Natrix_StatisticalAnalysis

Jan Taubenheim

7/21/2020

```
phy <- readRDS("../data/PE_denoise_dada2_physeq.RDS")
#phy <- readRDS(snakemake@input[["physeq"]])
# filter samples which are not really sequenced
otu <- phy@otu_table
```

Filtering

To reduce the noise in the statistical analysis I will reduce the number of bacteria across samples and will remove the samples, which were not really sequences. My aim would be to keep around 95% of the data.

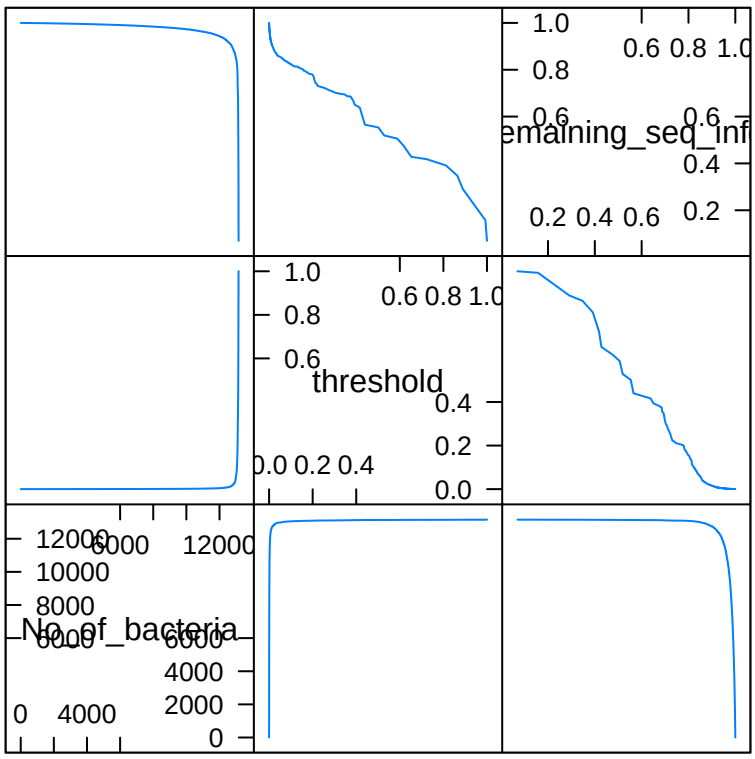
```
# find a good threshold to remove most of the bacteria which are only spuriously
# found calculate the maximum contribution (as fraction) to each sample of each
# ASV and take the maximum for each ASV
depth <- apply(otu,2,sum)
maxFrac <- apply(otu, 1, function(x) max(x/depth, na.rm=TRUE))

# take each of these maximum fractions and use them as a threshold for to
# exclude bacteria from the data set, which are lower than that specific
# fraction
maxFracTrhl <- function(x){
    nm <- names(sort(maxFrac))[1:x]
    sel <- !(rownames(otu) %in% nm)
    return(sum(otu[sel,])/sum(otu))
}


no_cores <- detectCores()-1
cl<- makeCluster(no_cores)
clusterExport(cl,"otu")
clusterExport(cl,"maxFrac")
clusterExport(cl,"maxFracTrhl")
steps <- seq(1,length(maxFrac), by = 2)
a <- parSapply(cl, steps, maxFracTrhl)
stopCluster(cl)

df <- data.frame(No_of_bacteria=steps,
                 threshold = sort(maxFrac)[steps],
                 remaining_seq_info = a)

splom(df, type = "l")
```



Scatter Plot Matrix

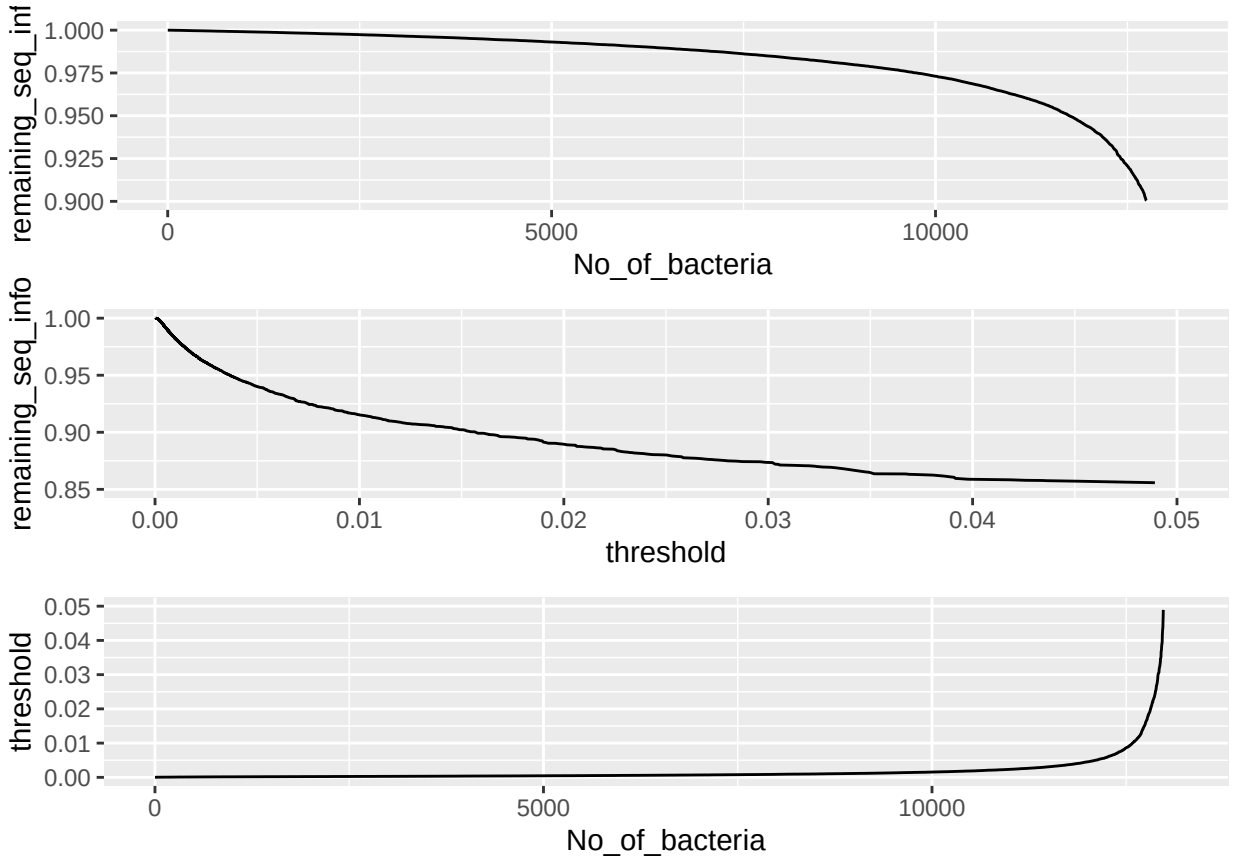
```
p_noASV <- ggplot(df, aes(y = remaining_seq_info, x= No_of_bacteria)) +
    geom_line() +
    ylim(0.9,1)
p_thr <- ggplot(df, aes( y = remaining_seq_info, x = threshold)) +
    geom_line() +
    ylim(0.85,1) +
    xlim(0,0.05)
p_thrBac <- ggplot(df, aes(x = No_of_bacteria, y = threshold)) +
    geom_line() +
    ylim(0,0.05)
p_all<- plot_grid(p_noASV, p_thr, p_thrBac, ncol = 1)
```

```
## Warning: Removed 204 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 88 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 88 row(s) containing missing values (geom_path).
```

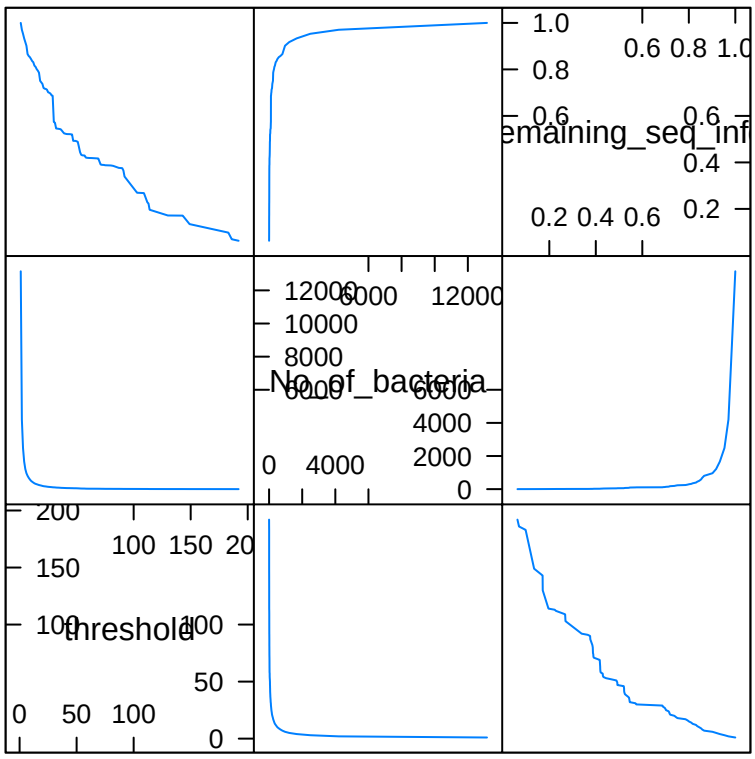
```
p_all
```



From the plots, it looks reasonable to use a threshold somewhere between 0.001 and 0.01 for contribution of the bacteria to at least one sample. This is the point in the curves, where the asymptotic phase starts, but also a relatively large amount of sequencing information can be lost. However, the asymptotic phase is a good point to filter at. To regard the amount of total information loss, I will add another criteria: the prevalence of the bacteria across samples.

```
# remove all bacteria, which contribute less than 1% to the microbial community,
# unless its prevalent in more than 3 samples and vice versa (bacteria that
# contribute less than 1% but is prevalent in more than 3 samples are not
# filtered)
prevalence <- apply(otu, 1, function(x) sum(x>0))
prevalence_uniq <- sort(unique(prevalence), decreasing = TRUE)
seqInfo <- sapply(prevalence_uniq, function(x){
        sel <- names(prevalence[prevalence >= x])
        return(sum(otu[sel,])/sum(otu))
})
numBac <- sapply(prevalence_uniq, function(x) length(prevalence[prevalence >=x]))
df2 <- data.frame(threshold = prevalence_uniq,
                  No_of_bacteria = numBac,
                  remaining_seq_info=seqInfo)

splom(df2, type = "l")
```



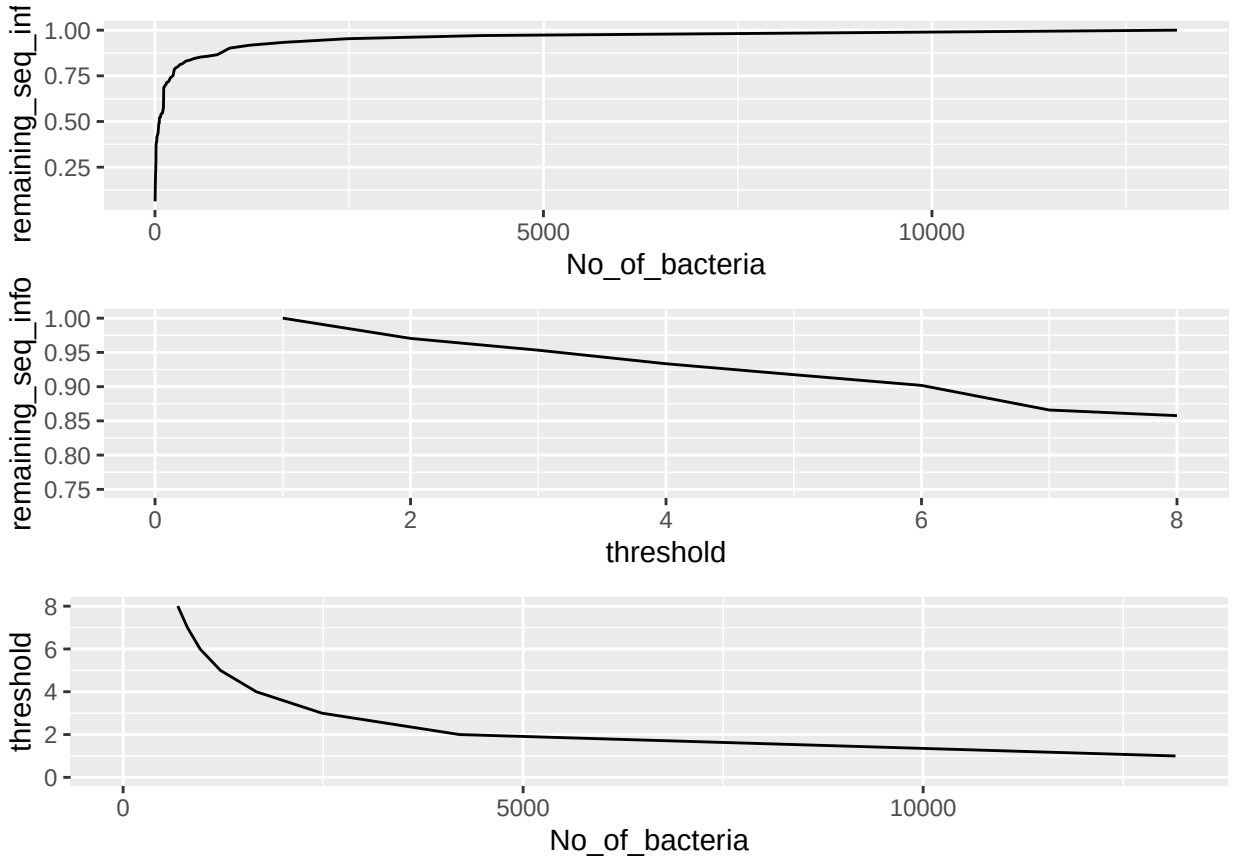
Scatter Plot Matrix

```
p_noASV2 <- ggplot(df2, aes(y = remaining_seq_info, x= No_of_bacteria)) +
    geom_line()
p_thr2 <- ggplot(df2, aes( y = remaining_seq_info, x = threshold)) +
    geom_line() +
    xlim (0,8) +
    ylim(0.75,1)
p_thrBac2 <- ggplot(df2, aes(x = No_of_bacteria, y = threshold)) +
    geom_line() +
    ylim(0,8)
p_all2<- plot_grid(p_noASV2, p_thr2, p_thrBac2, ncol = 1)
```

```
## Warning: Removed 73 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 73 row(s) containing missing values (geom_path).
```

```
p_all2
```



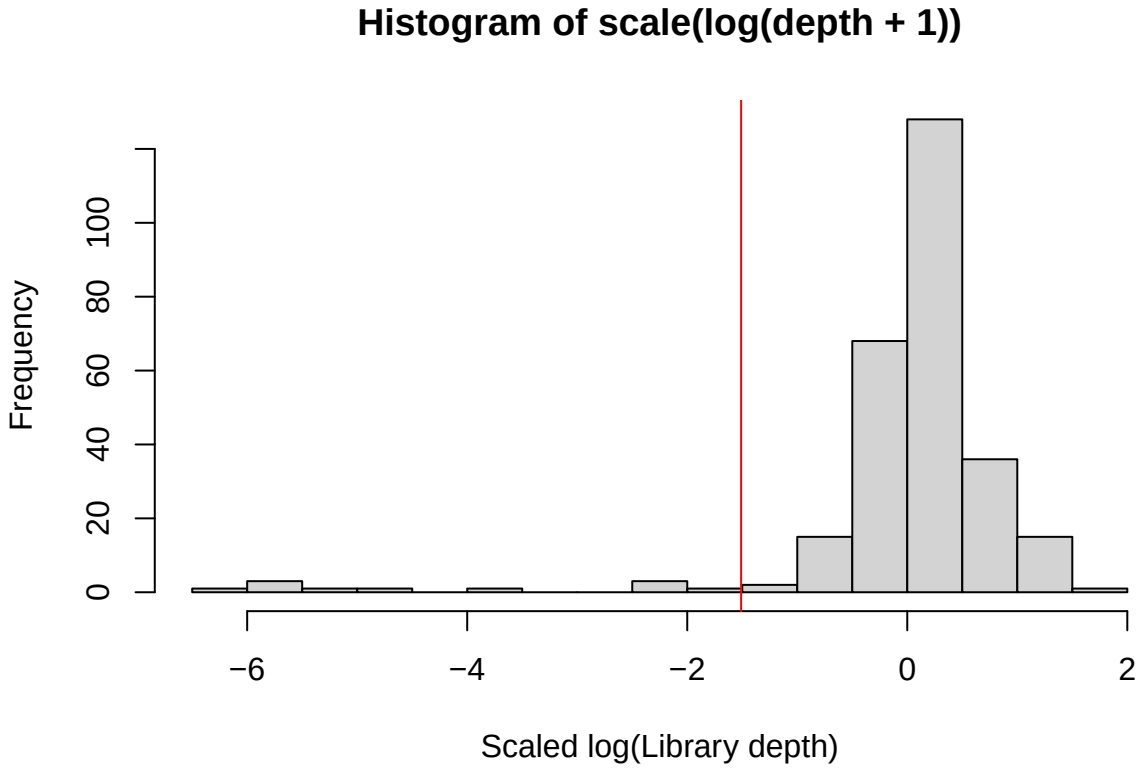
Following these plots, to keep a reasonable amount of data, I would set the threshold to a prevalance of 3.

```
sel <- ((prevalence >3 | maxFrac >0.01))
otu2 <- otu[sel,]
```

The two filters leave me with 1864 ASVs across all samples and 95.82% of sequencing information but filtered 11290 bacteria.

Eventually, it does not make sense to keep the samples which were not really sequenced. To that end, I will cut off the non normal tail of the logged library depth on the lower end and remove all sequences which ae not really expected for the sampling done at the lower tail.

```
# log the library size, scale it to center mean and by sd, take the upper bound,
# reverse the sign and apply it as lower bound filter
depth <- apply(otu2,2,sum)
hist(scale(log(depth+1)), xlab = "Scaled log(Library depth)", breaks = 20)
abline(v = -1*max(scale(log(depth+1))), col = "red")
```




```
sel <- scale(log(depth+1)) < -1*max(scale(log(depth+1)))
otu3 <- otu2[,!sel]
```

This filtering removes 11 samples and reduces the amount of sequencing information to 95.78% of the initial input. The sample with the least reads contains now 888 reads.