Script R4: Virome Taxonomy

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol outlines our bacteriophage/virus taxonomy analyses. This analysis includes profiles for the average order relative abundance for each site, the phage species relative abundance profiles for each patient, and the relative abundance of specific species between sites. Intermediate files are also included with the publication, and the paths are specified below. 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 R4: Virome Taxonomy. **protocols.io** dx.doi.org/10.17504/protocols.io.eiabcae
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(pgirmess)
packageVersion("pgirmess")

library(plyr)
packageVersion("plyr")

library(reshape2)
packageVersion("reshape2")
```

EXPECTED RESULTS

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.3-0
## [1] '2.3.0'
## [1] '1.0.1'
## [1] '1.6.0'
## [1] '1.8.2'
## [1] '1.4.1'
```

Step 2.

Import the data tables for order relative abundance.

cmd COMMAND

```
INPUT_ORDER <-
 read.delim("../../IntermediateOutput/Phage_Taxonomy/order_rel_abund.tsv", header=FALSE, se
p="\t")
INPUT_ORDER[c(1:5), ]
```

EXPECTED RESULTS

##	V1	V2	V3
## 1	No_hit	99626.40000	MG100098
## 2	Unclassified_Order	5596.65000	MG100098
## 3	Caudovirales	11968.50000	MG100098
## 4	Herpesvirales	7.61005	MG100098
## 5	Mononegavirales	49.08570	MG100098

Step 3.

Import the data tables for species relative abundance.

cmd COMMAND

```
INPUT_SPECIES <-
 read.delim("../../IntermediateOutput/Phage_Taxonomy/species_rel_abund.tsv", header=FALSE,
sep="\t")
INPUT_SPECIES[c(1:5), ]
```

EXPECTED RESULTS

```
##   V1                                V2       V3
## 1 Achromobacter_phage               0.00000  MG100098
## 3 Human_immunodeficiency_virus      1.08351  MG100098
## 4 Silicibacter_phage                0.00000  MG100098
## 5 Clavibacter_phage                 0.00000  MG100098
      Serratia_phage                   0.00000  MG100098
```

Step 4.

Input the mapping file.

cmd COMMAND

```
INPUT_MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv", h
eader=TRUE)
INPUT_MAP[c(1:4), c(1:5)]
```

EXPECTED RESULTS

```
##          NexteraXT_SampleID NexteraXT_RunName NexteraXT_Virome_SampleID
## 1                  MG100151       NexteraXT_007                  MG100102
## 2                  MG100150       NexteraXT_007                  MG100101
## 3                  MG100149       NexteraXT_007                      <NA>
## 4                  MG100146       NexteraXT_007                  MG100098
##   NexteraXT_Virome_RunName             SubjectID
## 1              NexteraXT_005                     1
## 2              NexteraXT_005                     1
## 3                      <NA>                      1
## 4              NexteraXT_005                     1
```

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 (c(“Ba”,“Ph”,“Vf”,“Neg”)).

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$SubjectID %in% c(2,3,9,11)), ]
SUBSET_MAP <- SUBSET_MAP[-which(SUBSET_MAP$NexteraXT_Virome_SampleID %in% NA), ]
```

Step 6.

Now we can easily plot the viral and phage order relative abundance information by skin site. Get a vector of the sample IDs that we want to pull out for analysis.

cmd COMMAND

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

Step 7.
For this analysis we are removing those contigs which did not have a hit. We are only look at those contigs with phage/virus hits.

cmd COMMAND
```
INPUT_SUBSET <- INPUT_SUBSET[-which(INPUT_SUBSET$V1 %in% c("No_hit")), ]
INPUT_MERGE <- merge(INPUT_SUBSET, SUBSET_MAP, by.x="V3", by.y="NexteraXT_Virome_SampleID")
```

Step 8.
These variables are called species but they are only named this. They really contain the order information.

cmd COMMAND
```
SPECIES_MEAN <- ddply(INPUT_MERGE, c("Site_Symbol","V1"), summarise, mean = mean(V2))
```

Step 9.
Also filter out those taxa that account for less than 0.5% of the mean relative abundance.

cmd COMMAND
```
MeanForExclusion <- ddply(SPECIES_MEAN, c("V1"), summarise, mean = mean(mean))
MeanForExclusion$Percent <- 100 * MeanForExclusion$mean / sum(MeanForExclusion$mean)
MeanForExclusionCut <- MeanForExclusion[c(MeanForExclusion$Percent > 0.5),]
```

Step 10.
Now only use the filtered taxa.

cmd COMMAND
```
SpeciesMeanFiltered <- SPECIES_MEAN[c(SPECIES_MEAN$V1 %in% MeanForExclusionCut$V1),]
```

Step 11.
Take a look at the data frame.

cmd COMMAND
```
head(SpeciesMeanFiltered)
```

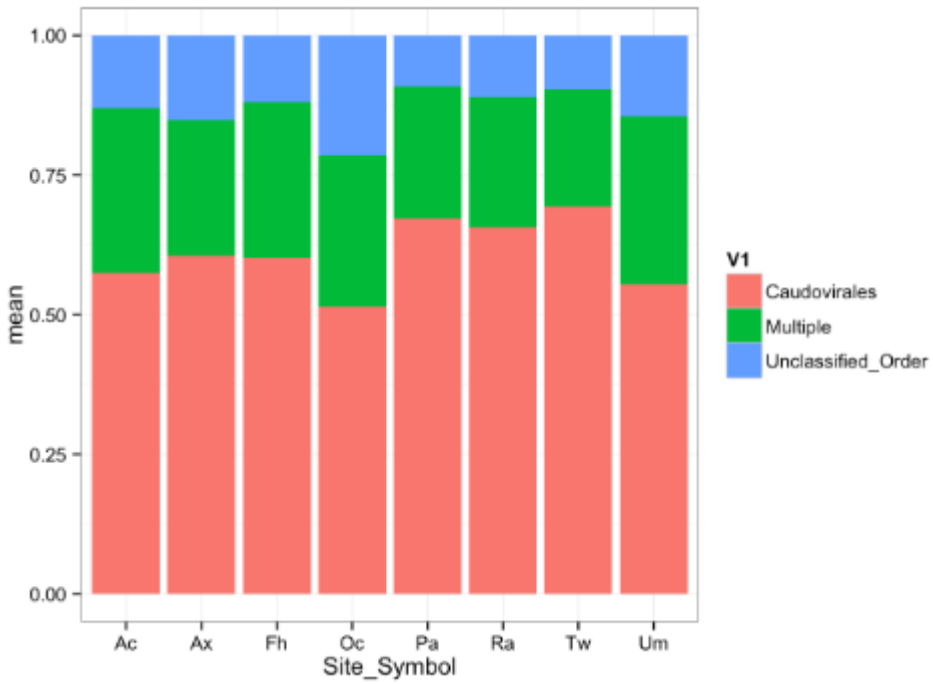
EXPECTED RESULTS

##	Site_Symbol	V1	mean
## 1	Ac	Caudovirales	6804.984
## 4	Ac	Multiple	3488.321
## 6	Ac	Unclassified_Order	1544.165
## 7	Ac	Caudovirales	8729.835
## 10	Ac	Multiple	3893.010
## 12	Ac	Unclassified_Order	2434.145

Step 12.
Plot the results.

cmd COMMAND
```
ggplot(SpeciesMeanFiltered, aes(x=Site_Symbol, y=mean, group=V1, fill=V1)) + theme_bw() + g
eom_bar(stat="identity", position="fill")
```

EXPECTED RESULTS



Step 13.

Now that we have looked at the information for taxonomic order, we also want to look at the species level. Here we are only going to look at the top 10 taxa, and include the remaining taxa relative abundance information as “Other”.

cmd COMMAND

```
#Same formatting and parsing as above
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")), ]
INPUT_MERGE <- merge(INPUT_SUBSET, SUBSET_MAP, by.x="V3", by.y="NexteraXT_Virome_SampleID")
#*
SPECIES_MEAN <- ddply(INPUT_MERGE, c("Site_Symbol","V1"), summarise, mean = mean(V2))
```

Step 14.

Get the top ten taxa so we can specifically look at them.

cmd COMMAND

```
TOP_TEN_MEAN <- ddply(SPECIES_MEAN, c("V1"), summarise, mean = mean(mean))
TOP_TEN_ORDER <- TOP_TEN_MEAN[c(order(TOP_TEN_MEAN$mean, decreasing=TRUE)),]
TOP_TEN <- TOP_TEN_ORDER[c(1:10),]
KEEP_TOP_TEN <- as.vector(TOP_TEN$V1)
#*
FINAL_TOP_TEN <- INPUT_MERGE[which(INPUT_MERGE$V1 %in% c(KEEP_TOP_TEN)), ]
FINAL_OTHER <- INPUT_MERGE[-which(INPUT_MERGE$V1 %in% c(KEEP_TOP_TEN)), ]
FINAL_OTHER$V3 <- factor(FINAL_OTHER$V3)
```

Step 15.

Get the rest of the relative abundance taxa into the "other" category.

cmd COMMAND

```
FINAL_OTHER_SUM <- data.frame(tapply(FINAL_OTHER$V2, INDEX=list(FINAL_OTHER$V3), FUN=sum))
FINAL_OTHER_SUM$SampleID <- c(row.names(FINAL_OTHER_SUM))
colnames(FINAL_OTHER_SUM) <- c("V2", "V3")
FINAL_OTHER_SUM$V2 <- as.numeric(as.character(FINAL_OTHER_SUM$V2))
FINAL_OTHER_SUM$V1 <- "Other"
FINAL_OTHER_SUM_FORMAT <- FINAL_OTHER_SUM[,c(2,3,1)]
FINAL_OTHER_MERGE <-
 merge(FINAL_OTHER_SUM_FORMAT, SUBSET_MAP, by.x="V3", by.y="NexteraXT_Virome_SampleID")
```

```
TOTAL_FINAL <- rbind(FINAL_TOP_TEN, FINAL_OTHER_MERGE)
head(TOTAL_FINAL)[,c(1:4)]
```

EXPECTED RESULTS

##	V3	V1	V2	NexteraXT_SampleID
## 21	MG100195	Multiple	6.39256e+03	MG100171
## 49	MG100195	Mycobacterium_phage	1.31141e+02	MG100171
## 57	MG100195	Propionibacterium_phage	1.69708e+00	MG100171
## 81	MG100195	Pseudomonas_phage	1.16539e+05	MG100171
## 85	MG100195	Streptococcus_phage	1.36360e+02	MG100171
## 87	MG100195	Human_papillomavirus	1.89501e+03	MG100171

Step 16.

Order according to relative abundance.

cmd COMMAND

```
TOP10_WITH_OTHER <- ddply(TOTAL_FINAL, c("V1"), summarise, mean=mean(V2))
TOP10_WITH_OTHER <- TOP10_WITH_OTHER[!c(TOP10_WITH_OTHER$V1=="Other"),]
TOP10_WITH_OTHER <- TOP10_WITH_OTHER[order(TOP10_WITH_OTHER$mean, decreasing=TRUE), ]
ORDER_MEAN_NAMES_WITH_OTHER <- as.vector(TOP10_WITH_OTHER$V1)
```

Step 17.

Append other to the vector.

cmd COMMAND

```
ORDER_MEAN_NAMES_WITH_OTHER <- c(ORDER_MEAN_NAMES_WITH_OTHER, "Other")
TOTAL_FINAL$V1 <- factor(TOTAL_FINAL$V1, levels=c(ORDER_MEAN_NAMES_WITH_OTHER))
```

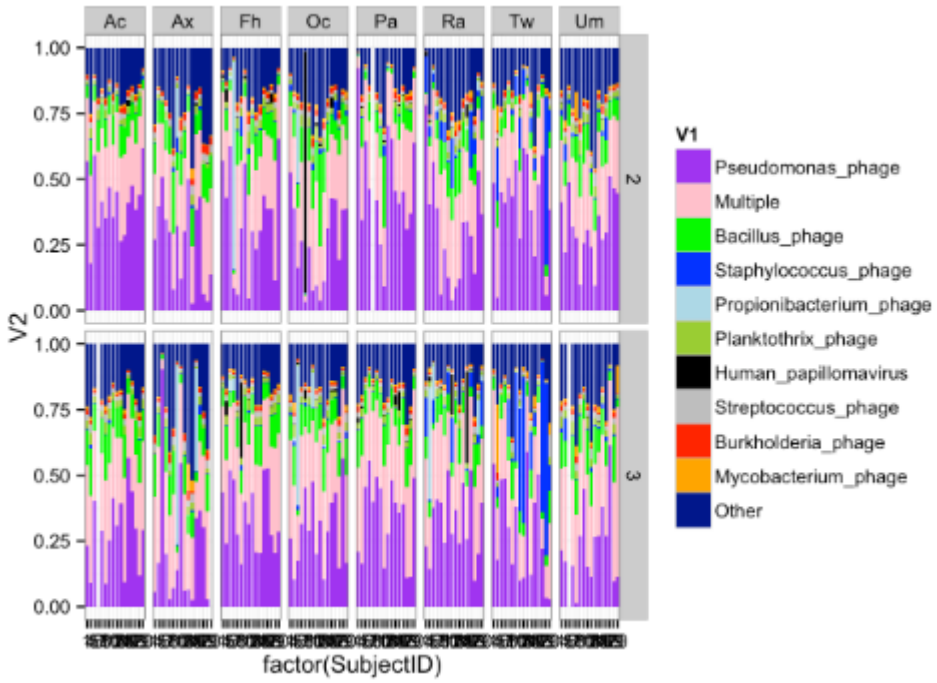
Step 18.

Get plotting of species by patient for graphing.

cmd COMMAND

```
ggplot(TOTAL_FINAL, aes(x=factor(SubjectID), y=V2, fill=V1, order=V1)) + theme_bw() + geom_
bar(stat="identity", position="fill") + facet_grid(TimePoint~Site_Symbol, scales="free") +
scale_fill_manual(values=c("purple","pink","green","blue","lightblue","yellowgreen", "black
","grey","red","orange", "darkblue")) + theme(legend.position="right")
```

EXPECTED RESULTS



NOTES

Geoffrey Hannigan 09 Feb 2016

The colors and other minor cosmetics can be fixed in illustrator.

Step 19.

We are also interested in some specific viral species relative abundances by site, after looking at the general subject profiles above. Here we look at the specific relative abundances of HPV, propionibacterium phages, and staphylococcus phages. First format the data. Get a list of the sample names.

cmd COMMAND

```
SAMPLE_NAMES <- as.vector(unique(INPUT_SUBSET$V3))
IN_SUBSET_REL_ABUND <- data.frame(lapply(SAMPLE_NAMES, function(i) {
  SUBSET <- INPUT_SUBSET[c(INPUT_SUBSET$V3==i),]
  SUM <- sum(SUBSET$V2)
  SUBSET$Rel_Abund <- 100 * SUBSET$V2 / SUM
  colnames(SUBSET) <- c("Taxa","Hits","Sample",i)
  return(SUBSET)
}))
IN_SUBSET_REL_ABUND_SUB <- IN_SUBSET_REL_ABUND[,c("Taxa",SAMPLE_NAMES)]
REL_ABUND_MELT <- melt(IN_SUBSET_REL_ABUND_SUB)
```

Step 20.

First we looked at the relative abundances of Staphylococcus phages. This is written as a set of copy/pasted script sections for taxonomic group.

cmd COMMAND

```
STAPH_REL_ABUND <- REL_ABUND_MELT[c(REL_ABUND_MELT$Taxa=="Staphylococcus_phage"), ]
STAPH_MERGE <-
 merge(STAPH_REL_ABUND, SUBSET_MAP, by.x="variable", by.y="NexteraXT_Virome_SampleID")
```

Step 21.

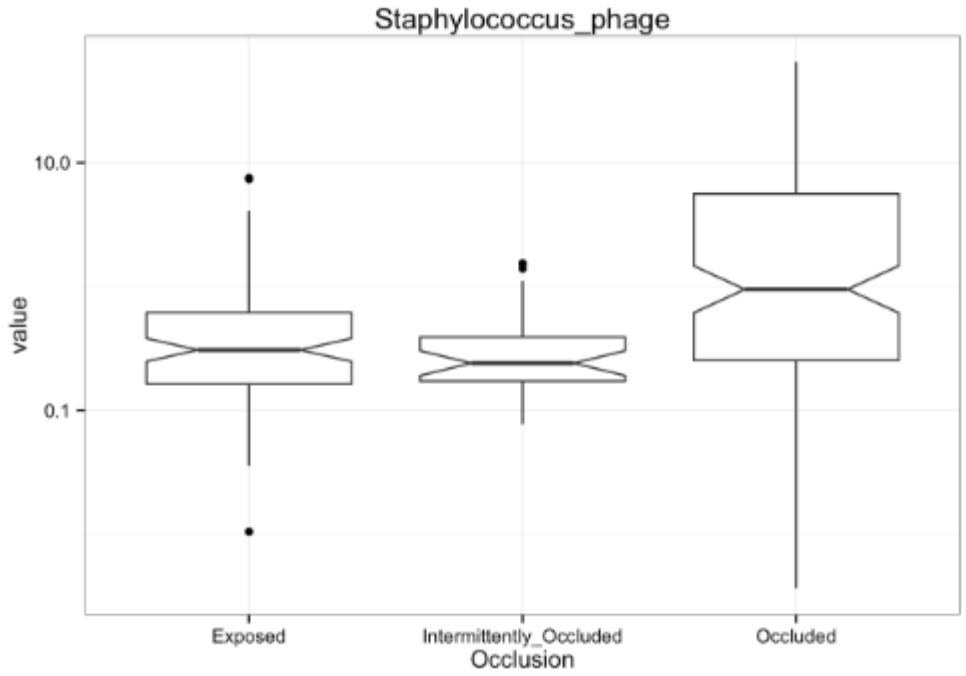
Plot the data by occlusion status.

cmd COMMAND

```
ggplot(STAPH_MERGE, aes(x=Occlusion, y=value)) + theme_bw() + geom_boxplot(notch=TRUE) + sc
ale_y_log10() + ggtitle("Staphylococcus_phage")
```

EXPECTED RESULTS

Warning in loop_apply(n, do.ply): Removed 1 rows containing non-finite values (stat_boxplot).



Step 22.
Perform stats using the kruskalmc subroutine.

cmd COMMAND
```
STAPH_MERGE$Occlusion <- factor(STAPH_MERGE$Occlusion)
kruskalmc(STAPH_MERGE$value, STAPH_MERGE$Occlusion)
```

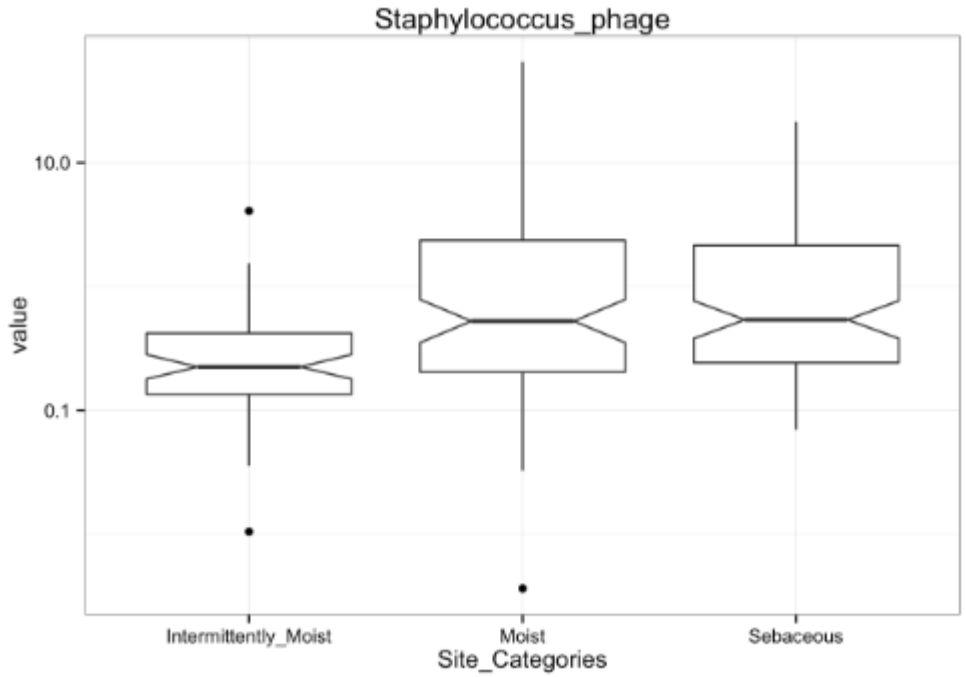
EXPECTED RESULTS
Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons

	obs.dif	critical.dif	difference
Exposed-Intermittently_Occluded	10.76401	36.09383	FALSE
Exposed-Occluded	44.70092	23.71043	TRUE
Intermittently_Occulded-Occluded	55.46493	34.98437	TRUE

Step 23.
Plot by site category.

cmd COMMAND
```
ggplot(STAPH_MERGE, aes(x=Site_Categories, y=value)) + theme_bw() + geom_boxplot(notch=TRUE
) + scale_y_log10() + ggtitle("Staphylococcus_phage")
```

EXPECTED RESULTS
Warning in loop_apply(n, do.ply): Removed 1 rows containing non-finite values (stat_boxplot).



Step 24.
Run the stats.

cmd COMMAND
```
STAPH_MERGE$Site_Categories <- factor(STAPH_MERGE$Site_Categories)
kruskalmc(STAPH_MERGE$value, STAPH_MERGE$Site_Categories)
```

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

	obs.dif	critical.dif	difference
Intermittently_Moist-Moist	51.958476	28.54913	TRUE
Intermittently_Moist-Sebaceous	53.065524	28.43069	TRUE
Moist-Sebaceous	1.107048	25.32023	FALSE

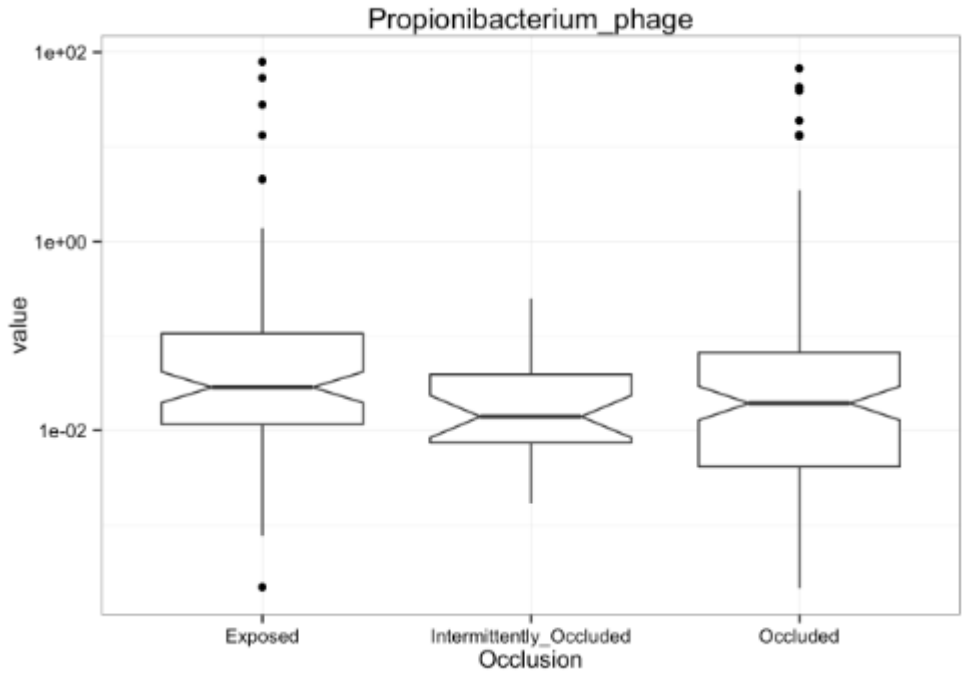
Step 25.
Get the Propionibacerium phage rows.

cmd COMMAND
```
PROP_REL_ABUND <- REL_ABUND_MELT[c(REL_ABUND_MELT$Taxa=="Propionibacterium_phage"), ]
PROP_MERGE <-
 merge(PROP_REL_ABUND, SUBSET_MAP, by.x="variable", by.y="NexteraXT_Virome_SampleID")
```

Step 26.
Plot by occlusion status.

cmd COMMAND
```
ggplot(PROP_MERGE, aes(x=Occlusion, y=value)) + theme_bw() + geom_boxplot(notch=TRUE) + sca
le_y_log10() + ggtitle("Propionibacterium_phage")
```

EXPECTED RESULTS
Warning in loop_apply(n, do.ply): Removed 27 rows containing non-finite values (stat_boxplot).



Step 27.
Run the stats.

cmd COMMAND
```
PROP_MERGE$Occlusion <- factor(PROP_MERGE$Occlusion)
kruskalmc(PROP_MERGE$value, PROP_MERGE$Occlusion)
```

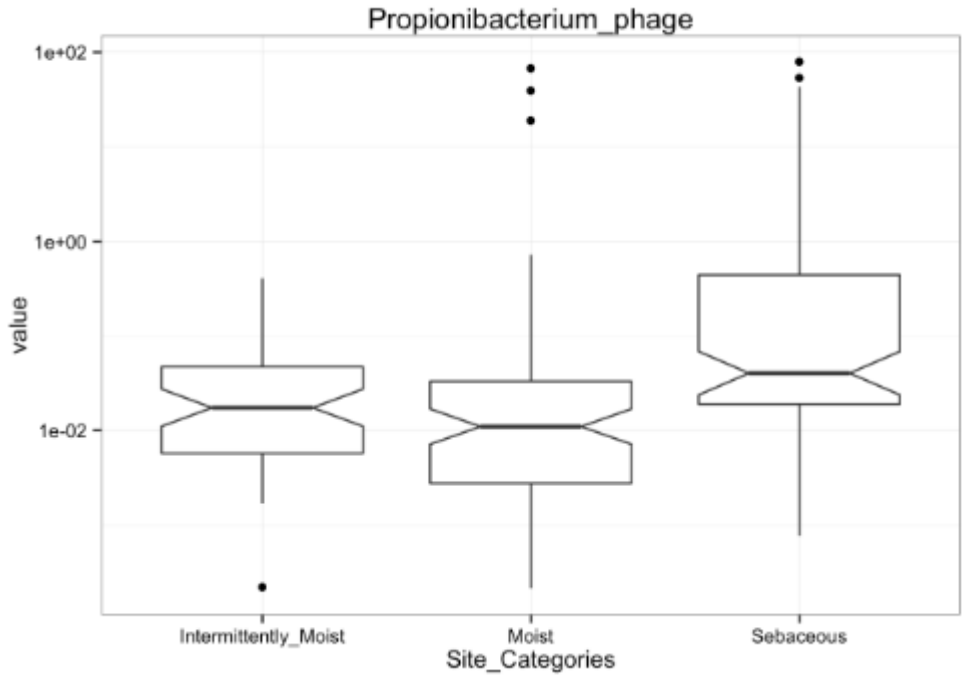
EXPECTED RESULTS
Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons

	obs.dif	critical.dif	difference
Exposed-Intermittently_Occluded	31.54397	36.09383	FALSE
Exposed-Occluded	20.64971	23.71043	FALSE
Intermittently_Occluded-Occluded	10.89427	34.98437	FALSE

Step 28.
Plot by site category.

cmd COMMAND
```
ggplot(PROP_MERGE, aes(x=Site_Categories, y=value)) + theme_bw() + geom_boxplot(notch=TRUE)
 + scale_y_log10() + ggtitle("Propionibacterium_phage")
```

EXPECTED RESULTS
Warning in loop_apply(n, do.ply): Removed 27 rows containing non-finite values (stat_boxplot).



Step 29.
Run the stats.

COMMAND
```
PROP_MERGE$Site_Categories <- factor(PROP_MERGE$Site_Categories)
kruskalmc(PROP_MERGE$value, PROP_MERGE$Site_Categories)
```

EXPECTED RESULTS

	obs.dif	critical.dif	difference
Intermittently_Moist-Moist	10.71380	28.54913	FALSE
Intermittently_Moist-Sebaceous	44.36156	28.43069	TRUE
Moist-Sebaceous	55.07535	25.32023	TRUE

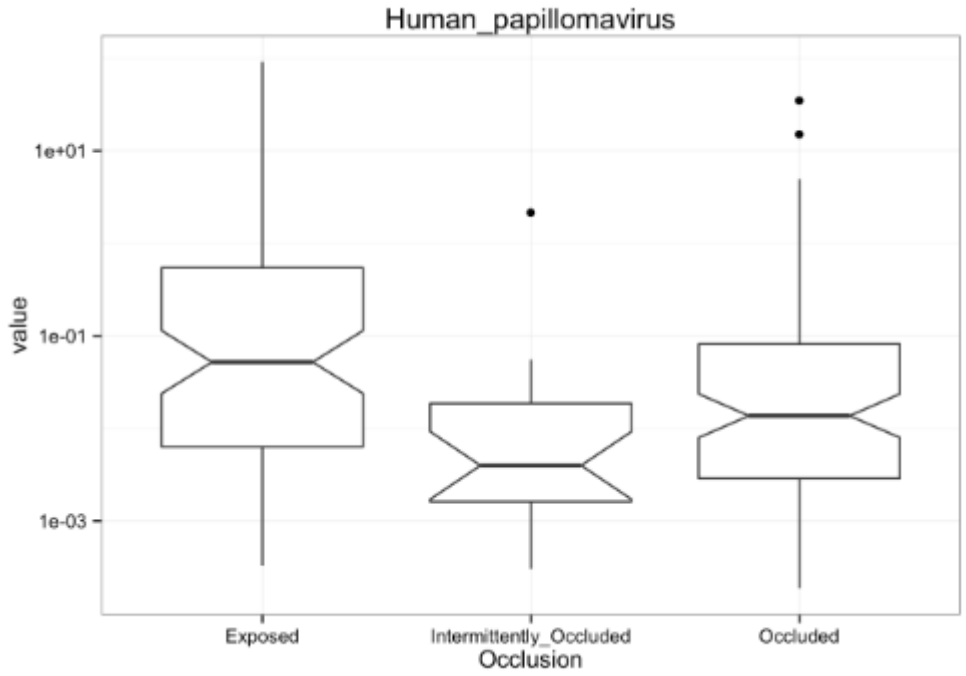
Step 30.
Finally do the same analysis for HPV.

COMMAND
```
HPV_REL_ABUND <- REL_ABUND_MELT[c(REL_ABUND_MELT$Taxa=="Human_papillomavirus"), ]
HPV_MERGE <-
 merge(HPV_REL_ABUND, SUBSET_MAP, by.x="variable", by.y="NexteraXT_Virome_SampleID")
```

Step 31.
Plot by occlusion status.

COMMAND
```
ggplot(HPV_MERGE, aes(x=Occlusion, y=value)) + theme_bw() + geom_boxplot(notch=TRUE) + scal
e_y_log10() + ggtitle("Human_papillomavirus")
```

EXPECTED RESULTS
Warning in loop_apply(n, do.ply): Removed 56 rows containing non-finite values (stat_boxplot).



Step 32.
Run the stats.

cmd COMMAND
```
HPV_MERGE$Occlusion <- factor(HPV_MERGE$Occlusion)
kruskalmc(HPV_MERGE$value, HPV_MERGE$Occlusion)
```

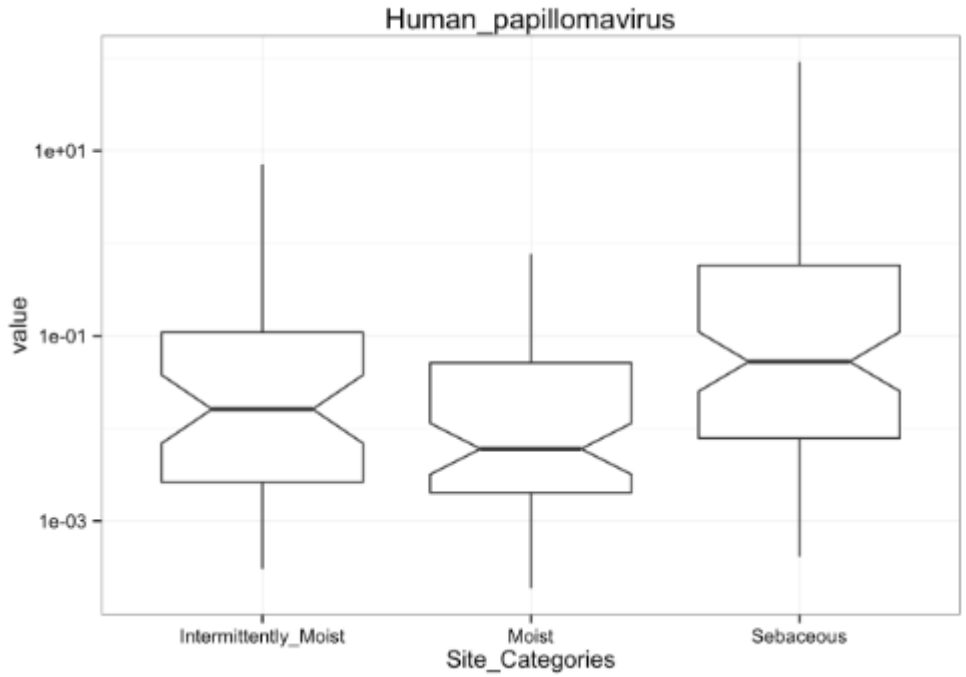
EXPECTED RESULTS

	obs.dif	critical.dif	difference
Exposed-Intermittently_Occluded	58.94431	36.09383	TRUE
Exposed-Occluded	33.67678	23.71043	TRUE
Intermittently_Occulded-Occluded	25.26754	34.98437	FALSE

Step 33.
Plot by site category.

cmd COMMAND
```
ggplot(HPV_MERGE, aes(x=Site_Categories, y=value)) + theme_bw() + geom_boxplot(notch=TRUE)
+ scale_y_log10() + ggtitle("Human_papillomavirus")
```

EXPECTED RESULTS
Warning in loop_apply(n, do.ply): Removed 56 rows containing non-finite values (stat_boxplot).



Step 34.
Run the stats.

cmd COMMAND

```
HPV_MERGE$Site_Categories <- factor(HPV_MERGE$Site_Categories)
kruskalmc(HPV_MERGE$value, HPV_MERGE$Site_Categories)
```

EXPECTED RESULTS

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

	obs.dif	critical.dif	difference
Intermittently_Moist-Moist	22.01579	28.54913	FALSE
Intermittently_Moist-Sebaceous	36.03696	28.43069	TRUE
Moist-Sebaceous	58.05275	25.32023	TRUE