Day 10: Statistical Analysis (Global Gut)

Back to Table of Contents

All of the code in this page is meant to be run in `R` unless otherwise specified.

Loading a genus table and the metadata into R

Before loading data into R, this QIIME command must be run on the command line to collapse OTU counts into genus (L6) and phylum (L2) count tables:

```
# (run on command line)
summarize_taxa.py -i otu_table.biom -L 6

# convert to JSON BIOM format to load into R using R biom package:
biom convert -i otu_table_L6.biom -o otu_table_L6_json.biom --to-json
```

Inside `R`, Install biom package and vegan package if not installed.

```
install.packages(c('biom','vegan'),repo='http://cran.wustl.edu')
```

Load biom package, vegan package; load data

```
library('biom')
library('vegan')

# load biom file
genus.biom <- read_biom('otu_table_L6_json.biom')

# Extract data matrix (genus counts) from biom table
genus <- as.matrix(biom_data(genus.biom))

# transpose so that rows are samples and columns are genera
genus <- t(genus)

# load mapping file
map <- read.table('map.txt', sep='\t', comment='', head=T, row.names=1)
```

It is extremely important to ensure that your genus table and metadata table sample IDs are lined up correctly.

```
# find the overlapping samples
common.ids <- intersect(rownames(map), rownames(genus))

# get just the overlapping samples
genus <- genus[common.ids,]
map <- map[common.ids,]
```

See dimensions of genus table. Then drop genera present in $< 10\%$ of samples.

```
dim(genus)
```

```
## [1]  66 205
```

```
genus <- genus[,colMeans(genus > 0) >= .1]
dim(genus)
```

```
## [1] 66 97
```

Show only the first ten genera in genus table

```
colnames(genus)[1:10]
```

```
##  [1] "k__Archaea;p__Euryarchaeota;c__Methanobacteria;o__Methanobacteriales;f__Methanobacteriaceae;g__
##  [2] "k__Archaea;p__Euryarchaeota;c__Methanobacteria;o__Methanobacteriales;f__Methanobacteriaceae;g__
##  [3] "k__Bacteria;p__Actinobacteria;c__Actinobacteria;o__Actinomycetales;f__Actinomycetaceae;g__Actin
##  [4] "k__Bacteria;p__Actinobacteria;c__Actinobacteria;o__Bifidobacteriales;f__Bifidobacteriaceae;g__B
##  [5] "k__Bacteria;p__Actinobacteria;c__Coriobacteriia;o__Coriobacteriales;f__Coriobacteriaceae;g__"
##  [6] "k__Bacteria;p__Actinobacteria;c__Coriobacteriia;o__Coriobacteriales;f__Coriobacteriaceae;g__Adl
##  [7] "k__Bacteria;p__Actinobacteria;c__Coriobacteriia;o__Coriobacteriales;f__Coriobacteriaceae;g__Col
##  [8] "k__Bacteria;p__Actinobacteria;c__Coriobacteriia;o__Coriobacteriales;f__Coriobacteriaceae;g__Sla
##  [9] "k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__;g__"
## [10] "k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Bacteroidaceae;g__Bacteroides"
```

Show the first 10 rows and first 2 columns of the genus table

```
genus[1:10,1:2]
```

```
##                k__Archaea;p__Euryarchaeota;c__Methanobacteria;o__Methanobacteriales;f__Methanobacteri
## h122M.1.418534
## h85M.1.418596
## h95M.1.418831
## h9M.1.418588
## k278A.2.418424
## h146M.1.418838
## h147M.1.418531
## h101M.1.418586
## h165M.1.418394
## h186M.1.418788
##                k__Archaea;p__Euryarchaeota;c__Methanobacteria;o__Methanobacteriales;f__Methanobacteri
## h122M.1.418534
## h85M.1.418596
## h95M.1.418831
## h9M.1.418588
## k278A.2.418424
## h146M.1.418838
## h147M.1.418531
## h101M.1.418586
## h165M.1.418394
## h186M.1.418788
```

See available columns in the metadata

```
colnames(map)
```

```
##  [1] "BarcodeSequence"              "LinkerPrimerSequence"
##  [3] "AGE"                          "AGE_GROUP"
##  [5] "BODY_SITE"                    "COUNTRY"
##  [7] "ELEVATION"                    "EXPERIMENT_DESIGN_DESCRIPTION"
##  [9] "FAMILY_RELATIONSHIP"          "HOST_COMMON_NAME"
## [11] "HOST_SUBJECT_ID"              "LATITUDE"
## [13] "LONGITUDE"                    "PCR_PRIMERS"
## [15] "REGION"                       "RUN"
## [17] "RUN_CENTER"                   "RUN_DATE"
## [19] "RUN_LANE"                     "SEX"
## [21] "STUDY_ABSTRACT"               "Description"
```

Show how many samples are from each Country

```
table(map$COUNTRY)
```

```
##
##                   GAZ:Malawi GAZ:United States of America
##                           22                           22
##                GAZ:Venezuela
##                           22
```

Basic association testing

Let's run some tests on Prevotella. First extract the Prevotella column and save it to a variable `prevotella` for convenience.

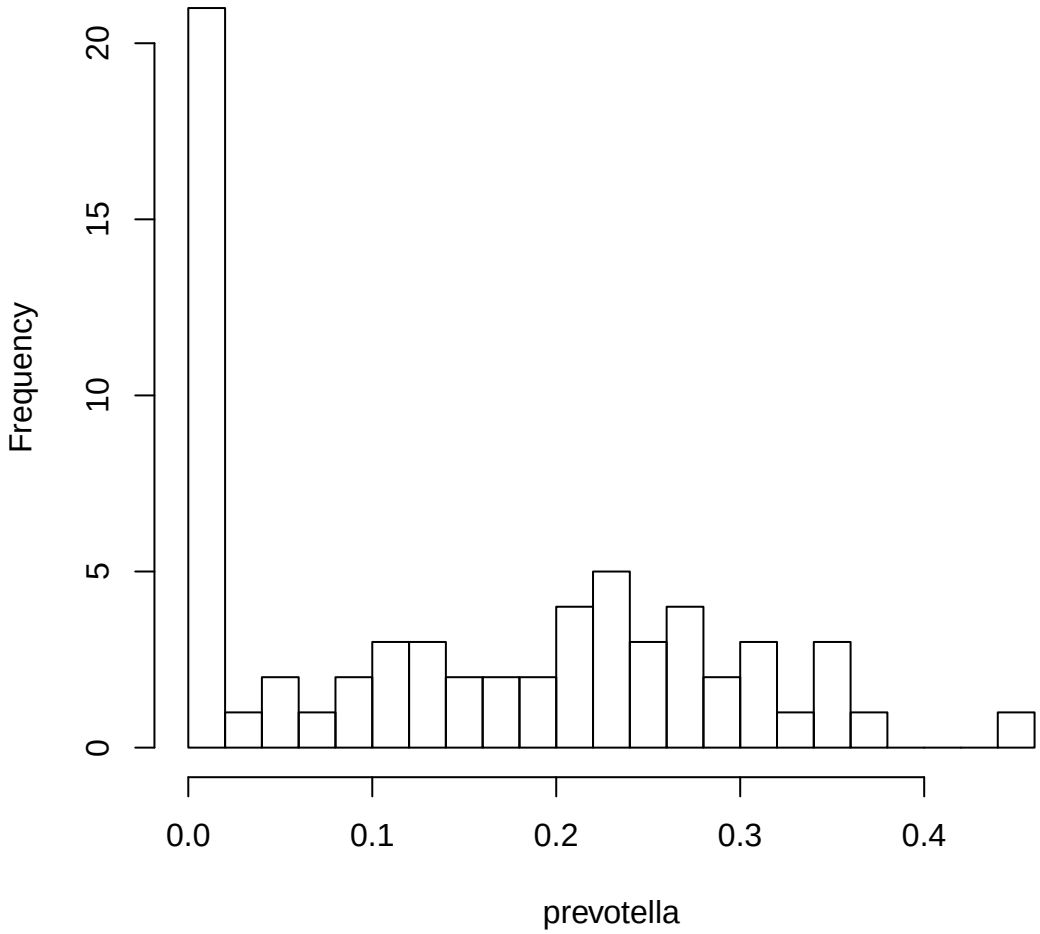
```
# find out what column Prevotella is in
grep('g__Prevotella',colnames(genus))
```

```
## [1] 13
```

```
# save that column in a variable
prevotella <- genus[,grep('g__Prevotella',colnames(genus))]
```

Visualize the distribution of Prevotella

```
# find out what column Prevotella is in
hist(prevotella, br=30)
```



Run a test of Pearson's correlation of Prevotella and age. Note that the result is not quite significant (p=0.0531).

```
cor.test(prevotella, map$AGE)
```

```
##
##  Pearson's product-moment correlation
##
## data:  prevotella and map$AGE
## t = -1.9704, df = 64, p-value = 0.05312
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.454858210  0.003055594
## sample estimates:
##       cor
## -0.239154
```

Now run a linear regression of prevotella against age. Notice that statistically this is equivalent to running the Pearson's correlation. The p-value in row 2 column 4 of the "Coefficients" table is the same as the p-value from the correlation test.

```
# fit a linear model. The "~" means "as a function of"
fit <- lm(prevotella ~ map$AGE)

# print a summary of the results
summary(fit)
```

```
##
## Call:
## lm(formula = prevotella ~ map$AGE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18946 -0.12186 -0.01805  0.10726  0.30337
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2009925  0.0319288   6.295 3.15e-08 ***
## map$AGE     -0.0019218  0.0009753  -1.970   0.0531 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.124 on 64 degrees of freedom
## Multiple R-squared:  0.05719,    Adjusted R-squared:  0.04246
## F-statistic: 3.883 on 1 and 64 DF,  p-value: 0.05312
```

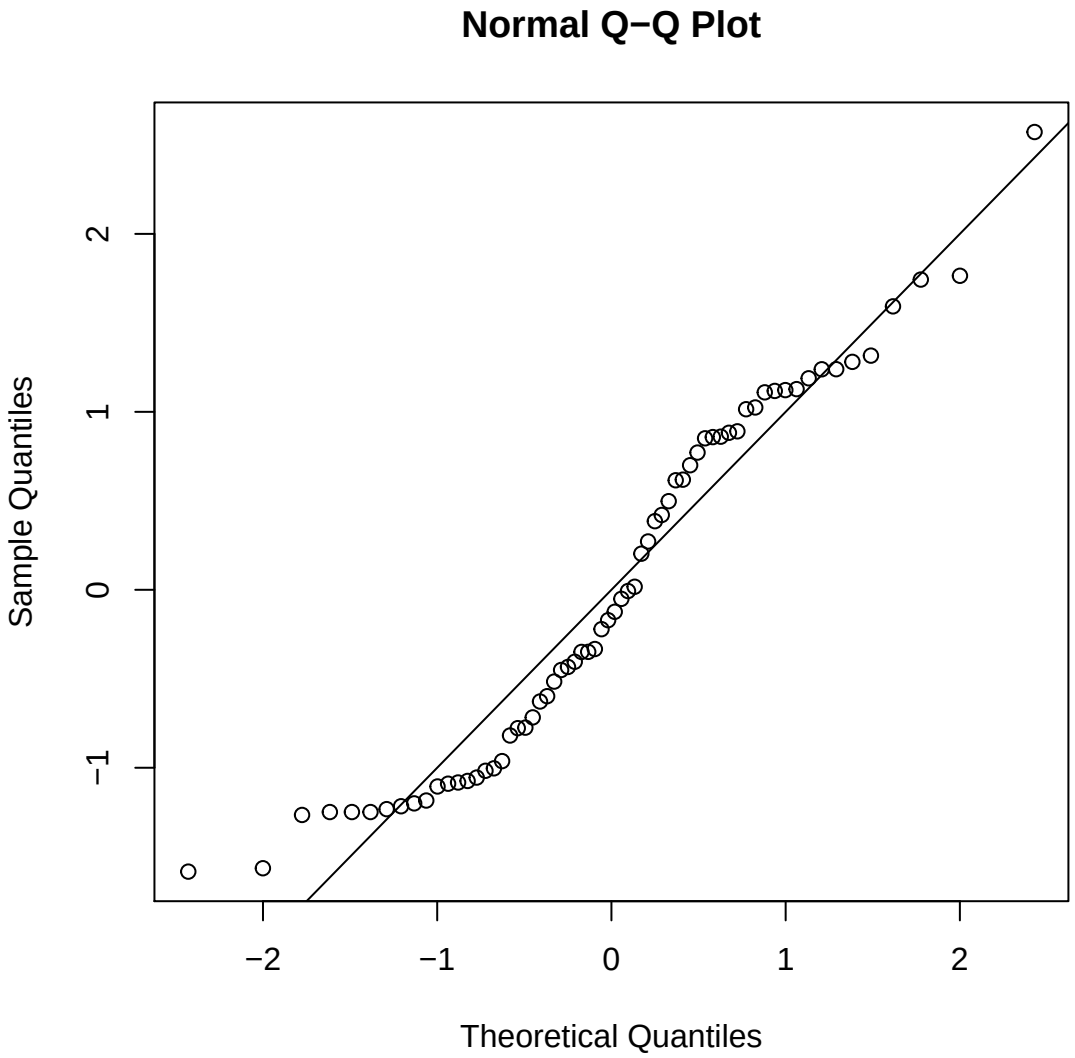
```
# A nice way to get the exact p-value for the age regression coefficient using the anova function
pval <- anova(fit)['map$AGE','Pr(>F)']
pval
```

```
## [1] 0.0531206
```

Testing for normally distributed data

We can test whether the residuals are normally distributed Kolmogorov-Smirnov test. If $p < 0.05$, we can reject the null hypothesis that the data came from a normal distribution, meaning that the linear test is not appropriate.

```
# Make a quantile-quantile plot of the (studentized) residuals vs. a normal distribution
qqnorm(rstudent(fit)); abline(0,1)
```




```
# Kolmogorov-Smirnov test
ks.test(rstudent(fit), pnorm, mean=mean(rstudent(fit)), sd=sd(rstudent(fit)))
```

```
## Warning in ks.test(rstudent(fit), pnorm, mean = mean(rstudent(fit)), sd =
## sd(rstudent(fit))): ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstudent(fit)
## D = 0.10218, p-value = 0.496
## alternative hypothesis: two-sided
```

Controling for confounders

Perhaps country of origin is a confounder that is obscuring the association of Prevotella and Age. Using lm() we can add confounders to the regression. Now after removing the effects of country, there is a strong association of Prevotella and age.

```
# fit a linear model. The "~" means "as a function of"
fit <- lm(prevotella ~ map$AGE + map$COUNTRY)

# print a summary of the results
summary(fit)
```

```
##
## Call:
## lm(formula = prevotella ~ map$AGE + map$COUNTRY)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.150373 -0.040611 -0.008627  0.033487  0.199252
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             0.2982044  0.0238895  12.483
## map$AGE                                -0.0016456  0.0006193  -2.657
## map$COUNTRYGAZ:United States of America -0.2311571  0.0237310  -9.741
## map$COUNTRYGAZ:Venezuela                -0.0843013  0.0237296  -3.553
##                                         Pr(>|t|)
## (Intercept)                              < 2e-16 ***
## map$AGE                                 0.010010 *
## map$COUNTRYGAZ:United States of America 4.08e-14 ***
## map$COUNTRYGAZ:Venezuela                0.000736 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07861 on 62 degrees of freedom
## Multiple R-squared:  0.6329, Adjusted R-squared:  0.6152
## F-statistic: 35.63 on 3 and 62 DF,  p-value: 1.641e-13
```

Testing multiple hypotheses

We have so far only tested one genus. Let's test them all using a loop.

```
# pvals is a vector initialized with zeroes
# with enough slots for the different genera
pvals <- numeric(ncol(genus))

# "name" the pvalues after the genera
names(pvals) <- colnames(genus)

# Loop through the columns of the genus table, testing each one
for(i in 1:ncol(genus)) {
    fit <- lm(genus[,i] ~ map$AGE + map$COUNTRY)
    pvals[i] <-  anova(fit)['map$AGE','Pr(>F)']
```

```
}

# note, you could put this all on one line with:
# for(i in 1:ncol(genus)) pvals[i] <-  anova(lm(genus[,i] ~ map$AGE + map$COUNTRY))['map$AGE','Pr(>F)']

# print the 10 smallest p-values:
sort(pvals)[1:10]
```

```
##                                                                                                  k__Bacteria;p__WPS-2;c__
##
##                        k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales;f__Prevotellaceae;g
##
##                                                  k__Bacteria;p__Verrucomicrobia;c__Verruco-5;o__WCHB1-41
##
##                                 k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Christensen
##
##                          k__Bacteria;p__Firmicutes;c__Bacilli;o__Lactobacillales;f__Streptococcaceae;g__
##
##                                                  k__Bacteria;p__Proteobacteria;c__Alphaproteobacteria;o__
##
## k__Archaea;p__Euryarchaeota;c__Methanobacteria;o__Methanobacteriales;f__Methanobacteriaceae;g__Methan
##
##                                              k__Bacteria;p__Bacteroidetes;c__Bacteroidia;o__Bacteroidales
##
##                     k__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Ruminococcaceae;g__Faec
##
##                                                    k__Bacteria;p__Proteobacteria;c__Alphaproteobacteria
##
```