Script R11: Replication Cycle

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This section outlines the analyses we used in our replication cycle section of our report. We first predict how many contigs are potentially of the temperate replication cycle and display this information using a Euler diagram. We then use a relative abundance approach by visualizing the percent of temperate phages present at each site. We end by visualizing the relative abundances of bacteria annotations of the phage contigs. 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 R11: Replication Cycle. **protocols.io** dx.doi.org/10.17504/protocols.io.ejfbcjn
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(venneuler)
packageVersion("venneuler")

library(reshape2)
packageVersion("reshape2")

library(ggplot2)
packageVersion("ggplot2")

library(plyr)
packageVersion("plyr")

library(pgirmess)
packageVersion("pgirmess")

library(VennDiagram)
packageVersion("VennDiagram")
```

EXPECTED RESULTS

```
## Loading required package: rJava

## [1] '1.1.0'


## [1] '1.4.1'


## [1] '1.0.1'


## [1] '1.8.2'


## [1] '1.6.0'


## [1] '1.6.9'
```

Step 2.
Read in the data.

cmd COMMAND

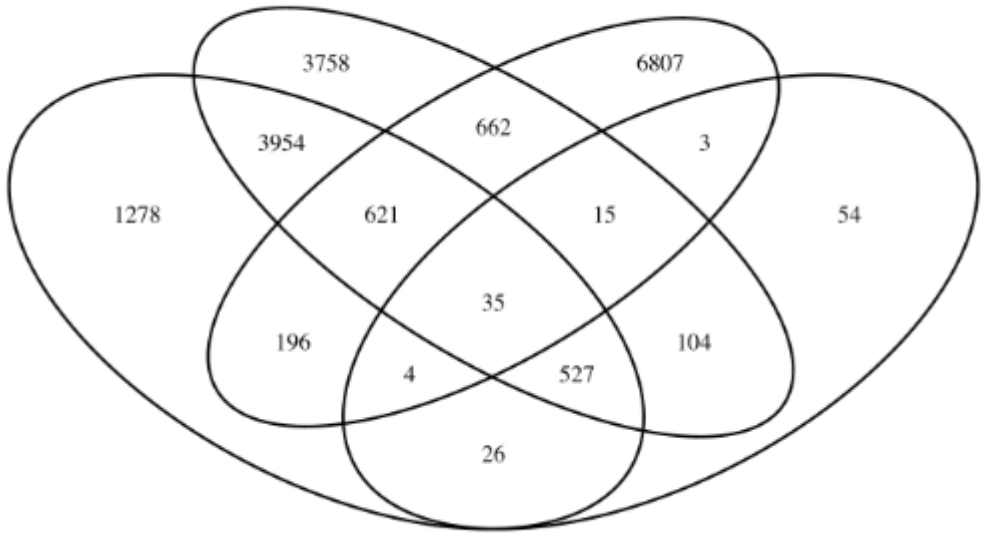
```
Phage <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/phage_contigs_no_negs_uniq.tx
t", header=FALSE, sep="\t")
Phage$V1 <- as.character(Phage$V1)
Integrase <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/int_contigs_no_negs_uniq.txt"
, header=FALSE, sep="\t")
Integrase$V1 <- as.character(Integrase$V1)
Aclame <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/ACLAME_contigs_no_negs_uniq.t
xt", header=FALSE, sep="\t")
Bacteria <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/bacteria_hits_contigs_no_negs
_uniq.txt", header=FALSE, sep="\t")
```

Step 3.
Make this Venn Diagram.

cmd COMMAND

```
draw.quad.venn(length( Phage[,1] ), length( Integrase[,1] ), length( Aclame[,1] ), length(
Bacteria[,1] ), length( (subset(Phage, is.element(V1, Integrase$V1) == TRUE)[,1] )), length
( (subset(Phage, is.element(V1, Aclame$V1) == TRUE)[,1] )), length( (subset(Phage, is.eleme
nt(V1, Bacteria$V1) == TRUE)[,1] )), length( (subset(Integrase, is.element(V1, Aclame$V1) =
= TRUE)[,1] )), length( (subset(Integrase, is.element(V1, Bacteria$V1) == TRUE)[,1] )), len
gth( (subset(Aclame, is.element(V1, Bacteria$V1) == TRUE)[,1] )), length( subset(subset(Pha
ge, is.element(V1, Integrase$V1) == TRUE), is.element(V1, Aclame$V1) == TRUE )[,1] ), lengt
h( subset(subset(Phage, is.element(V1, Integrase$V1) == TRUE), is.element(V1, Bacteria$V1)
== TRUE )[,1] ), length( subset(subset(Phage, is.element(V1, Aclame$V1) == TRUE), is.elemen
t(V1, Bacteria$V1) == TRUE )[,1] ), length( subset(subset(Integrase, is.element(V1, Aclame$
V1) == TRUE), is.element(V1, Bacteria$V1) == TRUE )[,1] ), length( subset(subset(subset(Pha
ge, is.element(V1, Integrase$V1) == TRUE), is.element(V1, Aclame$V1) == TRUE ), is.element(
V1, Bacteria$V1) == TRUE )[,1] ))
```

EXPECTED RESULTS



(polygon[GRID.polygon.4968], polygon[GRID.polygon.4969], polygon[GRID.polygon.4970],

polygon[GRID.polygon.4971], polygon[GRID.polygon.4972], polygon[GRID.polygon.4973], polygon[GRID.polygon.4974], polygon[GRID.polygon.4975], text[GRID.text.4976], text[GRID.text.4977], text[GRID.text.4978], text[GRID.text.4979], text[GRID.text.4980], text[GRID.text.4981], text[GRID.text.4982], text[GRID.text.4983], text[GRID.text.4984], text[GRID.text.4985], text[GRID.text.4986], text[GRID.text.4987], text[GRID.text.4988], text[GRID.text.4989], text[GRID.text.4990], text[GRID.text.4991], text[GRID.text.4992], text[GRID.text.4993], text[GRID.text.4994])

Step 4.

Next, we calculated the relative abundance of temperate and non-temperate phages across skin sites, which is done by quantifying the numbers of reads mapping contigs, instead of numbers of contigs (so a more accurate relative abundance instead of just counting reference contigs). First upload the needed input files.

cmd COMMAND

```
INPUT <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/phage_lifecycle_otu_table_for
_rel_abund.tsv", header=TRUE, sep="\t")
INPUT[c(1:4),c(1:4)]

MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv", s
ep="\t", header=TRUE)
MAP[c(1:4),c(1:4)]
```

Step 5.

To get the sums of the temperate and non-temperate relative abundances for each sample, first split the data frame based on whether the row is assigned to a temperate or non-temperate phage.

cmd COMMAND

```
SUMS_NON_TEMP <- as.data.frame(colSums(INPUT[c(INPUT$Contig_ID=="Non-
Temperate_Phage"), -1]))
SUMS_TEMP <- as.data.frame(colSums(INPUT[c(INPUT$Contig_ID=="Temperate_Phage"), -1]))
```

Step 6.

Set the column names.

cmd COMMAND

```
colnames(SUMS_NON_TEMP) <- c("Non-Temp")
colnames(SUMS_TEMP) <- c("Temp")
```

Step 7.

Set the row names.

cmd COMMAND

```
SUMS_NON_TEMP$SampleID <- row.names(SUMS_NON_TEMP)
SUMS_TEMP$SampleID <- row.names(SUMS_TEMP)
```

Step 8.

Melt for formatting required by ggplot2.

cmd COMMAND

```
NON_TEMP_MELT <- melt(SUMS_NON_TEMP)
TEMP_MELT <- melt(SUMS_TEMP)
```

Step 9.

Merge the data frames.

cmd COMMAND

```
SUMS_MERGED <- merge(TEMP_MELT, NON_TEMP_MELT, by="SampleID")
SUMS_MERGED$Percent_temperate <-
 100 * SUMS_MERGED$value.x / (SUMS_MERGED$value.x + SUMS_MERGED$value.y)
head(SUMS_MERGED, n=5)
```

EXPECTED RESULTS

##	SampleID	variable.x	value.x	variable.y	value.y	Percent_temperate
## 1	MG100098	Temp	20864.59	Non-Temp	3041.170	87.27851
## 2	MG100099	Temp	18828.67	Non-Temp	2349.366	88.90659
## 3	MG100100	Temp	10942.90	Non-Temp	2069.876	84.09351
## 4	MG100101	Temp	20092.34	Non-Temp	5941.926	77.17652
## 5	MG100102	Temp	15654.27	Non-Temp	3395.954	82.17368

Step 10.

Merge the mapping file to the relative abundance data frame.

cmd COMMAND

```
MAP_MERGED <- merge(SUMS_MERGED, MAP, by.x="SampleID", by.y="NexteraXT_Virome_SampleID")
```

Step 11.

We will not be using time point 1, the incomplete time point.

cmd COMMAND

```
MAP_MERGED_SUBSET <- MAP_MERGED[-which(MAP_MERGED$TimePoint %in% 1), ]
```

Step 12.

We will not be including these four locations.

cmd COMMAND

```
MAP_MERGED_SUBSET <- MAP_MERGED_SUBSET[-
which(MAP_MERGED_SUBSET$Site_Symbol %in% c("Ba","Ph","Vf","Neg")), ]
```

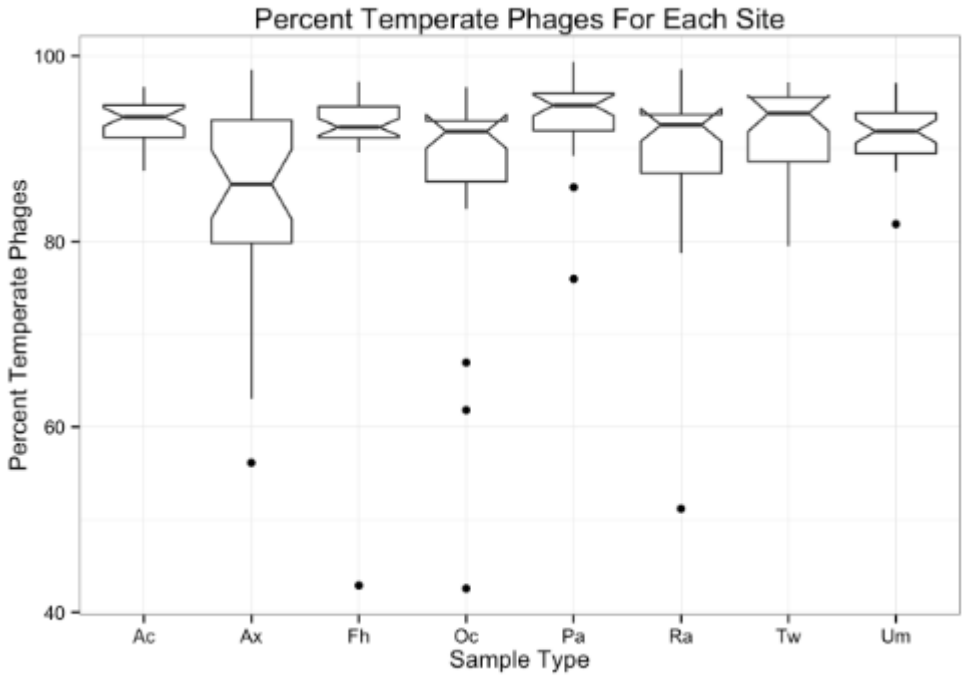
Step 13.

We then need to generate the boxplot we will use to visualize the data.

cmd COMMAND

```
ggplot(MAP_MERGED_SUBSET, aes(x=Site_Symbol, y=Percent_temperate)) + theme_bw() + geom_boxp
lot(notch=TRUE) + ggtitle("Percent Temperate Phages For Each Site") + ylab("Percent Tempera
te Phages") + xlab("Sample Type")
```

EXPECTED RESULTS



Step 14.

We can use the kruskalmc function to determine which sites are significantly different from each

other, after correction and other considerations.

cmd COMMAND

```
MAP_MERGED_SUBSET$Site_Symbol <- factor(MAP_MERGED_SUBSET$Site_Symbol)
kruskalmc(MAP_MERGED_SUBSET$Percent_temperate, MAP_MERGED_SUBSET$Site_Symbol)
```

The category needs to be factored to be used with the following function.

Step 15.

Finally, we will visualize the relative abundances of the bacteria that were also annotated as containing phages. First read in the input relative abundance file.

cmd COMMAND

```
INPUT_BAC_REL_ABUND <-
 read.delim("../../IntermediateOutput/Phage_replication_cycle/final_contig_quant_annotation
_ncbi.tsv", sep="\t", header=TRUE)
INPUT_BAC_REL_ABUND$Percent <-
 100 * INPUT_BAC_REL_ABUND$Number_Contigs / sum(INPUT_BAC_REL_ABUND$Number_Contigs)
INPUT_ORDER <- INPUT_BAC_REL_ABUND[c(order(INPUT_BAC_REL_ABUND$Bacterial_Phylum)), ]
head(INPUT_ORDER, n=5)
```

EXPECTED RESULTS

##	Number_Contigs	Bacterial_Genus	Bacterial_Phylum	Percent
## 14	3	Mobiluncus	Actinobacteria	0.3504673
## 16	42	Propionibacterium	Actinobacteria	4.9065421
## 4	1	Anaerococcus	Firmicutes	0.1168224
## 5	1	Anoxybacillus	Firmicutes	0.1168224
## 6	2	Bacillus	Firmicutes	0.2336449

Step 16.

We can plot the resulting relative abundance information as a pie chart.

cmd COMMAND

```
ggplot(INPUT_ORDER, aes(x="", y=Percent, fill=Bacterial_Phylum)) + theme_bw() + geom_bar(wi
dth=1, stat="identity") + coord_polar(theta="y")
```

EXPECTED RESULTS