Bacterial communities on classroom surfaces

Demo

The data used here are a small subset (first 20,000 quality-filtered sequences) of those used in Meadow et al. 2014. Microbiome 2:7 (Meadow et al. 2014).

This sequence dataset was processed using QIIME 1.8 (Caporaso et al. 2010) with a default MacQIIME installation http://www.wernerlab.org/software/macqiime. Scripts for processing raw data are in `../QIIME/` folder. To pick OTUs in that folder, you will execute the `pickTheseOTUs.sh` script sitting in that folder. This script wants to run MacQIIME, so if you are not using MacQIIME, you'll probably need to alter the top line.

```
# pick OTUs using the script in the QIIME folder.
./pickTheseOTUs
```

Getting started and importing data to R

To make things reproducible, the first step is to set the random number generator. R's random number generator is not actually random, but designed to look random while still being reproducible. Enter some integer - doesn't really matter what - in the `set.seed` command, and the results should turn out identical each time.

```
set.seed(42)
```

Load `phyloseq` to handle QIIME output files, and `vegan` and `labdsv` for multivariate ecology stats (McMurdie and Holmes 2013; Oksanen et al. 2011; Roberts 2010). Also load `xtable` package to convert tables to latex or html (Dahl 2013).

```
library(phyloseq)
library(vegan)
library(labdsv)
library(xtable)
```

setwd('~/Dropbox/SLP_Teaching/Rmd')

First use the `phyloseq` package to gracefully bring big QIIME/JSON-format dataset into R. This saves lots of code and testing, and also avoids having to change the file headers by hand.

```
surfaceTablePhyloseq <- import_biom("otu_table.biom", parseFunction = parse_taxonomy_greengenes)
surfaceMapPhyloseq <- import_qiime_sample_data("map.txt")
```

Once `phyloseq` has done the heavy lifting to input data, we can extract the parts we want. First is the OTU table. One sample gets excluded here since it was an internal control that is not used in this script. Then print out a bit to make sure it looks as expected.

```
surfaceTable.tmp <- t(otu_table(surfaceTablePhyloseq))
surfaceTable.tmp <- surfaceTable.tmp[!(row.names(surfaceTable.tmp) == "Swab.162.61"),
    ]
surfaceTable <- as(surfaceTable.tmp, "matrix")
surfaceTable[1:5, 1:5]
```

```
##             838843 259732 127012 185100 131115
## Swab.162.9       1      0      0      0      0
## Swab.162.58      1      0      0      0      0
## Swab.162.5       1      0      0      0      0
## Swab.162.16      0      1      0      0      0
## Swab.162.20      0      1      0      0      0
```

Extract the mapping file. This was in QIIME format, so the comment line needs to be removed to take out out of its `phyloseq` object.

```
surfaceMap <- data.frame(surfaceMapPhyloseq)[-1, ]
head(surfaceMap)
```

```
##            X.SampleID BarcodeSequence LinkerPrimerSequence           Study
## Swab.162.1 Swab.162.1    AGCTTACTAATG   TACNVGGGTATCTAATCC lillis_surfaces
## Swab.162.2 Swab.162.2    AGCTTACTGTTA   TACNVGGGTATCTAATCC lillis_surfaces
## Swab.162.3 Swab.162.3    AGCTTACATGTA   TACNVGGGTATCTAATCC lillis_surfaces
## Swab.162.4 Swab.162.4    AGCTTACACATC   TACNVGGGTATCTAATCC lillis_surfaces
## Swab.162.5 Swab.162.5    AGCTTACCTTAG   TACNVGGGTATCTAATCC lillis_surfaces
## Swab.162.6 Swab.162.6    AGCTTACGACTA   TACNVGGGTATCTAATCC lillis_surfaces
##            SurfaceType xcoord ycoord location location2 tier Description
## Swab.162.1        wall      6      1    south       low wall  Swab.162.1
## Swab.162.2        wall      6      1    south      high wall  Swab.162.2
## Swab.162.3        wall     20      1    south       low wall  Swab.162.3
## Swab.162.4        wall     20      1    south      high wall  Swab.162.4
## Swab.162.5       floor     17     11       f7         7  mid  Swab.162.5
## Swab.162.6       floor     13      5      f10        10 back  Swab.162.6
```

The taxonomic assignments are embedded in the OTU table. The output is a really convenient table with OTU numeric IDs as row names.

```
surfaceTaxa <- data.frame(tax_table(surfaceTablePhyloseq))
head(surfaceTaxa)
```

```
##          Kingdom         Phylum                 Class             Order
## 838843  Bacteria Proteobacteria   Alphaproteobacteria  Rhodospirillales
## 259732  Bacteria Proteobacteria   Alphaproteobacteria   Caulobacterales
## 127012  Bacteria  Bacteroidetes            Cytophagia      Cytophagales
## 185100  Bacteria Proteobacteria   Deltaproteobacteria Bdellovibrionales
## 131115  Bacteria Proteobacteria   Gammaproteobacteria   Pseudomonadales
## 4375688 Bacteria Proteobacteria Epsilonproteobacteria Campylobacterales
##                     Family         Genus       Species
## 838843    Acetobacteraceae          <NA>          <NA>
## 259732    Caulobacteraceae Brevundimonas      diminuta
## 127012       Cytophagaceae  Hymenobacter          <NA>
## 185100  Bacteriovoracaceae          <NA>          <NA>
## 131115       Moraxellaceae Acinetobacter rhizosphaerae
## 4375688 Campylobacteraceae Campylobacter          <NA>
```

Check to make sure things line up

After extracting separate objects, a few quick tests to make sure everything looks as expected. R does not check to make sure row names match, so you always have to.

```
identical(row.names(surfaceTaxa), colnames(surfaceTable))
```

```
## [1] TRUE
```

So all OTUs are present in both the OTU table and the taxonomic info table. And they are in the same order.

```
identical(sort(row.names(surfaceMap)), sort(row.names(surfaceTable)))
```

```
## [1] TRUE
```

All of the row names in the mapping file also match with the row names of the OTU table. So all of the samples are there, but they are not in the same order - notice the `sort` commands used above.

Rarefy to even sampling depth

Add up observations in each sample. For analysis like this, we should rarefy to even sampling depth so some samples are not biased just by having more or fewer observations.

```
sort(rowSums(surfaceTable), decreasing = FALSE)
```

```
##  Swab.162.7 Swab.162.26  Swab.162.6  Swab.162.1  Swab.162.2 Swab.162.27
##         115         129         135         145         151         163
## Swab.162.31  Swab.162.3  Swab.162.5 Swab.162.22  Swab.162.8 Swab.162.25
##         163         164         184         187         190         201
##  Swab.162.4 Swab.162.35 Swab.162.30 Swab.162.29 Swab.162.38 Swab.162.23
##         206         214         217         219         221         223
## Swab.162.19 Swab.162.32 Swab.162.51 Swab.162.42 Swab.162.36 Swab.162.34
##         233         248         249         279         280         283
## Swab.162.28 Swab.162.10 Swab.162.41 Swab.162.17 Swab.162.18 Swab.162.58
##         283         297         298         304         304         308
## Swab.162.40 Swab.162.21 Swab.162.33 Swab.162.57 Swab.162.59 Swab.162.45
##         312         315         321         322         324         325
## Swab.162.44  Swab.162.9 Swab.162.43 Swab.162.37 Swab.162.12 Swab.162.46
##         329         332         332         337         354         355
## Swab.162.20 Swab.162.49 Swab.162.24 Swab.162.39 Swab.162.14 Swab.162.15
##         356         358         366         371         378         378
## Swab.162.50 Swab.162.56 Swab.162.55 Swab.162.47 Swab.162.13 Swab.162.54
##         388         390         396         398         420         421
## Swab.162.60 Swab.162.11 Swab.162.48 Swab.162.52 Swab.162.53 Swab.162.16
##         437         455         482         495         511         552
```

It looks like we can cut them all off at 100 sequences, and not lose any samples to rarefaction. The other nice thing is that the counts in each cell double as percent counts.

```
tab <- rrarefy(surfaceTable, 100)
```

Fix a few things in the mapping table, and sort by sample name

Since the OTU table is sorted out for now, the map, or metadata table, can be sorted by the row names of the OTU table. Then colors get added by name for easy plotting later.

```
map <- surfaceMap[row.names(tab), ]
map$color <- "wheat"
map$color[map$SurfaceType == "floor"] <- "chocolate3"
map$color[map$SurfaceType == "chair"] <- "darkslateblue"
map$color[map$SurfaceType == "desk"] <- "goldenrod3"
```

When the samples were being processed initially, zeros were accidentally left out of single digit counts. So a sample named `Swab.162.2` actually gets sorted *after* `Swab.162.10`. Understandable but unacceptable. So run through and fix the offending names in a separate column that will act as a sorting index. It is probably not a good idea to mess with the actual row names since that could cause problems downstream when dealing with sequencing files or other previous versions of the data. First step is to remove one tiny piece of `phyloseq` baggage. The last command resorts the map by this new column and also cuts out some of the columns we won't use.

```
names(map)[1] <- gsub("X.", "", names(map)[1])

map$sortID <- as.character(map$SampleID)
for (i in 1:nrow(map)) {
    if (nchar(map$sortID[i]) == 10) {
        map$sortID[i] <- gsub("162.", "162.0", map$sortID[i])
    }
}

map <- map[order(map$sortID), c("sortID", "SurfaceType", "xcoord", "ycoord",
    "color")]
```

Since the map is final, one last step to reconcile the OTU table to the new mapping table row order. The same command also strips out OTUs that didn't make the rarefaction cut. Then reconcile the taxonomy table to the new trimmed OTU table, and everything is ready for analysis.

```
tab <- tab[row.names(map), which(colSums(tab) > 0)]
taxa <- surfaceTaxa[colnames(tab), ]
head(taxa)
```

```
##          Kingdom         Phylum                 Class             Order
## 838843  Bacteria Proteobacteria   Alphaproteobacteria  Rhodospirillales
## 259732  Bacteria Proteobacteria   Alphaproteobacteria   Caulobacterales
## 127012  Bacteria  Bacteroidetes            Cytophagia      Cytophagales
## 4375688 Bacteria Proteobacteria Epsilonproteobacteria Campylobacterales
## 4444760 Bacteria Actinobacteria        Actinobacteria   Actinomycetales
## 829373  Bacteria Actinobacteria        Actinobacteria   Actinomycetales
##                     Family          Genus  Species
## 838843    Acetobacteraceae           <NA>     <NA>
## 259732    Caulobacteraceae  Brevundimonas diminuta
## 127012       Cytophagaceae   Hymenobacter     <NA>
## 4375688 Campylobacteraceae  Campylobacter     <NA>
## 4444760     Micrococcaceae           <NA>     <NA>
## 829373  Pseudonocardiaceae Pseudonocardia     <NA>
```

```
dim(taxa)
```

```
## [1] 916   7
```

Analysis

To compare communities, make a matrix of pairwise multivariate distances (thus calculating beta-diversity). There are dozens of choices. The Canberra metric tends to work really well when communities share their most abundant OTUs, but have the strongest differences in a subset of relatively rare OTUs. Since we expect that to be the case in this dataset, the Canberra metric will be used here. Legendre & Legendre's *Numerical Ecology* is a terrific reference for choosing beta-diversity metrics that are appropriate for each problem.

```
distCanberra <- vegdist(tab, "canberra")
```

That distance matrix can go directly into many different multivariate analysis functions. To visualize the potential differences among sample types (walls, desks, floors and chairs), non-metric multidimensional scaling (NMDS) tends to be a mathematically satisfying visualization solution. There are several different ways to create an NMDS in R. This function is in the `labdsv` package, and uses random starting positions for all points before trying to fit the most parsimonious dissimilarity solution. Since we used `set.seed` at the top, it is not entirely random, but useful nonetheless.

```
nmdsCanberra <- bestnmds(distCanberra)
```

The plot will give us an indication of whether we should use discriminant analysis to test for differences among sample types.

```
par(mar = c(5, 4, 1, 5), las = 1, xpd = TRUE)
plot(nmdsCanberra, pch = 21, cex = 2, bg = map$color)
legend(par()$usr[2], par()$usr[4], pch = 21, pt.cex = 1.5, legend = c("walls",
    "floors", "chairs", "desks"), pt.bg = unique(map$color), bty = "n")
```

There is an apparent clustering by sample type (color in this case), so we should test to see if it is statistically worth discussing. The `adonis` function performs permutational multivariate analysis of variance (PERMANOVA), using 999 iterations as a default. The iterative nature is a must since our pairwise sample distances are technically not independent. Thus each iteration picks a few of them and test for a difference. Since we are only running 999 iterations, we can't reasonably report p-values lower than 0.001, since that is 1/1000.

```
surfaceTypeModel <- adonis(distCanberra ~ map$SurfaceType)$aov.tab
# print(xtable(surfaceTypeModel), type='html')
surfaceTypeModel
```

```
## Terms added sequentially (first to last)
##
##                 Df SumsOfSqs MeanSqs F.Model    R2 Pr(>F)
## map$SurfaceType  3      2.12   0.708    1.77 0.086  0.001 ***
## Residuals       56     22.44   0.401         0.914
## Total           59     24.57                 1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

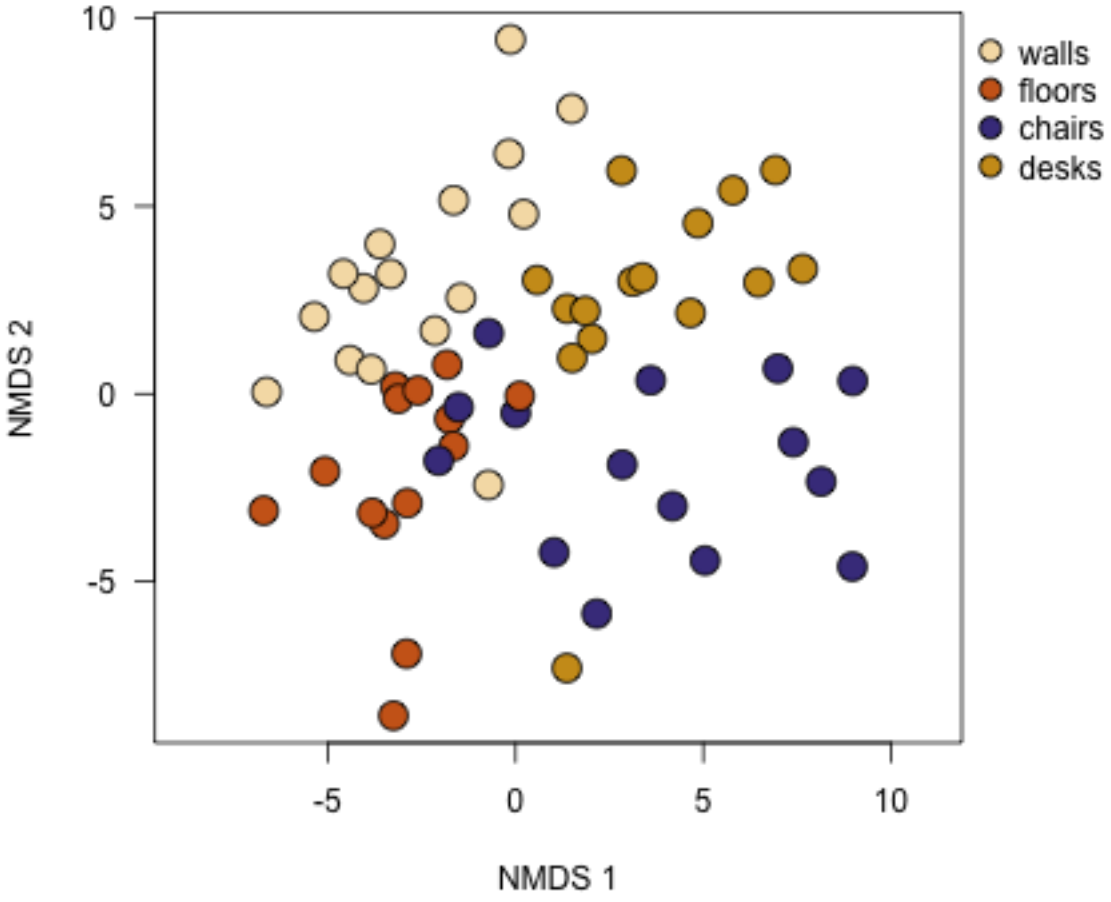

Figure 1: Samples cluster by the type of surface.

Yes - quite significant. So there is one reasonably strong result, but the R^2 value is pretty weak. That tells us that sample type certainly matters, but there is still lots of variability that cannot be explained just by that factor.

So we should check for a quasi-distance-decay relationship. This is the sort of pattern we see in just about every ecosystem with most forms of life. We even found this to be a stong predictor in the dust sampled from the entire building (Kembel et al. 2014). So we can use the x and y coordinates as a map of samples, and then calculate the Euclidean pairwise distance between all samples. Then that goes through a mantel test to determine if these distance are correlated with the community distances.

```
distSpatial <- dist(data.frame(map$xcoor, map$ycoor))
mantel(distCanberra, distSpatial)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = distCanberra, ydis = distSpatial)
##
## Mantel statistic r: -0.0401
##       Significance: 0.83
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0620 0.0765 0.0905 0.1051
##
## Based on 999 permutations
```

No - not even close. So proximity to other samples doesn't matter. But we are testing this with all sample types together. Is it still unimportant if each sample type is considered independently?

```
chair <- which(map$SurfaceType == "chair")
wall <- which(map$SurfaceType == "wall")
desk <- which(map$SurfaceType == "desk")
floor <- which(map$SurfaceType == "floor")

testMantel <- function(these) {
    mantel(vegdist(tab[these, ], "canberra"), dist(data.frame(map$xcoor, map$ycoor)[these,
        ]))
}

testMantel(chair)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = vegdist(tab[these, ], "canberra"), ydis = dist(data.frame(map$xcoor,      map$ycoor)[t
##
## Mantel statistic r: 0.0431
##       Significance: 0.33
##
## Upper quantiles of permutations (null model):
```

```
##   90%   95% 97.5%   99%
## 0.138 0.178 0.218 0.254
##
## Based on 999 permutations
```

```
testMantel(wall)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = vegdist(tab[these, ], "canberra"), ydis = dist(data.frame(map$xcoor,      map$ycoor)[tl
##
## Mantel statistic r: -0.032
##       Significance: 0.63
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0926 0.1342 0.1633 0.1976
##
## Based on 999 permutations
```

```
testMantel(desk)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = vegdist(tab[these, ], "canberra"), ydis = dist(data.frame(map$xcoor,      map$ycoor)[tl
##
## Mantel statistic r: 0.115
##       Significance: 0.19
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.170 0.212 0.256 0.298
##
## Based on 999 permutations
```

```
testMantel(floor)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = vegdist(tab[these, ], "canberra"), ydis = dist(data.frame(map$xcoor,      map$ycoor)[tl
##
## Mantel statistic r: -0.111
##       Significance: 0.81
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
```

```
## 0.161 0.218 0.259 0.308
##
## Based on 999 permutations
```

No. Not for any of the four surfaces.

So it looks like the type of surface, potentially as a proxy for human contact, explains a significant amount of variation, in the microbial communities on those surfaces, but their proximity to each other around the room doesn't matter at all.

References

Caporaso, J. G., J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, et al. 2010. "QIIME allows analysis of high-throughput community sequencing data." *Nature Methods* 7: 335–336.

Dahl, David B. 2013. *xtable: Export tables to LaTeX or HTML*. http://cran.r-project.org/package=xtable.

Kembel, Steven W., James F. Meadow, Timothy K. O'Connor, Gwynne Mhuireach, Dale Northcutt, Jeff Kline, Maxwell Moriyama, G. Z. Brown, Brendan J. M. Bohannan, and Jessica L. Green. 2014. "Architectural design drives the biogeography of indoor bacterial communities." *PLOS ONE* 9 (01): e87093. doi:10.1371/journal.pone.0087093.

McMurdie, Paul J., and Susan Holmes. 2013. "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data." Ed. Michael Watson. *PloS one* 8 (4) (jan): e61217. doi:10.1371/journal.pone.0061217. http://dx.plos.org/10.1371/journal.pone.0061217.

Meadow, James F., Adam E. Altrichter, Steven W. Kembel, Maxwell Moriyama, Timothy K. O'Connor, Ann M. Womack, G. Z. Brown, Jessica L. Green, and Brendan J. M. Bohannan. 2014. "Bacterial communities on classroom surfaces vary with human contact." *Microbiome* 2: 7.

Oksanen, Jari, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, and Helene Wagner. 2011. *vegan: Community Ecology Package*. http://cran.r-project.org/package=vegan.

Roberts, David W. 2010. *labdsv: Ordination and Multivariate Analysis for Ecology*. http://cran.r-project.org/package=labdsv.