Script R10: Whole Metagenome Beta Diversity

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol outlines the beta-diversity analysis that we performed on the whole metagenome samples. We measured the signifiance of the dissimilarity between samples that were grouped by biological occlusion status, microenvironment (sebaceous, moist, etc), and sampling time point. Here we also describe our intrapersonal vs interpersonal dissimilarity measurements. Based on 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 R10: Whole Metagenome Beta Diversity. **protocols.io** dx.doi.org/10.17504/protocols.io.ejebcje
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.
First load the required R packages.

cmd COMMAND
```
library(scatterplot3d)
packageVersion("scatterplot3d")

library(vegan)
packageVersion("vegan")

library(plyr)
packageVersion("plyr")

library(reshape2)
packageVersion("reshape2")

library(ggplot2)
packageVersion("ggplot2")
```

EXPECTED RESULTS
```
## [1] '0.3.35'



## [1] '2.3.0'



## [1] '1.8.2'



## [1] '1.4.1'



## [1] '1.0.1'
```

Step 2.
Because there were many samples, and especially many OTUs, it took substantial computing power to get the distance matrix from the data set. We saved an R image of the environment that contained the distance matrix and stats. Because the R session image provides quick access to the data, we load that here for the notebook. The actual matrix can be found in the intermediate files.

cmd COMMAND
```
load('../../IntermediateOutput/Whole_Microbiome_Beta_Div/distmatrix.RData')

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

Step 3.

We will want to format the data for visualization, and then use NMDS ordination to visualize the clusters formed by our categories of interest. 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$NexteraXT_SampleID %in% NA), ]
SUBSET_MAP <- SUBSET_MAP[which(SUBSET_MAP$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_SampleID)),]
```

Step 4.

Get only the samples described in the map subset.

cmd COMMAND

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

Step 5.

Visualize the distance matrix using NMDS.

cmd COMMAND

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

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

EXPECTED RESULTS

```
## Run 0 stress 0.1151716
## Run 1 stress 0.1198662
## Run 2 stress 0.1202445
## Run 3 stress 0.1196748
## Run 4 stress 0.1201702
## Run 5 stress 0.1170017
## Run 6 stress 0.117732
## Run 7 stress 0.1226832
## Run 8 stress 0.1211114
## Run 9 stress 0.1173711
## Run 10 stress 0.1162585
## Run 11 stress 0.1188054
## Run 12 stress 0.1163528
## Run 13 stress 0.1180277
## Run 14 stress 0.1191655
## Run 15 stress 0.1199955
## Run 16 stress 0.1193478
## Run 17 stress 0.1241531
## Run 18 stress 0.1182478
## Run 19 stress 0.115657
## ... procrustes: rmse 0.01284407 max resid 0.1230186
## Run 20 stress 0.1176689
```

Step 6.

Now we will visualize the data as 3D scatter plots.

cmd COMMAND

```
NMDS_AND_MAP$Site_Categories<-factor(NMDS_AND_MAP$Site_Categories)
NMDS_AND_MAP$TimePoint<-factor(NMDS_AND_MAP$TimePoint)
NMDS_AND_MAP$Occlusion<-factor(NMDS_AND_MAP$Occlusion)
SUBSET_MAP$SubjectID<-factor(SUBSET_MAP$SubjectID)
SUBSET_MAP$Site_Categories<-factor(SUBSET_MAP$Site_Categories)
SUBSET_MAP$Site_Symbol<-factor(SUBSET_MAP$Site_Symbol)
SUBSET_MAP$TimePoint<-factor(SUBSET_MAP$TimePoint)
SUBSET_MAP$Occlusion<-factor(SUBSET_MAP$Occlusion)
```

Step 7.

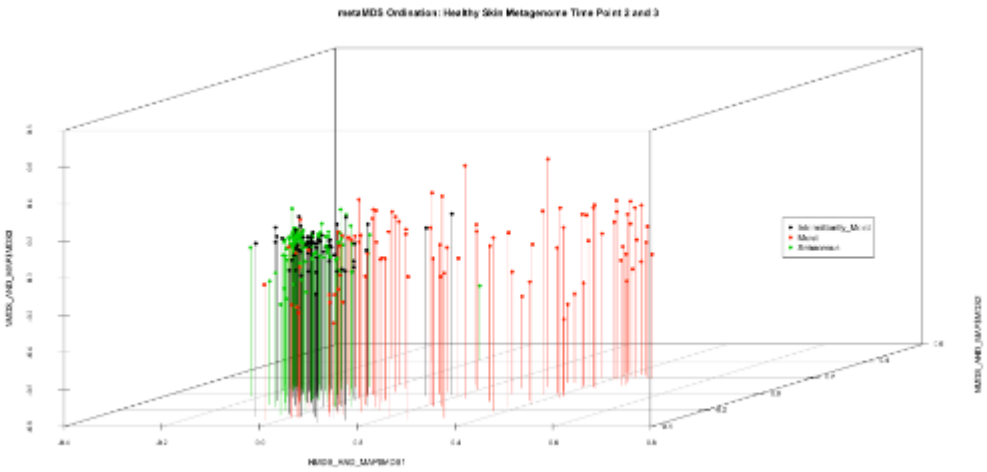
Plot site microenvironment.

cmd COMMAND

```
s3d<-
scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.integ
er(NMDS_AND_MAP$Site_Categories), type="h", main="metaMDS Ordination: Healthy Skin Metageno
me Time Point 2 and 3")
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 ~ SUBSET_MAP$Site_Categories, perm = 999, strata = SUBSET_M
AP$SubjectID)
```

EXPECTED RESULTS



Step 8.

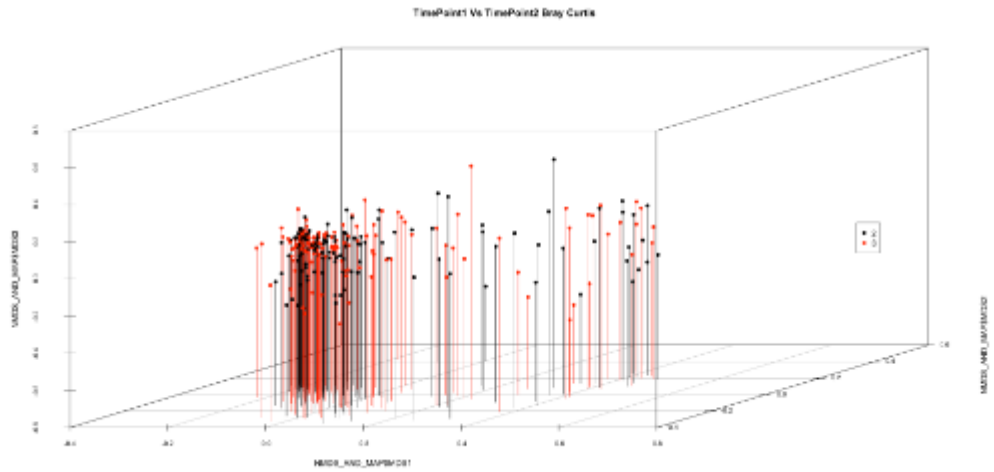
Plot time point.

cmd COMMAND

```
s3d<-
scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.integ
er(NMDS_AND_MAP$TimePoint), type="h", main="TimePoint1 Vs TimePoint2 Bray Curtis")
legend('right', pch = 16, legend = levels(NMDS_AND_MAP$TimePoint), col = seq_along(levels(N
MDS_AND_MAP$TimePoint)),inset=c(0.1,0))

adonis(INPUT_SUBSET_DIST_MATRIX ~ SUBSET_MAP$TimePoint, perm = 999, strata = SUBSET_MAP$Sub
jectID)
```

EXPECTED RESULTS



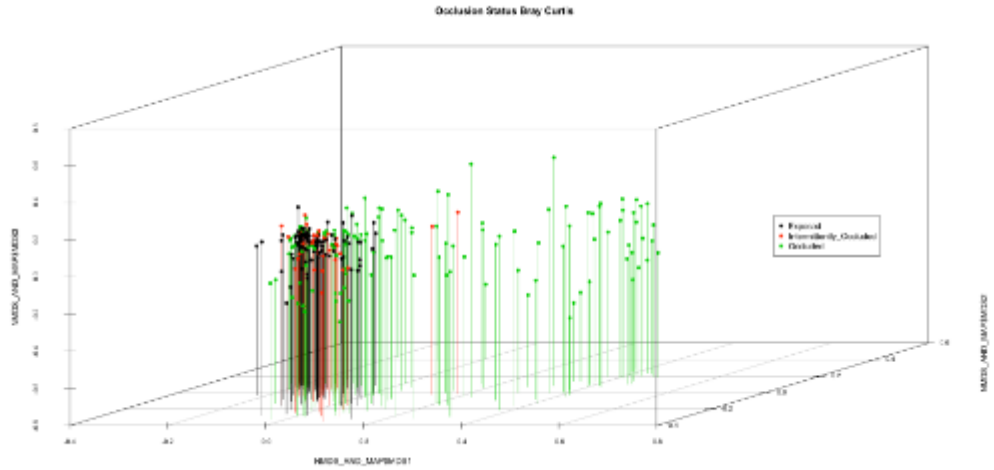
Step 9.
Plot occlusion.

cmd COMMAND

```
s3d<-
scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.integ
er(NMDS_AND_MAP$Occlusion), type="h", main="Occlusion Status Bray Curtis")
legend('right', pch = 16, legend = levels(NMDS_AND_MAP$Occlusion), col = seq_along(levels(N
MDS_AND_MAP$Occlusion)),inset=c(0.1,0))

adonis(INPUT_SUBSET_DIST_MATRIX ~ SUBSET_MAP$Occlusion, perm = 999, strata = SUBSET_MAP$Sub
jectID)
```

EXPECTED RESULTS

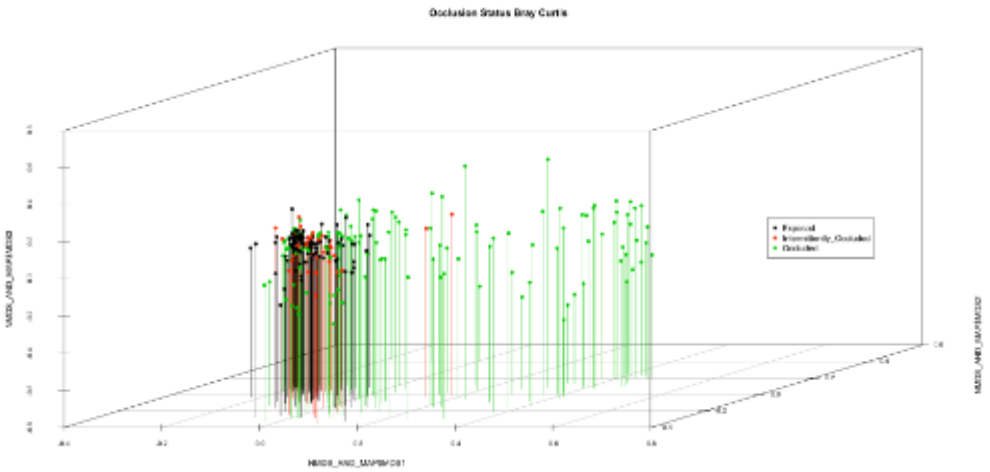


Step 10.
Plot occlusion with legend.

cmd COMMAND

```
s3d<-
scatterplot3d(NMDS_AND_MAP$MDS1,NMDS_AND_MAP$MDS2,NMDS_AND_MAP$MDS3, pch=16, color=as.integ
er(NMDS_AND_MAP$Occlusion), type="h", main="Occlusion Status Bray Curtis")
legend('right', pch = 16, legend = levels(NMDS_AND_MAP$Occlusion), col = seq_along(levels(N
MDS_AND_MAP$Occlusion)), inset=c(0.1,0))
```

EXPECTED RESULTS



Step 11.
Get intra-personal and interpersonal distance similarities, showing intra over time is more similar than that site compared to all other sites. INPUT_SUBSET_DIST_MATRIX was generated using the vegan command, vegdist (INPUT_SUB_FORMAT, method = "bray").

cmd COMMAND
```
INPUT_SUBSET_DIST_MATRIX_MATRIX <- data.frame(as.matrix(INPUT_SUBSET_DIST_MATRIX))
```
It is loaded from the RData image to save computational time

Step 12.
Data frame reference: "sample tp2" \t "sample tp3" using merge function.

cmd COMMAND
```
MAP_TP2 <- SUBSET_MAP[c(SUBSET_MAP$TimePoint==2),c(1,5:9)]
MAP_TP3 <- SUBSET_MAP[c(SUBSET_MAP$TimePoint==3),c(1,5:9)]
MAP_MERGE_REF <-
 merge(MAP_TP2, MAP_TP3, by=c("SubjectID","Site_Symbol","Site_Categories","Occlusion"))

SAMPLE_NAMES <- as.vector(MAP_MERGE_REF$NexteraXT_SampleID.x)

INTRAPERSONAL_DIST <- data.frame(lapply(SAMPLE_NAMES, function(i) {
  INTRAPERSON_DIST <-
 INPUT_SUBSET_DIST_MATRIX_MATRIX[c(row.names(INPUT_SUBSET_DIST_MATRIX_MATRIX)==i), as.vecto
r(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"NexteraXT_SampleID.y"])]
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"SubjectID"]
  SITE <- as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"Site_Symbol"])
  RESULT <- data.frame(X=c(i, SUBJECT, SITE, INTRAPERSON_DIST))
  return(RESULT)
}))

INTERPERSONAL_DIST_TP3 <- data.frame(lapply(SAMPLE_NAMES, function(i) {
  INTERPERSON_DIST_TP3 <-
 INPUT_SUBSET_DIST_MATRIX_MATRIX[c(row.names(INPUT_SUBSET_DIST_MATRIX_MATRIX)==i), as.vecto
r(MAP_MERGE_REF[-which(MAP_MERGE_REF$NexteraXT_SampleID.x %in% i),"NexteraXT_SampleID.y"])]
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"SubjectID"]
  SITE <- as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"Site_Symbol"])
  TRANS <- data.frame(t(INTERPERSON_DIST_TP3))
  RESULT <- data.frame(X=c(SUBJECT, SITE, INTERPERSON_DIST_TP3))
  return(TRANS)
}))

INTERPERSONAL_DIST_TP2 <- data.frame(lapply(SAMPLE_NAMES, function(i) {
  INTERPERSON_DIST_T2 <-
 INPUT_SUBSET_DIST_MATRIX_MATRIX[c(row.names(INPUT_SUBSET_DIST_MATRIX_MATRIX)==i), as.vecto
r(MAP_MERGE_REF[-which(MAP_MERGE_REF$NexteraXT_SampleID.x %in% i),"NexteraXT_SampleID.x"])]
```

```
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"SubjectID"]
  SITE <- as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_SampleID.x==i,"Site_Symbol"])
  TRANS <- data.frame(t(INTERPERSON_DIST_T2))
  RESULT <- data.frame(X=c(SUBJECT, SITE, INTERPERSON_DIST_T2))
  return(TRANS)
}))
```

Step 13.

Melt the two interpersonal distance data frames.

cmd COMMAND

```
INTER_TP2_MELT <- melt(INTERPERSONAL_DIST_TP2)

INTER_TP2_MELT$Type <- "Inter"
INTER_TP3_MELT <- melt(INTERPERSONAL_DIST_TP3)
```

EXPECTED RESULTS

```
## No id variables; using all as measure variables
```

Step 14.

Get intrapersonal values in same format.

cmd COMMAND

```
INTER_TP3_MELT$Type <- "Inter"
INTRA_TRANS <- data.frame(t(INTRAPERSONAL_DIST))
INTRA_TRANS_CUT <- INTRA_TRANS[,c("X1","X4")]
INTRA_TRANS_CUT$Type <- "Intra"
colnames(INTRA_TRANS_CUT) <- c("variable","value","Type")
INTRA_TRANS_CUT$value <- as.numeric(as.character(INTRA_TRANS_CUT$value))
row.names(INTRA_TRANS_CUT) <- NULL
```

Step 15.

Bind together all of these data frames.

cmd COMMAND

```
BOUND_DIST <- rbind(INTRA_TRANS_CUT, INTER_TP2_MELT, INTER_TP3_MELT)
BOUND_DIST <- BOUND_DIST[,c(2,3)]
```

Step 16.

Plot the resulting distances as means with stdev.

cmd COMMAND

```
BOUND_SUMMARY <-
 ddply(BOUND_DIST, c("Type"), summarise, N=length(value), mean=mean(value), sd=sd(value), s
e=sd/sqrt(N))
```

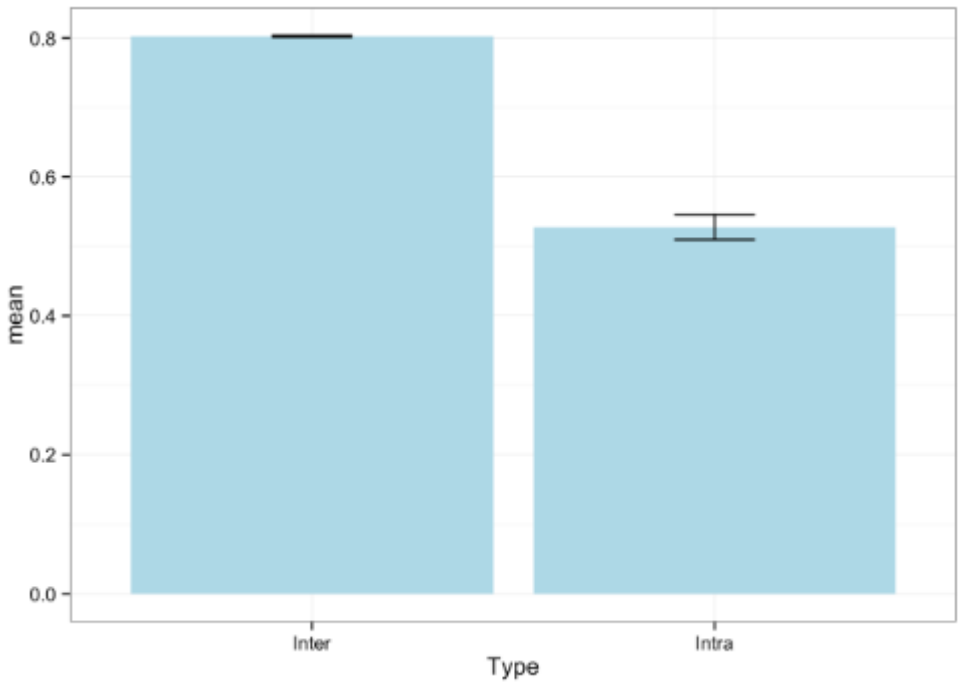
Step 17.

Now we can visualize our results.

cmd COMMAND

```
ggplot(BOUND_SUMMARY, aes(x=Type, y=mean)) + theme_bw() + geom_bar(position=position_dodge(
), stat="identity", fill="lightblue") + geom_errorbar(aes(ymin=mean-
se, ymax=mean+se), width=.2, position=position_dodge(.9))
```

EXPECTED RESULTS



Step 18.

Run a T-test to calculate the statistical significance of the differences in means.

cmd COMMAND

```
t.test(BOUND_DIST$value ~ BOUND_DIST$Type)
t.test(BOUND_DIST$value ~ BOUND_DIST$Type)$p.value
```