QBio5 :: Microbiome workshop

A glimpse into the microbial jungle in your mouth

A. Murat Eren[*†]
Evan Kieft[‡]

Our planet *is* microbial. The astonishing number of microbial organisms that occupy terrestrial and marine habitats of Earth represent a biomass that exceeds every living organism that can be seen by naked eye, combined. They can survive in a wide range of environmental and chemical gradients, so even the most extreme environments we can find on Earth have some microbial ambassadors, carrying the flag of life to places where you don't want to go. They also are the engine of our planet as they govern large and critical biogeochemical cycles that make Earth a habitable planet for much less talented organisms (such as ourselves) by doing the real tough jobs you don't want to do. Pretty much they are the best.

Our own body is microbial, too. Just to put things into perspective, for every human cell that make up our own body, there is one or more bacterial cells that live on us. Starting from the moment we are born, microbes are with us throughout our jouney in life and even a litle after that: they help us maintain our health by extracting energy from things we can't digest, by synthesizing vitamins or metabolizing xenobiotics for us, and help us return all the things we borrowed in pristine conditions so other things can be built from us. They are just beautiful like that.

Microbiologists who realized early on that **a complete understanding of life is only possible through a complete understanding of microbes** have been studying the evolution and ecology of microbes everywhere relentlessly for many decades. Of course, the emergence of advanced molecular and computational approaches had a huge influence on this quest, and allowed people like Meren and Evan to be relevant to fundamental questions of microbiology with their computational skills. **As a group (http://merenlab.org) we seek answers in very large sequencing datasets about microbial life**. How do they respond to changing environments? How do they evolve? What roles do they play in health and disease? To investigate these questions, we study all sorts of environments: from human gut to the surface ocean, from mosquito ovaries to sewage ecosystems.

Where does the oral cavity fit here? **The oral cavity is also colonized by bacteria** just like every other surface on your body. A complete microbial understanding of this environment has always been essential for medical reasons, but besides its immediate relevance for overall health, we believe **the oral cavity represents a fascinating environment to study the ecology of microbes**. Imagine how every microbial cell can go anywhere in the mouth due to the lack of any physical barriers, and continuous flow of saliva. But, their distribution is far from being random in that jungle of microbial life you all maintain in your mouths. And we are very curious to see whether we can make better sense of how they are distributed to later learn what makes them do that.

The purpose of this workshop is to give you a glimpse of that invisible jungle in your mouth by making sense of high-throughput sequencing data from the oral cavity using `R` and `ggplot`.

At the end of this workshop you will have an idea about how the community structures of naturally occurring microbes that live in a given environment can be studied with currently available molecular tools, sequencing technologies, and computational approaches. You will also gain more insights into the power of exploratory data visualization with `ggplot`, and the power of `R` in manipulating data.

When you are done here, you will know *whether the microbes live on your tongue are more similar to the ones that live on your cheek, or whether they are more similar to the ones that live on the tongue of the person next to you.*

[*]Department of Medicine, University of Chicago
[†]Marine Biological Laboratory
[‡]The Biophysical Sciences Program at the University of Chicago.

Setting the stage

The primary raw data we will be playing with throughout this tutorial come from the Human Microbiome Project (HMP, https://hmpdacc.org/), a National Institutes of Health (http://www.nih.gov) initiative that attempted to make sense of the 'normal microbiome' of healthy individuals. The HMP recruited many healthy volunteers, and collected multiple samples from each one of them to study microbes that lived in the healthy human gut, urogenitary tract, oral and nasal cavities, as well as skin.

Here we will focus only on the oral cavity to characterize the microbial communities of this particular environment (because the oral cavity is the best), and we will do this in a highly resolved manner using 'oligotypes'. The dataset we will re-analyze essentially comes from the supplementary tables of our 2014 study (http://www.pnas.org/content/111/28/E2875.short), which is available to you in PDF form in the `text` directory. In the same directory you can also find a copy of Carl Zimmer's take on our study, "The Zoo in the Mouth".

OK. First things first. We will need the following libraries throughout this tutorial for statistical analyses (`vegan`), re-formatting the input data (`reshape2`), and to visualize it (`ggplot2`):

```
library(vegan)
library(reshape2)
library(ggplot2)
```

If you are missing any of these libraries, you can install them using `install.packages("LIBRARY_NAME_HERE")` notation.

There are two data files for you to read in. The first one is the observation matrix that shows the distribution of each oligotype across each sample:

```
oligotypes <- read.table('../data/oligotypes.txt',
                         header = TRUE,
                         sep="\t")
```

The second data file contains data about our samples:

```
samples <- read.table('../data/samples.txt',
                      header = TRUE,
                      sep="\t")
```

Feel free to take a look at its format:

```
head(samples)
```

```
##             sample  individual environment site
## 1   s_147406386_BM s_147406386 ORAL_CAVITY   BM
## 2   s_147406386_HP s_147406386 ORAL_CAVITY   HP
## 3   s_147406386_KG s_147406386 ORAL_CAVITY   KG
## 4   s_147406386_PT s_147406386 ORAL_CAVITY   PT
## 5   s_147406386_ST s_147406386         GUT   ST
## 6 s_147406386_SUBP s_147406386 ORAL_CAVITY SUBP
```

Samples in our data describes two main environments:

```
levels(samples$environment)
```

```
## [1] "GUT"         "ORAL_CAVITY"
```

And more specifically, ten body sites:

```
levels(samples$site)
```

```
##  [1] "BM"   "HP"   "KG"   "PT"   "ST"   "SUBP" "SUPP" "SV"   "TD"   "TH"
```

While we have 148 individuals:

```
length(levels(samples$individual))
```

```
## [1] 148
```

We have a total of 1475 samples:

```
length(levels(samples$sample))
```

```
## [1] 1475
```

Fine. We have everything we need.

A visualization-driven exploration of the data

MDS

For this exploration, we're going to use multi-dimensional scaling, a commonly used technique to make sense of large dimensional data sets. In particular, MDS takes as input a "distance" matrix (say the distance between samples, sequences, etc.), and tries to find a projections onto few dimensions (typically 2) that can be used to find "clusters" of similar samples.

To see how this works, let's load some non-biological data.

```
load("../data/travel_times.RData")
```

The matrix `travel_time` contains the number of minutes it would take you to drive between two of the major US cities listed. You can take a look at the data by calling

```
View(travel_times)
```

Now we're going to apply MDS to the data, and find which two cities are "similar" (i.e., close to reach by car).

```
fit_mds <- cmdscale(travel_times, k = 2) # use two dimensions
```

Let's plot the results

```
cities_names <- rownames(travel_times)
plot(fit_mds)
text(fit_mds, labels = cities_names)
```

Not bad! Now, let's try with our microbial community.

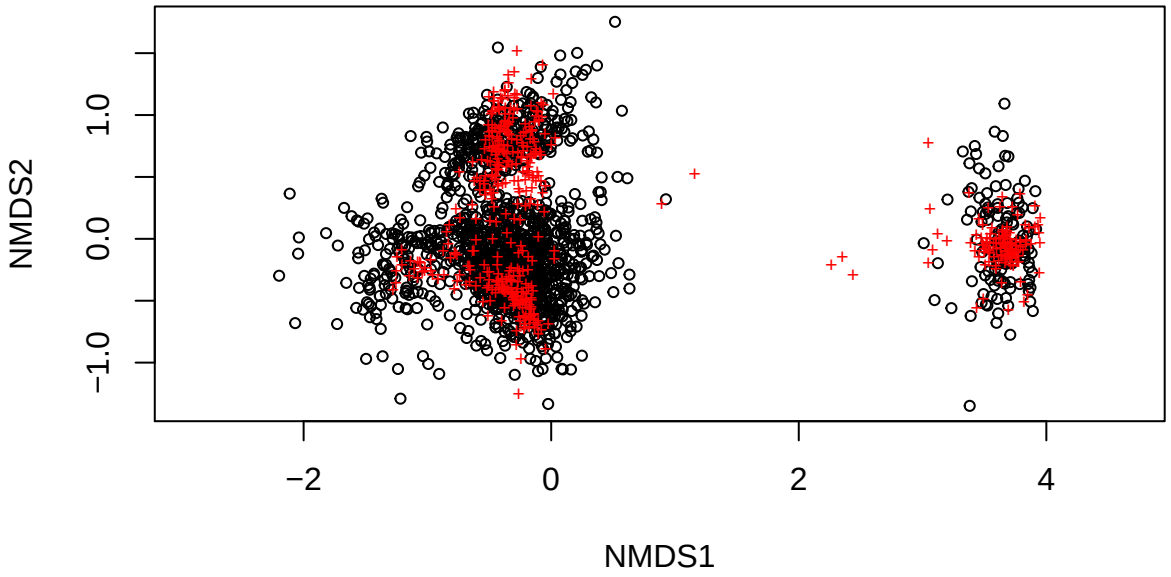
Back to the oral cavity

Here we create an ordination of our data using MDS:

```
# generate the mds object using the Morisita-Horn distance
# (this will take some time)
mds <- metaMDS(oligotypes[,-1], distance='horn')
```

We take a very quick look at the resulting ordination:

```
# show it
plot(mds)
```



We can all agree that this looks quite useless.

Instead of the `plot` function, we could use `ggplot` to have more control over our visualization needs by adding ad hoc information into our plots in an intuitive manner. But `ggplot` will not like the way `mds` object is formatted. But we can turn that object into a data frame rich with information:

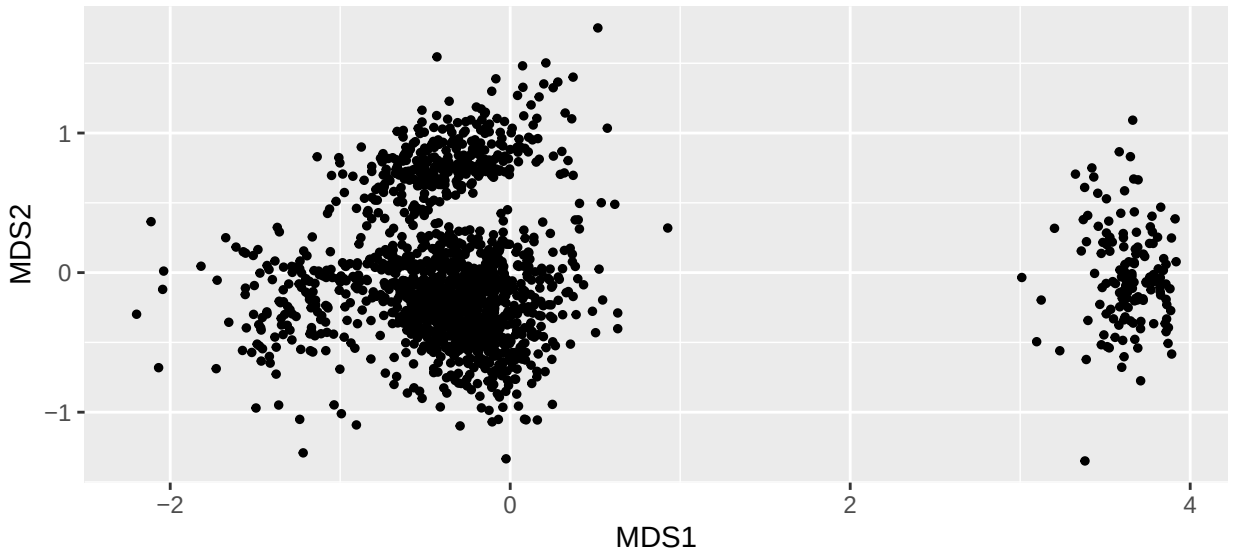
```
# generate a data frame
mds_df <- data.frame(MDS1 = mds$points[,1],
                     MDS2 = mds$points[,2],
                     individual=with(samples, get("individual")),
                     environment=with(samples, get("environment")),
                     site=with(samples, get("site")))

# take a peek
head(mds_df)
```

```
##         MDS1        MDS2  individual environment site
## 1 -0.9001296 -0.11533750 s_147406386 ORAL_CAVITY   BM
## 2 -0.6191344 -0.32741437 s_147406386 ORAL_CAVITY   HP
## 3 -1.2018615 -0.04347894 s_147406386 ORAL_CAVITY   KG
## 4 -0.1139542 -0.28778080 s_147406386 ORAL_CAVITY   PT
## 5  3.9113955  0.38500523 s_147406386         GUT   ST
## 6 -0.7747015  0.61157748 s_147406386 ORAL_CAVITY SUBP
```

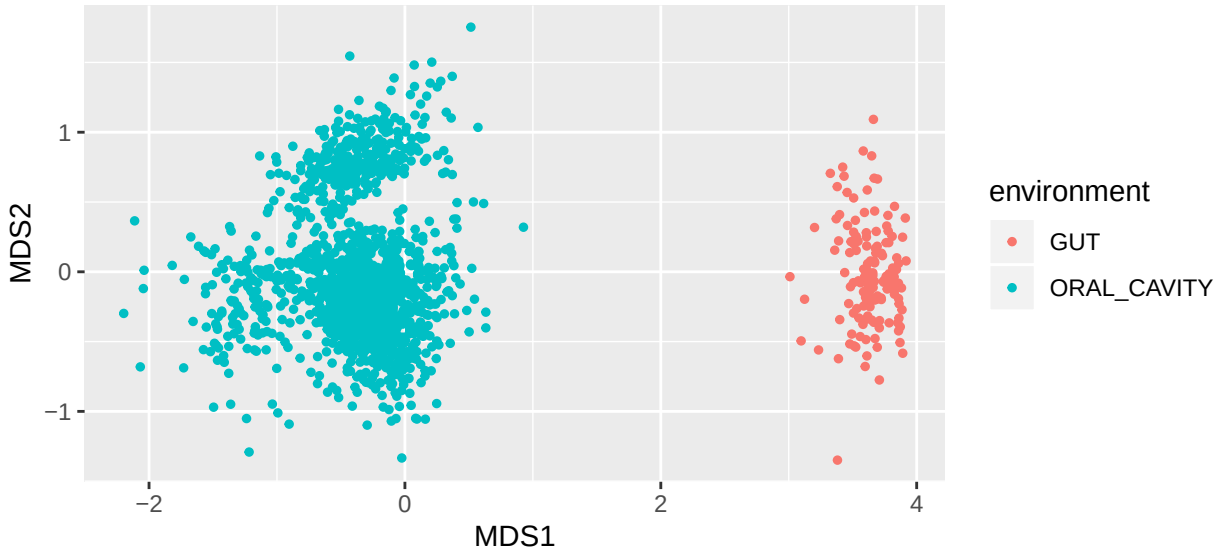
Well, this is more like it. Now we can take a look this with `ggplot`:

```
p <- ggplot(data = mds_df, aes(MDS1, MDS2))
p <- p + geom_point(size = 1)
p
```



This is not much better than the previous plot, but this time we can easily manipulate our visual objects. Let's say we color our points in this display by `environment` to ask this very question: *from the perspective of microbes, do our guts look like our mouths?*:

```
p <- ggplot(data = mds_df, aes(MDS1, MDS2))
p <- p + geom_point(aes(color=environment), size=1)
p
```



This is relieving.

Let's remove all gut samples to focus solely on oral microbes:

```
# take the subset of both data frames:
oral_oligotypes <- oligotypes[!grepl("_ST", oligotypes$sample), ]
oral_samples <- samples[samples$environment == "ORAL_CAVITY", ]

# set the factors straight:
oral_samples$sample <- factor(oral_samples$sample)
oral_samples$site <- factor(oral_samples$site)
```

Since we changed the shape of the data quite a bit, it is better to re-compute the ordination of our samples:

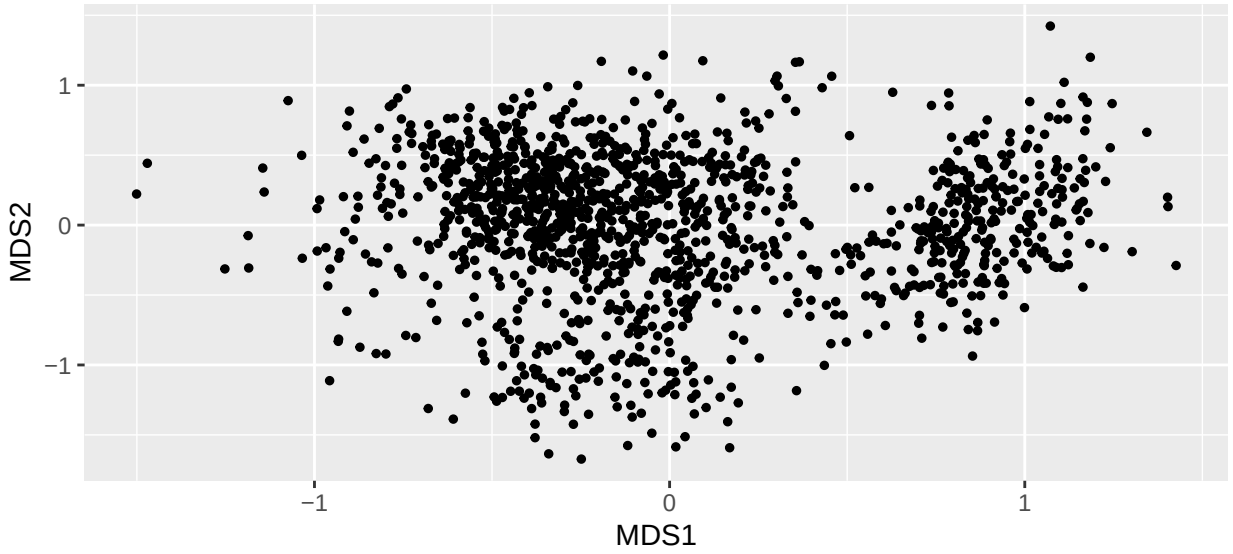
```
# new mds object:
oral_mds <- metaMDS(oral_oligotypes[,-1], distance='horn')

# generating a data frame from it:
oral_mds_df <- data.frame(MDS1 = oral_mds$points[,1],
                          MDS2 = oral_mds$points[,2],
                          individual=with(oral_samples, get("individual")),
                          site=with(oral_samples, get("site")))

# taking a quick look from it because why not:
head(oral_mds_df)
```

Alright! Let's take a quick look:

```
p <- ggplot(data = oral_mds_df, aes(MDS1, MDS2))
p <- p + geom_point(size=1)
p
```



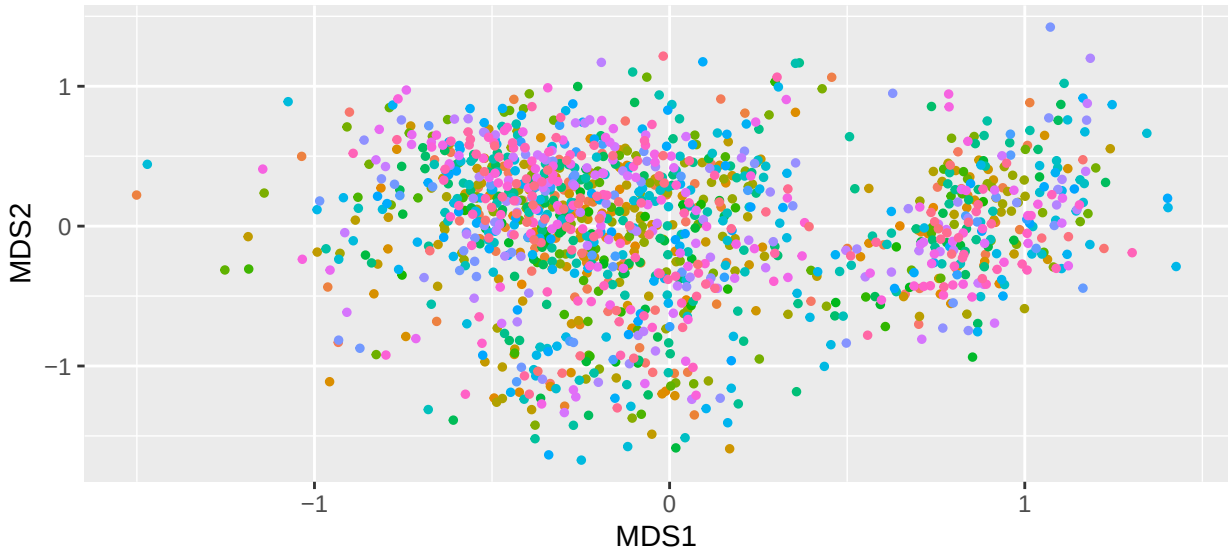
What chaos. What if we color based on individuals:

```
p <- ggplot(data = oral_mds_df, aes(MDS1, MDS2))
p <- p + geom_point(aes(color=individual), size=1)
p
```

)	s_158964549	s_159591683	s_160380657	s_160987560	s_370425937	s_763
1	s_159005010	s_159611913	s_160400887	s_161007791	s_404239096	s_763
5	s_159146620	s_159632143	s_160421117	s_161028021	s_414519462	s_763
5	s_159207311	s_159672603	s_160441347	s_161068481	s_432193348	s_763
5	s_159227541	s_159713063	s_160461578	s_161230322	s_441369442	s_763
5	s_159247771	s_159814214	s_160481808	s_161270782	s_451588811	s_763
5	s_159268001	s_159915365	s_160502038	s_161311242	s_492786515	s_764
5	s_159288231	s_160016515	s_160542498	s_161331472	s_514014184	s_764
5	s_159308461	s_160036745	s_160582958	s_161351702	s_517810313	s_764
5	s_159328691	s_160056975	s_160603188	s_161412393	s_533247696	s_764
7	s_159369152	s_160097436	s_160643649	s_161473083	s_604812005	s_764
7	s_159389382	s_160137896	s_160663879	s_161554003	s_612472597	s_764
7	s_159429842	s_160158126	s_160684109	s_206906765	s_638754422	s_764
7	s_159450072	s_160178356	s_160704339	s_208027353	s_650853796	s_764
3	s_159470302	s_160218816	s_160765029	s_246515023	s_668248235	s_764
3	s_159490532	s_160239046	s_160845950	s_257905678	s_682449369	s_764
9	s_159510762	s_160259276	s_160906640	s_295137534	s_686765762	s_764
9	s_159551223	s_160319967	s_160947100	s_336497421	s_737052003	s_764
9	s_159571453	s_160340197	s_160967330	s_370027359	s_763456073	s_764

Ouch. We don't see anything, because the legend takes the entire space. Let's disable the legend and try again:

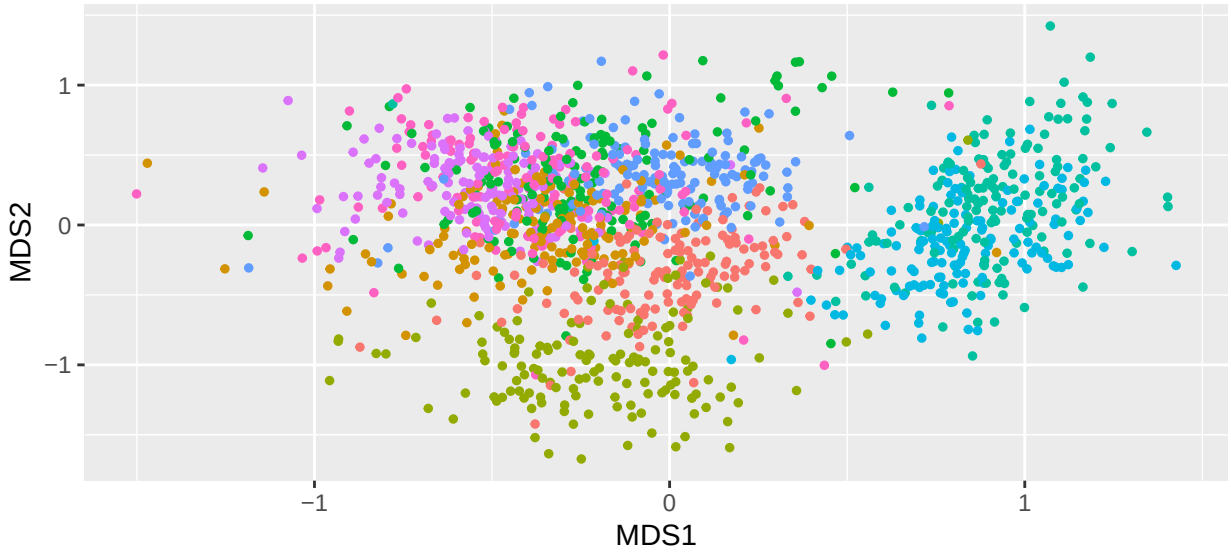
```
p <- ggplot(data = oral_mds_df, aes(MDS1, MDS2))
p <- p + geom_point(aes(color=individual), size=1)
p <- p + theme(legend.position="none")
p
```



Much better. But this doesn't seem to have any structure. Why?

OK. How about we color samples based on oral sites:

```
p <- ggplot(data = oral_mds_df, aes(MDS1, MDS2))
p <- p + geom_point(aes(color=site), size=1)
p <- p + theme(legend.position="none")
p
```



Aha!

What does this tell us?

(It would be great to do an ANOVA here to show oral sites explain a much more significant amount of variance in the dataset before moving on to the next chapter)

Making publication-ready visualizations with R

Let's say we wish to put some circles around our groups to help visualize their distribution and dispersal.

The function below will help us do that by returning all the `x` and `y` coordinates to draw a perfect ellipse on an ordination. It is coming from the depths of the library `vegan`, and here we simply are hacking it so we can use it to put ellipses on an ordination drawn by `ggplot`, rather than `vegan`:

```
veganCovEllipse <- function (cov_matrix, center){
  theta <- (0:100) * 2 * pi/100
  circle <- cbind(cos(theta), sin(theta))

  # here we have a perfect circle around the point zero, and the following line will
  # turn it into an ellipse by centering and multiplying that innocent circle with the
  # Choleski-decomposed input covariance matrix, which will represents the variation
  # among the distribution of samples that belong to a single group on the ordination
  # (this part will be much clear when you look at the for loop in the next step where
  # this function is called). if you are not familiar, the notation `%*%` is for
  # matrix multiplication. yes, you got it. this entire thing is absolute magic!
  ell <- t(center + t(circle %*% chol(cov_matrix)))

  return(as.data.frame(ell))
}
```

Using the magic up above, we will generate a new data frame, `ellipses_df`, to keep track of ellipses around our data points by going through each group in the for loop below:

```
ellipses_df <- data.frame()

# mighty for loop .. it looks ugly, but is very simple:
for(g in levels(oral_mds_df$site)){
  # get a smaller data frame just for site:
  s_df <- oral_mds_df[oral_mds_df$site==g, ]

  # calculate its center and its covariance matrix:
  center <- c(mean(s_df$MDS1), mean(s_df$MDS2))
  cov_matrix <- cov.wt(cbind(s_df$MDS1, s_df$MDS2))$cov

  # get the ellipse:
  ellipse <-veganCovEllipse(cov_matrix, center)

  # add the new ellipse to the data frame
  ellipses_df <- rbind(ellipses_df, cbind(ellipse, group=g))
}

# let's name the columns in our data frame more appropriately:
names(ellipses_df) <- c('x_coord', 'y_coord', 'group')
```

OK. You must be curious about what comes out of this black magic. Let's take a look at this new data frame:

```
head(ellipses_df)
```

```
##     x_coord    y_coord group
## 1 0.2267117 -0.1990243    BM
## 2 0.2262480 -0.1819480    BM
## 3 0.2248588 -0.1654502    BM
## 4 0.2225495 -0.1495960    BM
## 5 0.2193293 -0.1344480    BM
## 6 0.2152109 -0.1200659    BM
```

Don't let it fool you, this data frame has many entries since it is supposed to draw elliptic objects on our ordination:

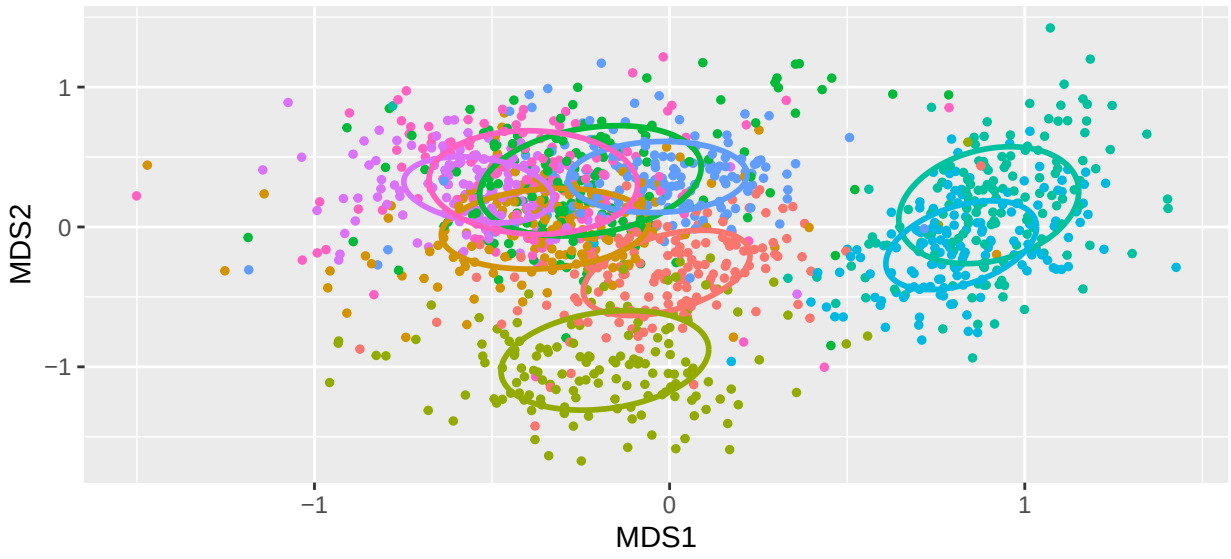
can you predict how many points should it have by looking at the function `veganCovEllipse`?

```
nrow(ellipses_df)
```

```
## [1] 909
```

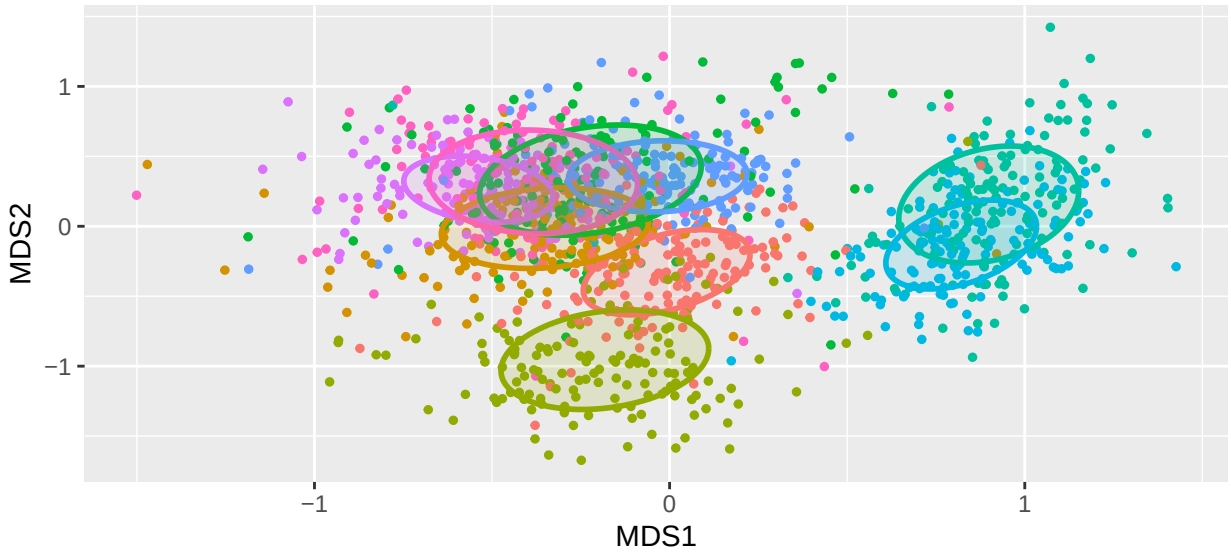
We still have the `ggplot` object in memory, let's add the data frame we just put together:

```
p <- p + geom_path(data=ellipses_df,
                   aes(x=x_coord, y=y_coord, colour=group),
                   size=1,
                   linetype=1)
p
```



There is always room for improvement:

```
p <- p + geom_polygon(data=ellipses_df,
                      aes(x=x_coord, y=y_coord, group=group, fill=group),
                      alpha=0.15)
p
```



It would have been great if we knew exactly what these ellipses represent. Let's add some labels at the center of each. For this, we first need to compute the group means of our samples:

```
oral_mds_group_means = aggregate(oral_mds_df[,1:2],
                                 list(group=with(oral_samples, get('site'))),
                                 mean)
```

Basically this is a new data frame that looks like this:

```
oral_mds_group_means
```

```
##   group         MDS1        MDS2
## 1    BM -0.008270321 -0.32853086
## 2    HP -0.350823982 -0.01267178
```

```
## 3    KG -0.182353841 -0.95353820
## 4    PT -0.222971056  0.32986660
## 5  SUBP  0.898656623  0.15570038
## 6  SUPP  0.821035565 -0.13045405
## 7    SV -0.033550862  0.35681564
## 8    TD -0.537879644  0.26518787
## 9    TH -0.383842482  0.31762440
```

And we can extend the `ggplot` object with one more layer:

```
p <- p + annotate("text",
                  x=oral_mds_group_means$MDS1,
                  y=oral_mds_group_means$MDS2,
                  label=oral_mds_group_means$group,
                  size=5,
                  fontface = 2)
p
```