Script R6: Virome Beta Diversity

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol outlines the Bray-Curtis dissimilarity NMDS ordination and significance analysis of our manuscript. Here we will look at the clustering of the virome samples using our reference independent OTU table based on contig hits. We will compare skin environments (sebaceous, etc), occlusion status, and time. We also look at the difference between the background controls and the rest of the samples. Based on the methods from the following publication:

Hannigan, Geoffrey D., et al. "The Human Skin Double-Stranded DNA Virome: Topographical and Temporal Diversity, Genetic Enrichment, and Dynamic Associations with the Host Microbiome." *mBio* 6.5 (2015): e01578-15.

Citation: HANNIGAN GD, GRICE EA, ET AL. Script R6: Virome Beta Diversity. **protocols.io** dx.doi.org/10.17504/protocols.io.einbcde
Published: 10 Mar 2016

Guidelines

sessionInfo()

```
## R version 3.2.0 (2015-04-16)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.4 (Yosemite)
## ## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## loaded via a namespace (and not attached):
## [1] magrittr_1.5    formatR_1.2    tools_3.2.0    htmltools_0.2.6
## [5] yaml_2.1.13    stringi_0.4-1    rmarkdown_0.7    knitr_1.10.5
## [9] stringr_1.0.0    digest_0.6.8    evaluate_0.7
```

Before start

Supplemental information available at:

https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Diversity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

Protocol

Step 1.
Load the required R packages.

cmd COMMAND
```
library(vegan)
packageVersion("vegan")

library(ggplot2)
packageVersion("ggplot2")


library(scatterplot3d)
packageVersion("scatterplot3d")

library(reshape2)
packageVersion("reshape2")

library(plyr)
packageVersion("plyr")

library(pgirmess)
packageVersion("pgirmess")
```

EXPECTED RESULTS
```
## [1] '2.3.0'



## [1] '1.0.1'



## [1] '0.3.35'



## [1] '1.4.1'



## [1] '1.8.2'



## [1] '1.6.0'
```

Step 2.
Import OTU table.

cmd COMMAND
```
INPUT <- read.delim("../../IntermediateOutput/Bray-
curtis_virome_analysis/contig_otu_table_transposed_formatted.txt", header=TRUE, sep="\t")
```

Step 3.
Remove last column because it contains NAs (due to transposing Python script upstream).

cmd COMMAND
```
INPUT_NO_FINAL <- INPUT[ ,c(1:74361)]
```

Step 4.
Import mapping file.

cmd COMMAND
```
INPUT_MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv", h
eader=TRUE)
```

Step 5.
Here we need to reformat the mapping files. This means only looking at the two time points for which we have a complete data set (we have only partial data for time point 1), as well as excluding the sites and subjects for which we only have partial sampling (as mentioned in previous notebooks).

Step 6.
We will be generating NMDS plots to visualize the distances between virome samples based on site environment, occlusion status, and time point. We also calculate the significance using the adonis test which was stratified across subjects.

Step 7.
Generate subset of mapping file for only the specific anatomic sites and all time pints (2 and 3). Remove rows in mapping file with ID of NA (meaning the mapping row belongs only to the metagenome dataset).

cmd COMMAND
```
SUBSET_MAP <- INPUT_MAP[-which(INPUT_MAP$NexteraXT_Virome_SampleID %in% NA), ]
SUBSET_MAP <- SUBSET_MAP[which(SUBSET_MAP$TimePoint %in% c(2,3)), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$Site_Symbol %in% c("Ba","Ph","Vf","Neg")), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$SubjectID %in% c(2,3,9,11)), ]
SUBSET_MAP <- SUBSET_MAP[c(order(SUBSET_MAP$NexteraXT_Virome_SampleID)),]
```

Step 8.
Get only the samples described in the map subset.

cmd COMMAND
```
KEEP_SAMPLES <- as.vector(SUBSET_MAP$NexteraXT_Virome_SampleID)
INPUT_SUBSET <- INPUT_NO_FINAL[which(INPUT_NO_FINAL$ContigID %in% c(KEEP_SAMPLES)), ]
row.names(INPUT_SUBSET) <- INPUT_SUBSET[,1]
INPUT_SUB_FORMAT <- INPUT_SUBSET[,-1]
head(INPUT_SUB_FORMAT)[,c(1:5)]
```

EXPECTED RESULTS

	X1	X2	X4	X5
## MG100195	0.0000	0	1.343610	2.71473
## MG100198	0.0000	0	0.363769	0.00000
## MG100199	0.0000	0	0.955394	0.00000
## MG100200	25.9488	0	0.457572	2.31129
## MG100201	0.0000	0	0.155531	0.00000
## MG100202	50.3225	0	1.361750	5.15885

Step 9.
Generate distance matrix using Bray Curtis for all time point (2 and 3).

cmd COMMAND
```
INPUT_SUBSET_DIST_MATRIX <- vegdist(INPUT_SUB_FORMAT, method = "bray")
```

Step 10.
Visualize the distance matrix using NMDS.

cmd COMMAND
```
BRAY_ORD_NMDS <- metaMDS(INPUT_SUBSET_DIST_MATRIX,k=3)
```

EXPECTED RESULTS
```
## Run 0 stress 0.1579314
## Run 1 stress 0.1611112
## Run 2 stress 0.1626692
## Run 3 stress 0.1582604
## ... procrustes: rmse 0.03032907 max resid 0.2032916
## Run 4 stress 0.1601818
## Run 5 stress 0.1595255
## Run 6 stress 0.1567941
## ... New best solution
## ... procrustes: rmse 0.04472259 max resid 0.3011937
## Run 7 stress 0.157843
## Run 8 stress 0.1617624
## Run 9 stress 0.165309
## Run 10 stress 0.1575912
## Run 11 stress 0.1580084
## Run 12 stress 0.1601537
## Run 13 stress 0.1579342
## Run 14 stress 0.1634542
## Run 15 stress 0.1602448
## Run 16 stress 0.1593178
## Run 17 stress 0.157845
## Run 18 stress 0.1563824
## ... New best solution
## ... procrustes: rmse 0.02635918 max resid 0.1612987
## Run 19 stress 0.1583989
## Run 20 stress 0.1600246
```

Step 11.
Record the stress value.

cmd COMMAND
```
BRAY_ORD_FIT = data.frame(MDS1 = BRAY_ORD_NMDS$points[,1], MDS2 = BRAY_ORD_NMDS$points[,2],
 MDS3 = BRAY_ORD_NMDS$points[,3])

BRAY_ORD_NMDS$stress
```

EXPECTED RESULTS
```
## [1] 0.1563824
```

Step 12.
Generate and visualize merged NMDS and MAP data.

cmd COMMAND
```
BRAY_ORD_FIT$SampleID <- rownames(BRAY_ORD_FIT)
NMDS_AND_MAP <-
 merge(BRAY_ORD_FIT, SUBSET_MAP, by.x="SampleID", by.y="NexteraXT_Virome_SampleID")
```

```
head(NMDS_AND_MAP)[,c(1:5)]
```

EXPECTED RESULTS

##	SampleID	MDS1	MDS2	MDS3	NexteraXT_SampleID
## 1	MG100195	-0.039764304	0.10493926	-0.27817131	MG100171
## 2	MG100198	-0.164123841	0.04221565	0.02443569	MG100174
## 3	MG100199	-0.034656783	0.06907268	-0.09141366	MG100175
## 4	MG100200	-0.037125834	0.08835104	-0.01446685	MG100176
## 5	MG100201	0.0102459480	0.09353651	0.11426126	MG100177
## 6	MG100202	0.002223516	0.10249228	0.02265875	MG100178

Step 13.

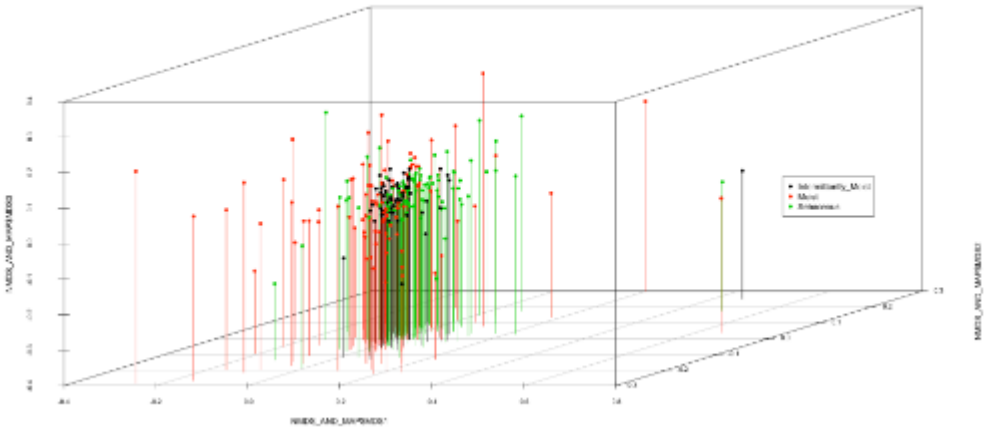
Start plotting the figures.

COMMAND

```
s3d <-
 scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.inte
ger(factor(NMDS_AND_MAP$Site_Categories)), type="h")
legend('right', pch = 16,legend = levels(factor(NMDS_AND_MAP$Site_Categories)),  col = seq_
along(levels(NMDS_AND_MAP$Site_Categories)), inset=c(0.1,0))

adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories), perm=999, strata =
factor(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS



```
##
## Call:
## adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories),
permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
## Blocks: strata
## Permutation: free ## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
```

	Df	SumsOfSqs	MeansSqs	F.Model
factor(NMDS_AND_MAP$Site_Categories)	2	2.562	1.28109	5.0118
Residuals	249	63.649	0.25562	
Total	251	66.211		

Residuals
Total

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

NOTES

Geoffrey Hannigan 11 Feb 2016

The legends will show up on top of the figures here, but running the first plot without the second legend line will generate the plot alone (this is what was used in publication)

Geoffrey Hannigan 11 Feb 2016

The legends will show up on top of the figures here, but running the first plot without the second legend line will generate the plot alone (this is what was used in publication)

Step 14.

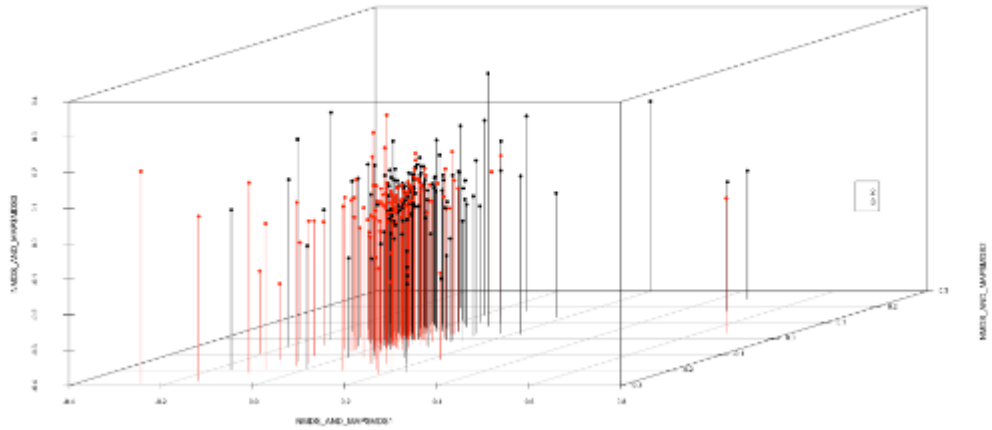
Plot time point.

cmd COMMAND

```
s3d <-
 scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.inte
ger(factor(NMDS_AND_MAP$TimePoint)), type="h")
legend('right', pch = 16,legend = levels(factor(NMDS_AND_MAP$TimePoint)),  col = seq_along(
levels(NMDS_AND_MAP$TimePoint)), inset=c(0.1,0))

adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$TimePoint), perm=999, strata = factor
(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS



##
Call:
adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories), permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
Blocks: strata
Permutation: free ## Number of permutations: 999
##
Terms added sequentially (first to last)
##

	Df	SumsOfSqs	MeansSqs	F.Model

factor(NMDS_AND_MAP$TimePoint)	1	2.279	2.27899	8.9118
Residuals	250	63.932	0.25573	
Total	251	66.211		

Residuals
Total

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step 15.

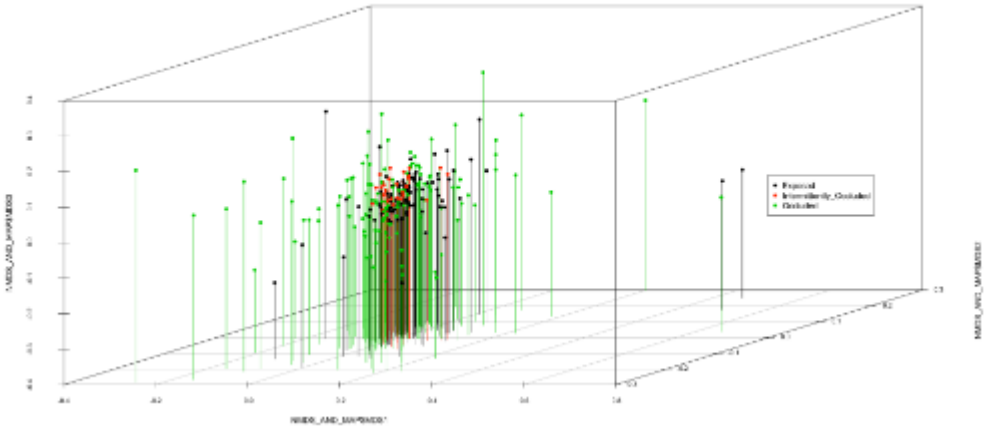
Plot ordination for occlusion site status.

cmd COMMAND

```
s3d <-
 scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.inte
ger(factor(NMDS_AND_MAP$Occlusion)), type="h")
legend('right', pch = 16,legend = levels(factor(NMDS_AND_MAP$Occlusion)),  col = seq_along(
levels(NMDS_AND_MAP$Occlusion)), inset=c(0.1,0))

adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Occlusion), perm=999, strata = factor
(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS



##
Call:
adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories), permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
Blocks: strata
Permutation: free ## Number of permutations: 999
##
Terms added sequentially (first to last)
##

	Df	SumsOfSqs	MeansSqs	F.Model
factor(NMDS_AND_MAP$TimePoint)	1	2.279	2.27899	8.9118
Residuals	250	63.932	0.25573	
Total	251	66.211		

```
## Residuals
## Total
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 16.

We will also perform the negative control ordination to validate we have samples over background, which are significantly different from the rest of the samples.

Step 17.

Environmental background contol in the analysis Generate subset of mapping file for only the specific anatomic sites and all time points (2 and 3).

cmd COMMAND

```
SUBSET_MAP <- INPUT_MAP[-which(INPUT_MAP$NexteraXT_Virome_SampleID %in%
    NA), ]
SUBSET_MAP <- SUBSET_MAP[which(SUBSET_MAP$TimePoint %in% c(2,
    3)), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$Site_Symbol %in% c("Ba",
    "Ph", "Vf")), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$SubjectID %in% c(2,
    3, 9, 11)), ]
SUBSET_MAP <- SUBSET_MAP[c(order(SUBSET_MAP$NexteraXT_Virome_SampleID)),
    ]
```

Step 18.

This will define each sample as a control or not:

cmd COMMAND

```
for (i in 1:length(SUBSET_MAP$Site_Categories)) {
    if (SUBSET_MAP$Site_Categories[i] == "Control") {
        SUBSET_MAP$CNTRL[i] = "Cntrl"
    } else {
        SUBSET_MAP$CNTRL[i] = "Smpl"
    }
}
SUBSET_MAP$CNTRL <- factor(SUBSET_MAP$CNTRL)
```

Step 19.

Get only the samples described in the map subset.

cmd COMMAND

```
KEEP_SAMPLES <- as.vector(SUBSET_MAP$NexteraXT_Virome_SampleID)
INPUT_SUBSET <- INPUT_NO_FINAL[which(INPUT_NO_FINAL$ContigID %in%
    c(KEEP_SAMPLES)), ]
row.names(INPUT_SUBSET) <- INPUT_SUBSET[, 1]
INPUT_SUB_FORMAT <- INPUT_SUBSET[, -1]
```

Step 20.

Generate distance matrix using Bray Curtis for all time points (2 and 3).

cmd COMMAND

```
INPUT_SUBSET_DIST_MATRIX <- vegdist(INPUT_SUB_FORMAT, method = "bray")
```

Step 21.

Visualize the distance matrix using NMDS.

cmd COMMAND

```
BRAY_ORD_NMDS <- metaMDS(INPUT_SUBSET_DIST_MATRIX, k = 2)
```

EXPECTED RESULTS

```
## Run 0 stress 0.2106771
## Run 1 stress 0.2152638
## Run 2 stress 0.2212825
## Run 3 stress 0.2131834
## Run 4 stress 0.2137311
## Run 5 stress 0.2188144
## Run 6 stress 0.2098836
## ... New best solution
## ... procrustes: rmse 0.02215487 max resid 0.2231707
## Run 7 stress 0.2292835
## Run 8 stress 0.218468
## Run 9 stress 0.2248454
## Run 10 stress 0.2147353
## Run 11 stress 0.2143595
## Run 12 stress 0.2191855
## Run 13 stress 0.2148725
## Run 14 stress 0.2248317
## Run 15 stress 0.2357207
## Run 16 stress 0.2133108
## Run 17 stress 0.217102
## Run 18 stress 0.2149535
## Run 19 stress 0.210627
## Run 20 stress 0.2231295
```

Step 22.

Record the stress value.

cmd COMMAND

```
BRAY_ORD_FIT = data.frame(MDS1 = BRAY_ORD_NMDS$points[, 1], MDS2 = BRAY_ORD_NMDS$points[,
    2])

BRAY_ORD_NMDS$stress
```

EXPECTED RESULTS

```
## [1] 0.2098836
```

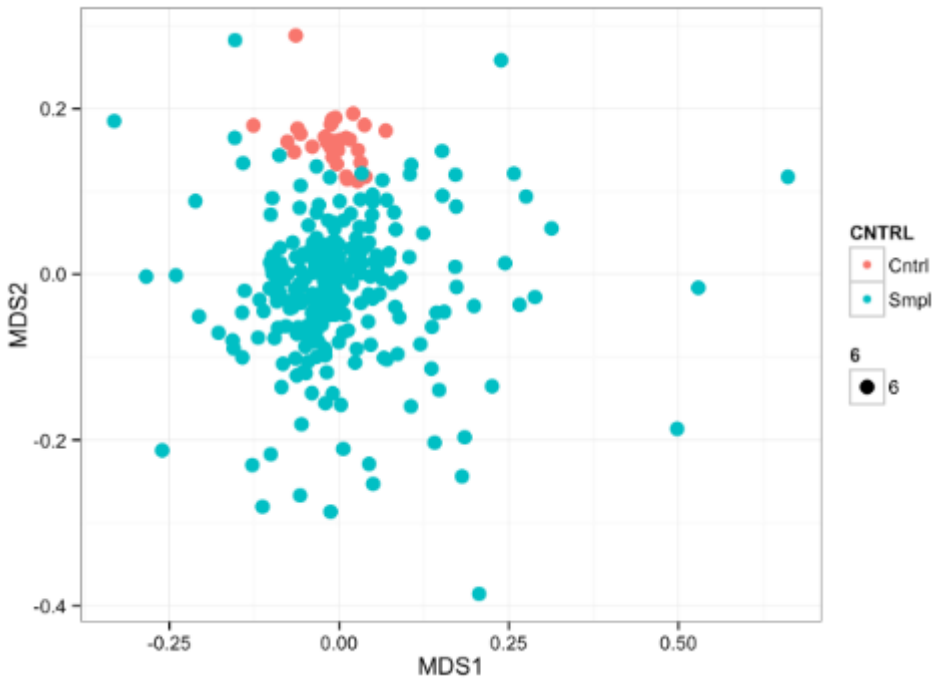
Step 23.

Plot MDS1 vs MDS2.

cmd COMMAND

```
RAY_ORD_FIT$SampleID <- rownames(BRAY_ORD_FIT)
NMDS_AND_MAP <- merge(BRAY_ORD_FIT, SUBSET_MAP, by.x = "SampleID",
    by.y = "NexteraXT_Virome_SampleID")
ggplot(NMDS_AND_MAP, aes(x = MDS1, y = MDS2, size = 6, group = CNTRL,
    colour = CNTRL)) + theme_bw() + geom_point()
```

EXPECTED RESULTS



Step 24.

Calculate the significance of the differences between the environmental background and the rest of the samples.

cmd COMMAND

```
adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$CNTRL),
    perm = 999, strata = factor(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS

```
##
## Call:
## adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories),
permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
## Blocks: strata
## Permutation: free ## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
```

	Df	SumsOfSqs	MeansSqs	F.Model
factor(NMDS_AND_MAP$CNTRL)	1	5.439	5.4391	21.422
Residuals	282	71.601	0.2539	
Total	283	77.040		

```
## Residuals
## Total
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 25.

In order to evaluate the utility of reference-independent methods, we also calculated alpha and beta

diversity metrics using our reference-dependent taxonomic data.

cmd COMMAND

```
INPUT_SPECIES <-
 read.delim("../../IntermediateOutput/Phage_Taxonomy/species_rel_abund.tsv",
    header = FALSE, sep = "\t")
```

Step 26.

Also import the mapping file.

cmd COMMAND

```
INPUT_MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv",
    header = TRUE)
```

Step 27.

Subset the mapping file.

cmd COMMAND

```
SUBSET_MAP <- INPUT_MAP[which(INPUT_MAP$TimePoint %in% c(2, 3)),
    ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$Site_Symbol %in% c("Ba",
    "Ph", "Vf", "Neg")), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$SubjectID %in% c(2,
    3, 9, 11)), ]
```

Step 28.

Same formatting and parsing as above.

cmd COMMAND

```
KEEP_SAMPLES <- as.vector(SUBSET_MAP$NexteraXT_Virome_SampleID)
INPUT_SUBSET <- INPUT_SPECIES[which(INPUT_SPECIES$V3 %in% c(KEEP_SAMPLES)),
    ]
INPUT_SUBSET <- INPUT_SUBSET[-which(INPUT_SUBSET$V1 %in% c("No_hit")),
    ]
```

NOTES

Geoffrey Hannigan 11 Feb 2016

From here we would calculate the mean across the various sites and merge in the mapping file data but we are going to want to use this file here for distance matrix calculations.

Step 29.

Convert the file to wide format so that it can be used for distance matrix calculations.

cmd COMMAND

```
inputWide <- dcast(INPUT_SUBSET, V1 ~ V3, value.var = "V2")
```

Step 30.

I pulled out a couple values to ensure the transformation was correct.

cmd COMMAND

```
inputWideTranspose <- as.data.frame(t(inputWide[, -1]))
INPUT_SUBSET_DIST_MATRIX <- vegdist(inputWideTranspose, method = "bray")
```

Step 31.

Visualize the distance matrix using NMDS.

cmd COMMAND

```
BRAY_ORD_NMDS <- metaMDS(INPUT_SUBSET_DIST_MATRIX, k = 3)
```

EXPECTED RESULTS

```
## Run 0 stress 0.08479484
## Run 1 stress 0.08752696
## Run 2 stress 0.0945716
## Run 3 stress 0.09044704
```

```
## Run 4 stress 0.09281868
## Run 5 stress 0.09634914
## Run 6 stress 0.09194107
## Run 7 stress 0.09115468
## Run 8 stress 0.08718591
## Run 9 stress 0.09208599
## Run 10 stress 0.08650345
## Run 11 stress 0.09094712
## Run 12 stress 0.09453678
## Run 13 stress 0.09349933
## Run 14 stress 0.08866572
## Run 15 stress 0.09333028
## Run 16 stress 0.09083254
## Run 17 stress 0.08825317
## Run 18 stress 0.0877584
## Run 19 stress 0.09264528
## Run 20 stress 0.09402247
```

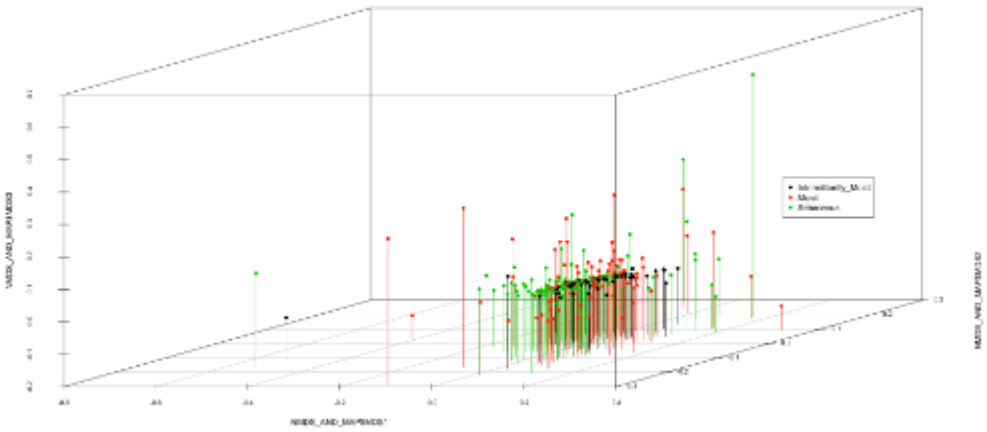
Step 32.

Plot the data.

cmd COMMAND

```
BRAY_ORD_FIT = data.frame(MDS1 = BRAY_ORD_NMDS$points[, 1], MDS2 = BRAY_ORD_NMDS$points[,
    2], MDS3 = BRAY_ORD_NMDS$points[, 3])
BRAY_ORD_NMDS_STRESS <- BRAY_ORD_NMDS$stress
NMDS_AND_MAP <- merge(BRAY_ORD_FIT, SUBSET_MAP, by.x = "row.names",
    by.y = "NexteraXT_Virome_SampleID")
s3d <- scatterplot3d(NMDS_AND_MAP$MDS1, NMDS_AND_MAP$MDS2, NMDS_AND_MAP$MDS3,
    pch = 16, color = as.integer(factor(NMDS_AND_MAP$Site_Categories)),
    type = "h")
legend("right", pch = 16, legend = levels(factor(NMDS_AND_MAP$Site_Categories)),
    col = seq_along(levels(NMDS_AND_MAP$Site_Categories)), inset = c(0.1,
        0))
```

EXPECTED RESULTS



Step 33.

Test the significance.

cmd COMMAND

```
adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories),
    perm = 999, strata = factor(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS

```
##
## Call:
## adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories),
permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
## Blocks: strata
## Permutation: free ## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
```

	Df	SumsOfSqs	MeansSqs	F.Model
factor(NMDS_AND_MAP$Site_Categories)	2	0.6099	0.304959	3.8099
Residuals	249	19.9311	0.080045	
Total	251	20.5410		

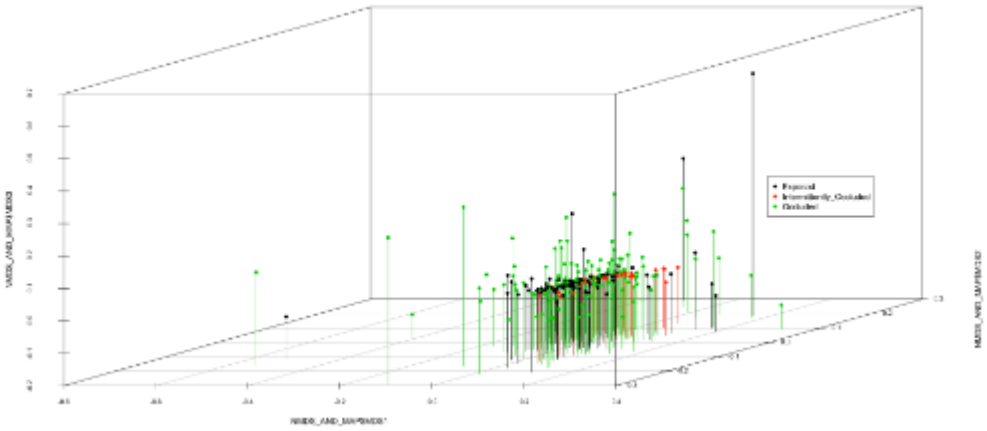
```
## Residuals
## Total
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 34.
Also plot occlusion.

cmd COMMAND

```
s3d <- scatterplot3d(NMDS_AND_MAP$MDS1, NMDS_AND_MAP$MDS2, NMDS_AND_MAP$MDS3,
    pch = 16, color = as.integer(factor(NMDS_AND_MAP$Occlusion)),
    type = "h")
legend("right", pch = 16, legend = levels(factor(NMDS_AND_MAP$Occlusion)),
    col = seq_along(levels(NMDS_AND_MAP$Occlusion)), inset = c(0.1,
        0))
```

EXPECTED RESULTS



Step 35.
Test the significance.

cmd COMMAND

```
adonis(INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Occlusion),
    perm = 999, strata = factor(NMDS_AND_MAP$SubjectID))
```

EXPECTED RESULTS

```
##
## Call:
## adonis(formula = INPUT_SUBSET_DIST_MATRIX ~ factor(NMDS_AND_MAP$Site_Categories),
permutations = 999, strata = factor(NMDS_AND_MAP$SubjectID))
##
## Blocks: strata
## Permutation: free ## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
```

	Df	SumsOfSqs	MeansSqs	F.Model
factor(NMDS_AND_MAP$Occlusion)	2	0.6008	0.300423	3.7515
Residuals	249	19.9402	0.080081	
Total	251	20.5410		

```
## Residuals
## Total
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

NOTES

Geoffrey Hannigan 11 Feb 2016

There is a significant difference between the sites, even when using the taxonomic profiles, suggests that even though the annotated portion of the data only represents a small portion of the community, it is actually able to detect the large trends for the overall community.

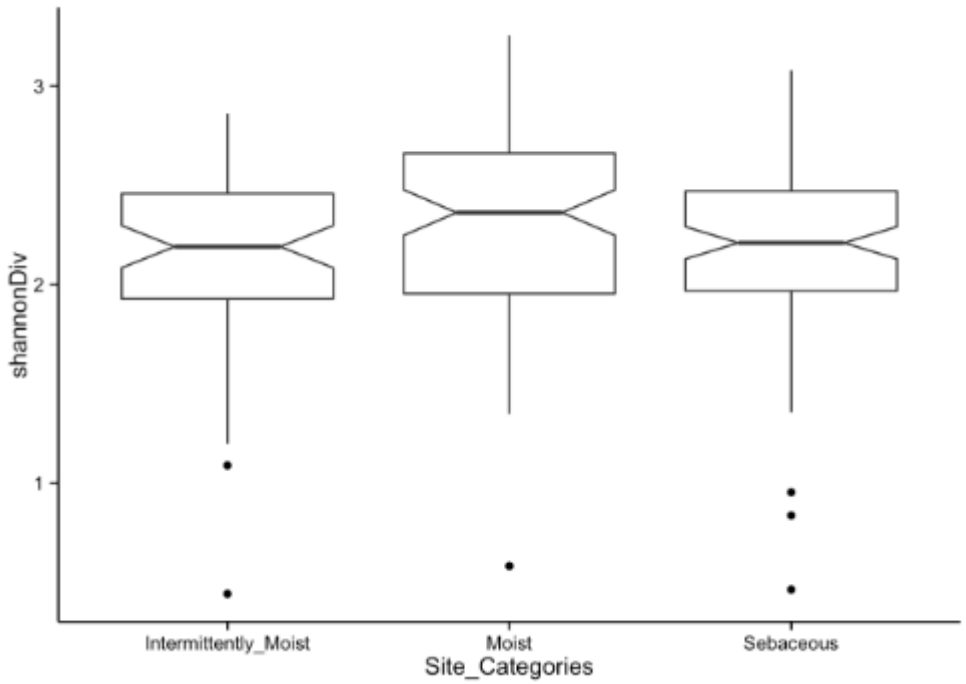
Step 36.

Also look at the alpha diversity.

COMMAND

```
shannonDiv <- as.data.frame(diversity(inputWideTranspose, index = "shannon"))
colnames(shannonDiv) <- c("shannonDiv")
alphaAndMap <- merge(shannonDiv, SUBSET_MAP, by.x = "row.names",
    by.y = "NexteraXT_Virome_SampleID")
MicroEnvPlot <- ggplot(alphaAndMap, aes(x = Site_Categories,
    y = shannonDiv)) + theme_classic() + geom_boxplot(notch = TRUE)
MicroEnvPlot
```

EXPECTED RESULTS



Step 37.
Check for significance of differences between the groups. Check levels of categories.

cmd COMMAND
```
alphaAndMap$Site_Categories <- factor(alphaAndMap$Site_Categories)
levels(alphaAndMap$Site_Categories)
```

EXPECTED RESULTS
```
## [1] "Intermittently_Moist" "Moist"                "Sebaceous"
```

Step 38.
Run Kruskal-Wallis on the dataset.

cmd COMMAND
```
kruskalmc(alphaAndMap$shannonDiv, alphaAndMap$Site_Categories)
```

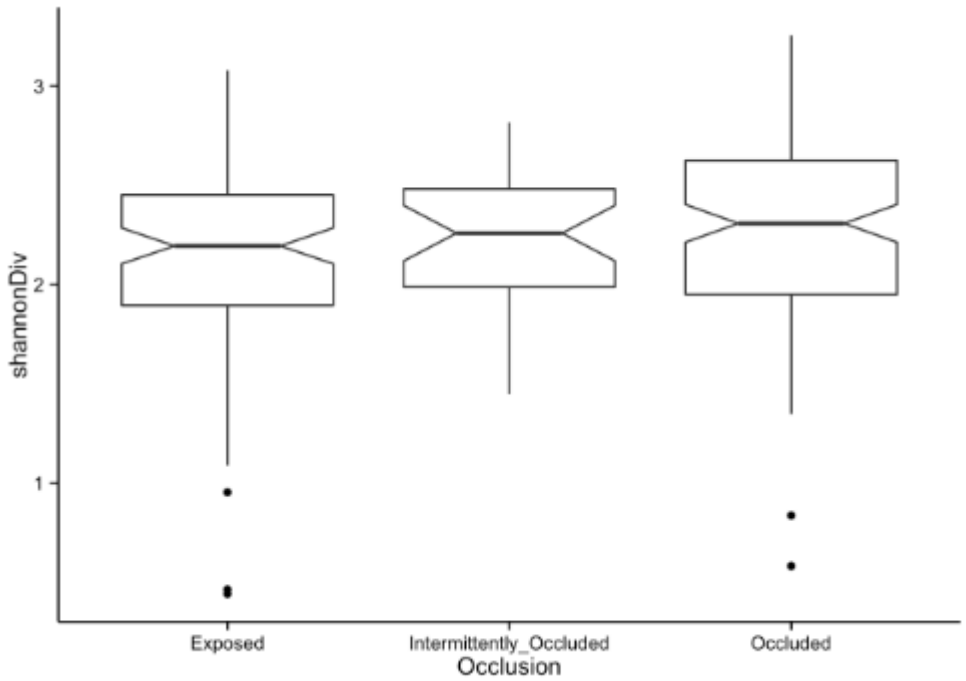
EXPECTED RESULTS
Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons

##	obs.dif	critical.dif	difference
## Intermittently_Moist-Moist	25.987989	28.54913	FALSE
Intermittently_Moist-Sebaceous	8.339718	28.43069	FALSE
Moist-Sebaceous	17.648271	25.32023	FALSE

Step 39.
Also look at occlusion status.

cmd COMMAND
```
OcclusionPlot <- ggplot(alphaAndMap, aes(x = Occlusion, y = shannonDiv)) +
    theme_classic() + geom_boxplot(notch = TRUE)
OcclusionPlot
```

EXPECTED RESULTS



Step 40.
Check for significance of differences between groups. Check levels of categories.

cmd COMMAND

```
alphaAndMap$Occlusion <- factor(alphaAndMap$Occlusion)
levels(alphaAndMap$Occlusion)
```

EXPECTED RESULTS

```
## [1] "Exposed"                  "Intermittently_Occluded"
## [3] "Occluded"
```

Step 41.
Run Kruskal-Wallis on the dataset.

EXPECTED RESULTS

Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons

##	obs.dif	critical.dif	difference
## Exposed-Intermittently_Occluded	13.804754	36.09383	FALSE
Exposed-Occluded	20.761487	23.71043	FALSE
Intermittently_Occluded-Occluded	6.959733	34.98437	FALSE