

WLS_analysis

Kim Dill-McFarland

April 20, 2017

Load data

Packages

```
#The vegan package provides tools for descriptive community ecology. It has most basic functions of diversity analysis.  
#In general, this package is used for Bray-Curtis and Jaccard analyses.  
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.4-6
```

```
#The phyloseq package seeks to address issues with multiple microbiome analysis packages by providing a unified interface.  
#In general, this package is used for UniFrac analyses.  
library(phyloseq)  
#Graphing package used in phyloseq. To edit the default setting of a plot, you need to use functions in the ggplot2 package.  
library(PMCMR)
```

```
## PMCMR is superseded by PMCMRplus and will be no longer maintained. You may wish to install PMCMRplus.  
#Graphing package used in phyloseq. To edit the default setting of a plot, you need to use functions in the ggplot2 package.  
library(ggplot2)  
#Figures  
library(lattice)  
library(gridExtra)
```

Functions

- R-krusk: Function to calculate Kruskal-Wallis for OTUs and variables identified in SIMPER analysis. From A. Steinberger
- Multiplot: for placing 2 or more ggplot2 graphs in one figure. From Cookbook for R
- Pairwise_adonis: Pairwise PERMANOVA (adonis) function to test within categorical variables with more than 2 levels. Modified from P. M. Arbizu

```
source("R_krusk.R")  
source("multiplot.R")  
source("Pairwise_adonis.R")
```

Data

Load environment from WLS_data_manipulation.Rmd

```
load("WLS_environment_final.rdata")
```

Gradutes-only

Beta-diversity

Tests performed on the graduate samples to avoid possible interactions between spouses and siblings. First, check if the graduate subset is different from the other sample types to ensure the subset is a good representation of the overall data set.

```
#Test if the overall microbiota differs by groups (g, p, s, e). We do not expect significance as these  
#Permutational analysis of variance (PERMANOVA)  
adonis(OTU ~ Link, data=Met, method="bray", permutations=1000)
```

```
##  
## Call:  
## adonis(formula = OTU ~ Link, data = Met, method = "bray", permutations = 1000)  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Terms added sequentially (first to last)  
##  
##          Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)  
## Link      3      0.728 0.24269 0.96004 0.00708 0.557  
## Residuals 404    102.129 0.25280      0.99292  
## Total    407    102.858      1.00000
```

```
adonis(OTU ~ Link, data=Met, method="jaccard", permutations=1000)
```

```
##  
## Call:  
## adonis(formula = OTU ~ Link, data = Met, method = "jaccard",      permutations = 1000)  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Terms added sequentially (first to last)  
##  
##          Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)  
## Link      3      0.99 0.32987 0.97191 0.00717 0.602  
## Residuals 404    137.12 0.33940      0.99283  
## Total    407    138.11      1.00000
```

```
adonis(wUF.dist ~ Link, data=Met, permutations=1000)
```

```
##  
## Call:  
## adonis(formula = wUF.dist ~ Link, data = Met, permutations = 1000)  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Terms added sequentially (first to last)  
##  
##          Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)  
## Link      3      0.1056 0.03521 1.0567 0.00779 0.35  
## Residuals 404    13.4613 0.03332      0.99221
```

```
## Total      407      13.5669                1.00000
adonis(uwUF.dist ~ Link, data=Met, permutations=1000)
```

```
##
## Call:
## adonis(formula = uwUF.dist ~ Link, data = Met, permutations = 1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Link           3      0.689 0.22952  1.0697 0.00788 0.218
## Residuals    404     86.682 0.21456           0.99212
## Total       407     87.370                1.00000
```

There are no differences so we move forward with the graduate subset data set. We will run a number of PERMANOVA (adonis) tests of variables of interest. All tests are run on Bray-Curtis, Jaccard, weighted and unweighted UniFrac distance measures. For variables with missing data, the data is subset to only complete.cases to avoid NA errors in adonis.

Sex and age

```
#Does sex/gender or age correlate to the gut microbiota?
adonis(BC.dist.g ~ gender+age, data = vars.g, permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g ~ gender + age, data = vars.g, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2  Pr(>F)
## gender         1      0.983 0.98276  3.9154 0.02164 0.000999 ***
## age            1      0.264 0.26409  1.0521 0.00581 0.348651
## Residuals    176     44.176 0.25100           0.97255
## Total       178     45.423                1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(J.dist.g ~ gender+age, data = vars.g, permutations=1000)
```

```
##
## Call:
## adonis(formula = J.dist.g ~ gender + age, data = vars.g, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
```

```
##
##              Df SumsOfSqs MeanSqs F.Model      R2    Pr(>F)
## gender        1      0.913 0.91259  2.7046 0.01505 0.000999 ***
## age           1      0.353 0.35336  1.0473 0.00583 0.331668
## Residuals    176     59.385 0.33742          0.97913
## Total        178     60.651          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
adonis(wUF.dist.g ~ gender+age, data = vars.g, permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g ~ gender + age, data = vars.g, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2    Pr(>F)
## gender        1      0.1888 0.188788  4.5515 0.02512 0.000999 ***
## age           1      0.0275 0.027540  0.6640 0.00366 0.861139
## Residuals    176      7.3001 0.041478          0.97122
## Total        178      7.5164          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
adonis(uwUF.dist.g ~ gender+age, data = vars.g, permutations=1000)
```

```
##
## Call:
## adonis(formula = uwUF.dist.g ~ gender + age, data = vars.g, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2    Pr(>F)
## gender        1      0.562 0.56205  2.59673 0.01447 0.000999 ***
## age           1      0.175 0.17473  0.80726 0.00450 0.874126
## Residuals    176     38.094 0.21644          0.98103
## Total        178     38.831          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since sex is a significant determinant of the microbiota and can impact a number of other variables of interest, we test if our variables vary by sex.

```
chisq.test(table(vars.g$gender, as.factor(vars.g$AB)))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$gender, as.factor(vars.g$AB))
## X-squared = 0.025893, df = 1, p-value = 0.8722
```

```

chisq.test(table(vars.g$RU57, vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$RU57, vars.g$gender)
## X-squared = 1.115, df = 1, p-value = 0.291
chisq.test(table(vars.g$RU11, vars.g$gender))

## Warning in chisq.test(table(vars.g$RU11, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$RU11, vars.g$gender)
## X-squared = 1.7123, df = 1, p-value = 0.1907
chisq.test(table(vars.g$iq, vars.g$gender))

## Warning in chisq.test(table(vars.g$iq, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$iq, vars.g$gender)
## X-squared = 32.254, df = 40, p-value = 0.8032
chisq.test(table(vars.g$yrs.edu, vars.g$gender))

## Warning in chisq.test(table(vars.g$yrs.edu, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$yrs.edu, vars.g$gender)
## X-squared = 9.7847, df = 7, p-value = 0.2011
chisq.test(table(vars.g$cohabYN, vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$cohabYN, vars.g$gender)
## X-squared = 13.675, df = 1, p-value = 0.0002174
chisq.test(table(vars.g$child.sum, vars.g$gender))

## Warning in chisq.test(table(vars.g$child.sum, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$child.sum, vars.g$gender)
## X-squared = 7.6711, df = 3, p-value = 0.05332

```

```

chisq.test(table(vars.g$groom, vars.g$gender))

## Warning in chisq.test(table(vars.g$groom, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$groom, vars.g$gender)
## X-squared = 4.5002, df = 7, p-value = 0.7207
chisq.test(table(vars.g$social.sum, vars.g$gender))

## Warning in chisq.test(table(vars.g$social.sum, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$social.sum, vars.g$gender)
## X-squared = 14, df = 22, p-value = 0.9015
chisq.test(table(vars.g$dog, vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$dog, vars.g$gender)
## X-squared = 0.56686, df = 1, p-value = 0.4515
chisq.test(table(vars.g$cat, vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$cat, vars.g$gender)
## X-squared = 0.98659, df = 1, p-value = 0.3206
chisq.test(table(vars.g$pet.other, vars.g$gender))

## Warning in chisq.test(table(vars.g$pet.other, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(vars.g$pet.other, vars.g$gender)
## X-squared = 0.0023825, df = 1, p-value = 0.9611
chisq.test(table(vars.g$clean.house, vars.g$gender))

## Warning in chisq.test(table(vars.g$clean.house, vars.g$gender)): Chi-
## squared approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$clean.house, vars.g$gender)
## X-squared = 3.0291, df = 6, p-value = 0.8052

```

```

chisq.test(table(vars.g$bmi, vars.g$gender))

## Warning in chisq.test(table(vars.g$bmi, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$bmi, vars.g$gender)
## X-squared = 168, df = 160, p-value = 0.3167
chisq.test(table(vars.g$srh11, vars.g$gender))

## Warning in chisq.test(table(vars.g$srh11, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data:  table(vars.g$srh11, vars.g$gender)
## X-squared = 2.5354, df = 4, p-value = 0.6383
chisq.test(table(as.factor(vars.g$hb11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(as.factor(vars.g$hb11), vars.g$gender)
## X-squared = 1.4052, df = 1, p-value = 0.2358
chisq.test(table(as.factor(vars.g$hbs11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(as.factor(vars.g$hbs11), vars.g$gender)
## X-squared = 0.31741, df = 1, p-value = 0.5732
chisq.test(table(as.factor(vars.g$heart11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(as.factor(vars.g$heart11), vars.g$gender)
## X-squared = 2.3549, df = 1, p-value = 0.1249
chisq.test(table(as.factor(vars.g$arh11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(as.factor(vars.g$arh11), vars.g$gender)
## X-squared = 0.77841, df = 1, p-value = 0.3776
chisq.test(table(as.factor(vars.g$cancer11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##

```

```
## data: table(as.factor(vars.g$cancer11), vars.g$gender)
## X-squared = 0.50722, df = 1, p-value = 0.4763
chisq.test(table(as.factor(vars.g$stroke11), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(as.factor(vars.g$stroke11), vars.g$gender)
## X-squared = 0.27776, df = 1, p-value = 0.5982
chisq.test(table(as.factor(vars.g$IBS), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(as.factor(vars.g$IBS), vars.g$gender)
## X-squared = 4.8242, df = 1, p-value = 0.02806
chisq.test(table(vars.g$walk.ave, vars.g$gender))

## Warning in chisq.test(table(vars.g$walk.ave, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data: table(vars.g$walk.ave, vars.g$gender)
## X-squared = 139.61, df = 135, p-value = 0.3753
chisq.test(table(as.factor(vars.g$smokeYN), vars.g$gender))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(as.factor(vars.g$smokeYN), vars.g$gender)
## X-squared = 0.0023952, df = 1, p-value = 0.961
chisq.test(table(vars.g$meat, vars.g$gender))

## Warning in chisq.test(table(vars.g$meat, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data: table(vars.g$meat, vars.g$gender)
## X-squared = 8.3147, df = 7, p-value = 0.3057
chisq.test(table(vars.g$poul, vars.g$gender))

## Warning in chisq.test(table(vars.g$poul, vars.g$gender)): Chi-squared
## approximation may be incorrect
##
## Pearson's Chi-squared test
##
## data: table(vars.g$poul, vars.g$gender)
## X-squared = 3.8028, df = 7, p-value = 0.8022
```



```
chisq.test(table(vars.g$pork, vars.g$gender))
```

```
## Warning in chisq.test(table(vars.g$pork, vars.g$gender)): Chi-squared  
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data: table(vars.g$pork, vars.g$gender)
```

```
## X-squared = 10.397, df = 7, p-value = 0.1672
```

```
chisq.test(table(vars.g$sea, vars.g$gender))
```

```
## Warning in chisq.test(table(vars.g$sea, vars.g$gender)): Chi-squared  
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data: table(vars.g$sea, vars.g$gender)
```

```
## X-squared = 3.631, df = 4, p-value = 0.4582
```

```
chisq.test(table(vars.g$prot.sum, vars.g$gender))
```

```
## Warning in chisq.test(table(vars.g$prot.sum, vars.g$gender)): Chi-squared  
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data: table(vars.g$prot.sum, vars.g$gender)
```

```
## X-squared = 18.766, df = 15, p-value = 0.2245
```

```
chisq.test(table(vars.g$fruit.sum, vars.g$gender))
```

```
## Warning in chisq.test(table(vars.g$fruit.sum, vars.g$gender)): Chi-squared  
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data: table(vars.g$fruit.sum, vars.g$gender)
```

```
## X-squared = 22.782, df = 20, p-value = 0.2996
```

```
chisq.test(table(vars.g$veg.sum, vars.g$gender))
```

```
## Warning in chisq.test(table(vars.g$veg.sum, vars.g$gender)): Chi-squared  
## approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

```
## data: table(vars.g$veg.sum, vars.g$gender)
```

```
## X-squared = 40.631, df = 39, p-value = 0.3984
```

And correct these P-values for multiple comparison testing.

```
chi.list= c(0.8722,0.291,0.1907,0.2011,0.8032,0.0002174,0.05332,0.7207,0.9015,0.4515,0.167,0.9611,0.805)
```

```
p.adjust(chi.list, method="BH")
```

```
## [1] 0.9611000000 0.6545133333 0.6545133333 0.6545133333 0.9244888889
## [6] 0.0033697000 0.4132300000 0.9244888889 0.9611000000 0.7382650000
## [11] 0.6545133333 0.9611000000 0.9244888889 0.6545133333 0.8603173913
## [16] 0.0000016957 0.8429181818 0.6545133333 0.6545133333 0.7013750000
## [21] 0.7382650000 0.8429181818 0.2899533333 0.9611000000 0.6545133333
## [26] 0.9244888889 0.6545133333 0.7382650000 0.6545133333 0.7264941176
## [31] 0.6545133333
```

For the variables that do differ by sex, we will stratify (`strata`) the model by sex. These include

- cohabYN
- walk.ave

Antibiotics

Does antibiotic use correlate to the gut microbiota?

- AB = antibiotic use in the last 6 months, YN

```
vars = data.frame(vars.g$AB)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(AB), data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(AB)  1      0.519 0.51945  2.0615 0.01184 0.004 **
## Residuals    172     43.340 0.25198      0.98816
## Total        173     43.860      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(AB), data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(AB)  1      0.533 0.53295  1.575 0.00907 0.004 **
## Residuals    172     58.200 0.33837      0.99093
```

```
## Total          173      58.733          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(AB), data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(AB)  1    0.1049 0.104882  2.5225 0.01445  0.003 **
## Residuals     172    7.1514 0.041578          0.98555
## Total         173    7.2563          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(AB), data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(AB)  1    0.478 0.47768  2.2207 0.01275  0.002 **
## Residuals     172   36.999 0.21511          0.98725
## Total         173   37.476          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Rural vs. urban

Does rural-urban upbringing or current residence correlate to the gut microbiota?

- RU57 = OCF357 = Father's occupation in 1957
- RU11 = hf017j1e = 1990 occupation code for graduates

```
vars = data.frame(vars.g$RU57, vars.g$RU11)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ RU57+RU11, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## RU57       1    0.2119 0.21189  0.8212 0.00696  0.734
## RU11       1    0.3152 0.31516  1.2215 0.01035  0.170
## Residuals 116   29.9307 0.25802          0.98270
## Total     118   30.4578          1.00000

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])])
~ RU57+RU11, data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## RU57       1    0.315 0.31482 0.91801 0.00778  0.674
## RU11       1    0.377 0.37725 1.10005 0.00932  0.233
## Residuals 116   39.781 0.34294          0.98290
## Total     118   40.473          1.00000

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])])
~ RU57+RU11, data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## RU57       1    0.0281 0.028060 0.65311 0.00554  0.854
## RU11       1    0.0516 0.051578 1.20049 0.01019  0.241
## Residuals 116   4.9838 0.042964          0.98427
## Total     118   5.0634          1.00000

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])])
~ RU57+RU11, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## RU57      1      0.1823 0.18225 0.82481 0.00697 0.828
## RU11      1      0.3381 0.33811 1.53017 0.01293 0.032 *
## Residuals 116     25.6316 0.22096      0.98010
## Total     118     26.1520      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Education

Does education or IQ correlate to the gut microbiota?

- yrs.edu = hb103red = years of education
- iq = IQ (Henmomm-Nelson score)

```
vars = data.frame(vars.g$yrs.edu, vars.g$iq)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ yrs.edu+iq, data = vars.g[complete.cases(vars),],
      permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## yrs.edu    1      0.285 0.28468  1.117 0.00627 0.280
## iq         1      0.283 0.28340  1.112 0.00624 0.253
## Residuals 176     44.855 0.25486      0.98749
## Total     178     45.423      1.00000
adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ yrs.edu+iq, data = vars.g[complete.cases(vars),],
      permutations=1000)
```

```
##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## yrs.edu      1      0.354 0.35372  1.0386 0.00583  0.352
## iq           1      0.354 0.35398  1.0393 0.00584  0.353
## Residuals 176    59.943 0.34059          0.98833
## Total      178    60.651          1.00000

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ yrs.edu+iq, data = vars.g[complete.cases(vars),],
      permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## yrs.edu      1      0.0395 0.039465 0.93469 0.00525  0.485
## iq           1      0.0459 0.045856 1.08605 0.00610  0.329
## Residuals 176      7.4311 0.042222          0.98865
## Total      178      7.5164          1.00000

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ yrs.edu+iq, data = vars.g[complete.cases(vars),],
      permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## yrs.edu      1      0.263 0.26332 1.20764 0.00678  0.122
## iq           1      0.191 0.19114 0.87661 0.00492  0.701
## Residuals 176    38.377 0.21805          0.98830
## Total      178    38.831          1.00000
```

Human interaction

Does human interaction correlate to the gut microbiota?

- cohabYN = cohab = Cohabiting with spouse, YN (strata by sex)
- child.sum = HD01701-06 = # of children cohabiting with graduate
- groom = ha103re = Personal grooming score (0-10)
- social.sum = jz023rer + jz024rer = # of social interactions with family and friends in the last 4 wks

```

#Full model
vars = data.frame(vars.g$cohabYN, vars.g$child.sum, vars.g$groom, vars.g$social.sum)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum,
      data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
      permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN        1    0.3703 0.37031 1.48420 0.01223 0.127
## child.sum       1    0.2256 0.22562 0.90427 0.00745 0.561
## groom          1    0.2490 0.24897 0.99787 0.00822 0.505
## social.sum      1    0.2219 0.22194 0.88952 0.00733 0.603
## cohabYN:social.sum 1    0.5316 0.53156 2.13049 0.01755 0.004 **
## Residuals     115    28.6926 0.24950      0.94723
## Total         120    30.2909      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum,
      data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
      permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN        1    0.424 0.42442 1.26222 0.01044 0.153
## child.sum       1    0.320 0.32003 0.95178 0.00788 0.522
## groom          1    0.339 0.33854 1.00681 0.00833 0.469
## social.sum      1    0.315 0.31474 0.93604 0.00775 0.606
## cohabYN:social.sum 1    0.572 0.57167 1.70014 0.01407 0.001 ***
## Residuals     115    38.668 0.33625      0.95154
## Total         120    40.638      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum,
      data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
      permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN        1    0.0674 0.067378 1.63138 0.01347 0.156
## child.sum       1    0.0349 0.034858 0.84401 0.00697 0.580
## groom           1    0.0371 0.037145 0.89937 0.00742 0.640
## social.sum      1    0.0354 0.035366 0.85630 0.00707 0.558
## cohabYN:social.sum 1    0.0787 0.078676 1.90494 0.01573 0.027 *
## Residuals     115    4.7496 0.041301          0.94935
## Total          120    5.0030          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum,
      data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
      permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN        1    0.2541 0.25415 1.19412 0.00979 0.239
## child.sum       1    0.2745 0.27450 1.28973 0.01057 0.086 .
## groom           1    0.2036 0.20363 0.95676 0.00784 0.557
## social.sum      1    0.2264 0.22640 1.06375 0.00872 0.277
## cohabYN:social.sum 1    0.5295 0.52952 2.48794 0.02039 0.001 ***
## Residuals     115   24.4760 0.21283          0.94268
## Total          120   25.9642          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#CohabYN alone
vars = data.frame(vars.g$cohabYN)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])

```



```

~ cohabYN,
data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])], row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),]), strata = vars.g[complete.cases(vars),]$gender, permutations=1000)
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN      1      0.549 0.54873  2.1644 0.01208 0.031 *
## Residuals 177      44.874 0.25353          0.98792
## Total      178      45.423          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])], row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),]), strata = vars.g[complete.cases(vars),]$gender, permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])], row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),]), strata = vars.g[complete.cases(vars),]$gender, permutations=1000)
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN      1      0.574 0.57433  1.6921 0.00947 0.024 *
## Residuals 177      60.077 0.33942          0.99053
## Total      178      60.651          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])], row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),]), strata = vars.g[complete.cases(vars),]$gender, permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])], row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),]), strata = vars.g[complete.cases(vars),]$gender, permutations=1000)
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999

```

```
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN      1    0.0990 0.098986  2.3621 0.01317  0.043 *
## Residuals 177    7.4174 0.041906      0.98683
## Total      178    7.5164      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars)],),row.names(vars.g[complete.cases(vars),])
      ~ cohabYN,
      data = vars.g[complete.cases(vars),], strata = vars.g[complete.cases(vars),]$gender,
      permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## cohabYN      1     0.386 0.38601  1.7772 0.00994  0.042 *
## Residuals 177   38.445 0.21720      0.99006
## Total      178   38.831      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Socialness alone
##Within cohabiting persons
vars = data.frame(vars.g$social.sum)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[com
      ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars) &      vars.g$cohabYN == "Y", ]), r
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum   1     0.3459 0.34587  1.4415 0.01435  0.05 *
## Residuals   99   23.7545 0.23994      0.98565
## Total      100   24.1004      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

adonis(J.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
##        data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum    1      0.419 0.41936  1.2769 0.01273 0.056 .
## Residuals   99     32.513 0.32841      0.98727
## Total      100     32.932      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
##        data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum    1      0.0487 0.048736  1.2118 0.01209 0.239
## Residuals   99     3.9816 0.040218      0.98791
## Total      100     4.0303      1.00000

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",]), ~ social.sum,
##        data = vars.g[complete.cases(vars) & vars.g$cohabYN == "Y",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum    1      0.2859 0.28593  1.3458 0.01341 0.064 .
## Residuals   99     21.0331 0.21246      0.98659
## Total      100     21.3190      1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##Within non-cohabiting persons
vars = data.frame(vars.g$social.sum)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]) ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum   1    0.4647 0.46470  1.7361 0.04062 0.016 *
## Residuals  41   10.9744 0.26767    0.95938
## Total       42   11.4391         1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
adonis(J.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]) ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum   1    0.5139 0.51385  1.4652 0.0345 0.011 *
## Residuals  41   14.3784 0.35069    0.9655
## Total       42   14.8923         1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]) ~ social.sum,
      data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]), data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)
##
## Permutation: free
## Number of permutations: 999
##

```

```
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum  1    0.07337 0.073372  1.7404 0.04072  0.046 *
## Residuals  41    1.72849 0.042158      0.95928
## Total      42    1.80186      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N",]), row.names(vars.g[c
~ social.sum,
data = vars.g[complete.cases(vars) & vars.g$cohabYN == "N",], permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars) & vars.g$cohabYN == "N", ]),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## social.sum  1    0.4314 0.43141  1.9549 0.04551  0.006 **
## Residuals  41    9.0478 0.22068      0.95449
## Total      42    9.4792      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Home environment

Does the home environment correlate to the gut microbiota?

- dog = dog in house
- cat = cat in house
- pet.other = Other pet (bird, reptile, fish) in house
- clean.house = hall4re = Residence cleanliness score (1-10)

```
vars = data.frame(vars.g$dog, vars.g$cat, vars.g$pet.other, vars.g$clean.house)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ dog+cat+pet.other+clean.house, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars), ]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## dog           1    0.178 0.17757 0.69904 0.00524  0.924
## cat           1    0.249 0.24906 0.98046 0.00735  0.442
```

```
## pet.other      1      0.208 0.20812 0.81930 0.00614 0.711
## clean.house    1      0.221 0.22134 0.87136 0.00653 0.665
## Residuals     130     33.023 0.25402          0.97473
## Total          134     33.879          1.00000
```

```
adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ dog+cat+pet.other+clean.house, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
```

```
## Call:
```

```
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])
```

```
##
```

```
## Permutation: free
```

```
## Number of permutations: 999
```

```
##
```

```
## Terms added sequentially (first to last)
```

```
##
```

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
## dog	1	0.276	0.27604	0.81305	0.00608	0.921
## cat	1	0.329	0.32871	0.96818	0.00725	0.532
## pet.other	1	0.311	0.31137	0.91711	0.00686	0.628
## clean.house	1	0.313	0.31302	0.92197	0.00690	0.657
## Residuals	130	44.137	0.33951		0.97291	
## Total	134	45.366			1.00000	

```
adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ dog+cat+pet.other+clean.house, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
```

```
## Call:
```

```
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])
```

```
##
```

```
## Permutation: free
```

```
## Number of permutations: 999
```

```
##
```

```
## Terms added sequentially (first to last)
```

```
##
```

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
## dog	1	0.0281	0.028096	0.66144	0.00497	0.855
## cat	1	0.0475	0.047492	1.11807	0.00840	0.298
## pet.other	1	0.0235	0.023481	0.55280	0.00415	0.902
## clean.house	1	0.0344	0.034381	0.80941	0.00608	0.647
## Residuals	130	5.5220	0.042477		0.97640	
## Total	134	5.6554			1.00000	

```
adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ dog+cat+pet.other+clean.house, data = vars.g[complete.cases(vars),],
permutations=1000)
```

```
##
```

```
## Call:
```

```
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.cases(vars),])
```

```
##
```

```
## Permutation: free
```

```
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## dog         1     0.1735 0.17350 0.79469 0.00594 0.916
## cat         1     0.1971 0.19712 0.90287 0.00675 0.647
## pet.other    1     0.2326 0.23263 1.06551 0.00797 0.297
## clean.house  1     0.2093 0.20932 0.95872 0.00717 0.488
## Residuals   130    28.3827 0.21833      0.97217
## Total       134    29.1952      1.00000
```

Health

Does health correlate to the gut microbiota?

- bmi = body mass index based on height and weight
- srh11 = Self-reported health score (1-5)
- walk.ave = (HX472RE+HX473RE)/2 = Average walking speed (sec) (strate by sex)
- hbs11 = High-blood sugar, YN
- hbp11 = High-blood pressure, YN
- heart11 = Heart disease, YN
- arth11 = Arthritis, YN
- cancer11 = Cancer, YN
- stroke11 = Stroke, YN
- IBS = irritable bowel syndrome, YN

```
#overall health
vars = data.frame(vars.g$bmi, vars.g$srh11, vars.g$walk.ave)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ bmi+srh11+walk.ave, data = vars.g[complete.cases(vars),],
strata = vars.g[complete.cases(vars),]$gender,
permuations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## bmi         1     0.365 0.36461 1.42693 0.00863 0.052 .
## srh11        1     0.276 0.27648 1.08202 0.00654 0.274
## walk.ave     1     0.226 0.22579 0.88365 0.00534 0.668
## Residuals   162    41.394 0.25552      0.97949
## Total       165    42.261      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ bmi+srh11+walk.ave, data = vars.g[complete.cases(vars),],
```

```

strata = vars.g[complete.cases(vars),]$gender,
permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## bmi         1      0.444 0.44406 1.30288 0.00788 0.039 *
## srh11        1      0.360 0.35992 1.05602 0.00639 0.290
## walk.ave     1      0.308 0.30780 0.90308 0.00546 0.740
## Residuals 162    55.214 0.34083          0.98026
## Total      165    56.326          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
~ bmi+srh11+walk.ave, data = vars.g[complete.cases(vars),],
strata = vars.g[complete.cases(vars),]$gender,
permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Blocks: strata
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## bmi         1      0.0472 0.047222 1.11408 0.00676 0.284
## srh11        1      0.0403 0.040271 0.95009 0.00576 0.481
## walk.ave     1      0.0317 0.031706 0.74803 0.00454 0.777
## Residuals 162      6.8666 0.042386          0.98294
## Total      165      6.9858          1.00000
adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
~ bmi+srh11+walk.ave, data = vars.g[complete.cases(vars),],
strata = vars.g[complete.cases(vars),]$gender,
permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Blocks: strata
## Permutation: free

```



```

## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## bmi          1      0.299 0.29876  1.3751 0.00831  0.055 .
## srh11         1      0.218 0.21768  1.0019 0.00605  0.371
## walk.ave      1      0.240 0.23992  1.1043 0.00667  0.267
## Residuals    162     35.197 0.21727      0.97896
## Total        165     35.954      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Specific diseases
vars = data.frame(vars.g$hbs11, vars.g$hbp11, vars.g$heart11, vars.g$arth11, vars.g$cancer11, vars.g$stroke11)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ as.factor(hbs11) + as.factor(hbp11) + as.factor(heart11) + as.factor(arth11) + as.factor(cancer11) + as.factor(stroke11),
      data = vars.g[complete.cases(vars),], permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
##        permutations = 999, data = vars.g[complete.cases(vars),])
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(hbs11)      1      0.506 0.50572  2.00886 0.01285  0.005 **
## as.factor(hbp11)      1      0.260 0.25952  1.03090 0.00659  0.376
## as.factor(heart11)    1      0.476 0.47644  1.89255 0.01210  0.006 **
## as.factor(arth11)     1      0.172 0.17169  0.68201 0.00436  0.909
## as.factor(cancer11)   1      0.216 0.21558  0.85636 0.00548  0.667
## as.factor(stroke11)   1      0.174 0.17412  0.69164 0.00442  0.919
## as.factor(IBS)        1      0.299 0.29913  1.18823 0.00760  0.215
## Residuals            148     37.258 0.25174      0.94659
## Total                 155     39.360      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ as.factor(hbs11) + as.factor(hbp11) + as.factor(heart11) + as.factor(arth11) + as.factor(cancer11) + as.factor(stroke11),
      data = vars.g[complete.cases(vars),], permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
##        permutations = 999, data = vars.g[complete.cases(vars),])
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)

```

```

## as.factor(hbs11)      1      0.549 0.54890 1.62396 0.01043 0.004 **
## as.factor(hbp11)      1      0.341 0.34113 1.00928 0.00648 0.399
## as.factor(heart11)     1      0.495 0.49529 1.46538 0.00941 0.024 *
## as.factor(arth11)      1      0.276 0.27642 0.81782 0.00525 0.900
## as.factor(cancer11)    1      0.297 0.29733 0.87968 0.00565 0.766
## as.factor(stroke11)    1      0.263 0.26348 0.77953 0.00501 0.955
## as.factor(IBS)         1      0.382 0.38206 1.13037 0.00726 0.175
## Residuals             148     50.024 0.33800          0.95051
## Total                  155     52.628          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
~ as.factor(hbs11) + as.factor(hbp11) + as.factor(heart11) + as.factor(arth11) + as.factor(cancer11),
data = vars.g[complete.cases(vars),], permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(hbs11)      1      0.0774 0.077362 1.87129 0.01201 0.025 *
## as.factor(hbp11)      1      0.0356 0.035631 0.86186 0.00553 0.594
## as.factor(heart11)     1      0.0714 0.071404 1.72718 0.01109 0.060 .
## as.factor(arth11)      1      0.0223 0.022300 0.53942 0.00346 0.949
## as.factor(cancer11)    1      0.0388 0.038782 0.93808 0.00602 0.491
## as.factor(stroke11)    1      0.0295 0.029487 0.71326 0.00458 0.776
## as.factor(IBS)         1      0.0470 0.047004 1.13697 0.00730 0.287
## Residuals             148      6.1185 0.041341          0.95001
## Total                  155      6.4405          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
~ as.factor(hbs11) + as.factor(hbp11) + as.factor(heart11) + as.factor(arth11) + as.factor(cancer11),
data = vars.g[complete.cases(vars),], permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(hbs11)      1      0.412 0.41164 1.89555 0.01213 0.002 **
## as.factor(hbp11)      1      0.230 0.23001 1.05914 0.00678 0.294
## as.factor(heart11)     1      0.361 0.36148 1.66454 0.01065 0.009 **

```

```
## as.factor(arth11)      1      0.207 0.20693 0.95288 0.00610 0.521
## as.factor(cancer11)   1      0.178 0.17792 0.81927 0.00524 0.834
## as.factor(stroke11)   1      0.179 0.17899 0.82421 0.00528 0.830
## as.factor(IBS)        1      0.223 0.22299 1.02685 0.00657 0.358
## Residuals             148     32.140 0.21716          0.94725
## Total                 155     33.930          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Smoking

Does smoking impact the microbiota?

- smokeYN = jx013rec = currently a smoker, YN

```
vars = data.frame(vars.g$smokeYN)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(smokeYN), data = vars.g[complete.cases(vars),],
permuations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(smokeYN)  1      0.270 0.26958  1.0582 0.00691  0.34
## Residuals          152     38.724 0.25476          0.99309
## Total              153     38.994          1.00000
```

```
adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(smokeYN), data = vars.g[complete.cases(vars),],
permuations=1000)
```

```
##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## as.factor(smokeYN)  1      0.362 0.36222  1.0642 0.00695  0.259
## Residuals          152     51.736 0.34037          0.99305
## Total              153     52.099          1.00000
```

```
adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ as.factor(smokeYN), data = vars.g[complete.cases(vars),],
permuations=1000)
```

```
##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## as.factor(smokeyN)   1    0.0370 0.036964  0.8926 0.00584  0.574
## Residuals          152    6.2945 0.041411      0.99416
## Total              153    6.3314      1.00000
adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ as.factor(smokeyN), data = vars.g[complete.cases(vars),],
      permutations=1000)
```

```
##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## as.factor(smokeyN)   1     0.215 0.21537 0.99058 0.00647  0.427
## Residuals          152   33.048 0.21742      0.99353
## Total              153   33.263      1.00000
```

Diet

Does diet impact the microbiota?

- prot.sum = Times consume protein / wk
- fruit.sum = # different fruits consume / week
- veg.sum = # different vegetables consume / week
- meat = Times consume (red) meat / week
- poul = Times consume poultry / week
- pork = Times consume pork / week
- sea = Times consume seafood / week

```
#Protein, veg and fruits
vars = data.frame(vars.g$prot.sum, vars.g$fruit.sum, vars.g$veg.sum)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])],
      ~ prot.sum + fruit.sum + veg.sum,
      data = vars.g[complete.cases(vars),],
      permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),
##
##
```

```

## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## prot.sum    1      0.482 0.48164 1.91298 0.01150 0.007 **
## fruit.sum   1      0.215 0.21509 0.85429 0.00514 0.681
## veg.sum     1      0.393 0.39348 1.56285 0.00940 0.045 *
## Residuals 162    40.787 0.25177      0.97397
## Total      165    41.877      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ prot.sum + fruit.sum + veg.sum,
      data = vars.g[complete.cases(vars),],
      permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## prot.sum    1      0.522 0.52242 1.54591 0.00932 0.008 **
## fruit.sum   1      0.308 0.30806 0.91161 0.00550 0.695
## veg.sum     1      0.456 0.45584 1.34889 0.00814 0.034 *
## Residuals 162    54.746 0.33794      0.97704
## Total      165    56.032      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
      ~ prot.sum + fruit.sum + veg.sum,
      data = vars.g[complete.cases(vars),],
      permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]), row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## prot.sum    1      0.1028 0.102773 2.4908 0.01491 0.004 **
## fruit.sum   1      0.0517 0.051650 1.2518 0.00750 0.201
## veg.sum     1      0.0525 0.052510 1.2726 0.00762 0.208

```

```
## Residuals 162    6.6842 0.041261          0.96997
## Total      165    6.8911          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])])
~ prot.sum + fruit.sum + veg.sum,
data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = uwUF.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## prot.sum   1      0.290 0.28977 1.33965 0.00808 0.072 .
## fruit.sum  1      0.197 0.19718 0.91161 0.00550 0.621
## veg.sum    1      0.313 0.31307 1.44737 0.00873 0.054 .
## Residuals 162    35.041 0.21630          0.97768
## Total      165    35.841          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Specific proteins
vars = data.frame(vars.g$meat, vars.g$poul, vars.g$pork, vars.g$sea)
adonis(BC.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])])
~ meat+poul+pork+sea+meat:poul,
data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = BC.dist.g[row.names(vars.g[complete.cases(vars),      ]), row.names(vars.g[complete
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## meat       1      0.251 0.25125 1.0030 0.00600 0.425
## poul       1      0.334 0.33406 1.3336 0.00798 0.093 .
## pork       1      0.370 0.36992 1.4767 0.00883 0.050 *
## sea        1      0.341 0.34135 1.3626 0.00815 0.089 .
## meat:poul  1      0.501 0.50053 1.9981 0.01195 0.007 **
## Residuals 160    40.080 0.25050          0.95709
## Total     165     41.877          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

adonis(J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ meat+poul+pork+sea+meat:poul,
data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = J.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## meat       1      0.342 0.34181  1.0140 0.00610  0.403
## poul       1      0.386 0.38579  1.1444 0.00689  0.178
## pork       1      0.429 0.42875  1.2719 0.00765  0.073 .
## sea        1      0.416 0.41641  1.2353 0.00743  0.073 .
## meat:poul   1      0.524 0.52377  1.5538 0.00935  0.005 **
## Residuals 160    53.935 0.33710      0.96258
## Total     165    56.032      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ meat+poul+pork+sea+meat:poul,
data = vars.g[complete.cases(vars),],
permutations=1000)

##
## Call:
## adonis(formula = wUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## meat       1      0.0723 0.072346  1.7716 0.01050  0.043 *
## poul       1      0.0623 0.062263  1.5247 0.00904  0.090 .
## pork       1      0.0785 0.078518  1.9228 0.01139  0.026 *
## sea        1      0.0427 0.042669  1.0449 0.00619  0.363
## meat:poul   1      0.1016 0.101576  2.4874 0.01474  0.004 **
## Residuals 160     6.5338 0.040836      0.94814
## Total     165     6.8911      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis(uwUF.dist.g[row.names(vars.g[complete.cases(vars),]),row.names(vars.g[complete.cases(vars),])
~ meat+poul+pork+sea+meat:poul,
data = vars.g[complete.cases(vars),],
permutations=1000)

```



```
## [29] 0.6040000 0.7757241 0.0570000 0.4813333 0.4180000 0.1786000 0.2850000
## [36] 0.1520000 0.6677143 0.0570000
```

```
p.adjust(uwUF.list, method="BH")
```

```
## [1] 0.01266667 0.91211503 0.01266667 0.91211503 0.13933333 0.33305882
## [7] 0.87637500 0.41420000 0.27360000 0.68526667 0.45104348 0.01266667
## [13] 0.17480000 0.19584615 0.03166667 0.93100000 0.91088235 0.41420000
## [19] 0.66172414 0.18309091 0.57152000 0.45104348 0.02850000 0.45104348
## [25] 0.04885714 0.66172414 0.91211503 0.91088235 0.54150000 0.58829630
## [31] 0.19584615 0.81393548 0.13933333 0.58829630 0.33305882 0.24700000
## [37] 0.40111111 0.03040000
```

Alpha-diversity

Tests performed on the graduate samples to avoid possible interactions between spouses and siblings. First, check if the graduate subset is different from the other sample types to ensure the subset is a good representation of the overall data set.

```
summary(aov(shannon ~ Link, data=Met))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Link          3    0.84   0.2804    1.491  0.216
## Residuals    404   75.96   0.1880
```

```
summary(aov((1/simpson) ~ Link, data=Met))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Link          3    486   161.99    2.114 0.0979 .
## Residuals    404   30964    76.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(chao ~ Link, data=Met))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Link          3  130351   43450    1.787  0.149
## Residuals    404  9825590   24321
```

```
summary(aov(ace ~ Link, data=Met))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Link          3   207106   69035    1.639   0.18
## Residuals    404 17014109   42114
```

There are no differences so we move forward with the graduate subset data set. We will run a number of ANOVA (aov) tests of variables of interest similar to those run on beta-diversity. All tests are run on the Shannon's diversity index and Chao richness estimate. For more information on variables, see the corresponding beta-diversity test.

Sex and age

```
summary(aov(Met.g$shannon ~ gender+age, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## gender        1    0.41   0.4073    2.055  0.153
```

```
## age          1    0.01  0.0130   0.066  0.798
## Residuals   176   34.87  0.1981
```

```
summary(aov(Met.g$chao ~ gender+age, data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## gender        1  115499  115499    5.620 0.0188 *
## age           1     390     390    0.019 0.8906
## Residuals     176 3616849   20550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Antibiotics

```
summary(aov(Met.g$shannon ~ as.factor(AB), data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(AB)  1  1.444  1.4439    7.906 0.0055 **
## Residuals     172 31.412   0.1826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ as.factor(AB), data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## as.factor(AB)  1 112743  112743    5.601 0.0191 *
## Residuals     172 3462316   20130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

Rural vs. urban

```
summary(aov(Met.g$shannon ~ RU57*RU11, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## RU57          1  0.354  0.3544    1.607 0.2075
## RU11          1  0.498  0.4983    2.260 0.1355
## RU57:RU11     1  0.637  0.6370    2.888 0.0919 .
## Residuals     115 25.363   0.2205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 60 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ RU57*RU11, data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## RU57          1   49357   49357    2.424 0.122275
## RU11          1  241472  241472   11.857 0.000802 ***
## RU57:RU11     1   89542   89542    4.397 0.038202 *
## Residuals     115 2342104   20366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 60 observations deleted due to missingness
```

Education

```
summary(aov(Met.g$shannon ~ yrs.edu+iq, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## yrs.edu      1    0.30  0.30411   1.530  0.218
## iq           1    0.01  0.01281   0.064  0.800
## Residuals   176   34.98  0.19873
```

```
summary(aov(Met.g$chao ~ yrs.edu+iq, data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## yrs.edu      1 124867 124867   6.125 0.0143 *
## iq           1  19584  19584   0.961 0.3284
## Residuals   176 3588287  20388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Human interaction

```
summary(aov(Met.g$shannon ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## cohabYN      1  0.322  0.3219   1.751 0.18839
## child.sum     1  0.032  0.0320   0.174 0.67750
## groom         1  0.102  0.1018   0.554 0.45840
## social.sum    1  0.077  0.0767   0.417 0.51954
## cohabYN:social.sum 1  1.341  1.3408   7.294 0.00797 **
## Residuals    115 21.140  0.1838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 58 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ cohabYN + child.sum + groom + social.sum + cohabYN:social.sum, data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## cohabYN      1  35424  35424   2.291 0.132911
## child.sum     1  25037  25037   1.619 0.205808
## groom         1   884   884   0.057 0.811443
## social.sum    1  2555  2555   0.165 0.685167
## cohabYN:social.sum 1 181452 181452 11.733 0.000852 ***
## Residuals    115 1778518  15465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 58 observations deleted due to missingness
```

```
summary(aov(Met.g$shannon ~ cohabYN, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## cohabYN      1   1.53  1.5300   8.021 0.00516 **
## Residuals   177  33.76  0.1907
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(aov(Met.g$chao ~ cohabYN, data = vars.g))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## cohabYN      1  160740  160740    7.965 0.00531 **
## Residuals   177 3571997   20181
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(aov(Met.g[vars.g$cohabYN == "Y",]$shannon ~ social.sum, data = vars.g[vars.g$cohabYN == "Y",]))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## social.sum    1   0.285   0.2853    1.511  0.222
## Residuals    99 18.695   0.1888
## 26 observations deleted due to missingness

summary(aov(Met.g[vars.g$cohabYN == "Y",]$chao ~ social.sum, data = vars.g[vars.g$cohabYN == "Y",]))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## social.sum    1   26420   26420    1.137  0.289
## Residuals    99 2301108   23244
## 26 observations deleted due to missingness

summary(aov(Met.g[vars.g$cohabYN == "N",]$shannon ~ social.sum, data = vars.g[vars.g$cohabYN == "N",]))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## social.sum    1   0.623   0.6227    2.84 0.0995 .
## Residuals    41   8.989   0.2193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness

summary(aov(Met.g[vars.g$cohabYN == "N",]$chao ~ social.sum, data = vars.g[vars.g$cohabYN == "N",]))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## social.sum    1   64103   64103    3.972 0.0529 .
## Residuals    41 661638   16138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

Home environment

```
summary(aov(Met.g$shannon ~ dog + cat + pet.other + clean.house, data = vars.g))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## dog           1   0.031   0.03098    0.138  0.710
## cat           1   0.000   0.00003    0.000  0.990
## pet.other     1   0.060   0.05983    0.267  0.606
## clean.house   1   0.191   0.19063    0.852  0.358
## Residuals   130 29.086   0.22374
## 44 observations deleted due to missingness

summary(aov(Met.g$chao ~ dog + cat + pet.other + clean.house, data = vars.g))

##              Df Sum Sq Mean Sq F value    Pr(>F)
```

```
## dog          1    31337    31337    1.353    0.247
## cat          1    13239    13239    0.572    0.451
## pet.other    1     5914     5914    0.255    0.614
## clean.house  1     1017     1017    0.044    0.834
## Residuals   130 3010549    23158
## 44 observations deleted due to missingness
```

Health

```
summary(aov(Met.g$shannon ~ bmi+srh11+walk.ave, data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## bmi          1   0.558   0.5581    2.958 0.0874 .
## srh11        1   0.645   0.6449    3.418 0.0663 .
## walk.ave     1   0.000   0.0004    0.002 0.9652
## Residuals   162 30.568   0.1887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 13 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ bmi+srh11+walk.ave, data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## bmi          1   94485   94485    4.445 0.0365 *
## srh11        1   28826   28826    1.356 0.2459
## walk.ave     1    4231    4231    0.199 0.6561
## Residuals   162 3443638   21257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 13 observations deleted due to missingness
```

```
summary(aov(Met.g$shannon ~ as.factor(hbs11)+ as.factor(hbp11)+ as.factor(heart11)+ as.factor(arth11)+ as.factor(cancer11)+ as.factor(stroke11)+ as.factor(ibs11), data = vars.g))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(hbs11)  1   2.195   2.1954  10.916 0.0012 **
## as.factor(hbp11)  1   0.360   0.3598   1.789 0.1831
## as.factor(heart11) 1   0.021   0.0211   0.105 0.7463
## as.factor(arth11)  1   0.027   0.0273   0.136 0.7133
## as.factor(cancer11) 1   0.100   0.0998   0.496 0.4823
## as.factor(stroke11) 1   0.065   0.0645   0.321 0.5720
## as.factor(ibs11)   1   0.041   0.0414   0.206 0.6505
## Residuals        148 29.765   0.2011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 23 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ as.factor(hbs11)+ as.factor(hbp11)+ as.factor(heart11)+ as.factor(arth11)+ as.factor(cancer11)+ as.factor(stroke11)+ as.factor(ibs11), data = vars.g))
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## as.factor(hbs11)  1   92522   92522   4.198 0.0422 *
## as.factor(hbp11)  1   17153   17153   0.778 0.3791
## as.factor(heart11) 1     214     214   0.010 0.9216
## as.factor(arth11)  1     35     35   0.002 0.9683
## as.factor(cancer11) 1   41512   41512   1.884 0.1720
## as.factor(stroke11) 1   15939   15939   0.723 0.3964
```

```
## as.factor(IFS)          1      27      27  0.001 0.9721
## Residuals              148 3261661  22038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 23 observations deleted due to missingness
```

Smoking

```
summary(aov(Met.g$shannon ~ as.factor(smokeYN), data = vars.g))
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(smokeYN)  1  0.035  0.0347   0.177  0.675
## Residuals          152 29.874  0.1965
## 25 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ as.factor(smokeYN), data = vars.g))
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## as.factor(smokeYN)  1  42360  42360   1.969  0.163
## Residuals          152 3269920  21513
## 25 observations deleted due to missingness
```

Diet

```
summary(aov(Met.g$shannon ~ prot.sum + fruit.sum + veg.sum, data = vars.g))
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## prot.sum        1  0.066  0.0657   0.351  0.5546
## fruit.sum        1  0.169  0.1687   0.900  0.3442
## veg.sum          1  0.643  0.6431   3.431  0.0658 .
## Residuals       162 30.364  0.1874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 13 observations deleted due to missingness
```

```
summary(aov(Met.g$chao ~ prot.sum + fruit.sum + veg.sum, data = vars.g))
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## prot.sum        1   3412   3412   0.161  0.688
## fruit.sum        1  44307  44307   2.095  0.150
## veg.sum          1  28453  28453   1.345  0.248
## Residuals       162 3426524  21151
## 13 observations deleted due to missingness
```

```
summary(aov(Met.g$shannon ~ meat+poul+pork+sea +meat:poul +poul:pork, data = vars.g))
```

```
##                Df Sum Sq Mean Sq F value    Pr(>F)
## meat            1  0.196   0.196   1.204    0.2743
## poul            1  0.056   0.056   0.342    0.5594
## pork            1  0.090   0.090   0.552    0.4588
## sea             1  0.003   0.003   0.019    0.8914
## meat:poul       1  4.009   4.009  24.627 1.77e-06 ***
## poul:pork       1  1.004   1.004   6.165   0.0141 *
## Residuals      159 25.884   0.163
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 13 observations deleted due to missingness

summary(aov(Met.g$chao ~ meat+poul+pork+sea +meat:poul, data = vars.g))

##              Df  Sum Sq Mean Sq F value Pr(>F)
## meat           1      624      624   0.030 0.8636
## poul           1      487      487   0.023 0.8794
## pork           1     1427     1427   0.068 0.7950
## sea            1    29595    29595   1.404 0.2378
## meat:poul       1    98236    98236   4.661 0.0323 *
## Residuals     160 3372326    21077
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 13 observations deleted due to missingness
```

FDR-correction

```
shannon.list = c(0.153,0.798,0.0055,0.2075,0.1355,0.0919,0.218,0.8,0.18839,0.6775,0.4584,0.51954,0.0079)
chao.list = c(0.0188,0.8906,0.0191,0.122275,0.000802,0.038202,0.0143,0.3284,0.132911,0.205808,0.811443,0.0079)

p.adjust(shannon.list, method="BH")

## [1] 0.4707692 0.8421053 0.0550000 0.4933333 0.4516667 0.3618182 0.4933333
## [8] 0.8421053 0.4933333 0.8152000 0.7889655 0.7889655 0.0637600 0.0550000
## [15] 0.4933333 0.3618182 0.8152000 0.8492308 0.8152000 0.6761905 0.3618182
## [22] 0.3290000 0.7889655 0.0240000 0.4933333 0.8292222 0.8152000 0.7889655
## [29] 0.7889655 0.8152000 0.8152000 0.7889655 0.6761905 0.3290000 0.5774737
## [36] 0.7889655 0.7889655 0.8914000 0.0000708 0.0940000

p.adjust(chao.list, method="BH")

## [1] 0.1241500 0.9648167 0.1241500 0.3777429 0.0166140 0.1645800 0.1241500
## [8] 0.5336500 0.3777429 0.4396364 0.9648167 0.8944000 0.0166140 0.0690300
## [15] 0.4900435 0.1875545 0.4396364 0.6066667 0.7451786 0.9648167 0.1645800
## [22] 0.4396364 0.3777429 0.1645800 0.5913960 0.9714162 0.9721000 0.3945882
## [29] 0.5946000 0.9721000 0.3945882 0.8944000 0.3900000 0.4396364 0.9648167
## [36] 0.9648167 0.9648167 0.4396364 0.1645800          NA
```

Spouses and siblings

Beta-diversity

Are spouses or siblings more similar? Compare beta-diversity (Bray-Curtis, Jaccard, weighted and unweighted UniFrac) of spouses, siblings, and unrelated pairs. The unrelated group is defined as pairs that were not siblings, spouses, or in-laws within the dataset. The unrelated group was subsampled to 100 pairs with sex ratios equivalent to the sibling dataset (57% same sex pairs) and P-values were averaged across 1000 iterative subsamples

Since sample sizes and variance are unequal, compare groups by Kruskal-Wallis. Ties exist (values occur more than once in a group) so use Chi-squared method to correct within pairwise comparisons.

```

KW.subsample.fxn = function(rep.number)
{library(PMCMR)
  repeat{
    BC.MF.sample = match.BC.MF[sample(1:length(match.BC.MF), 43, replace=FALSE)]
    BC.MMFF.sample = match.BC.MMFF[sample(1:length(match.BC.MMFF), 57, replace=FALSE)]
    J.MF.sample = match.J.MF[sample(1:length(match.J.MF), 43, replace=FALSE)]
    J.MMFF.sample = match.J.MMFF[sample(1:length(match.J.MMFF), 57, replace=FALSE)]
    wUF.MF.sample = match.wUF.MF[sample(1:length(match.wUF.MF), 43, replace=FALSE)]
    wUF.MMFF.sample = match.wUF.MMFF[sample(1:length(match.wUF.MMFF), 57, replace=FALSE)]
    uwUF.MF.sample = match.uwUF.MF[sample(1:length(match.uwUF.MF), 43, replace=FALSE)]
    uwUF.MMFF.sample = match.uwUF.MMFF[sample(1:length(match.uwUF.MMFF), 57, replace=FALSE)]

    dist.sample = data.frame(BC=c(match.BC.gp.se, match.BC.gs, BC.MF.sample, BC.MMFF.sample),
                              J=c(match.J.gp.se, match.J.gs, J.MF.sample, J.MMFF.sample),
                              wUF=c(match.wUF.gp.se, match.wUF.gs, wUF.MF.sample, wUF.MMFF.sample),
                              uwUF=c(match.uwUF.gp.se, match.uwUF.gs, uwUF.MF.sample, uwUF.MMFF.sample),
                              comp=c(rep(times=length(match.BC.gp.se), "1.gpse"),
                                     rep(times=length(match.BC.gs), "2.gs"),
                                     rep(times=100, "3.other")))

    #Bray-Curtis
    BC.kw = kruskal.test(dist.sample$BC ~ dist.sample$comp)
    posthoc.BC.kw.BH = p.adjust(posthoc.kruskal.nemenyi.test(x=dist.sample$BC, g=dist.sample$comp, dist="Chisq"))
    #Jaccard
    J.kw = kruskal.test(dist.sample$J ~ dist.sample$comp)
    posthoc.J.kw.BH = p.adjust(posthoc.kruskal.nemenyi.test(x=dist.sample$J, g=dist.sample$comp, dist="Chisq"))
    #wUF
    wUF.kw = kruskal.test(dist.sample$wUF ~ dist.sample$comp)
    posthoc.wUF.kw.BH = p.adjust(posthoc.kruskal.nemenyi.test(x=dist.sample$wUF, g=dist.sample$comp, dist="Chisq"))
    #uwUF
    uwUF.kw = kruskal.test(dist.sample$uwUF ~ dist.sample$comp)
    posthoc.uwUF.kw.BH = p.adjust(posthoc.kruskal.nemenyi.test(x=dist.sample$uwUF, g=dist.sample$comp, dist="Chisq"))

    #Print to table
    kw.results = data.frame(
      BC = c(BC.MF.sample, BC.MMFF.sample, BC.kw$p.value, posthoc.BC.kw.BH),
      J = c(J.MF.sample, J.MMFF.sample, J.kw$p.value, posthoc.J.kw.BH),
      wUF = c(wUF.MF.sample, wUF.MMFF.sample, wUF.kw$p.value, posthoc.wUF.kw.BH),
      uwUF = c(uwUF.MF.sample, uwUF.MMFF.sample, uwUF.kw$p.value, posthoc.uwUF.kw.BH),
      label = c(rep("dist",100), "KW.P", "Sp.Sib", "Sp.other", "NA", "Sib.other"))

    write.table(kw.results, file="KW.results.csv", append=TRUE, sep=",", col.names=FALSE)
    KW.results.final = read.table("KW.results.csv", sep=",")

    if(nrow(KW.results.final) >= 105*rep.number){break}
  }}

KW.subsample.fxn(1000)

#Curating results
KW.results.final = read.table("KW.results.csv", sep=",")
colnames(KW.results.final) = c("rep", "BC", "J", "wUF", "uwUF", "label")

#Average P-values

```



```
KW.P.ave = data.frame(
  KW.P = colMeans(KW.results.final[KW.results.final$label == "KW.P" & complete.cases(KW.results.final),
  Sp.Sib = colMeans(KW.results.final[KW.results.final$label == "Sp.Sib" & complete.cases(KW.results.fin
  Sp.other = colMeans(KW.results.final[KW.results.final$label == "Sp.other" & complete.cases(KW.results
  Sib.other = colMeans(KW.results.final[KW.results.final$label == "Sib.other"& complete.cases(KW.results
)

write.table(KW.P.ave, file="KW.P.ave.csv", append=TRUE, sep=",", col.names=TRUE)
```

```
## Warning in write.table(KW.P.ave, file = "KW.P.ave.csv", append = TRUE, sep
## = ",", : appending column names to file
```

Use linear models to assess if beta-diversity among spouse and/or sibling pairs is explained by variables of interest. Variables found to impact the overall microbiota among graduates (PERMANOVA $P < 0.1$, Dataset S2), those with considerably more variation in the full dataset (age), and those specific to sibling/spouse relationships (kinship, years cohabitating, closeness) were tested.

Variables specified as `diff` are the difference between values for individuals within a pair. For example, a spousal pair of person 1 = 70 yrs old and person 2 = 75 yrs old would have an `age.diff` of 5 yrs. In this way, we are testing if pairs being discordant in one or more variables causes their microbiota to be more or less similar.

Variables in spouses and siblings

Closeness

Self-reported relationship closeness.

- 1 = not at all
- 2 = not very
- 3 = somewhat
- 4 = very

If individuals within a pair did not give the same answer, the lower of the two was used.

```
summary(aov(match.BC.gp.se.gs ~ as.factor(floor(vars.gp.se.gs$closeness))))
```

```
##                               Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gp.se.gs$closeness))    3 0.0578 0.01927   1.312
## Residuals                                173 2.5412 0.01469
##                               Pr(>F)
## as.factor(floor(vars.gp.se.gs$closeness))    0.272
## Residuals
```

```
summary(aov(match.J.gp.se.gs ~ as.factor(floor(vars.gp.se.gs$closeness))))
```

```
##                               Df Sum Sq  Mean Sq F value
## as.factor(floor(vars.gp.se.gs$closeness))    3 0.0352 0.011746   1.488
## Residuals                                173 1.3652 0.007891
##                               Pr(>F)
## as.factor(floor(vars.gp.se.gs$closeness))    0.219
## Residuals
```

```
summary(aov(match.wUF.gp.se.gs ~ as.factor(floor(vars.gp.se.gs$closeness))))
```

```
##                               Df Sum Sq  Mean Sq F value
## as.factor(floor(vars.gp.se.gs$closeness))    3 0.0052 0.001729   0.481
## Residuals                                173 0.6213 0.003591
```

```

##                                Pr(>F)
## as.factor(floor(vars.gp.se.gs$closeness)) 0.696
## Residuals
summary(aov(match.uwUF.gp.se.gs ~ as.factor(floor(vars.gp.se.gs$closeness))))

##                                Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gp.se.gs$closeness)) 3 0.0834 0.027787 4.605
## Residuals                                173 1.0438 0.006034
##                                Pr(>F)
## as.factor(floor(vars.gp.se.gs$closeness)) 0.00397 **
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Pairwise between closeness levels
pairwise.t.test(match.BC.gp.se.gs, as.factor(floor(vars.gp.se.gs$closeness)), p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.BC.gp.se.gs and as.factor(floor(vars.gp.se.gs$closeness))
##
##    1    2    3
## 2 0.69 -    -
## 3 0.73 0.69 -
## 4 0.69 0.84 0.47
##
## P value adjustment method: BH
pairwise.t.test(match.J.gp.se.gs, as.factor(floor(vars.gp.se.gs$closeness)), p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.J.gp.se.gs and as.factor(floor(vars.gp.se.gs$closeness))
##
##    1    2    3
## 2 0.53 -    -
## 3 0.74 0.53 -
## 4 0.53 0.95 0.37
##
## P value adjustment method: BH
pairwise.t.test(match.wUF.gp.se.gs, as.factor(floor(vars.gp.se.gs$closeness)), p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.wUF.gp.se.gs and as.factor(floor(vars.gp.se.gs$closeness))
##
##    1    2    3
## 2 0.63 -    -
## 3 0.66 0.63 -
## 4 0.63 0.63 0.63
##
## P value adjustment method: BH

```

```

pairwise.t.test(match.uwUF.gp.se.gs, as.factor(floor(vars.gp.se.gs$closeness)), p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.uwUF.gp.se.gs and as.factor(floor(vars.gp.se.gs$closeness))
##
##      1      2      3
## 2 0.283 -      -
## 3 0.243 0.886 -
## 4 0.059 0.161 0.021
##
## P value adjustment method: BH
##Pairwise between closeness levels and spouses vs. siblings
vars.gp.se.gs$close.group = paste(as.factor(floor(vars.gp.se.gs$closeness)), vars.gp.se.gs$group, sep="")
pairwise.t.test(match.BC.gp.se.gs, vars.gp.se.gs$close.group, p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.BC.gp.se.gs and vars.gp.se.gs$close.group
##
##           1.sib 2.sib 3.sib 3.spouse 4.sib
## 2.sib      0.78 -      -      -      -
## 3.sib      0.78 0.78 -      -      -
## 3.spouse 0.78 0.78 0.78 -      -
## 4.sib      0.78 0.78 0.95 0.78 -
## 4.spouse 0.78 0.78 0.44 0.78 0.78
##
## P value adjustment method: BH
pairwise.t.test(match.J.gp.se.gs, vars.gp.se.gs$close.group, p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.J.gp.se.gs and vars.gp.se.gs$close.group
##
##           1.sib 2.sib 3.sib 3.spouse 4.sib
## 2.sib      0.83 -      -      -      -
## 3.sib      0.83 0.83 -      -      -
## 3.spouse 0.83 0.83 0.83 -      -
## 4.sib      0.83 0.83 0.83 0.83 -
## 4.spouse 0.83 0.83 0.38 0.83 0.83
##
## P value adjustment method: BH
pairwise.t.test(match.wUF.gp.se.gs, vars.gp.se.gs$close.group, p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.wUF.gp.se.gs and vars.gp.se.gs$close.group
##

```

```
##          1.sib 2.sib 3.sib 3.spouse 4.sib
## 2.sib    0.74  -      -      -      -
## 3.sib    0.82  0.74  -      -      -
## 3.spouse 0.82  0.82  0.82  -      -
## 4.sib    0.94  0.74  0.74  0.74  -
## 4.spouse 0.74  0.82  0.74  0.82  0.74
##
## P value adjustment method: BH
pairwise.t.test(match.uwUF.gp.se.gs, vars.gp.se.gs$close.group, p.adj = "BH")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: match.uwUF.gp.se.gs and vars.gp.se.gs$close.group
##
##          1.sib 2.sib 3.sib 3.spouse 4.sib
## 2.sib    0.370 -      -      -      -
## 3.sib    0.370 0.877 -      -      -
## 3.spouse 0.252 0.623 0.370 -      -
## 4.sib    0.328 0.815 0.698 0.713  -
## 4.spouse 0.080 0.140 0.005 0.328  0.140
##
## P value adjustment method: BH
```

Health

#Heart disease

```
summary(aov(match.BC.gp.se.gs ~ as.factor(vars.gp.se.gs$heart.diff.cat)))
```

```
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$heart.diff.cat)  1 0.0241 0.02411    1.7  0.194
## Residuals                               155 2.1975 0.01418
## 20 observations deleted due to missingness
```

```
summary(aov(match.J.gp.se.gs ~ as.factor(vars.gp.se.gs$heart.diff.cat)))
```

```
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$heart.diff.cat)  1 0.0152 0.01518    2.006  0.159
## Residuals                               155 1.1734 0.00757
## 20 observations deleted due to missingness
```

```
summary(aov(match.wUF.gp.se.gs ~ as.factor(vars.gp.se.gs$heart.diff.cat)))
```

```
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$heart.diff.cat)  1 0.0065 0.006485    2.061  0.153
## Residuals                               155 0.4878 0.003147
## 20 observations deleted due to missingness
```

```
summary(aov(match.uwUF.gp.se.gs ~ as.factor(vars.gp.se.gs$heart.diff.cat)))
```

```
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$heart.diff.cat)  1 0.0037 0.003710    0.628  0.429
## Residuals                               155 0.9154 0.005906
## 20 observations deleted due to missingness
```

```
#High blood sugar
```

```
summary(aov(match.BC.gp.se.gs ~ as.factor(vars.gp.se.gs$hbs.diff.cat)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$hbs.diff.cat)    1 0.0859 0.08594    6.081 0.0148 *
## Residuals                                150 2.1201 0.01413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 25 observations deleted due to missingness
```

```
summary(aov(match.J.gp.se.gs ~ as.factor(vars.gp.se.gs$hbs.diff.cat)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$hbs.diff.cat)    1 0.0434 0.04344    5.731 0.0179 *
## Residuals                                150 1.1369 0.00758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 25 observations deleted due to missingness
```

```
summary(aov(match.wUF.gp.se.gs ~ as.factor(vars.gp.se.gs$hbs.diff.cat)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$hbs.diff.cat)    1 0.0133 0.01334    4.195 0.0423 *
## Residuals                                150 0.4771 0.00318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 25 observations deleted due to missingness
```

```
summary(aov(match.uwUF.gp.se.gs ~ as.factor(vars.gp.se.gs$hbs.diff.cat)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gp.se.gs$hbs.diff.cat)    1 0.0159 0.015867    2.662 0.105
## Residuals                                150 0.8942 0.005961
## 25 observations deleted due to missingness
```

Diet

```
#Distance in protein consumption
```

```
summary(lm(match.BC.gp.se.gs ~ match.prot.gp.se.gs))
```

```
##
## Call:
## lm(formula = match.BC.gp.se.gs ~ match.prot.gp.se.gs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29716 -0.07575 -0.01612  0.08217  0.28846
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.643932   0.014968  43.021  <2e-16 ***
## match.prot.gp.se.gs 0.006465   0.005869   1.102    0.272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.122 on 173 degrees of freedom
```

```
## (2 observations deleted due to missingness)
## Multiple R-squared: 0.006966, Adjusted R-squared: 0.001226
## F-statistic: 1.214 on 1 and 173 DF, p-value: 0.2721
```

```
summary(lm(match.J.gp.se.gs ~ match.prot.gp.se.gs))
```

```
##
## Call:
## lm(formula = match.J.gp.se.gs ~ match.prot.gp.se.gs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.259843 -0.051366 -0.005535  0.063490  0.185048
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.777437   0.010991  70.733  <2e-16 ***
## match.prot.gp.se.gs 0.004467   0.004309   1.036   0.301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08958 on 173 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared: 0.006171, Adjusted R-squared: 0.0004263
## F-statistic: 1.074 on 1 and 173 DF, p-value: 0.3014
```

```
summary(lm(match.wUF.gp.se.gs ~ match.prot.gp.se.gs))
```

```
##
## Call:
## lm(formula = match.wUF.gp.se.gs ~ match.prot.gp.se.gs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.136859 -0.036391 -0.007626  0.032091  0.180736
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.229411   0.007156  32.057  <2e-16 ***
## match.prot.gp.se.gs 0.002438   0.002806   0.869   0.386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05832 on 173 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared: 0.004344, Adjusted R-squared: -0.001412
## F-statistic: 0.7547 on 1 and 173 DF, p-value: 0.3862
```

```
summary(lm(match.uwUF.gp.se.gs ~ match.prot.gp.se.gs))
```

```
##
## Call:
## lm(formula = match.uwUF.gp.se.gs ~ match.prot.gp.se.gs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.175107 -0.049752 -0.003172 0.049159 0.196814
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.594967  0.009619  61.855  <2e-16 ***
## match.prot.gp.se.gs 0.009683  0.003771   2.567  0.0111 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07839 on 173 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.0367, Adjusted R-squared:  0.03113
## F-statistic: 6.592 on 1 and 173 DF, p-value: 0.01109
```

```
summary(lm(match.uwUF.gp.se.gs ~ match.prot.gp.se.gs))
```

```
##
## Call:
## lm(formula = match.uwUF.gp.se.gs ~ match.prot.gp.se.gs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.175107 -0.049752 -0.003172  0.049159  0.196814
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.594967  0.009619  61.855  <2e-16 ***
## match.prot.gp.se.gs 0.009683  0.003771   2.567  0.0111 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07839 on 173 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.0367, Adjusted R-squared:  0.03113
## F-statistic: 6.592 on 1 and 173 DF, p-value: 0.01109
```

```
#Specific protein sources
```

```
summary(lm(match.BC.gp.se.gs ~ vars.gp.se.gs$meat.diff*vars.gp.se.gs$poul.diff))
```

```
##
## Call:
## lm(formula = match.BC.gp.se.gs ~ vars.gp.se.gs$meat.diff * vars.gp.se.gs$poul.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.289279 -0.071626 -0.006431  0.075048  0.285035
##
## Coefficients:
##              Estimate Std. Error
## (Intercept)    0.647063  0.014574
## vars.gp.se.gs$meat.diff -0.006476  0.010337
## vars.gp.se.gs$poul.diff -0.004548  0.013067
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff 0.015851  0.007321
##              t value Pr(>|t|)
## (Intercept)    44.398  <2e-16 ***
```

```
## vars.gp.se.gs$meat.diff          -0.627  0.5318
## vars.gp.se.gs$poul.diff          -0.348  0.7282
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff  2.165  0.0318 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1191 on 169 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared:  0.05315,    Adjusted R-squared:  0.03634
## F-statistic: 3.162 on 3 and 169 DF,  p-value: 0.02606
summary(lm(match.J.gp.se.gs ~ vars.gp.se.gs$meat.diff*vars.gp.se.gs$poul.diff))
```

```
##
## Call:
## lm(formula = match.J.gp.se.gs ~ vars.gp.se.gs$meat.diff * vars.gp.se.gs$poul.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.254621 -0.046544  0.001272  0.059397  0.182297
##
## Coefficients:
##              Estimate Std. Error
## (Intercept)      0.779554   0.010741
## vars.gp.se.gs$meat.diff -0.004700   0.007619
## vars.gp.se.gs$poul.diff -0.002873   0.009630
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff  0.011010   0.005396
##              t value Pr(>|t|)
## (Intercept)      72.576 <2e-16 ***
## vars.gp.se.gs$meat.diff -0.617   0.5381
## vars.gp.se.gs$poul.diff -0.298   0.7658
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff  2.041   0.0428 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08781 on 169 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared:  0.04762,    Adjusted R-squared:  0.03071
## F-statistic: 2.817 on 3 and 169 DF,  p-value: 0.04075
```

```
summary(lm(match.wUF.gp.se.gs ~ vars.gp.se.gs$meat.diff*vars.gp.se.gs$poul.diff))
##
## Call:
## lm(formula = match.wUF.gp.se.gs ~ vars.gp.se.gs$meat.diff * vars.gp.se.gs$poul.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.129991 -0.033924 -0.005739  0.037445  0.182249
##
## Coefficients:
##              Estimate Std. Error
## (Intercept)      0.231814   0.006991
## vars.gp.se.gs$meat.diff -0.002896   0.004958
## vars.gp.se.gs$poul.diff -0.006832   0.006268
```



```
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff 0.009259 0.003511
## t value Pr(>|t|)
## (Intercept) 33.161 < 2e-16 ***
## vars.gp.se.gs$meat.diff -0.584 0.55997
## vars.gp.se.gs$poul.diff -1.090 0.27724
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff 2.637 0.00915 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05715 on 169 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared: 0.06547, Adjusted R-squared: 0.04888
## F-statistic: 3.946 on 3 and 169 DF, p-value: 0.009409
summary(lm(match.uwUF.gp.se.gs ~ vars.gp.se.gs$meat.diff*vars.gp.se.gs$poul.diff))

##
## Call:
## lm(formula = match.uwUF.gp.se.gs ~ vars.gp.se.gs$meat.diff *
## vars.gp.se.gs$poul.diff)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.175056 -0.046957 -0.003657 0.051325 0.186175
##
## Coefficients:
## Estimate Std. Error
## (Intercept) 0.596806 0.009381
## vars.gp.se.gs$meat.diff 0.001576 0.006653
## vars.gp.se.gs$poul.diff 0.008810 0.008410
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff 0.006660 0.004712
## t value Pr(>|t|)
## (Intercept) 63.621 <2e-16 ***
## vars.gp.se.gs$meat.diff 0.237 0.813
## vars.gp.se.gs$poul.diff 1.048 0.296
## vars.gp.se.gs$meat.diff:vars.gp.se.gs$poul.diff 1.413 0.159
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07669 on 169 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared: 0.08156, Adjusted R-squared: 0.06526
## F-statistic: 5.002 on 3 and 169 DF, p-value: 0.002386
```

Variables only in spouses

Rural vs. urban

Not tested for siblings as all pairs resided together as children

```
summary(aov(match.BC.gp.se ~ vars.gp.se.all$RU.diff.cat))

## Df Sum Sq Mean Sq F value Pr(>F)
## vars.gp.se.all$RU.diff.cat 1 0.0043 0.004289 0.354 0.554
## Residuals 63 0.7635 0.012119
```

```
## 29 observations deleted due to missingness
summary(aov(match.J.gp.se ~ vars.gp.se.all$RU.diff.cat))

##              Df Sum Sq  Mean Sq F value Pr(>F)
## vars.gp.se.all$RU.diff.cat  1 0.0033 0.003346    0.48  0.491
## Residuals                63 0.4387 0.006964
## 29 observations deleted due to missingness
summary(aov(match.wUF.gp.se ~ vars.gp.se.all$RU.diff.cat))

##              Df Sum Sq  Mean Sq F value Pr(>F)
## vars.gp.se.all$RU.diff.cat  1 0.00245 0.002450    1.005  0.32
## Residuals                63 0.15363 0.002439
## 29 observations deleted due to missingness
summary(aov(match.uwUF.gp.se ~ vars.gp.se.all$RU.diff.cat))

##              Df Sum Sq  Mean Sq F value Pr(>F)
## vars.gp.se.all$RU.diff.cat  1 0.0049 0.004942    0.809  0.372
## Residuals                63 0.3847 0.006106
## 29 observations deleted due to missingness
```

Years cohabiting

```
summary(lm(match.BC.gp.se ~ vars.gp.se.all$yrs.cohab.marr))

##
## Call:
## lm(formula = match.BC.gp.se ~ vars.gp.se.all$yrs.cohab.marr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27365 -0.07187 -0.01203  0.06711  0.29172
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.694038   0.049805  13.935  <2e-16 ***
## vars.gp.se.all$yrs.cohab.marr -0.001267   0.001110  -1.141    0.257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1151 on 92 degrees of freedom
## Multiple R-squared:  0.01396,    Adjusted R-squared:  0.003244
## F-statistic: 1.303 on 1 and 92 DF,  p-value: 0.2567
summary(lm(match.J.gp.se ~ vars.gp.se.all$yrs.cohab.marr))

##
## Call:
## lm(formula = match.J.gp.se ~ vars.gp.se.all$yrs.cohab.marr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.241879 -0.050109 -0.003267  0.054601  0.192958
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.8184200  0.0373019  21.940  <2e-16 ***
## vars.gp.se.all$yrs.cohab.marr -0.0010279  0.0008314  -1.236   0.219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08623 on 92 degrees of freedom
## Multiple R-squared:  0.01635,    Adjusted R-squared:  0.005654
## F-statistic: 1.529 on 1 and 92 DF,  p-value: 0.2194
summary(lm(match.wUF.gp.se ~ vars.gp.se.all$yrs.cohab.marr))

##
## Call:
## lm(formula = match.wUF.gp.se ~ vars.gp.se.all$yrs.cohab.marr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.131040 -0.029903 -0.003345  0.025035  0.185109
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.2404884  0.0246612   9.752  7.6e-16 ***
## vars.gp.se.all$yrs.cohab.marr -0.0002728  0.0005496  -0.496   0.621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05701 on 92 degrees of freedom
## Multiple R-squared:  0.00267,    Adjusted R-squared:  -0.00817
## F-statistic: 0.2463 on 1 and 92 DF,  p-value: 0.6209
summary(lm(match.uwUF.gp.se ~ vars.gp.se.all$yrs.cohab.marr))

##
## Call:
## lm(formula = match.uwUF.gp.se ~ vars.gp.se.all$yrs.cohab.marr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15923 -0.05140 -0.01736  0.05751  0.18949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.6607260  0.0348422  18.963  <2e-16 ***
## vars.gp.se.all$yrs.cohab.marr -0.0015277  0.0007765  -1.967   0.0522 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08054 on 92 degrees of freedom
## Multiple R-squared:  0.04037,    Adjusted R-squared:  0.02994
## F-statistic: 3.87 on 1 and 92 DF,  p-value: 0.05216
```

Closeness

Test if closeness results hold true across just spouses

```
summary(lm(match.BC.gp.se ~ as.factor(floor(vars.gp.se.all$closeness))))
```

```
##
## Call:
## lm(formula = match.BC.gp.se ~ as.factor(floor(vars.gp.se.all$closeness)))
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.279084	-0.073207	-0.008726	0.076662	0.289892

```
##
## Coefficients:
```

	Estimate	Std. Error	t value
(Intercept)	0.66454	0.02643	25.145
as.factor(floor(vars.gp.se.all\$closeness))4	-0.03222	0.02959	-1.089

```
##
## Pr(>|t|)
## (Intercept)
## <2e-16 ***
## as.factor(floor(vars.gp.se.all$closeness))4
## 0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1152 on 92 degrees of freedom
## Multiple R-squared:  0.01272, Adjusted R-squared:  0.001993
## F-statistic: 1.186 on 1 and 92 DF, p-value: 0.279
summary(lm(match.J.gp.se ~ as.factor(floor(vars.gp.se.all$closeness))))
```

```
##
## Call:
## lm(formula = match.J.gp.se ~ as.factor(floor(vars.gp.se.all$closeness)))
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.246560	-0.050489	-0.000864	0.061090	0.190911

```
##
## Coefficients:
```

	Estimate	Std. Error	t value
(Intercept)	0.79339	0.01981	40.048
as.factor(floor(vars.gp.se.all\$closeness))4	-0.02477	0.02218	-1.117

```
##
## Pr(>|t|)
## (Intercept)
## <2e-16 ***
## as.factor(floor(vars.gp.se.all$closeness))4
## 0.267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08636 on 92 degrees of freedom
## Multiple R-squared:  0.01338, Adjusted R-squared:  0.002653
## F-statistic: 1.247 on 1 and 92 DF, p-value: 0.267
```

```
summary(lm(match.wUF.gp.se ~ as.factor(floor(vars.gp.se.all$closeness))))
```

```
##
## Call:
## lm(formula = match.wUF.gp.se ~ as.factor(floor(vars.gp.se.all$closeness)))
##
## Residuals:
```

```
##           Min           1Q       Median           3Q           Max
## -0.132328 -0.031450 -0.007465  0.029408  0.186277
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                   0.233670   0.013082  17.862
## as.factor(floor(vars.gp.se.all$closeness))4 -0.006352   0.014646  -0.434
##                                Pr(>|t|)
## (Intercept)                   <2e-16 ***
## as.factor(floor(vars.gp.se.all$closeness))4    0.666
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05702 on 92 degrees of freedom
## Multiple R-squared:  0.00204,    Adjusted R-squared:  -0.008807
## F-statistic: 0.1881 on 1 and 92 DF,  p-value: 0.6655
summary(lm(match.uwUF.gp.se ~ as.factor(floor(vars.gp.se.all$closeness))))

##
## Call:
## lm(formula = match.uwUF.gp.se ~ as.factor(floor(vars.gp.se.all$closeness)))
##
## Residuals:
##           Min           1Q       Median           3Q           Max
## -0.15874 -0.05017 -0.01153  0.05363  0.19457
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                   0.61675   0.01868  33.023
## as.factor(floor(vars.gp.se.all$closeness))4 -0.02832   0.02091  -1.354
##                                Pr(>|t|)
## (Intercept)                   <2e-16 ***
## as.factor(floor(vars.gp.se.all$closeness))4    0.179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08141 on 92 degrees of freedom
## Multiple R-squared:  0.01955,    Adjusted R-squared:  0.008888
## F-statistic: 1.834 on 1 and 92 DF,  p-value: 0.179
```

Variables only in siblings

Sex

Not tested for spouses because all spousal pairs were male-female

```
summary(aov(match.BC.gs ~ as.factor(vars.gs.all$gender.diff.cat)))

##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gs.all$gender.diff.cat)  1 0.0004 0.000437   0.027  0.869
## Residuals                               81 1.2952 0.015990

summary(aov(match.J.gs ~ as.factor(vars.gs.all$gender.diff.cat)))

##                                Df Sum Sq Mean Sq F value Pr(>F)
```

```
## as.factor(vars.gs.all$gender.diff.cat) 1 0.0002 0.000221 0.027 0.871
## Residuals 81 0.6712 0.008287
```

```
summary(aov(match.wUF.gs ~ as.factor(vars.gs.all$gender.diff.cat)))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gs.all$gender.diff.cat) 1 0.0001 0.000059 0.015 0.902
## Residuals 81 0.3177 0.003923
```

```
summary(aov(match.uwUF.gs ~ as.factor(vars.gs.all$gender.diff.cat)))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gs.all$gender.diff.cat) 1 0.0013 0.001326 0.261 0.611
## Residuals 81 0.4114 0.005079
```

Genetics

Genetic relatedness defined by a kinship score.

```
#Subset data to those with kinship scores available
```

```
match.BC.gs.relate = subset(match.BC.gs, names(match.BC.gs) %in% row.names(relatedness.gs.match))
match.J.gs.relate = subset(match.J.gs, names(match.J.gs) %in% row.names(relatedness.gs.match))
match.wUF.gs.relate = subset(match.wUF.gs, names(match.wUF.gs) %in% row.names(relatedness.gs.match))
match.uwUF.gs.relate = subset(match.uwUF.gs, names(match.uwUF.gs) %in% row.names(relatedness.gs.match))
```

```
summary(aov(match.BC.gs.relate ~ relatedness.gs.match$kinship))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## relatedness.gs.match$kinship 1 0.0063 0.006294 0.43 0.514
## Residuals 74 1.0823 0.014626
```

```
summary(aov(match.J.gs.relate ~ relatedness.gs.match$kinship))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## relatedness.gs.match$kinship 1 0.0050 0.004982 0.674 0.414
## Residuals 74 0.5467 0.007388
```

```
summary(aov(match.wUF.gs.relate ~ relatedness.gs.match$kinship))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## relatedness.gs.match$kinship 1 0.0006 0.000603 0.157 0.693
## Residuals 74 0.2848 0.003848
```

```
summary(aov(match.uwUF.gs.relate ~ relatedness.gs.match$kinship))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## relatedness.gs.match$kinship 1 0.0001 0.00005 0.011 0.917
## Residuals 74 0.3412 0.00461
```

Rural vs. urban

Not tested for spouses as all resided together at time of sampling.

```
summary(aov(match.BC.gs ~ as.factor(vars.gs.all$RU11.diff)))
```

```
## Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(vars.gs.all$RU11.diff) 1 0.0150 0.01496 1.25 0.271
## Residuals 35 0.4189 0.01197
## 46 observations deleted due to missingness
```

```
summary(aov(match.J.gs ~ as.factor(vars.gs.all$RU11.diff)))
```

```
##
## as.factor(vars.gs.all$RU11.diff)  1 0.00822 0.008223  1.345  0.254
## Residuals                        35 0.21401 0.006115
## 46 observations deleted due to missingness
```

```
summary(aov(match.wUF.gs ~ as.factor(vars.gs.all$RU11.diff)))
```

```
##
## as.factor(vars.gs.all$RU11.diff)  1 0.00082 0.000817  0.22  0.642
## Residuals                        35 0.13019 0.003720
## 46 observations deleted due to missingness
```

```
summary(aov(match.uwUF.gs ~ as.factor(vars.gs.all$RU11.diff)))
```

```
##
## as.factor(vars.gs.all$RU11.diff)  1 0.00345 0.003446  0.682  0.415
## Residuals                        35 0.17692 0.005055
## 46 observations deleted due to missingness
```

CohabYN

Cohabiting with a spouse or not

```
summary(aov(match.BC.gs ~ as.factor(vars.gs.all$cohab.diff.cat)))
```

```
##
## as.factor(vars.gs.all$cohab.diff.cat)  1 0.0546 0.05463  3.565 0.0626 .
## Residuals                        81 1.2410 0.01532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(match.J.gs ~ as.factor(vars.gs.all$cohab.diff.cat)))
```

```
##
## as.factor(vars.gs.all$cohab.diff.cat)  1 0.0304 0.030437  3.846 0.0533 .
## Residuals                        81 0.6410 0.007914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(match.wUF.gs ~ as.factor(vars.gs.all$cohab.diff.cat)))
```

```
##
## as.factor(vars.gs.all$cohab.diff.cat)  1 0.01185 0.011846  3.136 0.0803 .
## Residuals                        81 0.30594 0.003777
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(match.uwUF.gs ~ as.factor(vars.gs.all$cohab.diff.cat)))
```

```
##
## as.factor(vars.gs.all$cohab.diff.cat)  1 0.0092 0.009176  1.842  0.179
## Residuals                        81 0.4035 0.004982
```

Socialness

```
summary(lm(match.BC.gs ~ vars.gs.all$social.sum.diff))

##
## Call:
## lm(formula = match.BC.gs ~ vars.gs.all$social.sum.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.303079 -0.064872 -0.007305  0.091746  0.263556
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.689907   0.022991  30.008  <2e-16 ***
## vars.gs.all$social.sum.diff -0.004186   0.002636  -1.588   0.118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1232 on 51 degrees of freedom
## (30 observations deleted due to missingness)
## Multiple R-squared:  0.04713,    Adjusted R-squared:  0.02845
## F-statistic: 2.523 on 1 and 51 DF,  p-value: 0.1184

summary(lm(match.J.gs ~ vars.gs.all$social.sum.diff))

##
## Call:
## lm(formula = match.J.gs ~ vars.gs.all$social.sum.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.250890 -0.041559  0.002072  0.068782  0.172481
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.808751   0.016916  47.809  <2e-16 ***
## vars.gs.all$social.sum.diff -0.002799   0.001939  -1.443   0.155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09064 on 51 degrees of freedom
## (30 observations deleted due to missingness)
## Multiple R-squared:  0.03924,    Adjusted R-squared:  0.0204
## F-statistic: 2.083 on 1 and 51 DF,  p-value: 0.1551

summary(lm(match.wUF.gs ~ vars.gs.all$social.sum.diff))

##
## Call:
## lm(formula = match.wUF.gs ~ vars.gs.all$social.sum.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.097302 -0.038009 -0.006541  0.031277  0.145307
##
## Coefficients:
```



```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.242229   0.009722  24.915   <2e-16 ***
## vars.gs.all$social.sum.diff -0.001621   0.001115  -1.454    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05209 on 51 degrees of freedom
## (30 observations deleted due to missingness)
## Multiple R-squared:  0.03982, Adjusted R-squared:  0.02099
## F-statistic: 2.115 on 1 and 51 DF, p-value: 0.152
summary(lm(match.uwUF.gs ~ vars.gs.all$social.sum.diff))

##
## Call:
## lm(formula = match.uwUF.gs ~ vars.gs.all$social.sum.diff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.186028 -0.043849 -0.000991  0.032925  0.162362
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.6286073   0.0130334  48.230   <2e-16 ***
## vars.gs.all$social.sum.diff -0.0007563   0.0014941  -0.506    0.615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06983 on 51 degrees of freedom
## (30 observations deleted due to missingness)
## Multiple R-squared:  0.005, Adjusted R-squared: -0.01451
## F-statistic: 0.2563 on 1 and 51 DF, p-value: 0.6149
```

Closeness

Test if closeness results hold true across just siblings

```
summary(aov(match.BC.gs ~ as.factor(floor(vars.gs.all$closeness.ave))))

##               Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gs.all$closeness.ave))  3 0.0152 0.005065   0.312
## Residuals                                79 1.2805 0.016208
##               Pr(>F)
## as.factor(floor(vars.gs.all$closeness.ave))  0.816
## Residuals

summary(aov(match.J.gs ~ as.factor(floor(vars.gs.all$closeness.ave))))

##               Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gs.all$closeness.ave))  3 0.0109 0.003645   0.436
## Residuals                                79 0.6605 0.008361
##               Pr(>F)
## as.factor(floor(vars.gs.all$closeness.ave))  0.728
## Residuals
```

```
summary(aov(match.wUF.gs ~ as.factor(floor(vars.gs.all$closeness.ave))))

##                               Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gs.all$closeness.ave))  3 0.00952 0.003174  0.813
## Residuals                               79 0.30826 0.003902
##                               Pr(>F)
## as.factor(floor(vars.gs.all$closeness.ave))  0.49
## Residuals
```

```
summary(aov(match.uwUF.gs ~ as.factor(floor(vars.gs.all$closeness.ave))))

##                               Df Sum Sq Mean Sq F value
## as.factor(floor(vars.gs.all$closeness.ave))  3 0.0126 0.004212  0.832
## Residuals                               79 0.4001 0.005064
##                               Pr(>F)
## as.factor(floor(vars.gs.all$closeness.ave))  0.48
## Residuals
```

FDR-correction

```
BC.sib.spouse.list = c(0.554,0.257,0.869,0.514,0.271,0.0626,0.118,0.0319,0.544,0.272,0.194,0.0148,0.272
J.sib.spouse.list = c(0.491,0.219,0.871,0.414,0.254,0.0533,0.155,0.0305,0.519,0.219,0.159,0.0179,0.301,
wUF.sib.spouse.list = c(0.338,0.657,0.901,0.829,0.832,0.0717,0.176,0.0562,0.962,0.754,0.147,0.052,0.672
uwUF.sib.spouse.list = c(0.361,0.0512,0.609,0.911,0.427,0.177,0.61,0.0266,0.731,0.00405,0.424,0.106,0.0
```

```
p.adjust(BC.sib.spouse.list, method="BH")
```

```
## [1] 0.6331429 0.4352000 0.8690000 0.6331429 0.4352000 0.2504000 0.3776000
## [8] 0.1701333 0.6331429 0.4352000 0.4352000 0.1701333 0.4352000 0.6331429
## [15] 0.7767467 0.1701333
```

```
p.adjust(J.sib.spouse.list, method="BH")
```

```
## [1] 0.6149714 0.4380000 0.8710000 0.6021818 0.4515556 0.2132000 0.4240000
## [8] 0.2132000 0.6149714 0.4380000 0.4240000 0.2132000 0.4816000 0.6149714
## [15] 0.8168533 0.2132000
```

```
p.adjust(wUF.sib.spouse.list, method="BH")
```

```
## [1] 0.7608000 0.9508571 0.9610667 0.9508571 0.9508571 0.2868000 0.4693333
## [8] 0.2868000 0.9620000 0.9508571 0.4693333 0.2868000 0.9508571 0.9508571
## [15] 0.7608000 0.2868000
```

```
p.adjust(uwUF.sib.spouse.list, method="BH")
```

```
## [1] 0.6210909 0.2048000 0.7507692 0.9110000 0.6210909 0.4045714 0.7507692
## [8] 0.1418667 0.8354286 0.0648000 0.6210909 0.3392000 0.0952000 0.8672000
## [15] 0.6040000 0.4045714
```

Interactions

Do variables that explain beta-diversity differ between spouse and sibling groups?

```
#AB
chisq.test(table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$AB.diff.cat)))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$AB.diff.cat))
## X-squared = 1.7011, df = 1, p-value = 0.1921

#Closeness
chisq.test(table(vars.gp.se.gs$group, as.factor(floor(vars.gp.se.gs$closeness))))

## Warning in chisq.test(table(vars.gp.se.gs$group,
## as.factor(floor(vars.gp.se.gs$closeness)))): Chi-squared approximation may
## be incorrect

##
## Pearson's Chi-squared test
##
## data: table(vars.gp.se.gs$group, as.factor(floor(vars.gp.se.gs$closeness)))
## X-squared = 59.604, df = 3, p-value = 7.142e-13

#heart.diff
chisq.test(table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$heart.diff.cat)))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$heart.diff.cat))
## X-squared = 0.065908, df = 1, p-value = 0.7974

#hbs.diff
chisq.test(table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$hbs.diff.cat)))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(vars.gp.se.gs$group, as.factor(vars.gp.se.gs$hbs.diff.cat))
## X-squared = 0, df = 1, p-value = 1

#dist.prot
summary(aov(match.prot.gp.se.gs ~ vars.gp.se.gs$group))

##
##              Df Sum Sq Mean Sq F value    Pr(>F)
## vars.gp.se.gs$group  1   84.5   84.48    42.05 8.99e-10 ***
## Residuals          173  347.6    2.01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness

#meat.diff:poul.diff
summary(aov(vars.gp.se.gs$meat.diff ~ vars.gp.se.gs$group))

##
##              Df Sum Sq Mean Sq F value    Pr(>F)
## vars.gp.se.gs$group  1  40.28   40.28   29.38 1.99e-07 ***
## Residuals          172 235.86    1.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness

summary(aov(vars.gp.se.gs$poul.diff ~ vars.gp.se.gs$group))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## vars.gp.se.gs$group    1  16.21   16.214    19.71 1.61e-05 ***
## Residuals          172  141.49    0.823
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness
```

FDR-correction

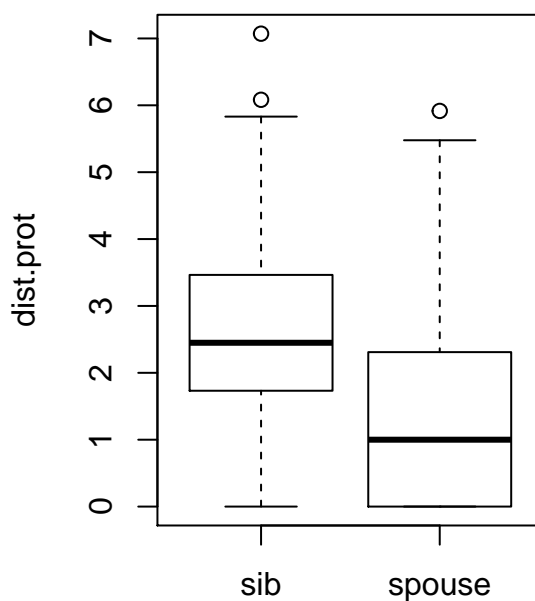
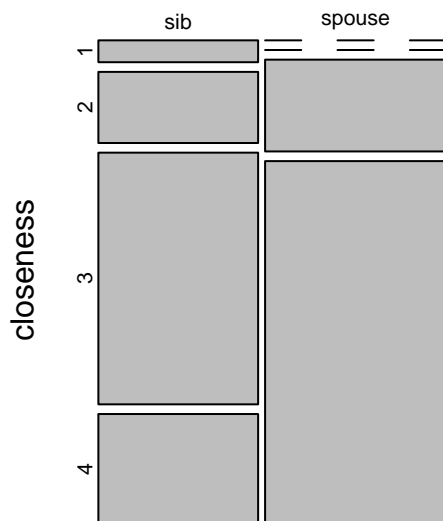
```
p.adjust(c(0.1921,7.14E-13,0.7974,1,8.99E-10,1.99E-07,1.61E-05), method="BH")
```

```
## [1] 2.689400e-01 4.998000e-12 9.303000e-01 1.000000e+00 3.146500e-09
## [6] 4.643333e-07 2.817500e-05
```

It appears that spouses tend to have closer relationships and more similar diets than siblings.

```
#Plots
par(mfrow = c(1, 2))
plot(table(vars.gp.se.gs$group, as.factor(floor(vars.gp.se.gs$closeness))), ylab="closeness")
boxplot(match.prot.gp.se.gs ~ vars.gp.se.gs$group, ylab="dist.prot")
```

e.gs\$group, as.factor(floor(vars.gp.



OTUs and health

OTUs commonly shared between siblings or spouses (> 50% of pairs) and rare in the overall dataset (< 70% of samples, < 49% of unrelated pairs) were correlated to health outcomes in the whole dataset by Kruskal-Wallis with the Benjamini-Hochberg correction for multiple comparisons.

We will co-op the `kruskal.pretty` function to run these tests by creating a fake SIMPER output. We will also need to load our unaltered `.cons.taxonomy` table from `mothur` as the function cannot take the altered version `Tax` we have used those far.

```
#Fake SIMPER output to get OTU list for health analyses
fake.simper = data.frame(
  X = c(1:22),
  Comparison = rep("1_2", 22),
  SIMPER = rep(0.001, 22),
  OTU = c("Otu00027", "Otu00035", "Otu00043", "Otu00007", "Otu00019", "Otu00025", "Otu00047", "Otu00004", "Otu00000", "Otu00001", "Otu00002", "Otu00003", "Otu00004", "Otu00005", "Otu00006", "Otu00007", "Otu00008", "Otu00009", "Otu00010", "Otu00011", "Otu00012")
)

#Unaltered mothur taxonomy table
Tax.orig = read.table("WLS.final.an.unique_list.0.02.cons.taxonomy", header=TRUE, row.names=1, sep="\t")
```

We then run Kruskal-Wallis tests for our health variables with NAs removed.

```
kruskal.pretty(OTU[!is.na(Pheno.sm$hbp11),], Pheno.sm[!is.na(Pheno.sm$hbp11),], fake.simper, c("hbp11"))

##
## Attaching package: 'dplyr'
## The following object is masked from 'package:gridExtra':
##
##      combine
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

kruskal.pretty(OTU[!is.na(Pheno.sm$hbs11),], Pheno.sm[!is.na(Pheno.sm$hbs11),], fake.simper, c("hbs11"))
kruskal.pretty(OTU[!is.na(Pheno.sm$heart11),], Pheno.sm[!is.na(Pheno.sm$heart11),], fake.simper, c("heart11"))
kruskal.pretty(OTU[!is.na(Pheno.sm$arthritis11),], Pheno.sm[!is.na(Pheno.sm$arthritis11),], fake.simper, c("arthritis11"))
kruskal.pretty(OTU[!is.na(Pheno.sm$cancer11),], Pheno.sm[!is.na(Pheno.sm$cancer11),], fake.simper, c("cancer11"))
kruskal.pretty(OTU[!is.na(Pheno.sm$stroke11),], Pheno.sm[!is.na(Pheno.sm$stroke11),], fake.simper, c("stroke11"))
```

The IBS variable was not part of the `Pheno.sm` survey so we curate it from the larger `Pheno` data set.

- `jx148rer` = graduates
- `px148rer` = siblings

```
vars.IBS = data.frame(
  IBS = c(Pheno$jx148rer, Pheno$px148rer),
  ID = c(paste(Pheno$idpriv, "g", sep=""), paste(Pheno$idpriv, "s", sep="")),
  row.names = c(paste(Pheno$idpriv, "g", sep=""), paste(Pheno$idpriv, "s", sep=""))
)

#Subset to data with microbiota samples and IBS data
vars.IBS.sub = as.data.frame(subset(vars.IBS, row.names(vars.IBS) %in% row.names(OTU)))
vars.IBS.all = as.data.frame(vars.IBS.sub[!is.na(vars.IBS.sub$IBS),])
OTU.IBS = subset(OTU, row.names(OTU) %in% row.names(vars.IBS.all))
```

Run Kruskal-Wallis test of new IBS variable.

```
kruskal.pretty(OTU.IBS, vars.IBS.all, fake.simper, c("IBS"), 'IBS', Tax.orig)
```

Figures

Figure 1

```
set.seed(51234)
#Calculate NMDS
uwUF.g.ordu = ordinate(physeq.tree.g, method="NMDS", distance="unifrac", weighted=FALSE)

## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu04244 --
## in the phylogenetic tree in the data you provided.

## Run 0 stress 0.1548755
## Run 1 stress 0.171804
## Run 2 stress 0.1630789
## Run 3 stress 0.173726
## Run 4 stress 0.1855051
## Run 5 stress 0.1834981
## Run 6 stress 0.1866751
## Run 7 stress 0.1641615
## Run 8 stress 0.1728711
## Run 9 stress 0.1937877
## Run 10 stress 0.1705475
## Run 11 stress 0.1643311
## Run 12 stress 0.1579658
## Run 13 stress 0.1838658
## Run 14 stress 0.1867786
## Run 15 stress 0.1578942
## Run 16 stress 0.177159
## Run 17 stress 0.173864
## Run 18 stress 0.1681529
## Run 19 stress 0.1841738
## Run 20 stress 0.1663627
## *** No convergence -- monoMDS stopping criteria:
##      20: stress ratio > sratmax

#Create data set of just significant variables
vars.envfit = data.frame(
  AB = ifelse(vars.g$AB == "Y", 2, 1), #Reverse code so that arrow points toward Y responses
  cohabYN = ifelse(vars.g$cohabYN == "N", 1, 2), #Reverse code so that arrow points toward Y responses
  socialness = vars.g$social.sum,
  hbs = ifelse(vars.g$hbs11 == "Y", 2, 1), #Reverse code so that arrow points toward Y responses
  heart = ifelse(vars.g$heart11 == 1, 2, 1), #Reverse code so that arrow points toward Y responses
  meat = vars.g$meat,
  poultry = vars.g$poul,
  prot.sum = vars.g$prot.sum,
  social.cohabY = ifelse(vars.g$cohabYN == "Y", vars.g$social.sum, NA),
  social.cohabN = ifelse(vars.g$cohabYN == "N", vars.g$social.sum, NA),
  row.names = row.names(vars.g))
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in cor(Hperm, take): the standard deviation is zero
```

```
ef.uwUF.socY = envfit(uwUF.g.ordu ~ social.cohabY, data=vars.envfit, na.rm=TRUE)
ef.uwUF.socN = envfit(uwUF.g.ordu ~ social.cohabN, data=vars.envfit, na.rm=TRUE)
ef.uwUF.hbs = envfit(uwUF.g.ordu ~ hbs, data=vars.envfit, na.rm=TRUE)
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

```
## Warning in cor(H, Pw): the standard deviation is zero
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

## Warning in cor(H, Pw): the standard deviation is zero

## Warning in cor(H, Pw): the standard deviation is zero

## Warning in cor(H, Pw): the standard deviation is zero

## Warning in cor(H, Pw): the standard deviation is zero

## Warning in cor(H, Pw): the standard deviation is zero

ef.uwUF.heart = envfit(uwUF.g.ordu ~ heart, data=vars.envfit, na.rm=TRUE)
ef.uwUF.meat = envfit(uwUF.g.ordu ~ meat, data=vars.envfit, na.rm=TRUE)
ef.uwUF.poul = envfit(uwUF.g.ordu ~ poultry, data=vars.envfit, na.rm=TRUE)

#Figure 1
postscript("FIGURE1.eps", width = 4.2, height = 4.5, horizontal = FALSE, onefile = FALSE, paper = "spec
plot(uwUF.g.ordu, type="n", main="", xlab="", ylab="")

## Warning in ordiplot(x, choices = choices, type = type, display = display, :
## Species scores not available
points(uwUF.g.ordu, pch=21, cex=0.75, display="sites", col="black", bg=c("red","blue")[as.factor(Pheno.
plot(ef.uwUF.AB, col="red")

## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
plot(ef.uwUF.socY, col="black")
plot(ef.uwUF.socN, col="red")
plot(ef.uwUF.hbs, col="red")

## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
plot(ef.uwUF.heart, col="red")
plot(ef.uwUF.meat, col="black")
plot(ef.uwUF.poul, col="black")
dev.off()

## pdf
## 2

```

Figure 2

```
#Figure 2
postscript("FIGURE2new.eps", width = 5, height = 4, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2), 1, 2, byrow = TRUE), widths=c(1,1), heights=c(1,1))

#Shannon
boxplot(Met.g$shannon ~ vars.g$cohabYN, ylab="Shannon's diversity", main="", xlab="Cohabiting", col=c("red", "blue"))

#Chao
boxplot(Met.g$chao ~ vars.g$cohabYN, ylab="Chao's richness", main="", xlab="Cohabiting", col=c("red", "blue"),
dev.off())

## pdf
## 2
```

Figure 3

```
#Load results from KW.subsample.fxn
KW.results.final = read.table("KW.results.csv", sep=",")

#Re-organize into data frame
gp.se.gs.boxplot = data.frame(
  BC = c(match.BC.gp.se.gs, KW.results.final[KW.results.final$V6 == "dist" & complete.cases(KW.results.final$V6)]),
  J = c(match.J.gp.se.gs, KW.results.final[KW.results.final$V6 == "dist" & complete.cases(KW.results.final$V6)]),
  wUF = c(match.wUF.gp.se.gs, KW.results.final[KW.results.final$V6 == "dist" & complete.cases(KW.results.final$V6)]),
  uwUF = c(match.uwUF.gp.se.gs, KW.results.final[KW.results.final$V6 == "dist" & complete.cases(KW.results.final$V6)]),
  group = c(rep("1.spouse", length(match.BC.gp.se)), rep("2.sibling", length(match.BC.gs)), rep("3.unrelated", length(match.BC.unrel)))

#Add variable for closeness:pair type
vars.gp.se.gs$closeness.group = ifelse(floor(vars.gp.se.gs$closeness) == 1 & vars.gp.se.gs$group == "spouse", "1",
ifelse(floor(vars.gp.se.gs$closeness) == 2 & vars.gp.se.gs$group == "spouse", "2",
ifelse(floor(vars.gp.se.gs$closeness) == 3 & vars.gp.se.gs$group == "spouse", "3",
ifelse(floor(vars.gp.se.gs$closeness) == 4 & vars.gp.se.gs$group == "spouse", "4",
ifelse(floor(vars.gp.se.gs$closeness) == 1 & vars.gp.se.gs$group == "sibling", "1",
ifelse(floor(vars.gp.se.gs$closeness) == 2 & vars.gp.se.gs$group == "sibling", "2",
ifelse(floor(vars.gp.se.gs$closeness) == 3 & vars.gp.se.gs$group == "sibling", "3",
ifelse(floor(vars.gp.se.gs$closeness) == 4 & vars.gp.se.gs$group == "sibling", "4",
ifelse(floor(vars.gp.se.gs$closeness) == 1 & vars.gp.se.gs$group == "unrelated", "1",
ifelse(floor(vars.gp.se.gs$closeness) == 2 & vars.gp.se.gs$group == "unrelated", "2",
ifelse(floor(vars.gp.se.gs$closeness) == 3 & vars.gp.se.gs$group == "unrelated", "3",
ifelse(floor(vars.gp.se.gs$closeness) == 4 & vars.gp.se.gs$group == "unrelated", "4", NA))

#Figure 3
postscript("FIGURE3.eps", width = 8, height = 5, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2,3), 1, 3, byrow = TRUE), widths=c(2,2.5,3), heights=c(1,1,1))

boxplot(gp.se.gs.boxplot$uwUF ~ gp.se.gs.boxplot$group, main = "", ylab="Unweighted UniFrac distance", col=c("blue", "red", "blue"))

boxplot(match.uwUF.gp.se.gs~floor(vars.gp.se.gs$closeness), xlab="", ylab="", main="", outpch=NA, ylim=c(0,1))

boxplot(match.uwUF.gp.se.gs~vars.gp.se.gs$closeness.group, col=c("blue", "blue", "blue", "red", "blue"),
dev.off())

## pdf
## 2
```

Figure 4

```

#Calculate relative abundances
abund = OTU/rowSums(OTU)

#Re-code variable to YN letters
Pheno.sm$hbs = ifelse(Pheno.sm$hbs11 == 1, "Y", ifelse(Pheno.sm$hbs11 == 2, "N", Pheno.sm$hbs11))

#Combine OTU abundance with health data
hbs.OTUs1 = data.frame(
  abundance = c(abund$Otu00160*100,abund$Otu00295*100),
  OTU = c(rep("OTU160", 408),rep("OTU295", 408)),
  hbs = rep(Pheno.sm$hbs, 2))
hbs.OTUs2 = data.frame(
  abundance = c(abund$Otu00004*100,abund$Otu00019*100,abund$Otu00074*100),
  OTU = c(rep("OTU004", 408),rep("OTU019", 408),rep("OTU074", 408)),
  hbs = rep(Pheno.sm$hbs, 3))

#Calculate means and standard deviations of OTU abundances within health groups
hbs.OTUs.data1 = aggregate(hbs.OTUs1$abundance, by=list(hbs.OTUs1$OTU, hbs.OTUs1$hbs), FUN = function(x)
hbs.OTUs.data1 = do.call(data.frame, hbs.OTUs.data1)
hbs.OTUs.data1$se = hbs.OTUs.data1$x.sd / sqrt(hbs.OTUs.data1$x.n)
colnames(hbs.OTUs.data1) = c("OTU", "hbs", "mean", "sd", "n", "se")

hbs.OTUs.data2 = aggregate(hbs.OTUs2$abundance, by=list(hbs.OTUs2$OTU, hbs.OTUs2$hbs), FUN = function(x)
hbs.OTUs.data2 = do.call(data.frame, hbs.OTUs.data2)
hbs.OTUs.data2$se = hbs.OTUs.data2$x.sd / sqrt(hbs.OTUs.data2$x.n)
colnames(hbs.OTUs.data2) = c("OTU", "hbs", "mean", "sd", "n", "se")

#Figure 4
limits1 <- aes(ymax = hbs.OTUs.data1$mean + hbs.OTUs.data1$se, ymin = hbs.OTUs.data1$mean - hbs.OTUs.da
p1 = ggplot(data = hbs.OTUs.data1, aes(x = factor(OTU), y = mean, fill = factor(hbs)))
limits2 <- aes(ymax = hbs.OTUs.data2$mean + hbs.OTUs.data2$se, ymin = hbs.OTUs.data2$mean - hbs.OTUs.da
p2 = ggplot(data = hbs.OTUs.data2, aes(x = factor(OTU), y = mean, fill = factor(hbs)))
plot1 = p1 + geom_bar(stat = "identity", position = position_dodge(0.9)) +
  geom_errorbar(limits1, position = position_dodge(0.9), width = 0.25) +
  labs(x = "", y = "Relative abundance (%)") + scale_fill_manual(name = "HBS", values=c("red", "blue"))
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor
plot2 = p2 + geom_bar(stat = "identity", position = position_dodge(0.9)) +
  geom_errorbar(limits2, position = position_dodge(0.9), width = 0.25) +
  labs(x = "", y = "") + scale_fill_manual(name = "HBS", values=c("red", "blue")) +
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor

postsript("FIGURE4.eps", width = 5, height = 2.5, horizontal = FALSE, onefile = FALSE, paper = "special")
multiplot(plot1, plot2, cols=2)
dev.off()

## pdf
## 2

```

Figure S1

```
set.seed(51234)
#Calculate NMDS
BC.g.MDS=metaMDS(OTU.g, distance="bray", k=2, trymax=500)

## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1778574
## Run 1 stress 0.1830169
## Run 2 stress 0.2165385
## Run 3 stress 0.1768809
## ... New best solution
## ... Procrustes: rmse 0.03254256  max resid 0.3843774
## Run 4 stress 0.1782439
## Run 5 stress 0.1757846
## ... New best solution
## ... Procrustes: rmse 0.01052766  max resid 0.1034541
## Run 6 stress 0.186635
## Run 7 stress 0.1768495
## Run 8 stress 0.1866905
## Run 9 stress 0.1961226
## Run 10 stress 0.175787
## ... Procrustes: rmse 0.0006641125  max resid 0.005862217
## ... Similar to previous best
## Run 11 stress 0.1870023
## Run 12 stress 0.1790881
## Run 13 stress 0.2053181
## Run 14 stress 0.1951932
## Run 15 stress 0.19719
## Run 16 stress 0.2008779
## Run 17 stress 0.1874823
## Run 18 stress 0.1899205
## Run 19 stress 0.1768658
## Run 20 stress 0.1768485
## *** Solution reached

J.g.MDS=metaMDS(OTU.g, distance="jaccard", k=2, trymax=500)

## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1778344
## Run 1 stress 0.1862218
## Run 2 stress 0.1830001
## Run 3 stress 0.1895922
## Run 4 stress 0.1790675
## Run 5 stress 0.1908328
## Run 6 stress 0.1956368
## Run 7 stress 0.1783647
## Run 8 stress 0.2001721
## Run 9 stress 0.1790354
## Run 10 stress 0.1928207
## Run 11 stress 0.1914063
## Run 12 stress 0.1883793
```

```

## Run 13 stress 0.1867394
## Run 14 stress 0.2097437
## Run 15 stress 0.1811278
## Run 16 stress 0.1856737
## Run 17 stress 0.2057145
## Run 18 stress 0.188236
## Run 19 stress 0.1915302
## Run 20 stress 0.1982333
## Run 21 stress 0.2017376
## Run 22 stress 0.1788227
## Run 23 stress 0.1943584
## Run 24 stress 0.1781236
## ... Procrustes: rmse 0.007399325  max resid 0.08542497
## Run 25 stress 0.19052
## Run 26 stress 0.1779312
## ... Procrustes: rmse 0.003673951  max resid 0.04211143
## Run 27 stress 0.1921566
## Run 28 stress 0.1790291
## Run 29 stress 0.1778557
## ... Procrustes: rmse 0.001420418  max resid 0.01682678
## Run 30 stress 0.215146
## Run 31 stress 0.1913911
## Run 32 stress 0.1885353
## Run 33 stress 0.1858878
## Run 34 stress 0.1913819
## Run 35 stress 0.1929349
## Run 36 stress 0.2047987
## Run 37 stress 0.1929133
## Run 38 stress 0.193596
## Run 39 stress 0.1758107
## ... New best solution
## ... Procrustes: rmse 0.03046111  max resid 0.3819559
## Run 40 stress 0.1776132
## Run 41 stress 0.1811713
## Run 42 stress 0.1801865
## Run 43 stress 0.1768579
## Run 44 stress 0.1884731
## Run 45 stress 0.1865361
## Run 46 stress 0.1946415
## Run 47 stress 0.1801935
## Run 48 stress 0.1768662
## Run 49 stress 0.1785746
## Run 50 stress 0.1830558
## Run 51 stress 0.2041382
## Run 52 stress 0.2010643
## Run 53 stress 0.1790965
## Run 54 stress 0.2140271
## Run 55 stress 0.2014982
## Run 56 stress 0.185326
## Run 57 stress 0.2158569
## Run 58 stress 0.1877491
## Run 59 stress 0.19494
## Run 60 stress 0.2141599
## Run 61 stress 0.1891321

```

```

## Run 62 stress 0.1931647
## Run 63 stress 0.1800636
## Run 64 stress 0.1875424
## Run 65 stress 0.1907144
## Run 66 stress 0.1759712
## ... Procrustes: rmse 0.007250359  max resid 0.08512437
## Run 67 stress 0.1758466
## ... Procrustes: rmse 0.003536395  max resid 0.04337283
## Run 68 stress 0.1788659
## Run 69 stress 0.1810522
## Run 70 stress 0.1875395
## Run 71 stress 0.1766162
## Run 72 stress 0.1962701
## Run 73 stress 0.1782816
## Run 74 stress 0.1919053
## Run 75 stress 0.1861325
## Run 76 stress 0.1979306
## Run 77 stress 0.1955984
## Run 78 stress 0.1926229
## Run 79 stress 0.1792036
## Run 80 stress 0.1948191
## Run 81 stress 0.1969628
## Run 82 stress 0.1920004
## Run 83 stress 0.1782967
## Run 84 stress 0.216145
## Run 85 stress 0.1967847
## Run 86 stress 0.1830781
## Run 87 stress 0.1885397
## Run 88 stress 0.1792427
## Run 89 stress 0.1894345
## Run 90 stress 0.1943004
## Run 91 stress 0.1959765
## Run 92 stress 0.1916804
## Run 93 stress 0.1790446
## Run 94 stress 0.1930767
## Run 95 stress 0.2020236
## Run 96 stress 0.1923937
## Run 97 stress 0.1866619
## Run 98 stress 0.1757793
## ... New best solution
## ... Procrustes: rmse 0.001693843  max resid 0.01223205
## Run 99 stress 0.1759575
## ... Procrustes: rmse 0.007292715  max resid 0.08513722
## Run 100 stress 0.1919335
## Run 101 stress 0.1961309
## Run 102 stress 0.1790366
## Run 103 stress 0.1903173
## Run 104 stress 0.1783425
## Run 105 stress 0.2090502
## Run 106 stress 0.4161178
## Run 107 stress 0.1920216
## Run 108 stress 0.178563
## Run 109 stress 0.1977078
## Run 110 stress 0.1800627

```

```

## Run 111 stress 0.1904304
## Run 112 stress 0.2059046
## Run 113 stress 0.1783447
## Run 114 stress 0.1860653
## Run 115 stress 0.1766072
## Run 116 stress 0.1782865
## Run 117 stress 0.1879055
## Run 118 stress 0.2143067
## Run 119 stress 0.1856557
## Run 120 stress 0.2143696
## Run 121 stress 0.1841323
## Run 122 stress 0.1966262
## Run 123 stress 0.2107032
## Run 124 stress 0.2024043
## Run 125 stress 0.1991685
## Run 126 stress 0.1843545
## Run 127 stress 0.1935549
## Run 128 stress 0.1758982
## ... Procrustes: rmse 0.006693901  max resid 0.08532005
## Run 129 stress 0.1938159
## Run 130 stress 0.1860952
## Run 131 stress 0.1989884
## Run 132 stress 0.1885009
## Run 133 stress 0.1862341
## Run 134 stress 0.1785989
## Run 135 stress 0.2049853
## Run 136 stress 0.1785976
## Run 137 stress 0.2047403
## Run 138 stress 0.2088977
## Run 139 stress 0.1888156
## Run 140 stress 0.1946764
## Run 141 stress 0.1765675
## Run 142 stress 0.1758444
## ... Procrustes: rmse 0.003334519  max resid 0.04317937
## Run 143 stress 0.1768781
## Run 144 stress 0.1987545
## Run 145 stress 0.1788026
## Run 146 stress 0.1779653
## Run 147 stress 0.1758474
## ... Procrustes: rmse 0.003402444  max resid 0.04342179
## Run 148 stress 0.1902268
## Run 149 stress 0.2039546
## Run 150 stress 0.1974484
## Run 151 stress 0.1765873
## Run 152 stress 0.1920247
## Run 153 stress 0.1803372
## Run 154 stress 0.1798656
## Run 155 stress 0.1868677
## Run 156 stress 0.1803026
## Run 157 stress 0.1796587
## Run 158 stress 0.1790663
## Run 159 stress 0.192147
## Run 160 stress 0.1906735
## Run 161 stress 0.1915894

```

```

## Run 162 stress 0.1792629
## Run 163 stress 0.1839238
## Run 164 stress 0.1958771
## Run 165 stress 0.201776
## Run 166 stress 0.1782835
## Run 167 stress 0.1785736
## Run 168 stress 0.2113934
## Run 169 stress 0.1971785
## Run 170 stress 0.1967032
## Run 171 stress 0.214885
## Run 172 stress 0.1758913
## ... Procrustes: rmse 0.006672755  max resid 0.08556301
## Run 173 stress 0.20161
## Run 174 stress 0.198531
## Run 175 stress 0.1950551
## Run 176 stress 0.1941418
## Run 177 stress 0.1782492
## Run 178 stress 0.1768658
## Run 179 stress 0.1783474
## Run 180 stress 0.1973157
## Run 181 stress 0.1857687
## Run 182 stress 0.1802025
## Run 183 stress 0.1801703
## Run 184 stress 0.1961555
## Run 185 stress 0.1919891
## Run 186 stress 0.1856448
## Run 187 stress 0.1971833
## Run 188 stress 0.1812108
## Run 189 stress 0.2149783
## Run 190 stress 0.1758009
## ... Procrustes: rmse 0.001199255  max resid 0.01186763
## Run 191 stress 0.2100291
## Run 192 stress 0.1941543
## Run 193 stress 0.1765786
## Run 194 stress 0.1785918
## Run 195 stress 0.1802696
## Run 196 stress 0.2043005
## Run 197 stress 0.1776268
## Run 198 stress 0.1858989
## Run 199 stress 0.2015495
## Run 200 stress 0.1783944
## Run 201 stress 0.4161477
## Run 202 stress 0.1883485
## Run 203 stress 0.1877381
## Run 204 stress 0.1922341
## Run 205 stress 0.1852034
## Run 206 stress 0.1782428
## Run 207 stress 0.1940847
## Run 208 stress 0.178364
## Run 209 stress 0.1872153
## Run 210 stress 0.2048338
## Run 211 stress 0.2143421
## Run 212 stress 0.1813106
## Run 213 stress 0.1765823

```



```
## Run 214 stress 0.1779442
## Run 215 stress 0.1792325
## Run 216 stress 0.1945764
## Run 217 stress 0.1779703
## Run 218 stress 0.4161317
## Run 219 stress 0.1788418
## Run 220 stress 0.1777212
## Run 221 stress 0.2019454
## Run 222 stress 0.200056
## Run 223 stress 0.1785676
## Run 224 stress 0.1934785
## Run 225 stress 0.1929416
## Run 226 stress 0.189122
## Run 227 stress 0.1867236
## Run 228 stress 0.1944087
## Run 229 stress 0.2089062
## Run 230 stress 0.1882561
## Run 231 stress 0.1859053
## Run 232 stress 0.1944056
## Run 233 stress 0.1864893
## Run 234 stress 0.1892151
## Run 235 stress 0.1850967
## Run 236 stress 0.1994451
## Run 237 stress 0.2068457
## Run 238 stress 0.1883604
## Run 239 stress 0.1963499
## Run 240 stress 0.1779844
## Run 241 stress 0.1942086
## Run 242 stress 0.21075
## Run 243 stress 0.1966056
## Run 244 stress 0.1779927
## Run 245 stress 0.1954545
## Run 246 stress 0.1855553
## Run 247 stress 0.4161297
## Run 248 stress 0.1790394
## Run 249 stress 0.2023709
## Run 250 stress 0.1782992
## Run 251 stress 0.1810201
## Run 252 stress 0.1903253
## Run 253 stress 0.1758363
## ... Procrustes: rmse 0.003480458  max resid 0.04327474
## Run 254 stress 0.1777257
## Run 255 stress 0.1905049
## Run 256 stress 0.1768649
## Run 257 stress 0.1880803
## Run 258 stress 0.1954373
## Run 259 stress 0.1792364
## Run 260 stress 0.1844688
## Run 261 stress 0.178364
## Run 262 stress 0.1797154
## Run 263 stress 0.1783397
## Run 264 stress 0.1765699
## Run 265 stress 0.2055737
## Run 266 stress 0.1779793
```

```

## Run 267 stress 0.192255
## Run 268 stress 0.1788459
## Run 269 stress 0.187432
## Run 270 stress 0.1759812
## ... Procrustes: rmse 0.007315288 max resid 0.08490335
## Run 271 stress 0.178841
## Run 272 stress 0.1785707
## Run 273 stress 0.2157835
## Run 274 stress 0.1942939
## Run 275 stress 0.1759012
## ... Procrustes: rmse 0.006838673 max resid 0.08537501
## Run 276 stress 0.206187
## Run 277 stress 0.1933202
## Run 278 stress 0.2020784
## Run 279 stress 0.1783465
## Run 280 stress 0.1928502
## Run 281 stress 0.1788609
## Run 282 stress 0.1782851
## Run 283 stress 0.1875743
## Run 284 stress 0.1887271
## Run 285 stress 0.1885436
## Run 286 stress 0.1811361
## Run 287 stress 0.1906411
## Run 288 stress 0.1765911
## Run 289 stress 0.1782554
## Run 290 stress 0.200926
## Run 291 stress 0.4161383
## Run 292 stress 0.1757835
## ... Procrustes: rmse 0.0006544988 max resid 0.007231332
## ... Similar to previous best
## *** Solution reached

wUF.g.ordu = ordinate(physeq.tree.g, method="NMDS", distance="unifrac", weighted=TRUE)

## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu04170 --
## in the phylogenetic tree in the data you provided.

## Run 0 stress 0.1110309
## Run 1 stress 0.1112774
## ... Procrustes: rmse 0.003638806 max resid 0.02286034
## Run 2 stress 0.1122669
## Run 3 stress 0.1114875
## ... Procrustes: rmse 0.005288243 max resid 0.02933534
## Run 4 stress 0.1113367
## ... Procrustes: rmse 0.004821078 max resid 0.03221992
## Run 5 stress 0.1118956
## Run 6 stress 0.1120194
## Run 7 stress 0.1114885
## ... Procrustes: rmse 0.01008123 max resid 0.03345407
## Run 8 stress 0.4160518
## Run 9 stress 0.1112226
## ... Procrustes: rmse 0.009058642 max resid 0.03666701
## Run 10 stress 0.1111704
## ... Procrustes: rmse 0.003355195 max resid 0.02518748
## Run 11 stress 0.1112287

```

```
## ... Procrustes: rmse 0.01122786 max resid 0.03989851
## Run 12 stress 0.1115027
## ... Procrustes: rmse 0.005896309 max resid 0.02518071
## Run 13 stress 0.4160153
## Run 14 stress 0.1110765
## ... Procrustes: rmse 0.004865465 max resid 0.02148985
## Run 15 stress 0.1113322
## ... Procrustes: rmse 0.006003153 max resid 0.03272031
## Run 16 stress 0.1112682
## ... Procrustes: rmse 0.006793857 max resid 0.03167451
## Run 17 stress 0.4161426
## Run 18 stress 0.1112634
## ... Procrustes: rmse 0.005501874 max resid 0.03138958
## Run 19 stress 0.1111517
## ... Procrustes: rmse 0.011615 max resid 0.03525381
## Run 20 stress 0.1113179
## ... Procrustes: rmse 0.005913694 max resid 0.03059023
## *** No convergence -- monoMDS stopping criteria:
## 20: stress ratio > sratmax
```

```
#Fit environmental variables
```

```
ef.BC.AB = envfit(BC.g.MDS ~ AB, data=vars.envfit, na.rm=TRUE)
```

```
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
## Warning in cor(Hperm, take): the standard deviation is zero
ef.BC.socY = envfit(BC.g.MDS ~ social.cohabY, data=vars.envfit, na.rm=TRUE)
ef.BC.socN = envfit(BC.g.MDS ~ social.cohabN, data=vars.envfit, na.rm=TRUE)
ef.BC.hbs = envfit(BC.g.MDS ~ hbs, data=vars.envfit, na.rm=TRUE)
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero

```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
ef.BC.heart = envfit(BC.g.MDS ~ heart, data=vars.envfit, na.rm=TRUE)
ef.BC.prot = envfit(BC.g.MDS ~ prot.sum, data=vars.envfit, na.rm=TRUE)
ef.BC.meat = envfit(BC.g.MDS ~ meat, data=vars.envfit, na.rm=TRUE)
ef.BC.poul = envfit(BC.g.MDS ~ poultry, data=vars.envfit, na.rm=TRUE)

ef.J.AB = envfit(J.g.MDS ~ AB, data=vars.envfit, na.rm=TRUE)

## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
ef.J.cohab = envfit(J.g.MDS ~ cohabYN, data=vars.envfit, na.rm=TRUE)
ef.J.socY = envfit(J.g.MDS ~ social.cohabY, data=vars.envfit, na.rm=TRUE)
ef.J.socN = envfit(J.g.MDS ~ social.cohabN, data=vars.envfit, na.rm=TRUE)
ef.J.hbs = envfit(J.g.MDS ~ hbs, data=vars.envfit, na.rm=TRUE)
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero
## Warning in cor(H, Pw): the standard deviation is zero

```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

#Figure S1

```
postscript("FIGURES1.eps", width = 9, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2,3), 1, 3, byrow = TRUE), widths=c(1,1,1), heights=c(1,1,1))
plot(BC.g.MDS, type="n", main="Bray-Curtis", xlab="", ylab="")
  points(BC.g.MDS, pch=21, cex=0.75, display="sites", col="black", bg=c("red","blue")[as.factor(Pheno.s)]
plot(ef.BC.AB, col="red")
```

```
## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
```

```
plot(ef.BC.hbs, col="red")
```

```
## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
```

```
plot(ef.BC.heart, col="red")
plot(ef.BC.prot, col="black")
plot(ef.BC.socY, col="black")
plot(ef.BC.socN, col="black")
plot(ef.BC.meat, col="black")
plot(ef.BC.poul, col="black")
plot(J.g.MDS, type="n", main="Jaccard", xlab="", ylab="")
  points(J.g.MDS, pch=21, cex=0.75, display="sites", col="black", bg=c("red","blue")[as.factor(Pheno.s)]
plot(ef.J.AB, col="red")
```

```
## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
```

```
plot(ef.J.cohab, col="black")
plot(ef.J.socY, col="black")
plot(ef.J.socN, col="black")
plot(ef.J.hbs, col="black")
```

```
## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
```

```

## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
plot(ef.J.heart, col="black")
plot(ef.J.prot, col="black")
plot(ef.J.meat, col="black")
plot(ef.J.poul, col="black")
plot(wUF.g.ordu, type="n", main="Weighted UniFrac", xlab="", ylab="", xlim=c(-0.03,0.03), ylim=c(-0.03,0.03))

## Warning in ordiplot(x, choices = choices, type = type, display = display, :
## Species scores not available
points(wUF.g.ordu, pch=21, cex=0.75, display="sites", col="black", bg=c("red","blue")[as.factor(Pheno
plot(ef.wUF.AB, col="black")

## Warning in min(u): no non-missing arguments to min; returning Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(u): no non-missing arguments to min; returning Inf
plot(ef.wUF.prot, col="black")
plot(ef.wUF.meat, col="black")
plot(ef.wUF.poul, col="black")
dev.off()

## pdf
## 2

```

Figure S2

```

#Re-code variables to YN letters
vars.g$AB = ifelse(vars.g$AB == 1, "Y", ifelse(vars.g$AB == 2, "N", vars.g$AB))
vars.g$hbs11 = ifelse(vars.g$hbs11 == 1, "Y", ifelse(vars.g$hbs11 == 2, "N", vars.g$hbs11))

#Figure S2ABCD
postscript("FIGURES2a.eps", width = 6, height = 6, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE), widths=c(1,1), heights=c(1,1))
#Shannon
boxplot(Met.g$shannon ~ vars.g$AB, ylab="Shannon's diversity", main="", xlab="Antibiotic use last 6mo")
stripchart(Met.g$shannon ~ vars.g$AB, col="black", bg="black", pch=21, vertical=TRUE, method="jitter")
boxplot(Met.g$shannon ~ vars.g$hbs11, main="", xlab="High blood sugar")
stripchart(Met.g$shannon ~ vars.g$hbs11, col="black", bg="black", pch=21, vertical=TRUE, method="jitter")
#Chao
boxplot(Met.g$chao ~ vars.g$RU57, ylab="Chao's richness", main="", xlab="1957")
stripchart(Met.g$chao ~ vars.g$RU57, col="black", bg="black", pch=21, vertical=TRUE, method="jitter")
boxplot(Met.g$chao ~ vars.g$RU11, main="", xlab="2011")

```

```

    stripchart(Met.g$chao ~ vars.g$RU11, col="black", bg="black", pch=21, vertical=TRUE, method="jitter",
dev.off()

## pdf
## 2

#Figure S2E
postscript("FIGURES2b.eps", width = 3, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1), 1, 1, byrow = TRUE), widths=c(1), heights=c(1))
levelplot(Met.g$shannon ~ vars.g$poul*vars.g$meat, col.regions=function(x)rev(gray.colors(x)), xlab="Por
dev.off()

## pdf
## 2

#Figure S2F
postscript("FIGURES2c.eps", width = 3, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1), 1, 1, byrow = TRUE), widths=c(1), heights=c(1))
levelplot(Met.g$shannon ~ vars.g$poul*vars.g$pork, col.regions=function(x)rev(gray.colors(x)), xlab="Por
dev.off()

## pdf
## 2

```

Figure S3

```

#Calculate linear models
shan.social.cohabY = glm(Met.g[vars.g$cohabYN == "Y",]$shannon ~ vars.g[vars.g$cohabYN == "Y",]$social.sum,
shan.social.cohabN = glm(Met.g[vars.g$cohabYN == "N",]$shannon ~ vars.g[vars.g$cohabYN == "N",]$social.sum,
chao.social.cohabY = glm(Met.g[vars.g$cohabYN == "Y",]$chao ~ vars.g[vars.g$cohabYN == "Y",]$social.sum,
chao.social.cohabN = glm(Met.g[vars.g$cohabYN == "N",]$chao ~ vars.g[vars.g$cohabYN == "N",]$social.sum,

#Figure S3
postscript("FIGURES3.eps", width = 6, height = 6, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE), widths=c(1,1), heights=c(1,1))

plot(Met.g$shannon ~ vars.g$social.sum, pch=21, col="black", bg=c("red","blue")[vars.g$cohabYN], ylab="
    abline(shan.social.cohabY, col="blue")
    abline(shan.social.cohabN, col="red")

#Chao
plot(Met.g$chao ~ vars.g$social.sum, pch=21, col="black", bg=c("red","blue")[vars.g$cohabYN], ylab="",
    abline(chao.social.cohabY, col="blue")
    abline(chao.social.cohabN, col="red")
dev.off()

## pdf
## 2

```

Figure S4

```

#Figure S4
postscript("FIGURES4.eps", width = 9, height = 9, horizontal = FALSE, onefile = FALSE, paper = "special")
layout(matrix(c(1,2,3,4,5,6,7,8,9), 3, 3, byrow = TRUE), widths=c(1,1,1), heights=c(1,1,1))

```

```
## pdf
## 2
```

Figure S5

```
postscript("FIGURES5.eps", width = 3, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special",
plot(match.uwUF.gp.se.gs ~ match.prot.gp.se.gs, bg=c(rep("red",94), rep("blue", 83)), col="black", pch=
uwUF.spouse = lm(match.uwUF.gp.se.gs[vars.gp.se.gs$group == "spouse"] ~ match.prot.gp.se.gs[vars.gp.se.gs
uwUF.sib = lm(match.uwUF.gp.se.gs[vars.gp.se.gs$group == "sib"] ~ match.prot.gp.se.gs[vars.gp.se.gs
uwUF.all = lm(match.uwUF.gp.se.gs ~ match.prot.gp.se.gs)
abline(uwUF.spouse, col="red")
abline(uwUF.sib, col="blue")
abline(uwUF.all, col="black")
dev.off()
```

```
## pdf
## 2
```

Figure S6

```
postscript("FIGURES6a.eps", width = 3, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special",
boxplot(match.prot.gp.se.gs~vars.gp.se.gs$group, ylab="Euclidean distances of protein consumption", main="Protein consumption",
dev.off())
```

```
## pdf
## 2
```

```
postscript("FIGURES6b.eps", width = 3, height = 3, horizontal = FALSE, onefile = FALSE, paper = "special")
plot(table(vars.gp.se.gs$group, floor(vars.gp.se.gs$closeness)), ylab="Closeness", main="")
dev.off()
```

pdf
2