Exploring the Data

Yoni Ackerman

October 25, 2016

Brief Summary

Data Shape + Meta Data

The dataset is made up of 101 samples, and 7234 different OTU features. Here is a data summary and meta data for the first OTU:

```
##                   taxonomy1                   taxonomy2
##               "k__Bacteria"         "p__Actinobacteria"
##                   taxonomy3                   taxonomy4
## "c__Actinobacteria (class)"        "o__Actinomycetales"
##                   taxonomy5                   taxonomy6
##        "f__Glycomycetaceae"             "g__Glycomyces"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      12    1106    2497    2595   22080
```

I haven't looked at how taxonomically similar all the OTU's are, but that's on my todo list.

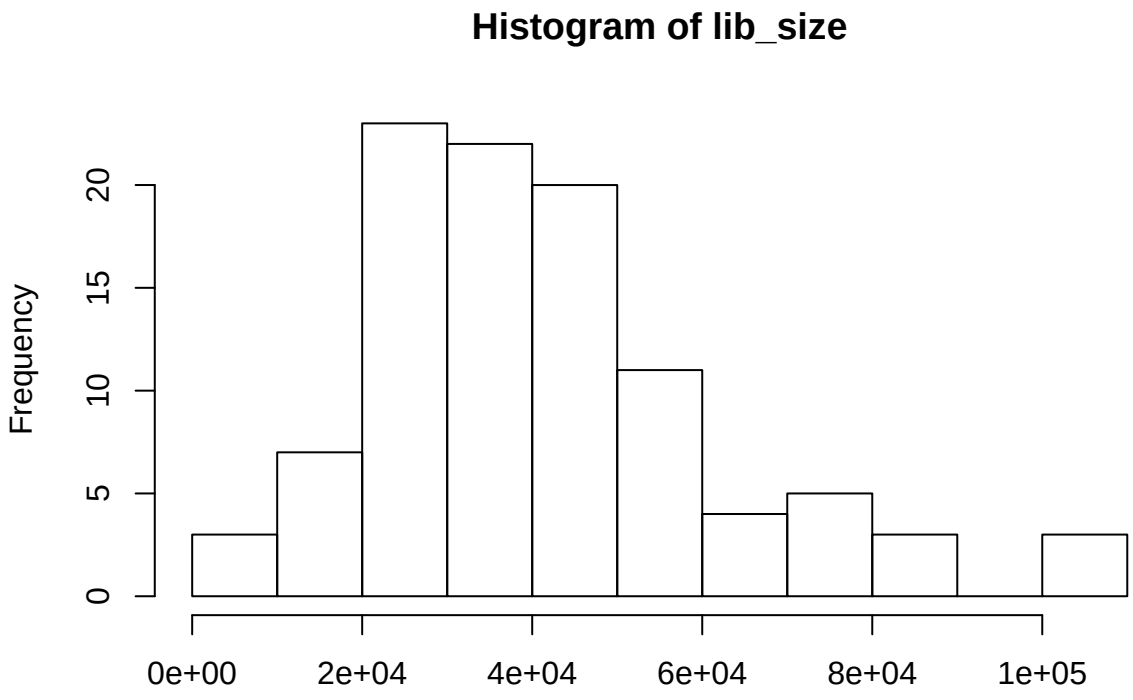
We've also got more information about the samples. Here are some of the sample id's:

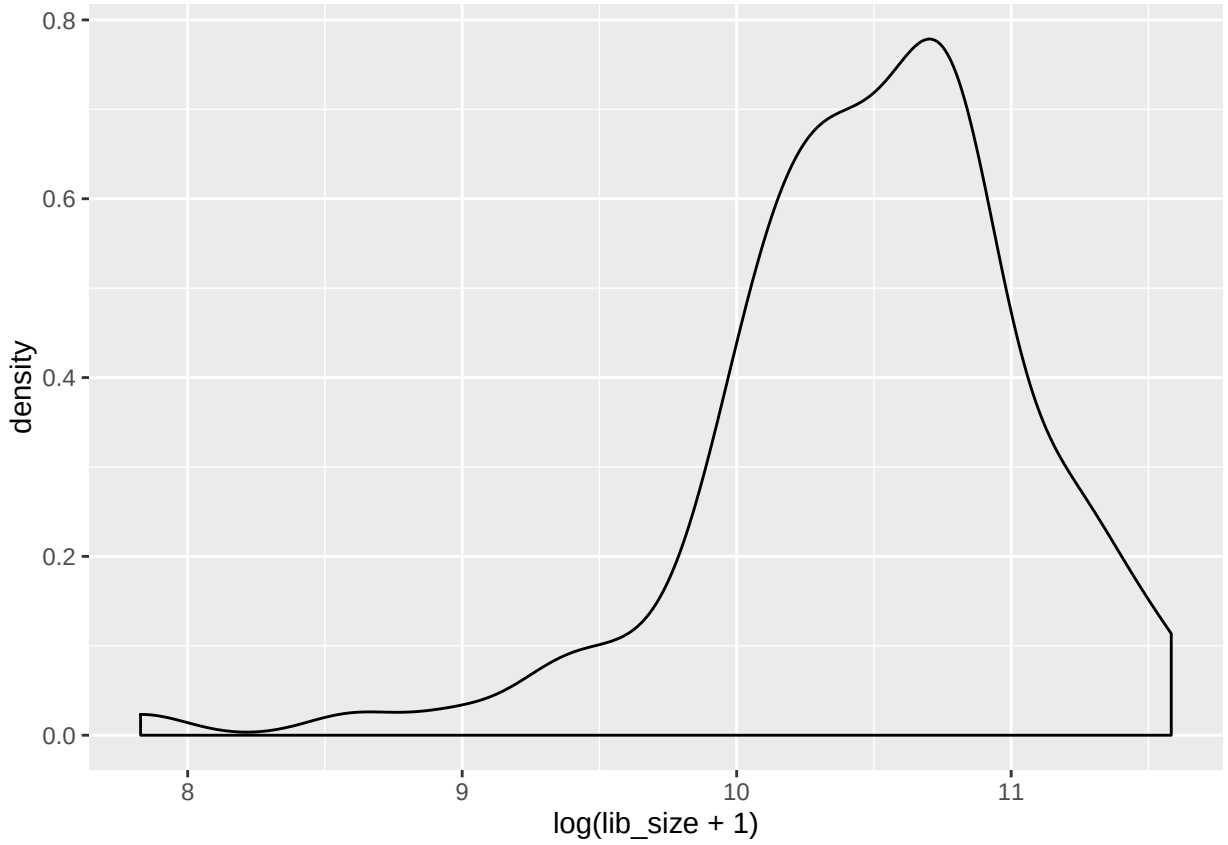
```
## [1] "TP1_B1_CTCC_R1_RH"   "TP1_B1_CTCC_R1_ROOT" "TP1_B1_CTCC_R1_S"
## [4] "TP1_B1_CTCC_R2_RH"   "TP1_B1_CTCC_R2_ROOT" "TP1_B1_CTCC_R2_S"
```

These give us information on Batch, Till, Covercrop, Replicate, and SampleType (Soil, Rhizosphere, or Root).

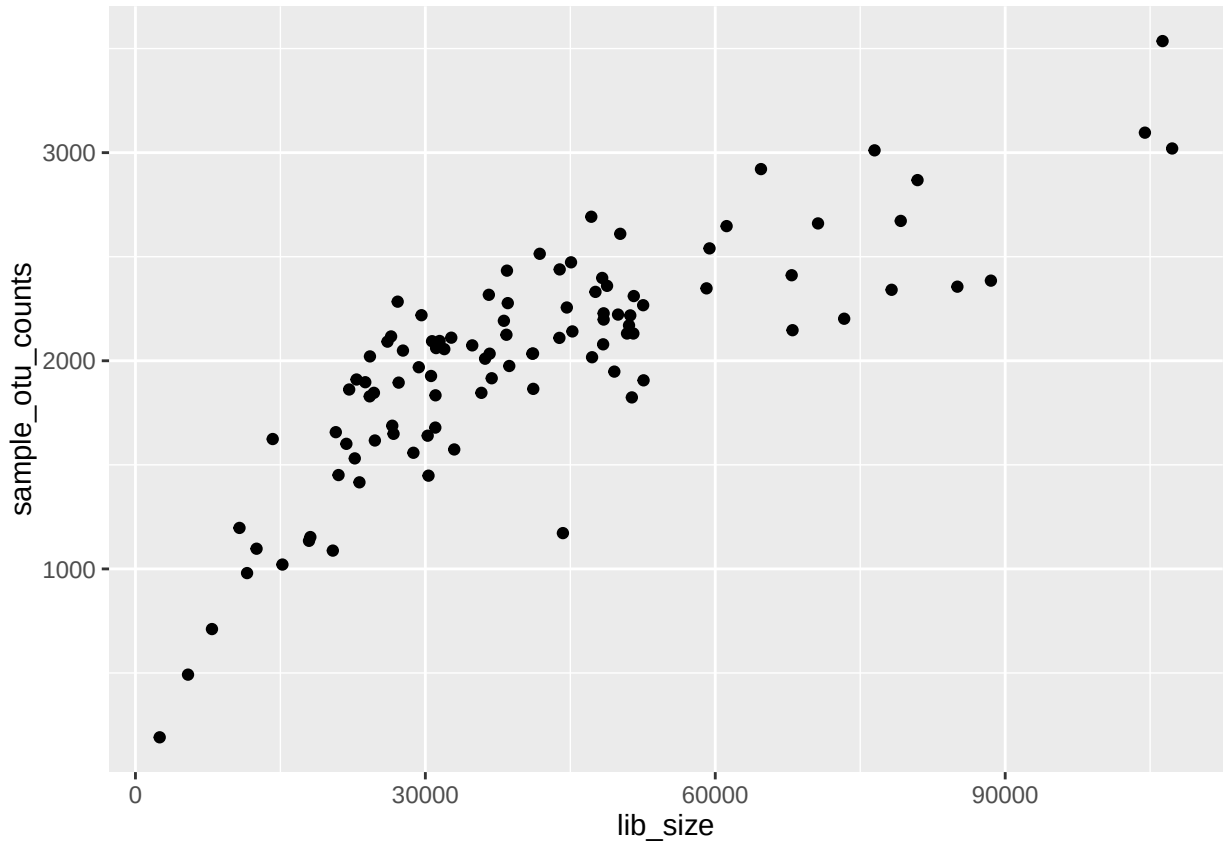
Library Sizes

The library size varies widely across the samples:

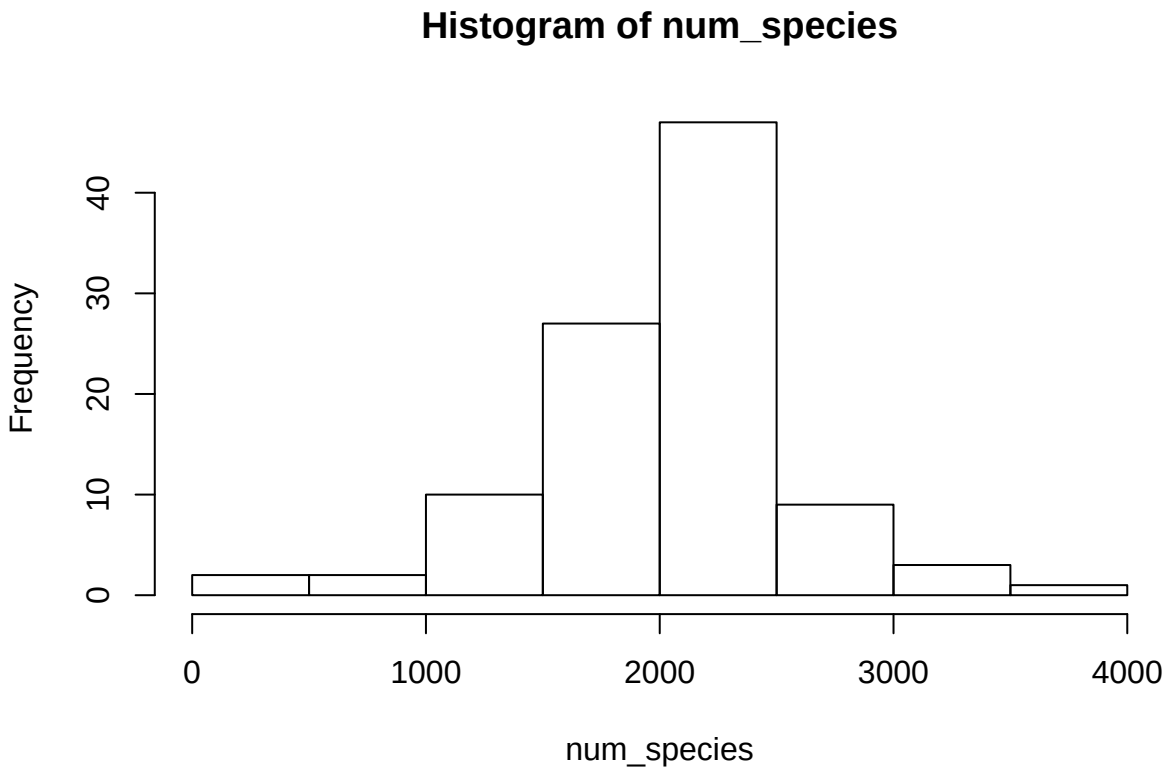




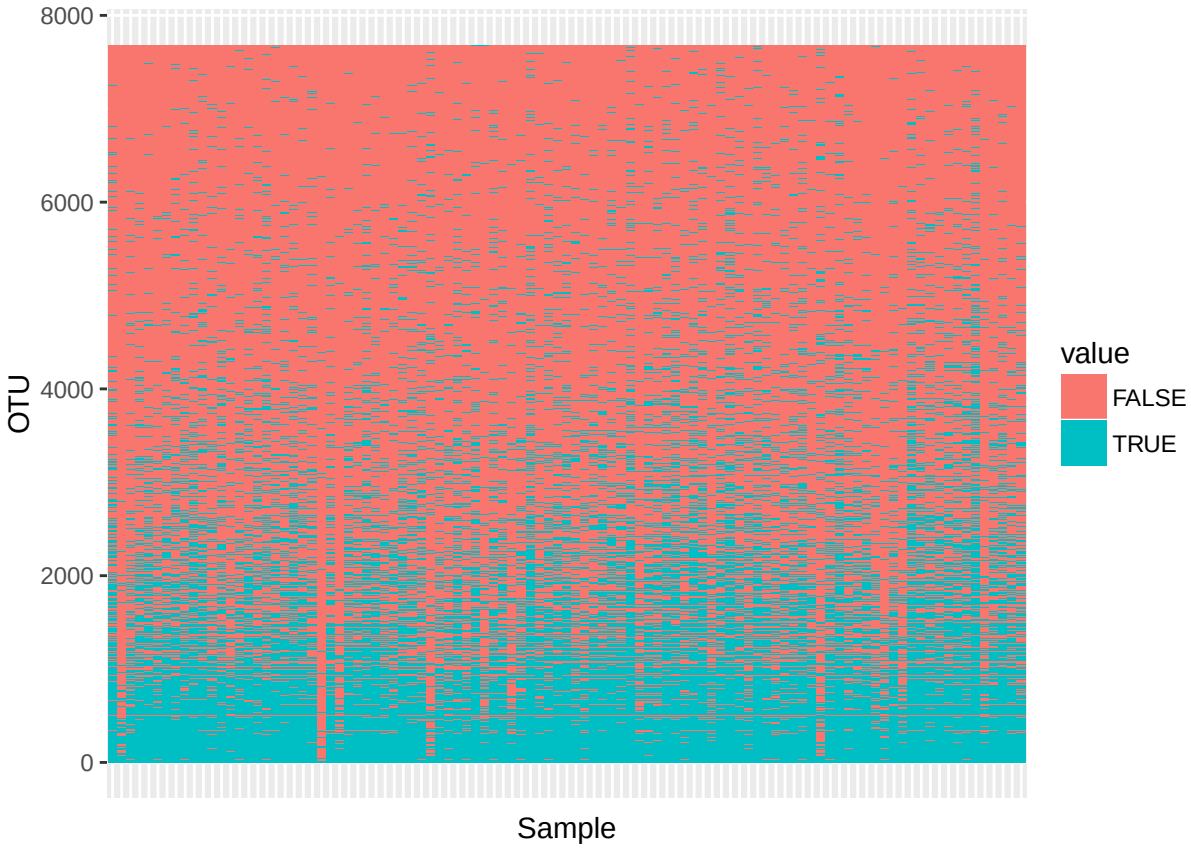
Library size vs. number of OTU's:



Along the same lines, we can ask about the distribution of unique species:



Sparcity



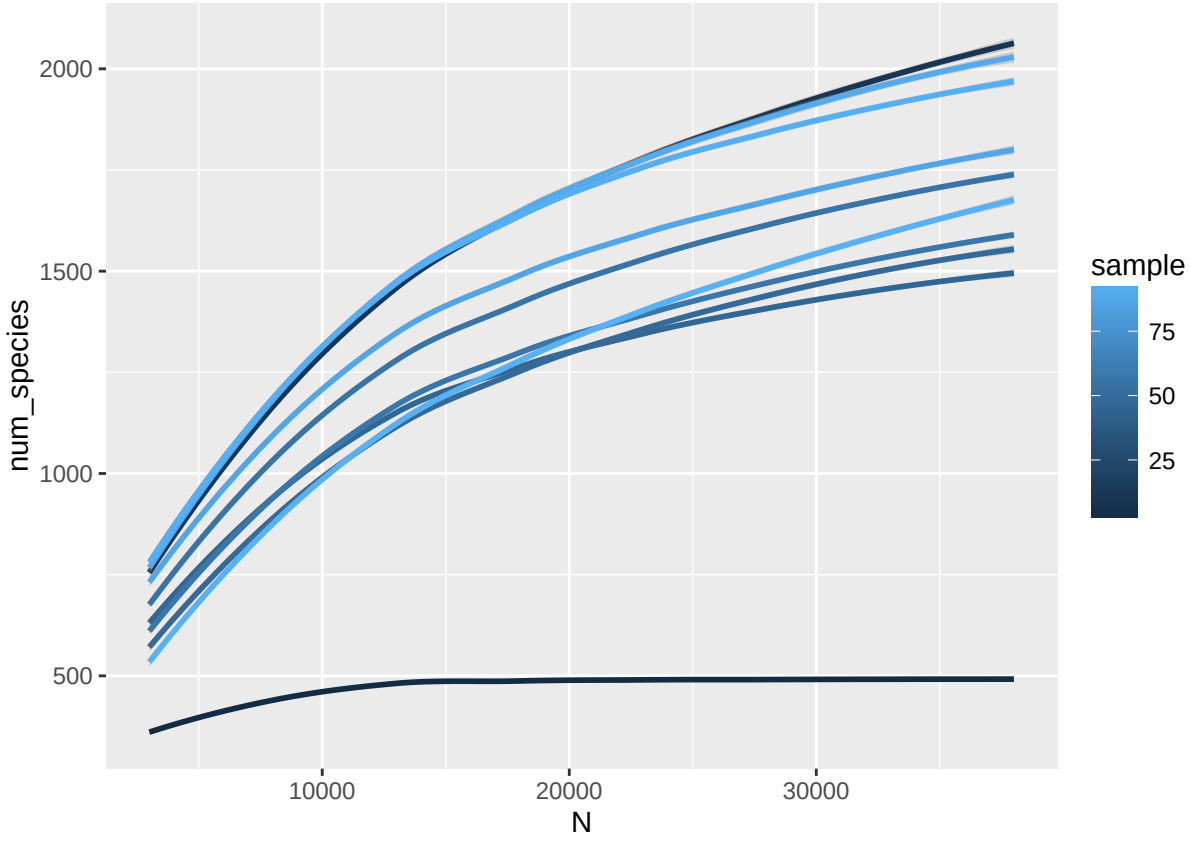
library size issues (normalization)

A common (bad) practice is rarefaction - down sampling each sample so that all samples have the same library size. Rarefaction is also a way to measure how well your samples capture the diversity of the species present. Here are some rarefaction curves:

```
set.seed(35)
i <- sample(1:101, 10)
N <- seq(from = 3000, to = 38000, by = 5000)
times <-  10
rarefaction_curves <- ldply(i, function(sample_i){
    ldply(1:times, function(iterate_j){
        num_unique <- sapply(N, function(rarefy_N){
            row <- biom_data[sample_i,]
            vals <- rep(1:7234, times = row)
            dwn_sample <- sample(vals,
                                 rarefy_N,
                                 replace = TRUE)
            length(unique(dwn_sample))
        })
        data.frame(sample = sample_i,
                   iterate = iterate_j,
                   N = N,
```

```
                    num_species = num_unique)
    })
})

ggplot(rarefaction_curves,
       aes(x = N,
           y = num_species,
           group = sample,
           color = sample)) +
    geom_smooth()
```



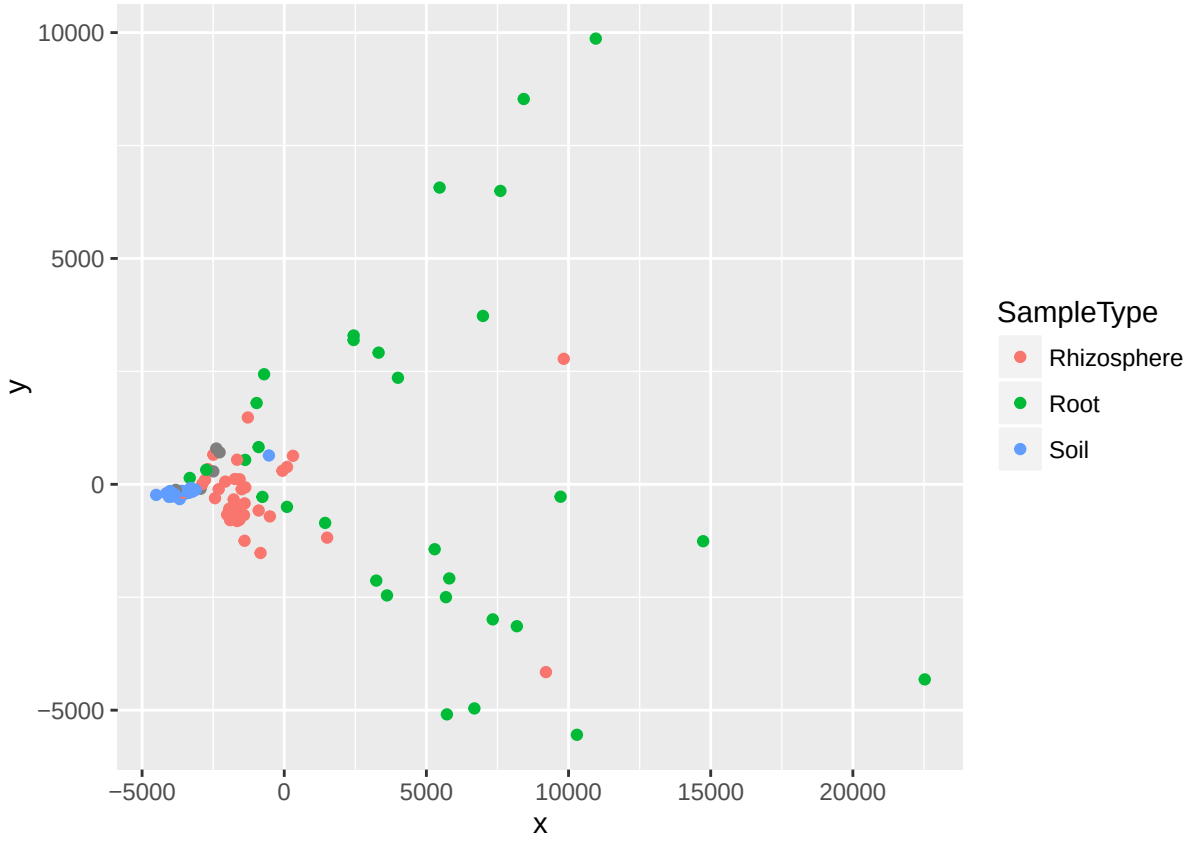
The problem is that rarefaction introduces artificial uncertainty. Furthermore, it means throwing away potentially large amounts of data.

An Example: MDS

Without rarefying:

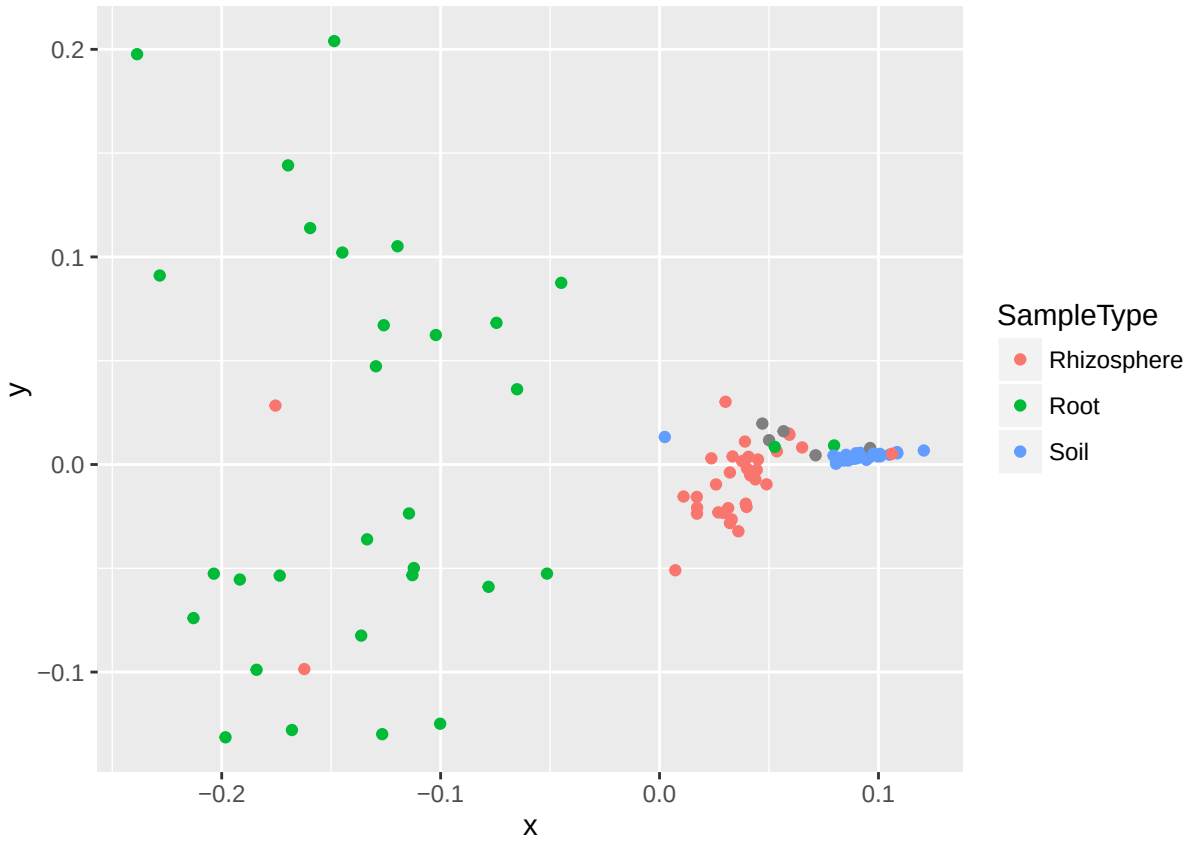
```
d <- dist(biom_data) # euclidean distances between the rows
fit <- cmdscale(d,eig=TRUE, k=2) # k is the number of dim
points <- as.data.frame(fit$points)
colnames(points) <- c("x", "y")
points <- cbind(all_data[, c("Replicate", "SampleType")], points)
```

```
ggplot(points, aes(x = x, y = y, color = SampleType)) +
    geom_point()
```



normalizing by library size:

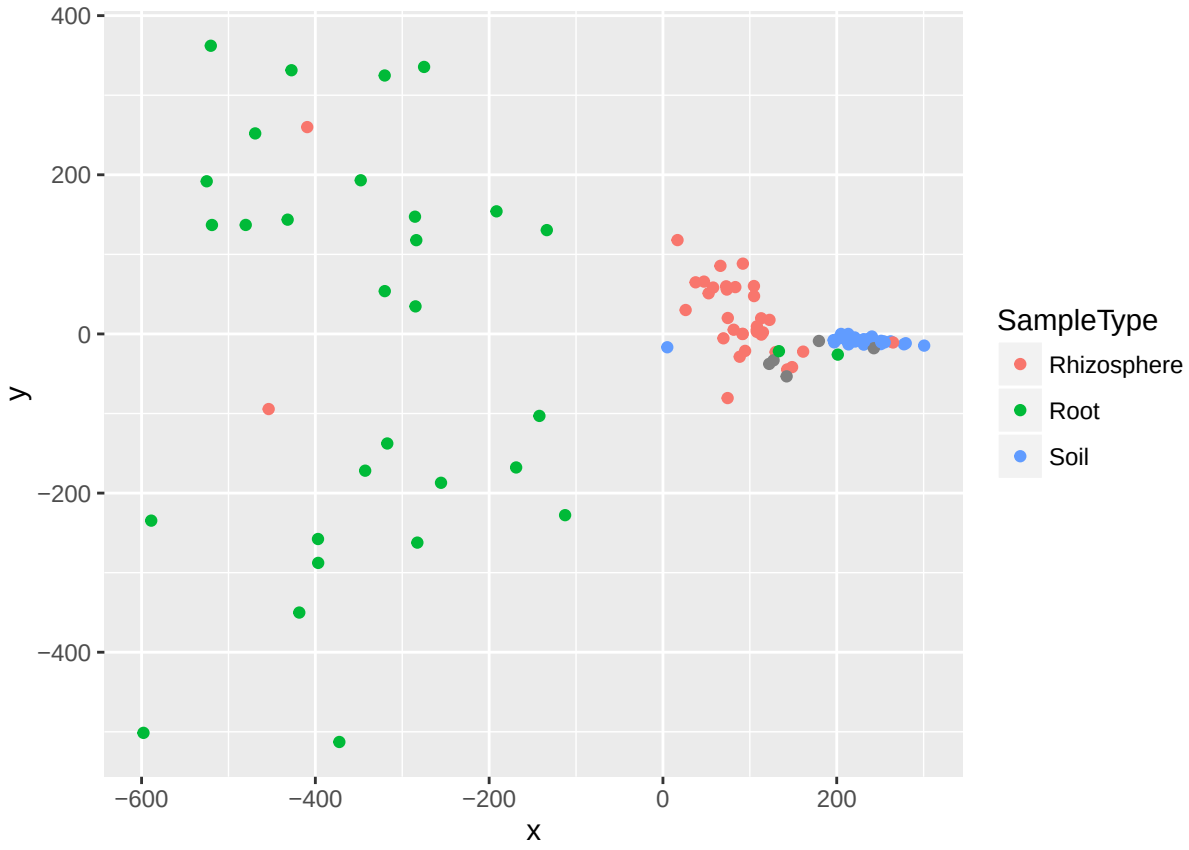
```
row_sums <- rowSums(biom_data)
ndata <- biom_data / row_sums
d <- dist(ndata)
fit <- cmdscale(d,eig=TRUE, k=2)
points <- as.data.frame(fit$points)
colnames(points) <- c("x", "y")
points <- cbind(all_data[, c("Replicate", "SampleType")], points)
ggplot(points, aes(x = x, y = y, color = SampleType)) +
    geom_point()
```



rarefying (down sampling to the smallest library size):

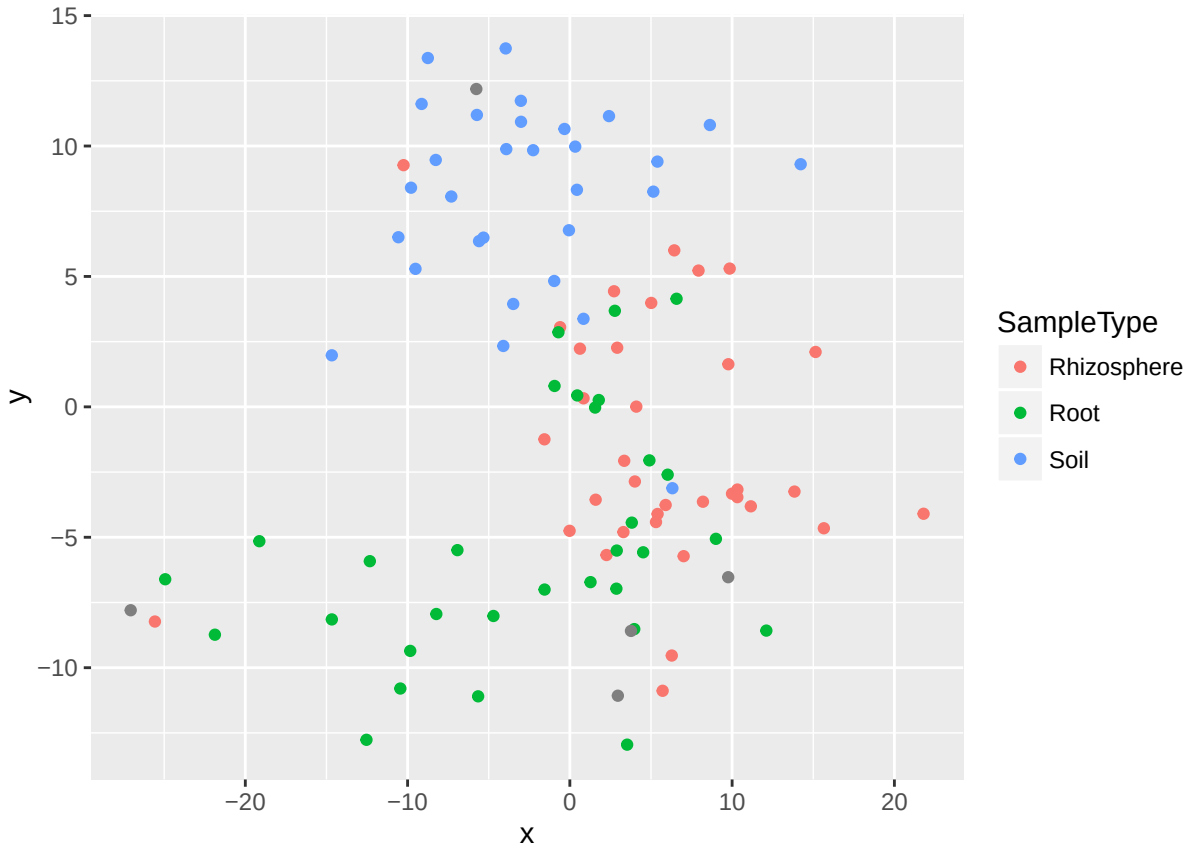
```
ds_data <- ldply(1:nrow(biom_data), function(i){
    row <- biom_data[i,]
    vals <- rep(1:7234, times = row)
    dwn_sample <- sample(vals,
                         2510, ## sample down to the smallest library size
                         replace = TRUE)
    dwn_sample <- table(dwn_sample)
    missing <- !(1:7234 %in% names(dwn_sample))
    missing_names <- (1:7234)[missing]
    missing_values <- rep(0, length(missing_names))
    names(missing_values) <- missing_names
    dwn_sample <- c(dwn_sample, missing_values)
    dwn_sample <- dwn_sample[order(as.integer(names(dwn_sample)))]
    ds_row <- dwn_sample
})

d <- dist(ds_data)
fit <- cmdscale(d,eig=TRUE, k=2)
points <- as.data.frame(fit$points)
colnames(points) <- c("x", "y")
points <- cbind(all_data[, c("Replicate", "SampleType")], points)
ggplot(points, aes(x = x, y = y, color = SampleType)) +
    geom_point()
```



Presence/Absence:

```
pa_data <- tmp_data > 0
d <- dist(pa_data)
fit <- cmdscale(d,eig=TRUE, k=2) # k is the number of dim
points <- as.data.frame(fit$points)
colnames(points) <- c("x", "y")
points <- cbind(all_data[, c("Replicate", "SampleType")], points)
ggplot(points, aes(x = x, y = y, color = SampleType)) +
    geom_point()
```


15
10
5
0
−5
−10
y
−20
−10
0
10
20
x
SampleType
Rhizosphere
Root
Soil