Documentation_Heatmap

Porthmeus

3/11/2020

This is the analysis of a recolonization experiment for *Nematostella vectensis* and how the bacteria community is re-established after antibiotic treatment. *Nematostella* will go through 3 major life stages after hatching from the egg, first is the planula stage (larva, L) where the larva are free swimming. After a short period, the planula will settle down and form a tube like body and first tentacles appear, the so colled juvenile (J) stage. From the juvenile stage, mesenteries and full (and more) tentacles will develop and the animal reaches its adult (A) stage. These developmental transitions are associated with changes in the microbial community living on the surface of *Nematostella* [Mortzfeld et al, 2015.]

Hanna investigated in her PhD thesis the question, how this microbial community is shaped by the host and to what extend the microbiome is resilient to disturbances. She removed the complete microbiota from adult *Nematostella* with anitbiotics and recolonized the animals with three different inocula: bacterial communities derived from larval (bL), juvenile (bJ) and adult polyps (bA). She then sampled at different timepoints to elucidate the dynamics of the recolonization and sequenced 16S-RNA.

The experiment results in a rather complex data set having several dimensions: first, the source of recolonizing bacteria (inocula), which served as a reference to what was used to recolonize the animals. This is another dimension in the adult recolonized animals. Third, there are different time points (2, 7, 14, 28 day post recolonization [dpr]).

To the data: you will find two data sets in the directory *data*. Both are relative abundance data of 16S-sequencing, one for ESV (**table_rel.tsv**) and one for OTUs (**otu_97_cluster.csv**). For the reader not familiar with the terms - ESV stands for Exact Sequence Variant, thus represents single bacterial species/strains, even with minimal sequence divergance in the 16S amplicon - OTU stands for Operational Taxonomic Units, which represent all ESVs after clustering at 97% identity level. The file **mapping_recolonization.tsv** contains meta data for the samples sequenced.

NOTE: the data files contain more samples, than given in the the meta data file, as Hanna sequnced several other conditions as well, e. g. wild types without the recolonization. We, however, exclude this data in the analysis, as it appeared not important at this point to us.

The idea of the whole plot is to show, that the recolonization event is recapitulating the bacterial community changes observed during the development in *Nematostella*, disregarding the origin of bacteria.

For now, we decided to analyse the ESV data, as it is state of the art. Since the names for the ESVs are unreadable, I will substitute them and simply number them from 1:n. A corresponding table can be found in the *data* directory, called **ESV2Number.csv**. So first thing is to load necessary libraries and data and further to recreate a matrix from the data.table.

```r
require(pheatmap)
require(data.table)
require(RColorBrewer)
require(ggplot2)
require(factoextra)
require(cowplot)

dataDir <- "data"
```

```r
meta <- fread(file.path(dataDir,"mapping_recolonization.tsv"))
mat <- fread(file.path(dataDir,"table_rel.tsv"))

# cast the data table into a matrix
mat2  <- as.matrix(mat[,-1])

# generate new names for the ESVs and add them as row.names
colnames(mat)[1] <- "OTU"
mat[,ESV := paste("ESV",1:nrow(mat), sep = "_")]
rownames(mat2) <- mat[,ESV]

# write a table as a meta file to the data directory for the conversion of ESV
# identifier to the ESV number
write.csv(mat[,.(ESV_ID = OTU,ESV=ESV)],
          file = file.path(dataDir, "ESV2Number.csv"),
          row.names=FALSE)
```

The first thing which will be necessary is to filter the data for the samples which are present in the meta data file and which are of importance for us. Further, I removed all ESVs which have no more reads in the remaining samples.

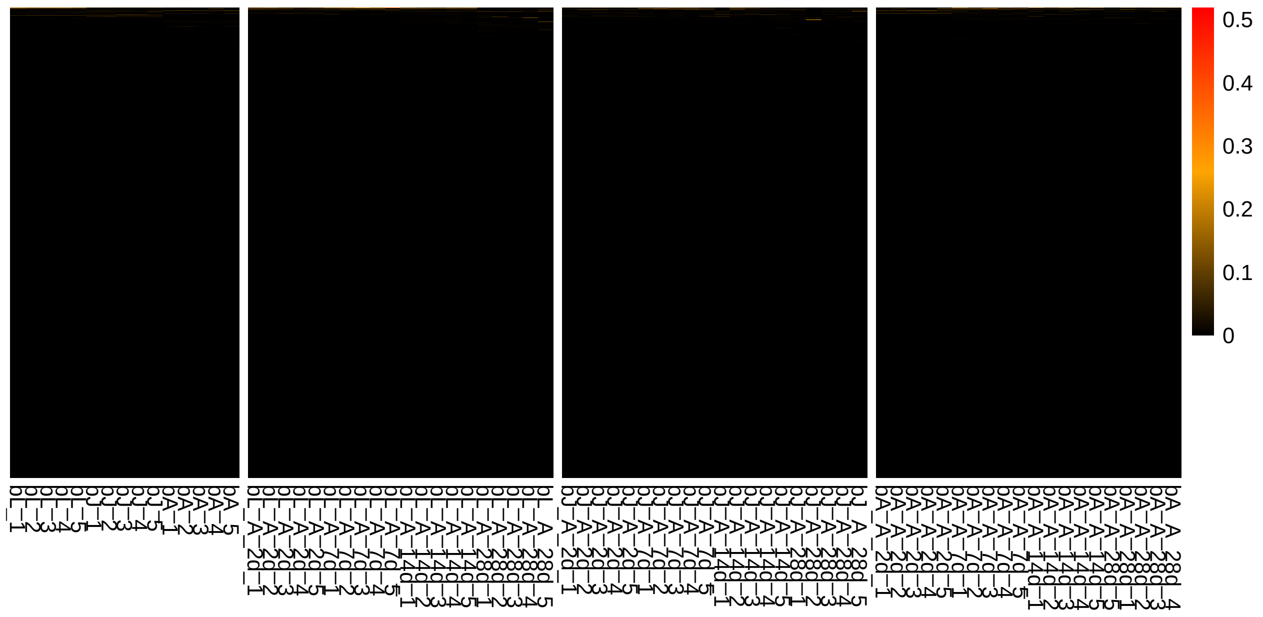
```r
mat2 <- mat2[,meta[,sampleID]]
mat2 <- mat2[apply(mat2,1,sum)>0,]
```

OK, next I prepare the pretty plotting, for this I will define the order of the samples and how they will appear in the final plot. I also define the color code and the gaps between the different sub heatmaps.

```r
# create a vector of the order, how the samples should appear in the data
cdt_order <- c("bL","bJ","bA",
               "bL_A_2d","bL_A_7d","bL_A_14d","bL_A_28d",
               "bJ_A_2d","bJ_A_7d","bJ_A_14d","bJ_A_28d",
               "bA_A_2d","bA_A_7d","bA_A_14d","bA_A_28d")

# make the meta column factors and level it according to the above order. Use
# the order as well to sort the samples in the matrix and the meta table itself.
# Rename the samples in the matrix so that they are more human readable.
meta[,sample := factor(sample, levels=cdt_order)]
mat2 <- mat2[,order(meta[,sample])]
setkey(meta,sample)
colnames(mat2) <- meta[,name]

# create a vector to introduce gaps in the columns of the heatmap and a
# different coloration
gaps <- c(15,35,55) # each condition was sequenced 5 times
colscale <- colorRampPalette(c("black","orange","red"))(255)
```

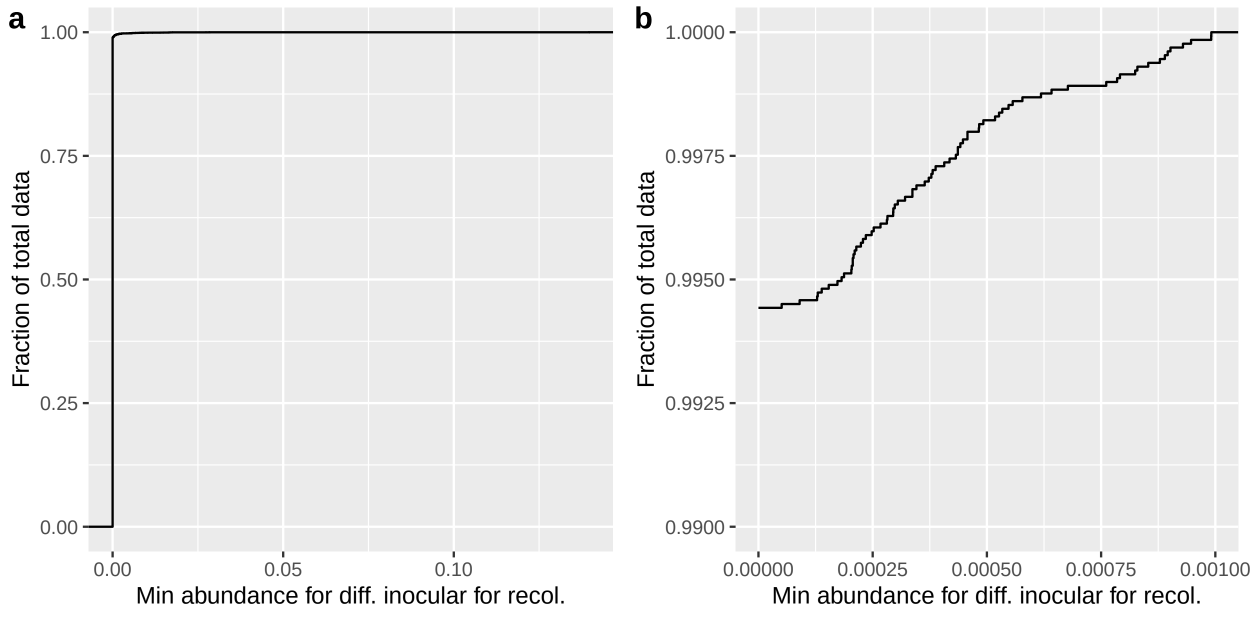
Basically, now a simple plot can be generated without any further filtering or data mingling.

```r
pheatmap(mat2,
         show_rownames = FALSE,
         cluster_cols=FALSE,
         cluster_rows = FALSE,
         gaps_col = gaps,
         color = colscale,
         border_color = NA
)
```



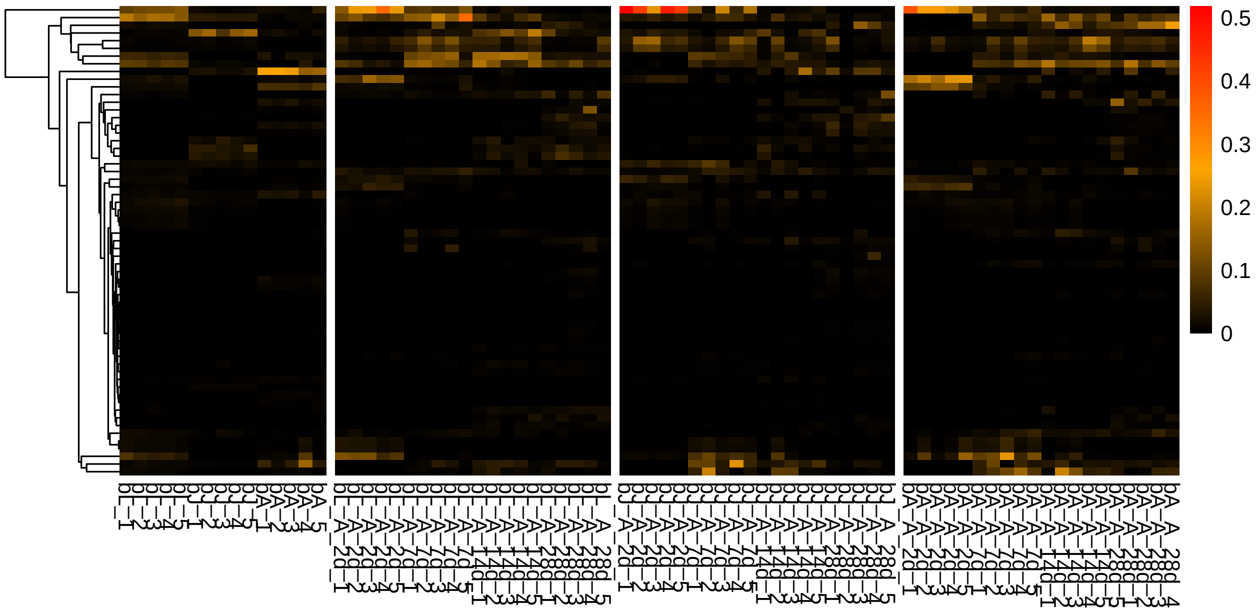
OK this is not informative at all so we need to apply some filtering. We are interested in the recolonization dynamics, I will thus use these samples to define a suitable filter only. We are less interested in the bacteria which can be only spourisly found and we are also not interested in bacteria which can be found in only one or two samples in high abundance. I thus could use the median relative abundance to filter, but this might reduce the bacteria which are more important in only one recolonization setup (for instance are only present in the recolonization if the inocular from adult animals [bA_A]). I tried different filters (see v3) and ended up using the minimal relative abundance for the different timepoints of recolonization >0. This means, to pass the filter, a bacteria has to have a certain amount relative abundance in at least one (2dpr, 7dpr, 14dpr **or** 28dpr) recolonization time point across all samples

Ok to define a filter threshold I plot the ECDF of these values across all samples. This needs some data mining in the first place, by extracting the minimal values across the data. And than it is simply plotting the ECDF of the data.



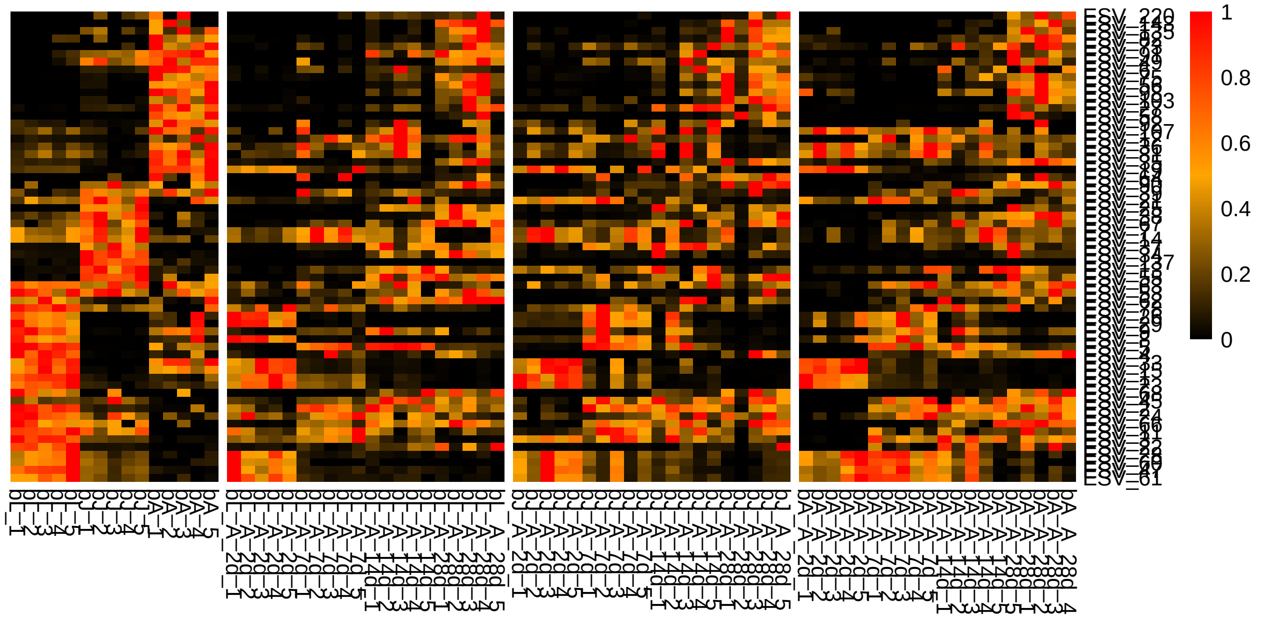
The plots indicate the that only ~0.6% of the data is >0. Thus I used this as a threshold to plot only these bacteria.

```r
# filter the matrix for only those bacteria with at least some reads in one of
# the conditions
sel <- unique(minDf[minAb >0, rn])
pheatmap(mat2[sel,],
         show_rownames = FALSE,
         cluster_cols=FALSE,
         cluster_rows = T,
         gaps_col = gaps,
         color = colscale,
         border_color = NA
)
```



Finally I normalized the abundance data to range between 1 and 0 for each inocula-recolonization sub heatmap. This gives a nice pattern and shows the timely recapitulation of the different timepoints to the different developmental stages. I also add a custom clustering pattern (using kendall's distance measure and clustering by median). The clustering is only performed on the inocula samples, as these recapitulate the microbiome changes according the developmental stages in *Nematostella*.

```r
# helper function
minMaxNorm<-function(x){
    # normalizes a vector of numeric values to min=0 and max=1
    # returns 0.5 if x is a single value
    if(length(x)==1){
        return(0.5)
    }else{
        res<-c()
        mi<-min(x)
        ma<-max(x)
        for(i in x){
            res<-c(res,(i-mi)/(ma-mi))
        }
    }
    res[is.na(res)] <- 0

    return(res)

}
# construct a vector to indicate the order for normalization of the OTUs
plotMeta <- meta[,treatment]
inocL <- nrow(meta[identity == "inoculate",])
plotMeta[1:inocL] <- meta[identity == "inoculate",identity]
meta[["plotMeta"]] <- plotMeta

# do the actual normalization
matMinMax <- mat2[sel,]
for(i in unique(meta[,plotMeta])){
    smpl <- meta[plotMeta == i,name]
    matMinMax[,smpl] <- t(apply(mat2[sel,smpl], 1, minMaxNorm))
}

# do the clustering on the inoculate samples only
clustered_samples <- meta[identity == "inoculate", name]

value_order <- hclust(

    get_dist(mat2[sel,clustered_samples],
             method = "spearman",
             stand = FALSE),

    method = "median")[["order"]]

# create the plot
p_minMax <- pheatmap(matMinMax[value_order,],
                  show_rownames = TRUE,
                  cluster_cols=FALSE,
                  cluster_rows = FALSE,
                  gaps_col = gaps,
                  color = colscale,
                  border_color = NA)
```



Just because I am curious and the data is there: I want to know, how much data is represented by this 61 ESVs. Since I am calculating with relative abundances simply summing all abundences devided by the total number of samples should result in the fraction of data represented by the ESVs given in the data.

```r
print(sum(mat2[sel,])/ncol(mat2))
```

```
## [1] 0.7802754
```

Thus 78.03% of the total data is represented by the 61 ESVs in the data. This is quite high and indicates a good filter.