Lab 1: Using data output from Qiime, transformations, quality control

Where to find useful output from QIIME

- OTU table:
  wf_da/uclust_picked_otus/rep_set/pynast_aligned_seqs/rdp_assigned_taxonomy/otu_table/seqs_otu_table.txt
- Mapping file:
  Fasting_Map.txt
- Tree file:
  wf_da/uclust_picked_otus/rep_set/pynast_aligned_seqs/fasttree_phylogeny/seqs_rep_set.tre

OTU table

OTU ID	CM01_16W	CM01_26W	...	PM12_I	Consensus Lineage
0	0	0	...	0	"Root;Bacteria;Firmicutes;""Clostridia"";Clostridiales;""Lachnospiraceae"""
1	0	0	...	0	Root;Bacteria;Bacteroidetes;Bacteroidetes;Bacteroidales
2	0	0	...	0	Root;Bacteria;Firmicutes;""Clostridia"";Clostridiales;""Lachnospiraceae"""

```
d = read.table('otu_table.txt', header=T, row.names=1, sep='\t')

taxa.names = d$Consensus.Lineage
d = as.matrix(d[,-dim(d)[2]])
```

Absolute abundance to relative abundance

Because the number of sequences for each sample is different, we need to standardize each sample to make them comparable. The most natural normalization is to go from absolute abundance to relative abundances. So each number in the OTU table will represent the proportion of sequences from that samples belonging to that OTU.

```
d = scale(d, center=F, scale=colSums(d))
d = t(d)
```

Writing an R function

- A function is an R object that can be called (executed)
- Function has inputs: parameters
- Function returns an output: value
- A value is either through explicit `return` command or by default the value of the last function called from within that function

```
extract.name.level = function(x, level){
   a=c(unlist(strsplit(x,';')),'Other')
   paste(a[1:min(level,length(a))],collapse=';')
}
```

Execute the new function

Call the function that we have just created and see what the results are.

```
a = as.character(d$Consensus.Lineage[1])
a
extract.name.level(a, 2)
extract.name.level(a, 3)
extract.name.level(a, 5)
```

Function to summarize data at different taxonomic levels

```
otu2taxonomy = function(x, level, taxa=NULL){
  if(is.null(taxa)){
   taxa = colnames(x)
  }
  if(length(taxa)!=dim(x)[2]){
   print("ERROR: taxonomy should have the same length
  as the number of columns in OTU table")
   return;
  }
  level.names = sapply(as.character(taxa),
   function(x)
      extract.name.level(x,level=level))
  t(apply(x, 1,
   function(y)
      tapply(y,level.names,sum)))
}
```

Summarize the OTU table

Explore the OTU tables, summarized at different taxonomic levels by running the following commands.
How many Phyla, Classes, Orders, Families, Genera, OTUs are represented in the dataset? Hint: use `dim` function.

```
d.genus = otu2taxonomy(d,level=7,taxa=taxa.names)
d.family = otu2taxonomy(d,level=6,taxa=taxa.names)
d.order = otu2taxonomy(d,level=5,taxa=taxa.names)
d.class = otu2taxonomy(d,level=4,taxa=taxa.names)
d.phylum = otu2taxonomy(d,level=3,taxa=taxa.names)
```

Mapping file

SampleID	…	Treatment	Gender	Week	Location	Mouse	…
CM01_C	…	C	M	30	Cecal	C1	…
CM02_C	…	C	M	30	Cecal	C2	…
CM03_C	…	C	M	30	Cecal	C3	…
PM11_I	…	P	M	30	Ileal	P11	…
PM12_I	…	P	M	30	Ileal	P12	…

```
ff = read.table('mapping.txt', header=T)
```

Matching up the rows of the mapping file with the OTU table

Run the following commands:

```
ff$SampleID
rownames(d.phylum)
```

Notice that the rows in the OTU table are in a different order (sorted alphanumerically) then in the mapping table. The following command orders the rows of the mapping table:

```
ff=ff[order(ff$SampleID),]
```

Check that the rows now match up:

```
ff$SampleID == rownames(d.phylum)
```

Subsetting the data

```
# select columns by name
vars = c("Root;Bacteria;Bacteroidetes",
   "Root;Bacteria;Firmicutes",
   "Root;Bacteria;Proteobacteria")
d.phylum[,vars]
# select columns by number
cnum = c(1,3,6)
d.phylum[,cnum]
# dropping columns
d.phylum[,!colnames(d.phylum) %in% vars]
# selecting/dropping rows
d.phylum[1:3,]
d.phylum[-(1:90),]
# subset function
week16fecal = subset(d.phylum, ff$Location ==
'Fecal' & ff$Week == 16, select = vars)
```

Basic plotting: explore on your own

```
plot(week16fecal)
week16map = subset(ff, ff$Location == 'Fecal' &
  ff$Week == 16)

plot(week16fecal ~ week16map$Treatment)

plot(week16fecal[,1] ~ week16map$Treatment,
  main=colnames(week16fecal)[1],
  xlab='Treatment', ylab='Relative abundance
  (fraction of the total)')

plot(week16fecal[,2] ~ week16map$Treatment, main
  = colnames(week16fecal)[2], xlab='Treatment',
  ylab='Relative abundance (fraction of the
  total)')
```

Plotting (cont'd)

```
# two plots on one figure
par(mfrow=c(1,2))

plot(week16fecal[,1] ~ week16map
  $Treatment, main = colnames(week16fecal)
  [1], xlab='Treatment', ylab='Relative
  abundance (fraction of the total)')

plot(week16fecal[,2] ~ week16map
  $Treatment, main = colnames(week16fecal)
  [2], xlab='Treatment', ylab='Relative
  abundance (fraction of the total)')
```

Plotting (cont'd)

```
#saving the plot
pdf(file='bf.pdf')
par(mfrow=c(1,2))

plot(week16fecal[,1] ~ week16map$Treatment, main
  = colnames(week16fecal)[1], xlab='Treatment',
  ylab='Relative abundance (fraction of the
  total)')

plot(week16fecal[,2] ~ week16map$Treatment, main
  = colnames(week16fecal)[2], xlab='Treatment',
  ylab='Relative abundance (fraction of the
  total)')

dev.off()
```

Heatmaps

```
# heatmaps
ma.phylum = subset(d.phylum, ff
  $Location=='Fecal', colMeans(d.phylum) >
  0.01)
ma.ff = subset(ff, ff$Location=='Fecal')

library(gplots)

heatmap.2(ma.phylum, dendrogram="both",
  scale="col", col=redgreen(75), distfun =
  function(x) dist(x, method='euclidean'),
  key=T, tracecol=NULL)
```

Heatmaps (factors as side colors)

```
heatmap.2(ma.phylum, dendrogram="both",
  scale="col", col=redgreen(75), distfun =
  function(x) dist(x, method='euclidean'),
  key=T, tracecol=NULL,
  RowSideColors=rainbow(2)
  [as.integer(ma.ff$Treatment)])

heatmap.2(ma.phylum, dendrogram="both",
  scale="col", col=redgreen(75), distfun =
  function(x) dist(x, method='euclidean'),
  key=T, tracecol=NULL,
  RowSideColors=rainbow(3)
  [as.integer(factor(ma.ff$Week))])
```

On your own

- Using plot() and heatmap.2() visually explore the data at a different taxonomic level (Class, Order, Family or Phylum)
- Tip: Focus only on highly abundant taxa (>1%, >5%)

Useful commands: find help pages on these and use for exploring the data

- colnames(), rownames()
- dim()
- max(), min()
- mean(), sd()
- scale()
- boxplot()
- heatmap.2()