Loading Microbiome Data

Back to Table of Contents

After you have installed QIIME and generated an OTU table, you can load the OTU table into R.

Install biom package

```
install.packages('biom',repo='http://cran.wustl.edu')
```

Load biom package

```
library('biom')
```

Convert BIOM file to JSON format

If you have data in a "new" BIOM format (HDF5), you first need to convert to JSON format first.

```
# make a JSON-formatted OTU table for loading into R
biom convert -i otus_closed_ref/otu_table.biom -o otu_table_json.biom --to-json
```

Load global gut data using biom package

```
gg.otus.biom <- read_biom('../data/globalgut/otu_table_json.biom')
```

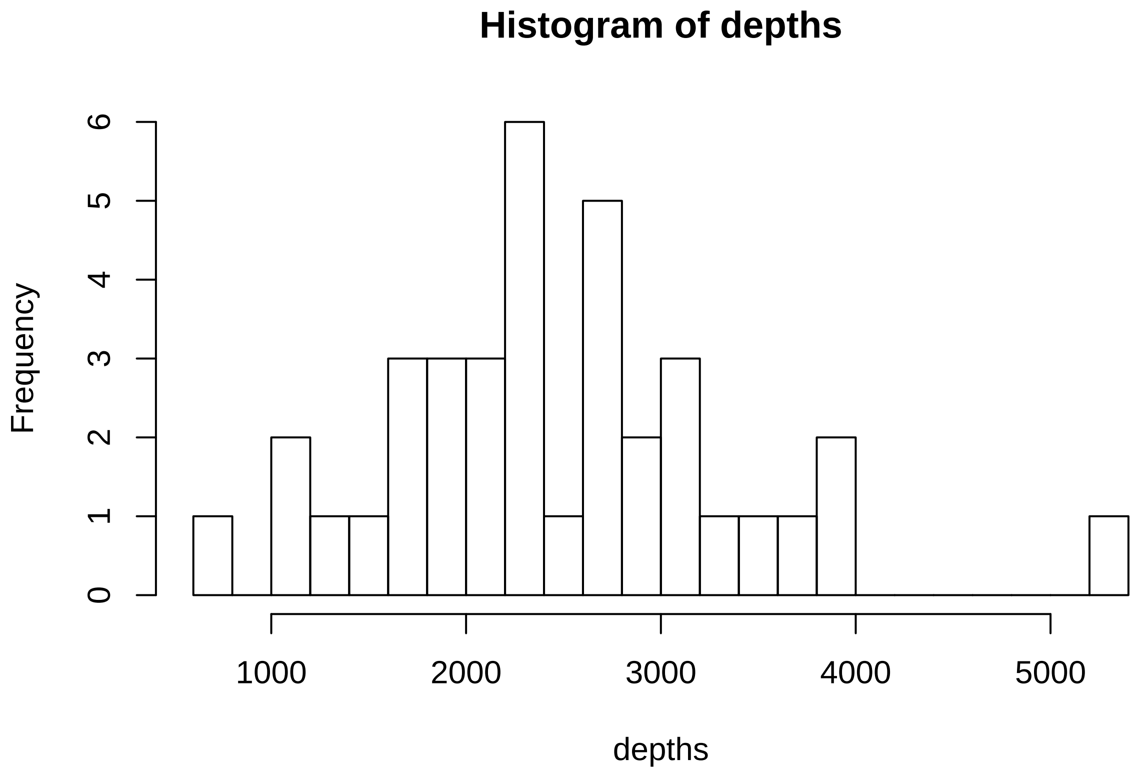
Extract data matrix (OTU counts) from biom table

```
gg.otus <- as.matrix(biom_data(gg.otus.biom))

# transpose so that rows are samples and columns are OTUs
gg.otus <- t(gg.otus)
```

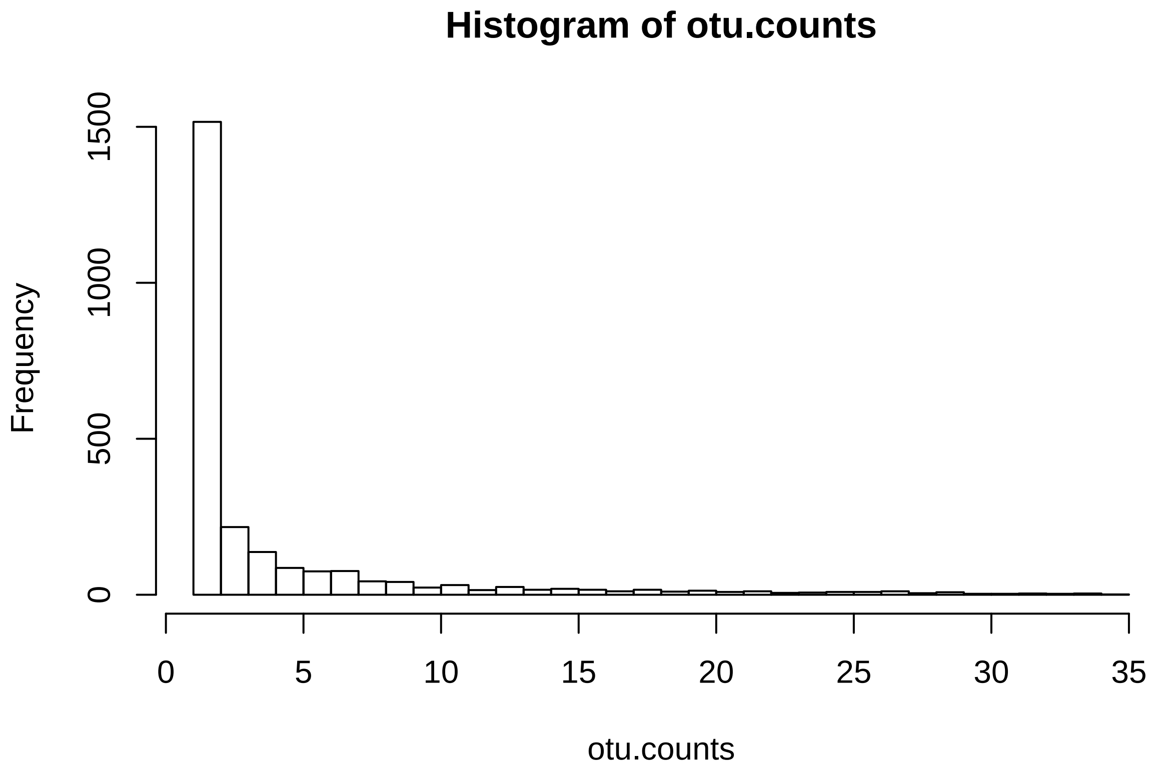
Plot histogram of sample depths

```
depths <- rowSums(gg.otus)
hist(depths,breaks=30)
```



Plot histogram of OTU frequencies

```
otu.counts <- colSums(gg.otus > 0)
hist(otu.counts,breaks=30)
```



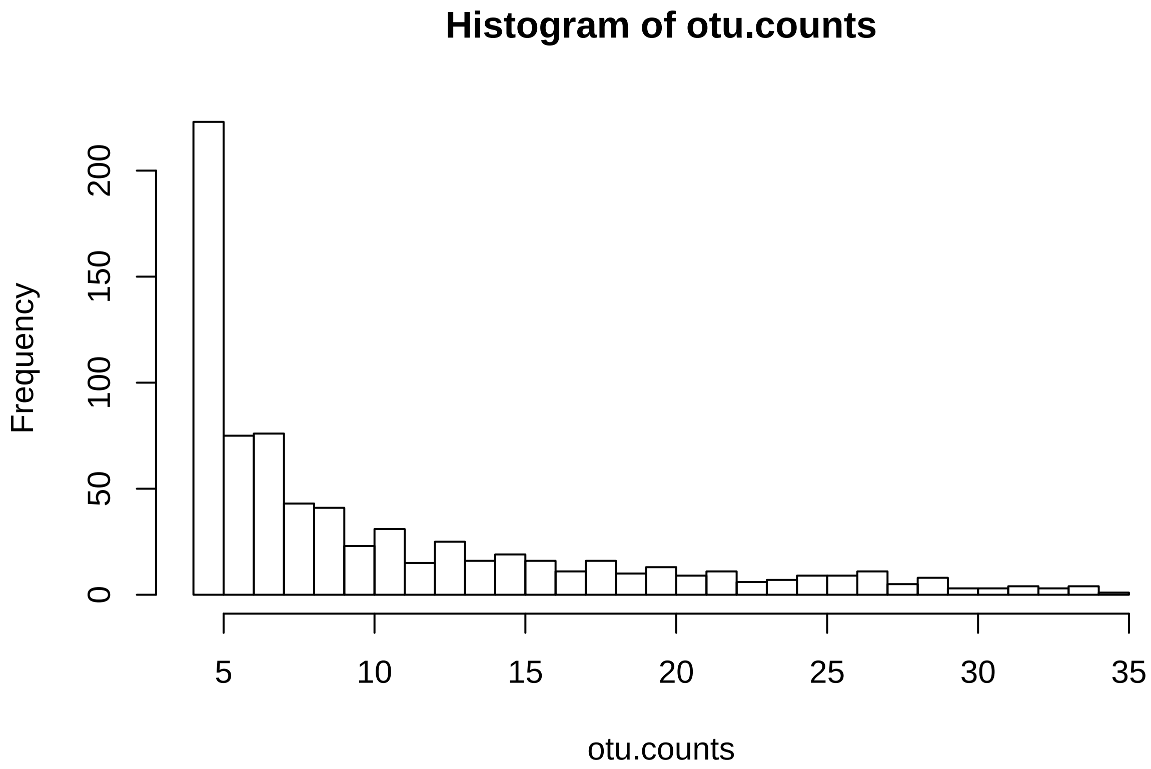
Remove OTUs present in $<$ 10% of samples

```
gg.otus <- gg.otus[,colMeans(gg.otus > 0) >= .1]
depths <- rowSums(gg.otus)
dim(gg.otus)
```

```
## [1]  37 746
```

Re-plot histogram of OTU frequencies now that we removed singletons

```
otu.counts <- colSums(gg.otus > 0)
hist(otu.counts,breaks=30)
```



Remove any samples with very low depth

```
sort(depths)[1:10]
```

```
##   AmzC3adltM.418849  AmzC30adltF.418737   AmzC9adltM.418480
##                 731                1025                1050
##  AmzC22chldM.418704  AmzC28chldM.418506 AmzC11adltF3.418391
##                1239                1439                1503
##       h35A.3.418524  AmzC31adltF.418549       h47A.1.418772
##                1548                1603                1635
##  AmzC2chldM1.418776
##                1729
```

```
gg.otus <- gg.otus[depths >= 1000,]
dim(gg.otus)
```

```
## [1]  36 746
```

Load mapping file

```
gg.map <- read.table('../data/globalgut/map.txt',sep='\t',head=T,row=1,check=F,comment='')
```

Ensure that mapping file and OTU table contain the sample samples in the same order

```
sample.ids <- intersect(rownames(gg.otus), rownames(gg.map))

# might as well put the samples in alphabetical order
sample.ids <- sort(sample.ids)

# in R you can subset using sample IDs or numerical indices. Most languages only use indices.
gg.otus <- gg.otus[sample.ids,]
gg.map <- gg.map[sample.ids,]
dim(gg.otus)
```

```
## [1]  36 746
```

```
dim(gg.map)
```

```
## [1] 36 22
```