Script R7: Intra and Inter Personal Dissimilarity

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol outlines the analysis used for intra and interpersonal diversity of the virome and whole metagenome using the Bray Curtis dissimilarity metric. We visualize the differences as bar plots and calculate the statistical significance of the differences using a t-test. 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 R7: Intra and Inter Personal Dissimilarity. **protocols.io** dx.doi.org/10.17504/protocols.io.eipbcdn
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(reshape2)
packageVersion("reshape2")

library(plyr)
packageVersion("plyr")
```

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



## [1] '1.0.1'



## [1] '1.4.1'



## [1] '1.8.2'
```

Step 2.
Load in the OTU table (here they are for the virome samples)

cmd COMMAND
```
INPUT <-
 read.delim("../../IntermediateOutput/Interpersonal_intrapersonal_dissimilarity/contig_otu_
table_transposed_formatted.txt", header=TRUE, sep="\t")
head(INPUT)[,c(1:6)]
```

EXPECTED RESULTS

##	ContigID	X1	X2	X3	X4	X5
## 1	MG100098	11.91350	0.00000	0	0	28.7690
## 2	MG100099	28.72340	0.00000	0	0	0.0000
## 3	MG100100	73.04680	5.90828	0	0	34.0190
## 4	MG100101	4.18674	0.00000	0	0	0.0000
## 5	MG100102	22.93320	4.99401	0	0	0.0000
## 6	MG100103	13.01880	9.21378	0	0	17.6838

Step 3.
Remove last column because it contains NAs (artifact of the python script used to transpose this file.)

cmd COMMAND

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

Step 4.

Input mapping file.

cmd COMMAND

```
INPUT_MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv", h
eader=TRUE)
head(INPUT_MAP)[,c(1:6)]
```

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
## 5                  MG100157       NexteraXT_007                  MG100107
## 6                  MG100153       NexteraXT_007                  MG100104
##   NexteraXT_Virome_RunName SubjectID TimePoint
## 1             NexteraXT_005         1         1
## 2             NexteraXT_005         1         1
## 3                      <NA>         1         1
## 4             NexteraXT_005         1         1
## 5             NexteraXT_005         1         1
## 6             NexteraXT_005         1         1
```

Step 5.

Now we need to format the data and generate a dissimilarity matrix using the Bray-Curtis metric. We will use this matrix to extract the dissimilarity values between our samples below.

Step 6.

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

Step 7.

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(1:6)]
```

EXPECTED RESULTS

```
##              X1       X2 X3       X4       X5       X6
## 1 MG100195 0.0000 0 0.508673 1.343610 2.71473 0
## 2 MG100198 0.0000 0 0.000000 0.363769 0.00000 0
```

```
## 3 MG100199  0.0000   0   0.059621  0.955394  0.00000  0
## 4 MG100200 25.9488   0   0.618683  0.457572  2.31129  0
## 5 MG100201  0.0000   0   0.126176  0.155531  0.00000  0
## 6 MG100202 50.3225   0   0.552366  1.361750  5.15885  0
```

Step 8.

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 9.

We need to obtain a data frame with all of the distance information between specific combinations of sample distances. Intrapersonal dissimilarities will be between the same location and subject, but at time point 2 vs 3. Interpersonal distance will be between the sample and any other given sample of that time point.

Step 10.

Get intrapersonal and interpersonal distance similarities, showing intra over time is more similar than that site compared to all other sites.

cmd COMMAND

```
INPUT_SUBSET_DIST_MATRIX_MATRIX <- data.frame(as.matrix(INPUT_SUBSET_DIST_MATRIX))
```

Step 11.

Data frame reference: "sample tp2" \t "sample tp3". Using merge function.

cmd COMMAND

```
MAP_TP2 <- SUBSET_MAP[c(SUBSET_MAP$TimePoint==2), ]
MAP_TP3 <- SUBSET_MAP[c(SUBSET_MAP$TimePoint==3), ]
MAP_MERGE_REF <- merge(MAP_TP2, MAP_TP3, by=c("SubjectID","Site_Symbol"))

SAMPLE_NAMES <- as.vector(MAP_MERGE_REF$NexteraXT_Virome_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$NexteraXT_Virome_SampleID.x==i,"NexteraXT_Virome_SampleID.y"]
)]
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"SubjectID"]
  SITE <-
 as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"Site_Symbol"])
  RESULT <- data.frame(X=c(i, SUBJECT, SITE, INTRAPERSON_DIST))
  return(RESULT)
}))
head(INTRAPERSONAL_DIST)[,c(1:4)]

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$NexteraXT_Virome_SampleID.x %in% i),"NexteraXT_Virome_SampleID.y"])]
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"SubjectID"]
  SITE <-
 as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"Site_Symbol"])
  TRANS <- data.frame(t(INTERPERSON_DIST_TP3))
  RESULT <- data.frame(X=c(SUBJECT, SITE, INTERPERSON_DIST_TP3))
  return(TRANS)
}))
head(INTERPERSONAL_DIST_TP3)[,c(1:4)]
```

```
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$NexteraXT_Virome_SampleID.x %in% i),"NexteraXT_Virome_SampleID.x"])]
  SUBJECT <- MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"SubjectID"]
  SITE <-
 as.vector(MAP_MERGE_REF[MAP_MERGE_REF$NexteraXT_Virome_SampleID.x==i,"Site_Symbol"])
  TRANS <- data.frame(t(INTERPERSON_DIST_T2))
  RESULT <- data.frame(X=c(SUBJECT, SITE, INTERPERSON_DIST_T2))
  return(TRANS)
}))
head(INTERPERSONAL_DIST_TP2)[,c(1:4)]
```

EXPECTED RESULTS

```
##    X                 X.1                 X.2                 X.3
## 1  MG100425          MG100283            MG100251            MG100267
## 2  1                 1                   1                   1
## 3  Ac                Ax                  Fh                  Oc
## 4  0.68106733910108  0.627045291391653   0.38613104099571    0.606457024500615
```

```
##           MG100425  MG100283  MG100251  MG100267
## MG100632  0.8905969 0.7000370 0.6604504 0.6138396
## MG100457  0.5052325 0.7546691 0.8898742 0.8971785
## MG100616  0.7112431 0.7428203 0.6193057 0.3955118
## MG100647  0.7544944 0.7635985 0.7276292 0.7268521
## MG100441  0.6827725 0.8227614 0.6277412 0.5954542
## MG100473  0.5301470 0.7833859 0.4859976 0.5672776
```

```
##           MG100425  MG100283  MG100251  MG100267
## MG100283  0.7811457 0.7811457 0.4822732 0.5395027
## MG100251  0.4822732 0.7636557 0.7636557 0.7703925
## MG100267  0.5395027 0.7703925 0.3816939 0.3816939
## MG100299  0.8537536 0.8952908 0.7917489 0.7957845
## MG100195  0.8723981 0.9160835 0.8147091 0.8071813
## MG100235  0.5958880 0.7292550 0.4572400 0.5011740
```

Step 12.

Melt the two interpersonal distance data frames.

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 13.

Get intrapersonal values in same format.

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 14.

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)]
head(BOUND_DIST)
```

EXPECTED RESULTS

```
##      value      Type
## 1 0.6810673   Intra
## 2 0.6270453   Intra
## 3 0.3861310   Intra
## 4 0.6064570   Intra
## 5 0.8635676   Intra
## 6 0.8629480   Intra
```

Step 15.

Plot the resulting distances as means with standard error.

cmd COMMAND

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

EXPECTED RESULTS

```
##    Type     N      mean       sd          se
## 1 Inter 30504 0.7217809 0.1400811 0.0008020493
## 2 Intra   124 0.6310340 0.1350353 0.0121265309
```

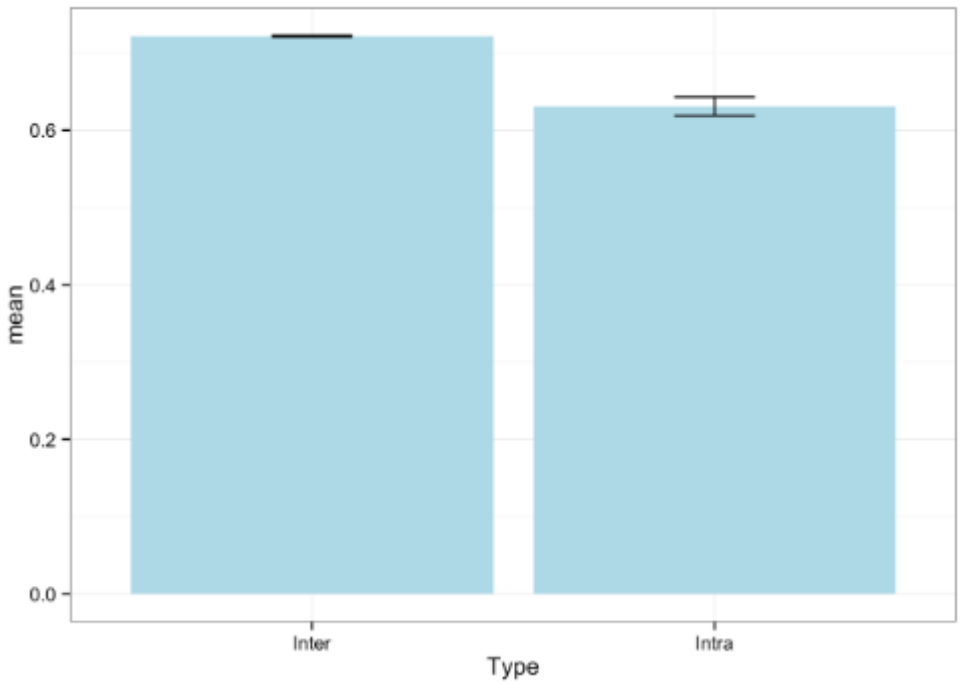
Step 16.

Visualize the data as a nice blue bar plot.

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 17.
Perform a t-test to determine the statistical significance of the difference between the two populations.

cmd COMMAND
```
t.test(BOUND_DIST$value ~ BOUND_DIST$Type)
```

EXPECTED RESULTS
```
##
## Welch Two Sample t-test
##
## data: BOUND_DIST$value by BOUND_DIST$Type
## t = 7.467, df = 124.08, p-value = 1.264e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 0.0666928 0.1148010
## sample estimates:
## mean in group Inter mean in group Intra
## 0.7217809 0.6310340
```

Step 18.
Finally, look at the temporal variation using the Jaccard Similarity Index.

cmd COMMAND
```
BOUND_DIST$simlr <- 1- BOUND_DIST$value
BOUND_DIST_SUB <- BOUND_DIST[c(which(BOUND_DIST$Type %in% "Intra")),]
JaccardPlot <-
 ggplot(BOUND_DIST_SUB, aes(x=Type, y=simlr)) + theme_classic() + geom_jitter(position = po
sition_jitter(width = .2))

JaccardPlot
```

EXPECTED RESULTS