**Exercise: 16S rRNA gene sequencing – Quantative Analysis**

**1 General questions and data description**

**Q1: In broad terms, what is the application of 16S rRNA gene amplicon sequencing analyses?**

**Q2: Please describe the computational pipeline used to process 16S rRNA gene amplicon sequencing data until the creation of the OTU table and representative sequences.**

**Q3: Describe with your own words, what is alpha diversity?**

**Q4: Describe with your own words, what is beta diversity?**

**Q5: Explain with your own words the importance of considering sample sizes**

The following exercise is based on samples from a study of the human gut microbiome and the response to diet. Here, two individuals are included. The individuals were asked to replace the grain products with standard high whole grain foods for eight weeks, followed by at least six weeks of wash out period. After the wash out period, the individuals were put on a control diet, where they consumed grain products with low whole grain content (refined grain).



The individuals handed in a fecal sample before and after the two dietary interventions, which makes t1 and t3 baseline and t2 and t4 are sampled after the interventions. During the wash out period, the participants could eat whatever they chose.

Microbial DNA was extracted from the fecal samples and the V3 and V4 region of the 16S rRNA gene were amplified with PCR using universal primers. Further, the amplified region is approximately 440 bp long and the amplicons were sequenced on an Illumina Miseq as 300 bp paired end data.

The UPARSE pipieline was used to process the data and the:

- OTU table  
- Taxonomic annotation

Is available on the server for download at

/home/27626/exercises/qmetagenomics/16S

So please create a separate folder for this exercise.

The statistical framework R contains modules, which enable diversity analyses of count data.

Open R Studio and install the needed libraries:

install.packages("vegan")

install.packages("GUniFrac")

install.packages("stringr")

Set the working directory to point to your exercise folder:

setwd("INSERT YOUR EXERCISE FOLDER PATH HERE")

Load the OTU table into R:

otu\_m <- as.matrix(read.table("otu\_table.txt", header = TRUE, row.names = 1, sep = '\t',comment.char=""))

**2 Sample depth**

To investigate if we covered the full species diversity, you can count the number of low abundant OTUs and look at rarefaction curves. A histogram of the OTU count can give an idea of the number of low abundant OTUs. Here we produce a histogram of OTUs that are observed 10 times or less:

tot\_count <- rowSums(otu\_m)  
low <- subset(tot\_count, tot\_count <= 10)

hist(low, xlab = "OTU counts")

Also, the shape of the rarefaction curve will reveal how many new OTUs are added to the sample, when including more reads. Here, the rarefaction curves are produced by subsampling the reads with a step size of 100 and counting the number of unique OTUs observed:

Create the rarefaction curves using the following commands:

library(vegan)  
raremax <- min(colSums(otu\_m))

rarecurve(t(otu\_m), step = 100, sample = raremax, col = "blue", cex = 0.6)

dev.print("rarecurve.pdf", device = pdf)

The ‘raremax’ variable includes the size of the smallest sample and this is marked with the vertical line in the plot

**Q6: Did we cover the whole microbial diversity in the sample, with the current sequencing depth? Please explain how you came to that conclusion.**

**3 Sample sizes and alpha diversity**

During the lectures we talked about the importance of sample size. Here we will try and interpret the effect of normalization and rarefaction on alpha diversity including the measures: observed species and the Shannon Index.

Look at the sample sizes using the command:

colSums(otu\_m)

The smallest sample is id2\_t2, which contains 29,798 reads.

We will normalize the samples according to library sizes by dividing each OTU count with the total count in the sample. The result is the proportion of a specific OTU out of the whole sample. We can times this with 100 to get it in percentage:

norm\_otu <- sweep(otu\_m, 2, colSums(otu\_m),"/")\*100

We can also choose to rarefy the OTU table. We will randomly pick 29,798 reads from each sample (29,798 reads is the size of the smallest library and the maximum reads we can rarefy, without discarding any samples):

library(GUniFrac)

rare\_otu <- t(Rarefy(t(otu\_m), depth=min(colSums(otu\_m)))$otu.tab.rff)

To observe the effect of these methods, accounting for varying sample sizes, we will count the number of observed species in each sample and for the raw otu table, the normalized otu table and the rarefied otu table. The number of observed species is displayed in a barchart:

raw\_obs\_spec <- specnumber(t(otu\_m))

norm\_obs\_spec <- specnumber(t(norm\_otu))

rare\_obs\_spec <- specnumber(t(rare\_otu)) # number of unique OTUs

counts <- data.frame(raw=raw\_obs\_spec,normalized = norm\_obs\_spec, rarefied = rare\_obs\_spec)

barplot(t(as.matrix(counts)), beside = TRUE, xlab = "Samples", ylab = "Observed Species", col = c("red", "blue", "green"), las=2)

legend("topright", legend = c("Raw","Normalized","Rarefied"), col = c("red", "blue", "green"), pch = 20)

dev.print("observed\_species.pdf", device=pdf)

The same procedure can be done for the Shannon Index:

raw\_shan <- diversity(t(otu\_m))

norm\_shan <- diversity(t(norm\_otu))

rare\_shan <- diversity(t(rare\_otu))

shan <- data.frame(raw=raw\_shan,normalized = norm\_shan, rarefied = rare\_shan)

barplot(t(as.matrix(shan)), beside = TRUE, xlab = "Samples", ylab = "Shannon Index", col = c("red", "blue", "green"), las=2)

legend("topright", legend = c("Raw","Normalized","Rarefied"), col = c("red", "blue", "green"), pch = 20)

dev.print("shannonID.pdf", device=pdf)

Please view the two figures: 'observed\_species.pdf' and 'shannonID.pdf'.

**Q7: Is there a difference in richness and alpha diversity between the two individuals?**

**Q8: What is the effect of the different methods to account for differences in sample sizes?**

**Q9: In the case where you want to compare samples, which normalization method would you choose and why?**

HINT: Try also to look at the connection between richness and sample size (here in id1), do you see any connection?

par(mfrow=c(1,2))

plot(norm\_obs\_spec[1:4], colSums(otu\_m)[1:4], xlab="Richness", ylab = "Sample size", main = "Normalisation")

plot(rare\_obs\_spec[1:4], colSums(otu\_m)[1:4], xlab="Richness", ylab = "Sample size", main = "Rarefaction")

par(mfrow=c(1,1))

**4 Beta diversity**

To investigate the similarity between samples, we will calculate the beta diversity. In this exercise we will use the Bray-Curtis dissimilarity metric and plot the results as a hierarchal tree and in a NMDS plot:

library(vegan)  
bc <- vegdist(t(rare\_otu))

h\_bc <- hclust(bc)

plot(h\_bc)

dev.print("hclust.pdf", device=pdf)

bc\_pcoa <- pcoa(bc)

plot(bc\_pcoa$vectors[,2] ~ bc\_pcoa$vectors[,1], col = c(rep("red",4),rep("blue",4)), main="PCoA Bray-Curtis distances", xlab = "PCoA1", ylab = "PCoA2", pch=16)

dev.print("PCoA.pdf", device=pdf)

**Q10: How do the samples cluster? What does it tell you about the dynamics of the microbial community structure in the individuals?**

**5 Summarizing the taxonomic composition**

To get an overview of the taxonomic composition in the samples we can summarize the taxonomic composition on phylum level. The taxonomic annotation should be imported to R:

tax <- read.table("rep\_seq.utax.txt", row.names = 1, header = FALSE, sep = '\t')

Then a regex is used in R to extract the information about which phylum a specific OTU is annotated to and then the taxonomic list are sorted according to the OTU table.

library(stringr)  
phylum\_tmp <- str\_extract(tax[,2], "p:(.\*),c")

phylum\_un <- substr(phylum\_tmp, 1, nchar(phylum\_tmp)-2)

names(phylum\_un) <- rownames(tax)  
phylum <- phylum\_un[rownames(otu\_m)]

Now we can use the phylum vector to summarize all OTUs from the same phylum using the aggregate function and display the resulting data frame in a stacked barchart

p\_df <- aggregate(data.frame(otu\_m), list(phylum), FUN = sum)

barplot(as.matrix(p\_df[,2:9]), col=rainbow(6), las=2)

legend("topleft", legend=p\_df[,1], fill=rainbow(6))

**Q11: Are there any differences between the two individuals?**