance, fill = Genus)) +  
 geom\_bar(stat = "identity", alpha=0.8) + theme\_pander(base\_size = 10) +  
 scale\_fill\_stata() + theme(axis.text.x = element\_text(angle = 90))

Results side by side

#show plots side by side  
dada\_plot+vs97\_plot+vs99\_plot



#### Observations

In this sample set, where we evidently have pure microbial cultures, each method gave exactly same result. Low diversity might explain why dada2 error profile looks clean. In more complex communities, it’s possible that denoising doesn’t function as well. We already have previous evidence that dada2 can function accurately in mock communities consisting few microbes.

On the other hand, clustering methods produce lot more low abundance noise, but it doesn’t show in the end results (with two digits accuracy). There are two discrepancies in 10 most commond features between denoising and clustering. Dada picked up fonticella and clustering methods acidithiobacillus.