Reviewer Questions

XCM & Cecilia Wang

30 May 2018

1. Load and filter primary taxonomy data

```
#Piece of code for loading taxonomy.tsv / modules.pcl / pathways.pcl and calculating bad samples.
# E. coli IQR is calculated for each age group, and  with > 1.5 IQR for each age group are filtered out

NZGL_taxonomy<-import_qiime_sample_data("~/PIP2018/primary_data/taxonomy.tsv")

# the imported taxonomy data should have each sample as a row and each variable or taxonomy as a column
Taxonomy_filter_file<-NZGL_taxonomy # make a copy

NZGL_taxonomy$time<-as.factor(NZGL_taxonomy$time)


# Find the interquartile range for E. coli &filter out samples with E. coli abundance  > 1.5 IQR
Taxonomy_filter_file$E_coli<-NZGL_taxonomy$k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o__Ente
# Split the dataset by time/age
E_coli_abundance_AtBirth<-subset(Taxonomy_filter_file, time==0)
E_coli_abundance_3_month<-subset(Taxonomy_filter_file, time==3)
E_coli_abundance_12_month<-subset(Taxonomy_filter_file, time==12)
E_coli_abundance_24_month<-subset(Taxonomy_filter_file, time==24)

# Calculate IQR by each time
E_coli_abundance_IOR_AtBirth<-IQR(E_coli_abundance_AtBirth$E_coli)
E_coli_abundance_IQR_3_month<-IQR(E_coli_abundance_3_month$E_coli)
E_coli_abundance_IQR_12_month<-IQR(E_coli_abundance_12_month$E_coli)
E_coli_abundance_IQR_24_month<-IQR(E_coli_abundance_24_month$E_coli)

# Filter the whole dataset at each time on E.coli > 1.5IQR
Taxonomy_filtered_AtBirth<-subset(E_coli_abundance_AtBirth, E_coli<=(1.5*E_coli_abundance_IOR_AtBirth))
Taxonomy_filtered_3_month<-subset(E_coli_abundance_3_month, E_coli<=(1.5*E_coli_abundance_IQR_3_month))
Taxonomy_filtered_12_month<-subset(E_coli_abundance_12_month, E_coli<=(1.5*E_coli_abundance_IQR_12_mont
Taxonomy_filtered_24_month<-subset(E_coli_abundance_24_month, E_coli<=(1.5*E_coli_abundance_IQR_24_mont
Taxonomy_filtered<-rbind(Taxonomy_filtered_AtBirth,Taxonomy_filtered_3_month,Taxonomy_filtered_12_month
```

Review question - is b. animalis different in formula vs bf infants?

```
meta<-read.table("~/PIP2018/primary_data/bf.txt", header=TRUE, sep="\t")
Taxonomy_filtered$bf <-meta$bfduration[match(Taxonomy_filtered$Studyid, meta$Studyid)]
Taxonomy_filtered$formula<-meta$ageanyformula[match(Taxonomy_filtered$Studyid, meta$Studyid)]

#Bind only the data we care about to answer this question
gg<-cbind( Taxonomy_filtered$k__Bacteria.p__Actinobacteria.c__Actinobacteria.o__Bifidobacteriales.f__Bi

colnames(gg) = c("B.animalis", "bfduration", "formula", "time", "studygroup")
gg<-as.data.frame(gg)

#Define "formulanow" by comparing age at which formula was first used in weeks (NA=never) to time at wh

gg = within(gg, {
```

```
formulanow = ifelse(formula/4.3 <= as.numeric(as.character(time)), 1, 0)
})

#Summarize formulanow by time & studygroup
ddply(gg,~time + studygroup,summarise, T=length(which(formulanow==1)), F=length(which(formulanow==0)) +
```

```
##    time studygroup  T  F
## 1     0          1  2 12
## 2     0          2  6 52
## 3     0          3 14 52
## 4     3          1  6  3
## 5     3          2 21 30
## 6     3          3 27 36
## 7    12          1 14  2
## 8    12          2 57 10
## 9    12          3 65 11
## 10   24          1  6  1
## 11   24          2 36  7
## 12   24          3 40  8
```

```
#Since all the NAs are equivalent to "No" & power is better if we don't split them, combine them so eit
gg$formulanow<-ifelse(is.na(gg$formula), 0, ifelse(gg$formulanow==1, 1, 0))


# is there a difference in B. animalis x formula exposure, stratified by time + treatment?
gg %>%
  group_by(time, studygroup) %>%
  do(tidy(wilcox.test(B.animalis ~ formulanow, data= .)))
```

```
## Warning in wilcox.test.default(x = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.01828, 0.1139, 0.00948), y = c(0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.05539, 0, 0, 0, 0, 0.01738, 0, 0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0, 0, 0, 56.08798, 0, 1.83176, 0, 0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = 0, y = c(0, 0.17996, 0.79488,
## 10.11564, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0, 0, 0, 0, 0, 0, 0), y = c(0.04515, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0, 0, 0, 0, 0.87543, 0.00816, 0, 0), :
## cannot compute exact p-value with ties
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## # A tibble: 12 x 6
## # Groups:   time, studygroup [12]
##     time studygroup statistic  p.value method                  alternative
##    <dbl>      <dbl>     <dbl>    <dbl> <chr>                   <fct>
##  1     0          1         6   0.0247 Wilcoxon rank sum test~ two.sided
##  2     0          2       156   NaN    Wilcoxon rank sum test~ two.sided
##  3     0          3       364   NaN    Wilcoxon rank sum test~ two.sided
##  4     3          1         8     0.897  Wilcoxon rank sum test~ two.sided
##  5     3          2       333     0.473  Wilcoxon rank sum test~ two.sided
##  6     3          3       526.    0.132  Wilcoxon rank sum test~ two.sided
##  7    12          1         4     0.15   Wilcoxon rank sum test  two.sided
##  8    12          2       320     0.445  Wilcoxon rank sum test~ two.sided
##  9    12          3       421     0.245  Wilcoxon rank sum test~ two.sided
## 10    24          1         0.5   0.313  Wilcoxon rank sum test~ two.sided
## 11    24          2        87.5   0.103  Wilcoxon rank sum test~ two.sided
## 12    24          3       157     0.930  Wilcoxon rank sum test~ two.sided
```

```
# is there a difference in B. animalis x formula exposure, stratified by time?
gg %>%
  group_by(time) %>%
  do(tidy(wilcox.test(B.animalis ~ formulanow, data= .)))
```

```
## # A tibble: 4 x 5
## # Groups:   time [4]
##    time statistic p.value method                                alternative
##   <dbl>     <dbl>   <dbl> <fct>                                 <fct>
## 1     0      1218  0.0228 Wilcoxon rank sum test with continu~ two.sided
## 2     3      1932  0.527  Wilcoxon rank sum test with continu~ two.sided
## 3    12      1730  0.342  Wilcoxon rank sum test with continu~ two.sided
## 4    24       522  0.112  Wilcoxon rank sum test with continu~ two.sided
```

```
#What's mean abundance of B. animalis in each stratified group?
#time only
ddply(gg,~time,summarise,mean=mean(B.animalis))
```

```
##   time         mean
## 1    0 0.0003578986
## 2    3 0.4807531707
## 3   12 0.3015864151
## 4   24 0.2111182653
```

```r
# time x studygroup
ddply(gg,~time + studygroup,summarise,mean=mean(B.animalis))
```

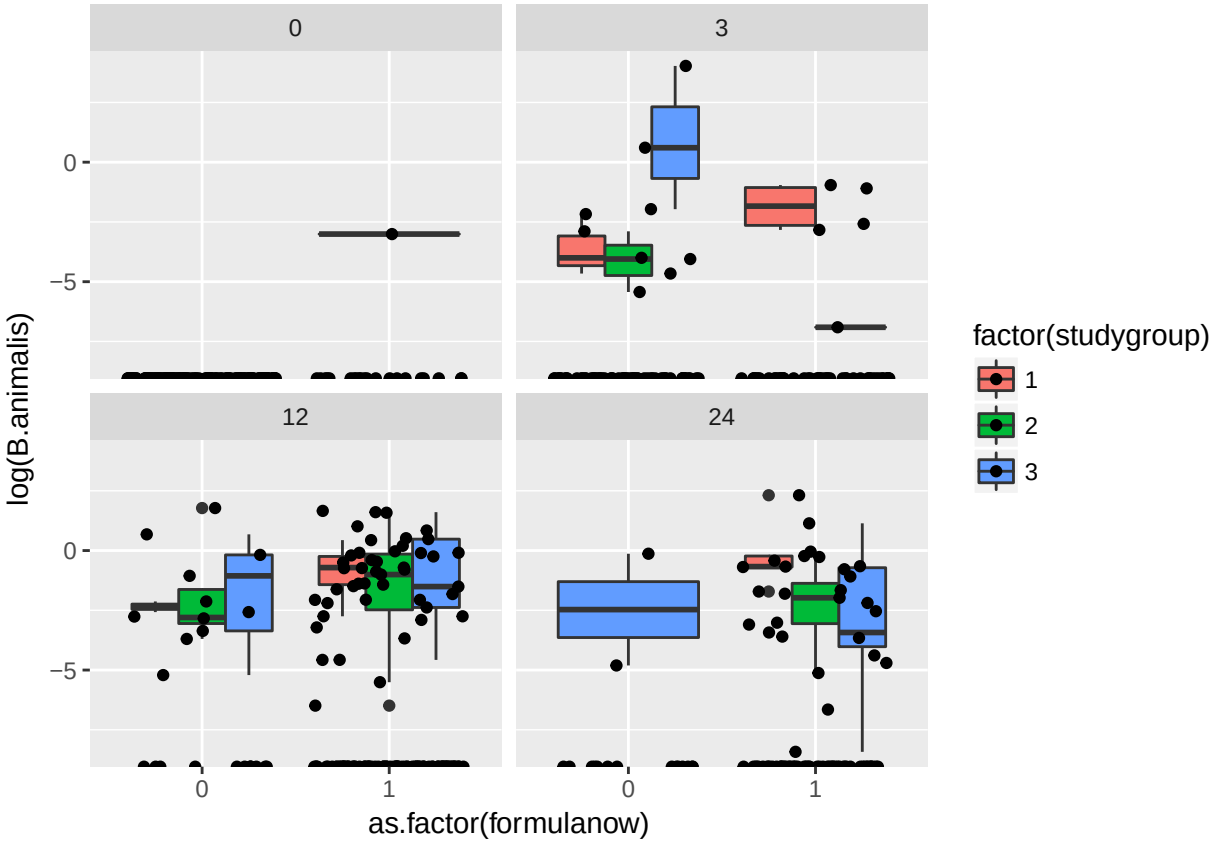
```
##    time studygroup        mean
## 1     0          1 0.003527857
## 2     0          2 0.000000000
## 3     0          3 0.000000000
## 4     3          1 0.110518889
## 5     3          2 0.001532353
## 6     3          3 0.921584444
## 7    12          1 0.493912500
## 8    12          2 0.281427313
## 9    12          3 0.278868553
## 10   24          1 1.728965714
## 11   24          2 0.061924186
## 12   24          3 0.123418542
```

```r
#time x studygroup x formula use
ddply(gg,~time + studygroup + formulanow,summarise,mean=mean(B.animalis))
```

```
##    time studygroup formulanow         mean
## 1     0          1          0 0.000000e+00
## 2     0          1          1 2.469500e-02
## 3     0          2          0 0.000000e+00
## 4     0          2          1 0.000000e+00
## 5     0          3          0 0.000000e+00
## 6     0          3          1 0.000000e+00
## 7     3          1          0 4.722000e-02
## 8     3          1          1 1.421683e-01
## 9     3          2          0 2.571667e-03
## 10    3          2          1 4.761905e-05
## 11    3          3          0 1.612773e+00
## 12    3          3          1 0.000000e+00
## 13   12          1          0 9.748000e-02
## 14   12          1          1 5.505457e-01
## 15   12          2          0 6.061770e-01
## 16   12          2          1 2.244537e-01
## 17   12          3          0 2.899473e-01
## 18   12          3          1 2.769937e-01
## 19   24          1          0 0.000000e+00
## 20   24          1          1 2.017127e+00
## 21   24          2          0 0.000000e+00
## 22   24          2          1 7.396500e-02
## 23   24          3          0 1.104488e-01
## 24   24          3          1 1.260125e-01
```

```r
ggplot(gg, aes(x=as.factor(formulanow), y=log(B.animalis), fill=factor(studygroup))) + geom_boxplot() +
```

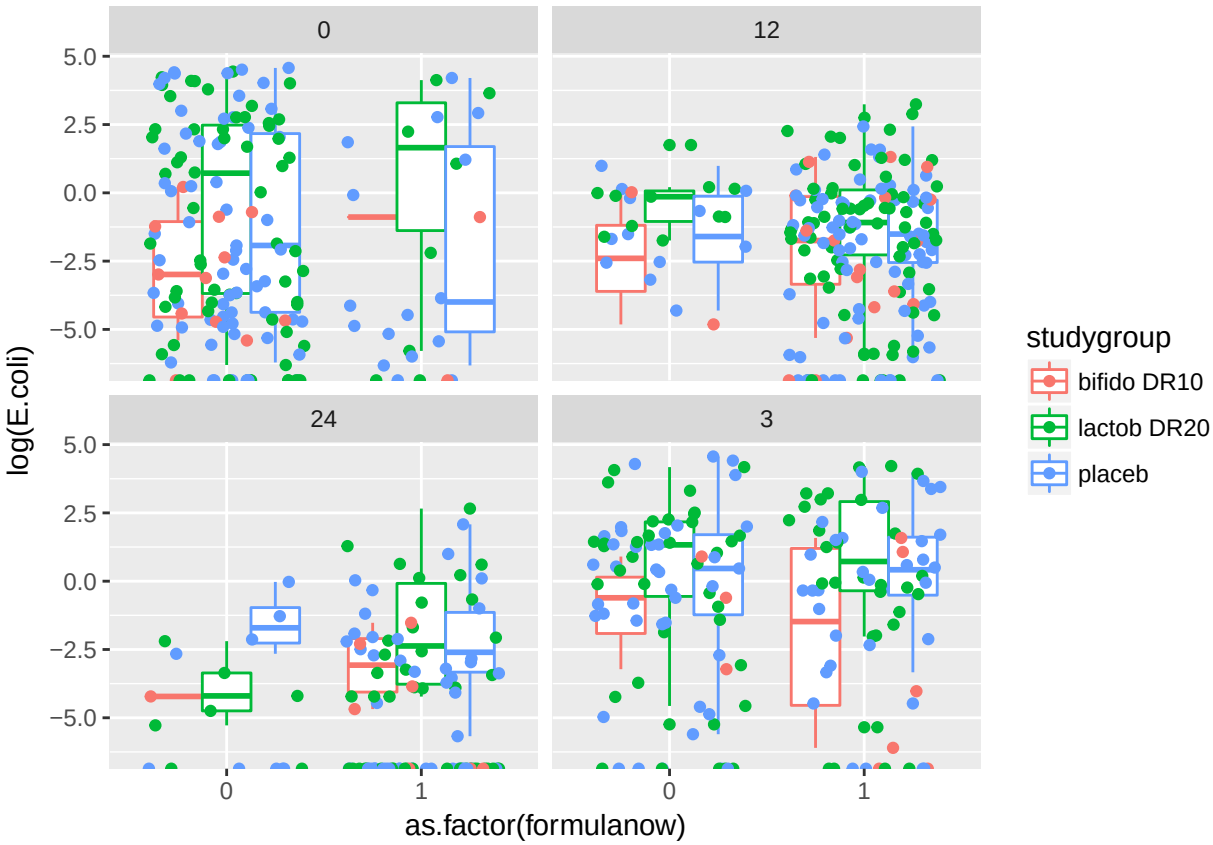
```
## Warning: Removed 418 rows containing non-finite values (stat_boxplot).
```



Reviewer question - does feeding influence E. coli? Using unfiltered data

```
#analyze only data we're interested in
foo<-cbind(NZGL_taxonomy$k__Bacteria.p__Proteobacteria.c__Gammaproteobacteria.o__Enterobacteriales.f__E
foo<-as.data.frame(foo)
foo$V1 = as.numeric(as.character(foo$V1))
colnames(foo) = c("E.coli", "time", "studygroup", "Studyid")

#add bf + formula data
foo$bf <-meta$bfduration[match(foo$Studyid, meta$Studyid)]
foo$formula<-meta$ageanyformula[match(foo$Studyid, meta$Studyid)]

#Examine the subset for which bf metadata isn't missing (32 samples)
bar<-subset(foo, !is.na(foo$bf))

#make formulanow variable
bar = within(bar, {formulanow = ifelse(is.na(formula), 0, ifelse(formula/4.3 <= as.numeric(as.character

# plot E. coli abundance stratified by formula use, studygroup, and time
ggplot(bar, aes(x=as.factor(formulanow), y=log(E.coli), colour=studygroup)) + geom_boxplot() + geom_jit
```

```
## Warning: Removed 121 rows containing non-finite values (stat_boxplot).
```




```
#is there difference in e. coli x formula stratified by time & studygroup?
bar %>%
  group_by(time, studygroup) %>%
  do(tidy(wilcox.test(E.coli ~ formulanow, data= .)))
```

```
## Warning in wilcox.test.default(x = c(0.09421, 0.00931, 0.41609, 0.05039, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(1.02187, 0.00809), y = c(0.017, 0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = 0.01471, y = c(0.21811, 0, 0.10072, 0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0, 0.01499, 0.11127, 0, 0.0051,
## 0.00867, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.278, 0.11836, 0, 0, 0.97401, 0, 0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(0.54446, 2.4653, 0.03987), y = c(0, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(4.30935, 0.64904, 2.82009, 8.94504, :
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(x = c(1.70607, 0, 0.23618, 48.79095,
## 5.19235, : cannot compute exact p-value with ties
```

```
## # A tibble: 12 x 6
```

```
## # Groups:   time, studygroup [12]
##    time  studygroup  statistic p.value method                  alternative
##    <fct> <fct>           <dbl>   <dbl> <fct>                   <fct>
##  1 0     bifido DR10      14.5   0.715 Wilcoxon rank sum test~ two.sided
##  2 0     lactob DR20     248     0.893 Wilcoxon rank sum test~ two.sided
##  3 0     placeb          607     0.304 Wilcoxon rank sum test~ two.sided
##  4 12    bifido DR10      17     1     Wilcoxon rank sum test~ two.sided
##  5 12    lactob DR20     496     0.105 Wilcoxon rank sum test~ two.sided
##  6 12    placeb          503     0.367 Wilcoxon rank sum test~ two.sided
##  7 24    bifido DR10       4     1     Wilcoxon rank sum test~ two.sided
##  8 24    lactob DR20     159     0.977 Wilcoxon rank sum test~ two.sided
##  9 24    placeb          212.    0.990 Wilcoxon rank sum test~ two.sided
## 10 3     bifido DR10      12     0.517 Wilcoxon rank sum test~ two.sided
## 11 3     lactob DR20     531     0.767 Wilcoxon rank sum test~ two.sided
## 12 3     placeb          628     0.896 Wilcoxon rank sum test~ two.sided
```

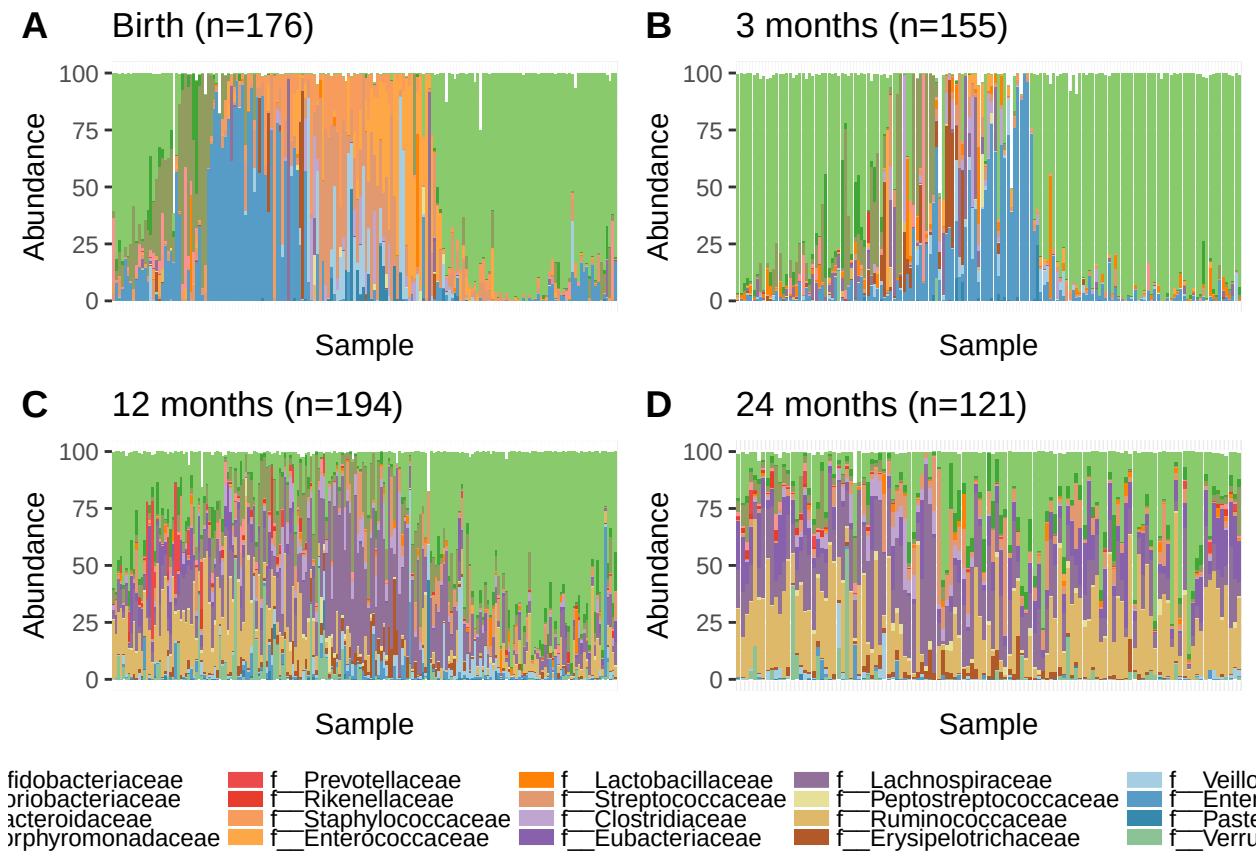
```
# is there a difference in Ecoli x formula exposure, stratified by time?
bar %>%
  group_by(time) %>%
  do(tidy(wilcox.test(E.coli ~ formulanow, data= .)))
```

```
## # A tibble: 4 x 5
## # Groups:   time [4]
##   time  statistic p.value method                                  alternative
##   <fct>     <dbl>   <dbl> <fct>                                   <fct>
## 1 0         2020.  0.411  Wilcoxon rank sum test with continu~ two.sided
## 2 12        2413   0.0753 Wilcoxon rank sum test with continu~ two.sided
## 3 24         846.  0.977  Wilcoxon rank sum test with continu~ two.sided
## 4 3         2726   0.983  Wilcoxon rank sum test with continu~ two.sided
```

Make barplot fig

```
#send to physeq
Taxonomy_filtered_num<-NZGL_taxonomy[,c(-1:-28)]
meta<-NZGL_taxonomy[,1:28]
phy<-otu_table(t(Taxonomy_filtered_num), taxa_are_rows = TRUE)
a<-rownames(phy)
b<-str_split_fixed(a, "\\.", 8)
c<-tax_table(b)
colnames(c) = c("domain", "phylum", "class", "order", "family", "genus", "species", "strain")
rownames(phy) = rownames(c)
ps<-phyloseq(phy, c)
sample_data(ps) = meta
# keep only genus level defined taxa
g<-which(tax_table(ps)[,6] != "")
h<-rownames(tax_table(ps))[g]
i<-prune_taxa(h, ps)

#ignore species & strain
j<-which(tax_table(i)[,7] == "")
k<-rownames(tax_table(i))[j]
l<-prune_taxa(k, ps)

#glom to family
l_class <- l %>%
  tax_glom(taxrank = "family")
```

```
tax_table(l_class) = tax_table(l_class)[,1:5]
#get the 20 most abundant families
F20 = names(sort(taxa_sums(l_class), TRUE)[1:20])
pruned = prune_taxa(F20, l_class)

# design a less-disgusting color palette
Mycolors <- colorRampPalette(brewer.pal(12, "Paired"))(20)
names(Mycolors) <- levels(as.factor(rownames(otu_table(pruned))))


pseq <- subset_samples(pruned, time==0)
a<-plot_composition(pseq, x.label="time", plot.type="barplot", sample.sort="neatmap")
first<-a + scale_fill_manual( values=Mycolors, labels=tax_table(pseq)[,5]) + ggtitle("Birth (n=176)") +
theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())


pseq <- subset_samples(pruned, time==3)
a<-plot_composition(pseq, x.label="time", plot.type="barplot", sample.sort="neatmap")
second<-a + scale_fill_manual( values=Mycolors, labels=tax_table(pseq)[,5]) + ggtitle("3 months (n=155)
theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())


pseq <- subset_samples(pruned, time==12)
a<-plot_composition(pseq, x.label="time", plot.type="barplot", sample.sort="neatmap")
third<-a + scale_fill_manual( values=Mycolors, labels=tax_table(pseq)[,5]) + ggtitle("12 months (n=194)
theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())


pseq <- subset_samples(pruned, time==24)
a<-plot_composition(pseq, x.label="time", plot.type="barplot", sample.sort="neatmap")
fourth<-a + scale_fill_manual( values=Mycolors, labels=tax_table(pseq)[,5]) + ggtitle("24 months (n=121
theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())

ggarrange(first, second, third, fourth, labels=c("A", "B", "C", "D"), ncol=2, nrow=2, common.legend=TRU
```




```
ggsave("~/PIP2018/results/stupid_barplots.pdf", plot = last_plot(), device = NULL, path = NULL,
  scale = 1, width = 9, height = 6, units = c("in", "cm", "mm"),
  dpi = 300, limitsize = FALSE)
```