Script R14: CARD Antibiotic Resistance Analysis

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol outlines our analysis of the potential antibiotic resistance genes found within the skin virome. We start by visualizing the relative abundances of the top 10 antibiotic resistance categories (according to the CARD). We then quantify the number of unique antibiotic resistance gene categories annotated in the samples at each skin site, and also look at the percent CARD annotated ORFs of the overall ORFs present in each sample at each skin site.

Citation: HANNIGAN GD, GRICE EA, ET AL. Script R14: CARD Antibiotic Resistance Analysis. **protocols.io** dx.doi.org/10.17504/protocols.io.ejnbcme
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(plyr)
packageVersion("plyr")

library(reshape2)
packageVersion("reshape2")

library(ggplot2)
packageVersion("ggplot2")

library(vegan)
packageVersion("vegan")

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

EXPECTED RESULTS

```
## [1] '1.8.2'



## [1] '1.4.1'



## [1] '1.0.1'



## [1] '2.3.0'



## [1] '1.6.0'
```

Step 2.
Load in the data files that will be analyzed.

cmd COMMAND

```
INPUT <-
 read.delim("../../IntermediateOutput/CARD_abx_resistance/CARD_annotated_orfs_in_otu_table.
tsv", sep="\t", header=TRUE)
#See a quick summary of the data frame we just loaded into R
INPUT[c(1:5),c(1:5)]

MAP <-
 read.delim("../../IntermediateOutput/Mapping_files/SkinMet_and_Virome_001_metadata.tsv", s
ep="\t", header=TRUE)
MAP[c(1:5),c(1:5)]
```

Step 3.

Format the input dataframes and prepare for visualization. To do this, we determine which categories have the top 10 relative abundance, and sum the other category relative abundances into the “other” category.

Here we also 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.

cmd COMMAND

```
MAP_SUBSET <- MAP[-which(MAP$TimePoint %in% 1), ]
MAP_SUBSET <- MAP_SUBSET[-which(MAP_SUBSET$NexteraXT_Virome_SampleID %in% NA), ]
MAP_SUBSET <- MAP_SUBSET[-which(MAP_SUBSET$Site_Symbol %in% c("Ba","Ph","Vf","Neg")), ]
```

Step 4.

Sum the columns be rows, such that, for example, I sum all of the column values whose rows correspond to tetracycline_inactivation_enzyme.

cmd COMMAND

```
UNIQ_ORF_NAMES <- as.vector(unique(INPUT$ORF_ID))
SUM_BY_NAMES <- data.frame(lapply(UNIQ_ORF_NAMES, function(i) {
  SUBSET <- data.frame(INPUT[c(INPUT$ORF_ID==i), ])
  SUMS <- data.frame(colSums(SUBSET[,-1]))
  colnames(SUMS) <- c(i)
  return(SUMS)
}))
SUM_BY_NAMES$SampleID <- rownames(SUM_BY_NAMES)
MELT_SUM <- melt(SUM_BY_NAMES)

MEAN_NAMES <- ddply(MELT_SUM, c("variable"), summarise, mean=mean(value))
SORT_MEAN_NAMES <- MEAN_NAMES[order(-MEAN_NAMES$mean),]
TOP10 <- as.vector(SORT_MEAN_NAMES[1:10,1])

TOP10_MELT <- MELT_SUM[which(MELT_SUM$variable %in% TOP10),]
OTHER_MELT <- MELT_SUM[-which(MELT_SUM$variable %in% TOP10),]
```

Step 5.

Sum values by sampleID.

cmd COMMAND

```
OTHER_MELT_SUM <- tapply(OTHER_MELT$value, INDEX=list(OTHER_MELT$SampleID), FUN=sum)
ROWNAMES <- c(row.names(OTHER_MELT_SUM))
```

Step 6.

You can confirm the merge here.

cmd COMMAND

```
OTHER_MELT_SUM_WITH_NAMES <- data.frame(cbind(ROWNAMES, OTHER_MELT_SUM))
colnames(OTHER_MELT_SUM_WITH_NAMES) <- c("SampleID", "value")
OTHER_MELT_SUM_WITH_NAMES$value <-
 as.numeric(as.character(OTHER_MELT_SUM_WITH_NAMES$value))
OTHER_MELT_SUM_WITH_NAMES$variable <- "Other"
```

Step 7.

Reorder the columns to match the TOP10_MELT

cmd COMMAND

```
OTHER_MELT_FOR_MERGE <- OTHER_MELT_SUM_WITH_NAMES[ ,c(1,3,2)]
```

Step 8.

Confirm the other is properly merged.

cmd COMMAND

```
unique(rbind(TOP10_MELT, OTHER_MELT_FOR_MERGE)$variable)

TOP10_WITH_OTHER <- rbind(TOP10_MELT, OTHER_MELT_FOR_MERGE)
```

EXPECTED RESULTS

```
##  [1] antibiotic_resistant_DNA_topoisomerase_subunit_gyrA
##  [2] antibiotic_resistant_gene_variant_or_mutant
##  [3] antibiotic_resistant_DNA_topoisomerase_subunit_parC
##  [4] elfamycin_resistance_gene
##  [5] subunit_of_efflux_pump_conferring_antibiotic_resistance
##  [6] OXA_beta.lactamase
##  [7] antibiotic_resistant_DNA_topoisomerase_subunit_parE
##  [8] rifampin_inactivation_enzyme
##  [9] tetracycline_resistance_MFS_efflux_pump
## [10] TEM_beta.lactamase
## [11] Other
## 30 Levels: antibiotic_resistant_DNA_topoisomerase_subunit_gyrA ...
```

Step 9.
Test that the summing worked.

cmd COMMAND

```
sum(OTHER_MELT[c(OTHER_MELT$SampleID == "MG100098"),"value"])

TOP10_MERGE <-
 merge(TOP10_WITH_OTHER, MAP_SUBSET, by.x="SampleID", by.y="NexteraXT_Virome_SampleID")

TOP10_MERGE_SORT <- TOP10_MERGE[order(TOP10_MERGE$variable), ]
```

EXPECTED RESULTS

```
## [1] 28.1389
```

Step 10.
Order according to relative abundance.

cmd COMMAND

```
TOP10_WITH_OTHER <- ddply(TOP10_MERGE_SORT, c("variable"), summarise, mean=mean(value))
TOP10_WITH_OTHER <- TOP10_WITH_OTHER[!c(TOP10_WITH_OTHER$variable=="Other"),]
TOP10_WITH_OTHER <- TOP10_WITH_OTHER[order(TOP10_WITH_OTHER$mean, decreasing=TRUE), ]
ORDER_MEAN_NAMES_WITH_OTHER <- as.vector(TOP10_WITH_OTHER$variable)
```

Step 11.
Append other to the vector.

cmd COMMAND

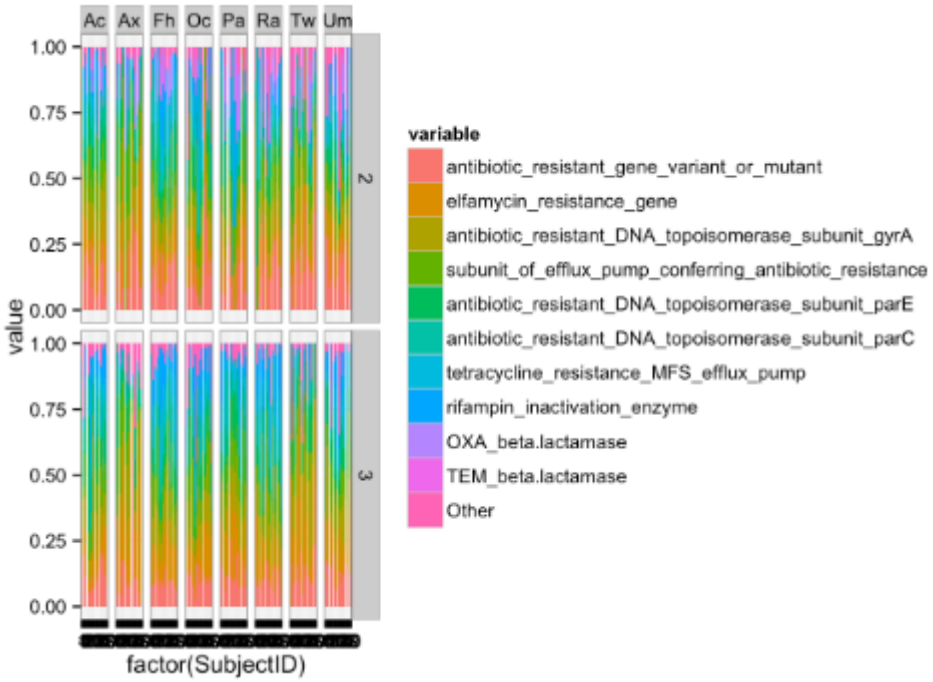
```
ORDER_MEAN_NAMES_WITH_OTHER <- c(ORDER_MEAN_NAMES_WITH_OTHER, "Other")
TOP10_MERGE_SORT$variable <-
 factor(TOP10_MERGE_SORT$variable, levels=c(ORDER_MEAN_NAMES_WITH_OTHER))
```

Step 12.
Use ggplot2 to visualize the antibiotic resistance gene relative abundance information (top 10 taxa).

cmd COMMAND

```
ggplot(TOP10_MERGE_SORT, aes(x=factor(SubjectID), y=value, fill=variable, order=variable))
+ theme_bw() + geom_bar(stat="identity", position="fill") + facet_grid(TimePoint~Site_Symbo
l, scales="free")
```

EXPECTED RESULTS



Step 13.

Plot it again without the legend so we can actually see what is going on.

cmd COMMAND

```
ggplot(TOP10_MERGE_SORT, aes(x=factor(SubjectID), y=value, fill=variable, order=variable))
+ theme_bw() + theme(legend.position="none") + geom_bar(stat="identity", position="fill") +
 facet_grid(TimePoint~Site_Symbol, scales="free")
```

EXPECTED RESULTS