MA-GenTA Analysis - Downstream Analysis

Jacquelynn Benjamino

August 3, 2020

This code is for the downstream analysis of the MA-GenTA assay. All code is split by figures. The JAX and Allegro probe sets are designated throughout as V4 and V2, respectively.

Load packages used for this analysis

```
library(dplyr)
library(tibble)
library(ggplot2)
library(ggpubr)
library(reshape2)
library(ggpubr)
library(phyloseq)
library(patchwork)
library(VennDiagram)
library(RColorBrewer)
library(vegan)
```

Figure 2

Figure 2a

Import count tables

```
V2_mapping<-read.csv("V2_controls.csv", header = TRUE)
V4_mapping<-read.csv("V4_controls.csv", header = TRUE)
```

Convert to percent abundance

```
V2_meta<-V2_mapping[,1:2]
V2_counts<-V2_mapping[,3:12]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mapping_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_mapping[,1:2]
V4_counts<-V4_mapping[,3:12]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mapping_prop<-cbind(V4_meta,V4_prop)
```

***E. coli* plot**
Select the *E. coli* samples from the dataframe and filter probes with different percent abundance thresholds: 0.001%, 0.00025%, 0.0005%, 0.001%, 0.0025%, 0.005%.

```
V2_0001<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>% select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_00025<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
          group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))
%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025")%>%
          add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0005<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005")%>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_001<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>%
        summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0025<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="Allegro")

V2_005<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>% add_count(set, name="number_of_mags") %>%
select(set,number_of_mags) %>% mutate(design="Allegro")

V4_0001<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>
%
         select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="JAX")

V4_00025<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
          group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>%
          mutate(abundance=sum(probe_mean))%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025") %>%
          add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% distinct() %>% mutate(design="JAX")

V4_0005<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="JAX")

V4_001<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")

V4_0025<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>
%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="JAX")

V4_005<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")
```

Combine the different probe threshold data into a single plot for each probe set, and then combine the two probe sets into one dataframe.

```
V2_all<-rbind(V2_0001,V2_00025,V2_0005,V2_001,V2_0025,V2_005) %>% distinct()
V4_all<-rbind(V4_0001,V4_00025,V4_0005,V4_001,V4_0025,V4_005) %>% distinct()

all<-rbind(V2_all,V4_all)
all$set<-factor(all$set, levels = c("0.0001","0.00025","0.0005","0.001","0.0025","0.005"))
```

Plot the graph

```
theme_set(theme_bw())
ggplot(all, aes(x=set, y=number_of_mags, fill=design))+
  geom_dotplot(binaxis='y', stackdir='center', position="dodge",  dotsize = 1.3, stackratio = .7)+
  scale_fill_manual(values=c("#274b69","#94ae3f"))+
  xlab("Abundance Threshold")+
  ylab("Number of MAGs")+
  theme(legend.position = "none")+
  ylim(0,15)
```

Mock plot

Select the Mock samples from the dataframe and filter probes with different percent abundance thresholds: 0.001%, 0.00025%, 0.0005%, 0.001%, 0.0025%, 0.005%.

```
V2_0001<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>
%
         select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>% add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_00025<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
          group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))
%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025")%>% add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0005<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005")%>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_001<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001") %>%
        add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0025<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_005<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
V2_005<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="Allegro")

V4_0001<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>
%
         select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="JAX")

V4_00025<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
          group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>%
          mutate(abundance=sum(probe_mean))%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025")%>%
          add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% mutate(design="JAX")

V4_0005<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005")%>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="JAX")

V4_001<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")

V4_0025<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>% mutate(abundance=sum(probe_mean))%>
%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="JAX")

V4_005<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>%
        add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")
```

Combine the different probe threshold data into a single plot for each probe set, and then combine the two probe sets into one dataframe.

```
V2_all<-rbind(V2_0001,V2_00025,V2_0005,V2_001,V2_0025,V2_005) %>% distinct()
V4_all<-rbind(V4_0001,V4_00025,V4_0005,V4_001,V4_0025,V4_005) %>% distinct()

all<-rbind(V2_all,V4_all)
all$set<-factor(all$set, levels = c("0.0001","0.00025","0.0005","0.001","0.0025","0.005"))
```

Plot the graph

```
theme_set(theme_bw())
ggplot(all, aes(x=set, y=number_of_mags, fill=design))+
  geom_dotplot(binaxis='y', stackdir='center', position="dodge", dotsize = 1.3, stackratio = .7)+
  scale_fill_manual(values=c("#274b69","#94ae3f"))+
  xlab("Abundance Threshold")+
  ylab("Number of MAGs")+
  theme(legend.position = "none")+
  ylim(0,15)
```

NTC plot

Select the NTC samples from the dataframe and filter probes with different percent abundance thresholds: 0.001%, 0.00025%, 0.0005%, 0.001%, 0.0025%, 0.005%.

```
V2_0001<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>% melt() %>
%
         group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         add_count(set, name="number_of_mags") %>% select(set,number_of_mags) %>%
         mutate(design="Allegro")

V2_00025<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
          melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>%
          mutate(abundance=sum(probe_mean))%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025")%>%
          add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0005<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
         melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005")%>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_001<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="Allegro")

V2_0025<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
         melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="Allegro")

V2_005<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>% add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="Allegro")

V4_0001<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
         melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0001, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>% distinct()%>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         ungroup %>% filter(ppM>=10) %>% mutate(set="0.0001") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="JAX")

V4_00025<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
          melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
          add_tally(probe_mean>=0.00025, name="ppM") %>% group_by(Bin) %>%
          mutate(abundance=sum(probe_mean))%>%
          select(Bin,ppM,abundance)%>%distinct()%>%
          ungroup %>% filter(ppM>=10)%>% mutate(set="0.00025")%>%
          add_count(set, name="number_of_mags") %>%
          select(set,number_of_mags) %>% mutate(design="JAX")

V4_0005<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
         melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0005, name="ppM") %>%group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0005")%>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="JAX")

V4_001<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001, name="ppM") %>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.001")%>%
        add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")

V4_0025<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
         melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
         add_tally(probe_mean>=0.0025, name="ppM") %>% group_by(Bin) %>%
         mutate(abundance=sum(probe_mean))%>%
         select(Bin,ppM,abundance)%>%distinct()%>%
         ungroup %>% filter(ppM>=10)%>% mutate(set="0.0025") %>%
         add_count(set, name="number_of_mags") %>%
         select(set,number_of_mags) %>% mutate(design="JAX")

V4_005<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.005, name="ppM")%>% group_by(Bin) %>%
        mutate(abundance=sum(probe_mean))%>%
        select(Bin,ppM,abundance)%>%distinct()%>%
        ungroup %>% filter(ppM>=10)%>% mutate(set="0.005")%>%
        add_count(set, name="number_of_mags") %>%
        select(set,number_of_mags) %>% mutate(design="JAX")
```

Combine the different probe threshold data into a single plot for each probe set, and then combine the two probe sets into one dataframe.

```
V2_all<-rbind(V2_0001,V2_00025,V2_0005,V2_001,V2_0025,V2_005) %>% distinct()
V4_all<-rbind(V4_0001,V4_00025,V4_0005,V4_001,V4_0025,V4_005) %>% distinct()

all<-rbind(V2_all,V4_all)
all$set<-factor(all$set, levels = c("0.0001","0.00025","0.0005","0.001","0.0025","0.005"))

#had to export and edit names and re-upload because V2 has 0-values and was removed from the dataframe
#write.csv(all,"all.csv")
all<-read.csv("all.csv", header = TRUE,row.names = 1)
all$set<-factor(all$set, levels = c("above0.0001","above0.00025","above0.0005","above0.001","above0.0025",
                                    "above0.005"))
```

Plot the graph

```
theme_set(theme_bw())
ggplot(all, aes(x=set, y=number_of_mags, fill=design))+
  geom_dotplot(binaxis='y', stackdir='center', position="dodge", dotsize = 1.3, stackratio = .7)+
  scale_fill_manual(values=c("#274b69","#94ae3f"))+
  xlab("Abundance Threshold")+
  ylab("Number of MAGs")+
  scale_y_continuous(breaks = c(0,1))+
  theme(legend.position = "none")
```

Figure 2b

Import count tables

```
V2_mapping<-read.csv("V2_controls.csv", header = TRUE)
V4_mapping<-read.csv("V4_controls.csv", header = TRUE)
```

convert to percent abundance

```
V2_meta<-V2_mapping[,1:2]
V2_counts<-V2_mapping[,3:12]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mapping_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_mapping[,1:2]
V4_counts<-V4_mapping[,3:12]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mapping_prop<-cbind(V4_meta,V4_prop)
```

NTC Plot
Select the NTC samples and filter using no thresholds, combine V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point()+
  scale_color_manual(values=c("gray","gray"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

Select the NTC samples and filter using 0.001% probe abundance threshold, combine the V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V2,JNC000.NTC_HL44_P1V2) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,JNC000.NTC_CCF_VNDR_KOMP_P2V4,JNC000.NTC_HL44_P1V4) %>%
        melt() %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point(color="black")+
  scale_color_manual(values=c("#274b69","#94ae3f"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

Mock Plot

Select the Mock samples and filter using no thresholds, combine V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point()+
  scale_color_manual(values=c("gray","gray"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

Select the Mock samples and filter using a 0.001% probe abundance threshold, combine the V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,J00YQD.zymo_mock_P1V2,J00YT0.zymo_mock_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,J00YP2.mock_P1V4,J00YRP.zymo_mock_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point()+
  scale_color_manual(values=c("#274b69","#94ae3f"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

E. coli Plot

Select the *E. coli* samples and filter using no thresholds, combine the V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>0) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point()+
  scale_color_manual(values=c("gray","gray"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

Select the *E. coli* samples and filter using a 0.001% probe abundance thresholds, combine the V2 and V4 tables, and plot

```
V2_all<-select(V2_mapping_prop, Bin,Probe,J00YQB.Ecoli_P1V2, J00YSX.ecoli_P2V2) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean)) %>% mutate(set="V2")
        colnames(V2_all)<-c("Bin","probe_counts","abund","set")

V4_all<-select(V4_mapping_prop, Bin,Probe,J00YP0.ecoli_P1V4,J00YRM.ecoli_P2V4) %>% melt() %>%
        group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))%>% mutate(set="V4")
        colnames(V4_all)<-c("Bin","probe_counts","abund","set")

combined_all<-rbind(V2_all,V4_all)
combined_all[is.na(combined_all)]<-0

ggplot(combined_all, aes(x=abund, y=probe_counts, color=set))+
  geom_point()+
  scale_color_manual(values=c("#274b69","#94ae3f"))+
  scale_x_continuous(trans="log10", name = "MAG Abundance (%)")+
  scale_y_continuous(name = "Probes per MAG")+
  geom_hline(yintercept = 10, color="#777777")+
  geom_count()+scale_size(trans = "log2", name = element_text("Count of Bins"))+
  expand_limits(y=c(0,20))+
  theme(legend.position = "none")
```

Figure 2c

Import mapping stats

```
BWA_mapping_stats<-read.csv("Mapping_stats_95.5_50plot.csv", header = TRUE)
mouse_only_stats<- BWA_mapping_stats %>% filter(!Sample %in% c("ecoli_P1V4",
                   "human_stool_P1V4","mock_P1V4",
                   "salmon_sperm_P1V4","Ecoli_P1V2","human_stool_P1V2",
                   "zymo_mock_P1V2","salmon_sperm_P1V2","salmon_sperm_P2V4",
                   "human_stool_P2V4","zymo_mock_P2V4","ecoli_P2V2",
                   "salmon_sperm_P2V2","human_stool_P2V2","zymo_mock_P2V2",
                   "NTC_HL44_P1V2","NTC_HL44_P1V4","NTC_CCF_VNDR_KOMP_P2V2",
                   "NTC_CCF_VNDR_KOMP_P2V4","ecoli_P2V4"))
```

Combine mapping stats for facets

```
melted<-melt(mouse_only_stats)

totalreads<- melted %>% filter(variable=="Total.Reads") %>% mutate(stat="Total") %>%
             mutate(other="Number of Reads")
mappedreads<- melted %>% filter(variable=="Mapped.Reads") %>% mutate(stat="Mapped")%>%
              mutate(other="Number of Reads")
uniquereads<- melted %>% filter(variable=="Uniquely.Mapped.Reads") %>%
              mutate(stat="Uniquely Mapped") %>%
              mutate(other="Number of Reads")
ontargetreads<- melted %>% filter(variable=="On.Target.Reads") %>% mutate(stat="On-Target")%>%
                mutate(other="Number of Reads")
mappedpercent<- melted %>% filter(variable=="Percent.Mapped.Reads") %>% mutate(stat="Mapped")%>%
                mutate(other="Fraction of Reads")
uniquepercent<- melted %>% filter(variable=="Percent.Uniquely.Mapped.Reads") %>%
                mutate(stat="Uniquely Mapped") %>% mutate(other="Fraction of Reads")
ontargetpercent<- melted %>% filter(variable=="Percent.On.Target.Reads") %>%
                  mutate(stat="On-Target") %>%
                  mutate(other="Fraction of Reads")

combined<-rbind(totalreads,mappedreads,uniquereads,ontargetreads,mappedpercent,uniquepercent,
          ontargetpercent)
combined$other<-factor(combined$other, levels = c("Number of Reads","Fraction of Reads"))
combined$stat<-factor(combined$stat, levels = c("Total","Mapped","Uniquely Mapped","On-Target"))
```

Plot data

```r
theme_set(theme_bw())
ggplot(combined, aes(x=stat, y=value, fill=stat)) +
  geom_boxplot(alpha=0.9)+
  scale_fill_manual(values = c("#6d6e41","#972426","#A9845C","#3B7277"))+
  xlab(element_blank())+
  ylab(element_blank())+
  theme(legend.position = "bottom")+
  theme(legend.title = element_blank())+
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())+
  facet_grid(other~Assay, scales = "free")+
  theme(strip.background = element_blank(), strip.text = element_blank())
```

Figure 2d

Import count tables

```r
V4_HLB<-read.csv("V4_mouse_nameonly.csv", header = TRUE)
V2_HLB<-read.csv("V2_mouse_nameonly.csv", header = TRUE)
```

Convert to percent abundance

```r
V2_meta<-V2_HLB[,1:2]
V2_counts<-V2_HLB[,3:79]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mouse_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_HLB[,1:2]
V4_counts<-V4_HLB[,3:79]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mouse_prop<-cbind(V4_meta,V4_prop)
```

Filter dataframes for no threshold and 0.001% probe abundance and 10ppM thresholds, combine V2 and V4 tables

```
test_melt_V4<-melt(V4_mouse_prop) %>% as_tibble
probe_bin_counts_V4<- test_melt_V4 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V4_0<- probe_bin_counts_V4 %>% group_by(Bin,variable) %>%
                summarize(abundance=mean(Bin_abund)) %>% ungroup %>% group_by(variable) %>%
                add_tally(abundance>0, name="MAGS") %>%
                mutate(set="None") %>% mutate(design="JAX")

filtered_V4_001_10<- probe_bin_counts_V4 %>% filter(probes_per_bin>=10) %>% filter(value>=0.001)
                     %>% group_by(Bin,variable) %>%
                     summarize(abundance=mean(Bin_abund))%>% ungroup %>% group_by(variable)%>%
                     add_tally(abundance>0, name="MAGS") %>%mutate(set="Thresh")%>%
                     mutate(design="JAX")

test_melt_V2<-melt(V2_mouse_prop) %>% as_tibble
probe_bin_counts_V2<- test_melt_V2 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))
filtered_V2_0<- probe_bin_counts_V2 %>% group_by(Bin,variable) %>%
                summarize(abundance=mean(Bin_abund)) %>% ungroup %>% group_by(variable) %>%
                add_tally(abundance>0,
                name="MAGS") %>% mutate(set="None")%>%
                mutate(design="Allegro")

filtered_V2_001_10<- probe_bin_counts_V2 %>% filter(probes_per_bin>=10) %>% filter(value>=0.001)
                     %>%group_by(Bin,variable) %>%
                     summarize(abundance=mean(Bin_abund))%>% ungroup %>% group_by(variable)%>%
                     add_tally(abundance>0, name="MAGS") %>%mutate(set="Thresh")%>%
                     mutate(design="Allegro")

combined<-rbind(filtered_V2_0,filtered_V4_0,filtered_V2_001_10,filtered_V4_001_10)

ggplot(combined, aes(x=set, y=MAGS, fill=design, color=design))+
  geom_boxplot()+
  scale_fill_manual(values=c("#597387","#94ae3f","#bfc0bd","#A9845C"))+
  scale_color_manual(values=c("#364450","#536222","#818181","#6D583F"))+
  xlab("Thresholds")+
  ylab("Number of MAGs")+
  theme(legend.position = "none")
```

Figure 2e

Select the percent of reads present above and below the applied thresholds, and plot

```
percents<- combined %>% filter(set=="Thresh") %>% group_by(variable,design) %>%
           mutate(above=sum(abundance)) %>%
           mutate(below=100-above) %>% select(variable,design,above,below) %>% distinct()
percents_melted<-melt(percents, by=variable)
colnames(percents_melted)<-c("sample","design","set","reads")

theme_set(theme_bw())
ggplot(percents_melted, aes(x=set, y=reads, fill=design, color=design))+
  geom_boxplot()+
  scale_fill_manual(values=c("#597387","#94ae3f","#bfc0bd","#A9845C"))+
  scale_color_manual(values=c("#364450","#536222","#818181","#6D583F"))+
  xlab("Thresholds")+
  ylab("Percent of Reads")+
  theme(legend.position = "none")+
  ylim(0,100)
```

Figure 3

Figure 3a and 3b

Import count tables

```
V2_mouse_strains<-read.csv("V2_mouse_nameonly.csv", header = TRUE, check.names = FALSE)
V4_mouse_strains<-read.csv("V4_mouse_nameonly.csv", header = TRUE, check.names = FALSE)
```

Convert to percent abundance

```
V2_meta<-V2_mouse_strains[,1:2]
V2_counts<-V2_mouse_strains[,3:79]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mouse_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_mouse_strains[,1:2]
V4_counts<-V4_mouse_strains[,3:79]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mouse_prop<-cbind(V4_meta,V4_prop)
```

Melt into tibbles

```
V2_melt<-melt(V2_mouse_prop) %>% as_tibble
V4_melt<-melt(V4_mouse_prop) %>% as_tibble
```

Add 0.001% probe abundance threshold

```
V2_all<-V2_melt %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))
        colnames(V2_all)<-c("Bin","V2_probe_counts","V2_abund")

V4_all<-V4_melt %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
        add_tally(probe_mean>=0.001) %>% ungroup %>%
        group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))
        colnames(V4_all)<-c("Bin","V4_probe_counts","V4_abund")
```

Make a column designating the 10ppM threshold

```
V2_all_aboevelow10<- V2_all %>% mutate(ten=V2_probe_counts>=10)
V4_all_abovebelow10<- V4_all %>% mutate((ten=V4_probe_counts>=10))
combined_all<-V2_all_aboevelow10 %>% left_join(V4_all_abovebelow10, by=c('Bin'))
combined_all[is.na(combined_all)]<-0
colnames(combined_all)<-c("Bin","V2_probe_counts","V2_abund","V2_ten","V4_probe_counts","V4_abund","V4_
                        ten")
combined_all<- combined_all %>% mutate(same= V2_ten==TRUE & V4_ten==TRUE)
```

Plot the graphs

```
ggplot(combined_all, aes(x=V2_abund, y=V4_abund, color=same))+
  geom_point(size=.5)+
  scale_color_manual(values = c("light gray","#274b69"))+
  scale_x_continuous(trans = 'log10', name = "Allegro MAG Abundance")+
  scale_y_continuous(trans = 'log10', name = "JAX MAG Abundance")+
  theme(legend.position = "none")

ggplot(combined_all, aes(x=V2_probe_counts, y=V4_probe_counts, color=same))+
  geom_point()+
  scale_color_manual(values = c("light gray","#274b69"))+
  scale_x_continuous(name = "Allegro Probes per MAG")+
  scale_y_continuous(name = "JAX Probes per MAG")+
  #geom_hline(yintercept = 10, color="red")+
  #geom_vline(xintercept = 10, color="red")+
  geom_count()+scale_size(trans = "log2", range=c(0,5), name = element_text("Count of Bins\n      (Log2)"))+
  theme(legend.position ="none")
```

Pearson Correlation for 3a

```
for_corr<- combined_all %>% filter(same == TRUE) %>% select(V2_abund,V4_abund)
ggscatter(for_corr, x = "V2_abund", y = "V4_abund", size=.5,
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson", cor.coef.size = 3, conf.int.level =0.95,
          xlab = "V2 Abundance", ylab = "V4 Abundance")
```

Figure 3c

Import and adjust metagenome data

```
metag<-read.csv("mwgs_HLB_CCF_new_mapping.csv",check.names = FALSE)
Bin<-metag[,1:1]
meta_prop<-metag[,2:71]
meta_means<-as.data.frame(rowMeans(meta_prop, dims=1))
meta_prop_sum<-cbind(Bin,meta_means)
colnames(meta_prop_sum)<-c("Bin","Meta_abund")
```

Import MA-GenTA count tables

```
V2_mapping<-read.csv("V2_HLB_CCF.csv", header = TRUE, check.names = FALSE)
V4_mapping<-read.csv("V4_HLB_CCF.csv", header = TRUE, check.names = FALSE)
```

Convert to percent abundance

```
V2_meta<-V2_mapping[,1:2]
V2_counts<-V2_mapping[,3:70]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mapping_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_mapping[,1:2]
V4_counts<-V4_mapping[,3:71]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mapping_prop<-cbind(V4_meta,V4_prop)
```

Make tibbles

```
tb2corr<-melt(V2_mapping_prop) %>% as_tibble
tb4corr<-melt(V4_mapping_prop) %>% as_tibble
```

Calculate MAG abundance counts for MA-GenTA data

```
tbv2_bin_sums_counts<-tb2corr %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
                      add_tally(probe_mean>0) %>% ungroup %>%
                      group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))
colnames(tbv2_bin_sums_counts)<-c("Bin","V2_probe_counts","V2_abund")


tbv4_bin_sums_counts<-tb4corr %>% group_by(Bin,Probe) %>% summarize(probe_mean=mean(value)) %>%
                      add_tally(probe_mean>0) %>% ungroup %>%
                      group_by(Bin,n) %>% summarize(BinSum=sum(probe_mean))
colnames(tbv4_bin_sums_counts)<-c("Bin","V4_probe_counts","V4_abund")
```

Combine MA-GenTA and mWGS data and convert NA values to 0

```
combined_corr<- meta_prop_sum %>% left_join(tbv2_bin_sums_counts, by=c('Bin')) %>%
                left_join(tbv4_bin_sums_counts, by=c('Bin'))
combined_corr[is.na(combined_corr)]<-0
```

Calculate correlation values

```
combined_table<-combined_corr %>% group_by(V2_probe_counts) %>%
                mutate(V2_corr=cor(Meta_abund,V2_abund)) %>%
                add_count(V2_probe_counts, name = "V2_bin_counts") %>% ungroup %>%
                group_by(V4_probe_counts) %>% mutate(V4_corr=cor(Meta_abund,V4_abund)) %>%
                add_count(V4_probe_counts, name = "V4_bin_counts") %>%
                ungroup %>% distinct(.keep_all = TRUE)
#Export to excel and then change the corr values to the ones done by the dotplots below (more accurate)
write.csv(combined_table,"combined_table.csv")

#Get correlation values for Allegro and JAX vs mWGS
V2_melt<-melt(V2_mapping_prop)
V2_per_sample_bincounts<-V2_melt %>% group_by(Bin,Probe,variable) %>%
                          add_tally(value>0, name = "V2_probes") %>% ungroup %>%
                          group_by(Bin,variable) %>%
                          mutate(V2_probes_per_bin=sum(V2_probes)) %>% ungroup
V2_count_table<- V2_per_sample_bincounts %>% group_by(Bin,variable) %>%
                 mutate(V2_abund=sum(value)) %>% ungroup %>%
                 select(Bin,variable,V2_probes_per_bin,V2_abund) %>% distinct()
colnames(V2_count_table)[2]<-"Sample"
```

```
V4_melt<-melt(V4_mapping_prop)
V4_per_sample_bincounts<-V4_melt %>% group_by(Bin,Probe,variable) %>%
                         add_tally(value>0, name = "V4_probes") %>% ungroup %>%
                         group_by(Bin,variable) %>%
                         mutate(V4_probes_per_bin=sum(V4_probes)) %>% ungroup
V4_count_table<- V4_per_sample_bincounts %>% group_by(Bin,variable) %>%
                 mutate(V4_abund=sum(value)) %>% ungroup %>%
                 select(Bin,variable,V4_probes_per_bin,V4_abund) %>% distinct()
colnames(V4_count_table)[2]<-"Sample"

#melt the metagenome data and rename columns
meta_melted<-melt(metag)
colnames(meta_melted)[1:3]<-c("Bin","Sample","meta_abund")

#combine meta, V2, V4 data
combined_for_corr<- meta_melted %>% full_join(V2_count_table, by=c("Bin","Sample")) %>%
                    full_join(V4_count_table, by=c("Bin","Sample"))
combined_for_corr[is.na(combined_for_corr)]<-0
#produce correlation plots to obtain correlation values
theme_set(theme_bw())
ggscatter(combined_for_corr, x="meta_abund", y="V4_abund", size = 0.5,
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE,
          cor.coeff.args= list(method= "pearson", label.x.npc="left", label.y.npc="top"),
          cor.coef.size =4,
          xlab = "mWGS Abundance (%)", ylab = "V4 Abundance (%)")+

  facet_wrap(combined_for_corr$V4_probes_per_bin, scales = "free") +
  theme(plot.title = element_text(hjust = 0.5, size = 14))+
  theme_linedraw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        strip.background = element_rect(fill="#BCBABE"),
        strip.text = element_text(colour = 'black', size = 14),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 14))

ggscatter(combined_for_corr, x="meta_abund", y="V2_abund", size = 0.5,
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE,
          cor.coeff.args= list(method= "pearson", label.x.npc="left", label.y.npc="top"),
          cor.coef.size =4,
          xlab = "mWGS Abundance (%)", ylab = "V2 Abundance (%)")+
          facet_wrap(combined_for_corr$V2_probes_per_bin, scales = "free") +
          theme(plot.title = element_text(hjust = 0.5, size = 14))+
          theme_linedraw() +
          theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
          strip.background = element_rect(fill="#BCBABE"),
          strip.text = element_text(colour = 'black', size = 14),
          axis.text = element_text(size = 12),
          axis.title = element_text(size = 14))
#Use correlation values from plots above to fill in exported table and import
combined_table<-read.csv("combined_table.csv", row.names = 1, header = TRUE)
```

Plots graphs

```
theme_set(theme_bw())
ggplot(combined_table, aes(x=V2_probe_counts, group=V2_corr)) +
  geom_histogram(binwidth = 1, color="black", size= 0.3, aes(fill=V2_corr))+
  scale_fill_gradient("Pearson\nCorrelation", limits=c(0,1))+
  labs(x= "Number of Probes", y= "Number of MAGs", title = "Allegro-mWGS\nCorrelations")+
  theme(plot.title = element_text(hjust = 0.5))+
  ylim(c(0,250))

ggplot(combined_table, aes(x=V4_probe_counts, group=V4_corr)) +
  geom_histogram(binwidth = 1, color="black", size= 0.3, aes(fill=V4_corr))+
  scale_fill_gradient("Pearson\nCorrelation", limits=c(0,1))+
  labs(x= "Number of Probes", y= "Number of MAGs", title = "JAX-mWGS\nCorrelations")+
  theme(plot.title = element_text(hjust = 0.5))+
  ylim(c(0,250))
```

Supplementary Figure 1

```
V2_2<-ggplot(combined_table, aes(x=V2_probe_counts, group=V2_abund)) +
      geom_histogram(binwidth = 1, aes(fill=V2_abund))+
      scale_fill_viridis_c("V2 Design\nPercent Abundance", alpha = 0.9, limits=c(0,.01),
      na.value = "#FDE725FF")+
      labs(x= "Number of Probes", y= "Number of MAGs", title = "Allegro\nAbundance")+
      theme(legend.position = "none",plot.title = element_text(hjust = 0.5))+
      ylim(c(0,250))

V2_3<-ggplot(combined_table, aes(x=V2_probe_counts, group=Meta_abund)) +
      geom_histogram(binwidth = 1, aes(fill=Meta_abund))+
      scale_fill_viridis_c("Percent\nAbundance", alpha = 0.9, limits=c(0,.01), na.value = "#FDE725FF")+
      labs(x= "Number of Probes", y= "Number of MAGs", title = "mWGS\nAbundance")+
      theme(plot.title = element_text(hjust = 0.5))+
      ylim(c(0,250))

library(patchwork)
V2_2 + V2_3

V4_2<-ggplot(combined_table, aes(x=V4_probe_counts, group=V4_abund)) +
      geom_histogram(binwidth = 1, aes(fill=V4_abund))+
      scale_fill_viridis_c("V4 Design\nPercent Abundance", alpha = 0.9, limits=c(0,.01),
      na.value = "#FDE725FF")+
      labs(x= "Number of Probes", y= "Number of MAGs", title = "JAX\nAbundance")+
      theme(legend.position = "none",plot.title = element_text(hjust = 0.5))+
      ylim(c(0,250))

V4_3<-ggplot(combined_table, aes(x=V4_probe_counts, group=Meta_abund)) +
      geom_histogram(binwidth = 1, aes(fill=Meta_abund))+
      scale_fill_viridis_c("Percent\nAbundance", alpha = 0.9, limits=c(0,.01), na.value = "#FDE725FF")+
      labs(x= "Number of Probes", y= "Number of MAGs", title = "mWGS\nAbundance")+
      theme(plot.title = element_text(hjust = 0.5))+
      ylim(c(0,250))

V4_2 + V4_3
```

Figure 3d

Import tables for Venn-diagrams

```
above_0_venn<-read.csv("above0_venn.csv", header = TRUE)
above_001_venn<-read.csv("above001_venn.csv",header = TRUE)
above_01_venn<-read.csv("above01_venn.csv",header = TRUE)
above_.1_venn<-read.csv("above.1_venn.csv",header = TRUE)
```

Plot the Venn-diagrams for the MAG abundance thresholds: 0.1%, 0.01%, 0.001%, No threshold

```
venn.diagram(x=list(above_0_venn$mWGS,above_0_venn$Allegro,above_0_venn$JAX),
             category.names = c("mWGS","Allegro","JAX"),
             filename = 'above_0_venndiagram.png',
             imagetype = "png",
             height = 550,
             width = 550,
             resolution = 800,
             lwd=.5,
             fill=c(alpha("#bfc0bd",.5),alpha("#597387",.5),alpha("#94ae3f",.5)),
             col=c("#bfc0bd","#597387","#94ae3f"),
             cex=.4,
             cat.cex=.4,
             cat.fontface="bold",
             cat.pos=c(-15,15,180),
             cat.dist=c(0.075,0.075,0.075),)

venn.diagram(x=list(above_001_venn$mWGS,above_001_venn$Allegro,above_001_venn$JAX),
             category.names = c("mWGS","Allegro","JAX"),
             filename = 'above_001_venndiagram.png',
             imagetype = "png",
             height = 550,
             width = 550,
             resolution = 800,
             lwd=.5,
             fill=c(alpha("#bfc0bd",.5),alpha("#597387",.5),alpha("#94ae3f",.5)),
             col=c("#bfc0bd","#597387","#94ae3f"),
             cex=.4,
             cat.cex=.4,
             cat.fontface="bold",
             cat.pos=c(-15,15,180),
             cat.dist=c(0.075,0.075,0.075),)

venn.diagram(x=list(above_01_venn$mWGS,above_01_venn$Allegro,above_01_venn$JAX),
             category.names = c("mWGS","Allegro","JAX"),
             filename = 'above_01_venndiagram.png',
             imagetype = "png",
             height = 550,
             width = 550,
             resolution = 800,
             lwd=.5,
             fill=c(alpha("#bfc0bd",.5),alpha("#597387",.5),alpha("#94ae3f",.5)),
             col=c("#bfc0bd","#597387","#94ae3f"),
             cex=.4,
             cat.cex=.4,
             cat.fontface="bold",
             cat.pos=c(-15,15,180),
             cat.dist=c(0.075,0.075,0.075),)

venn.diagram(x=list(above_.1_venn$mWGS,above_.1_venn$Allegro,above_.1_venn$JAX),
             category.names = c("mWGS","Allegro","JAX"),
             filename = 'above_.1_venndiagram.png',
             imagetype = "png",
             height = 550,
             width = 550,
             resolution = 800,
             lwd=.5,
             fill=c(alpha("#bfc0bd",.5),alpha("#597387",.5),alpha("#94ae3f",.5)),
             col=c("#bfc0bd","#597387","#94ae3f"),
             cex=.4,
             cat.cex=.4,
             cat.fontface="bold",
             cat.pos=c(-15,15,180),
             cat.dist=c(0.075,0.075,0.075),)
```

Figure 3e

Import 16S, mWGS, and MA-GenTA data

```
otus_16s<-read.csv("16S_mouse_only.csv", header = TRUE, check.names = FALSE)
metag<-read.csv("HLB_new_mapping.csv", header = TRUE, check.names = FALSE)
V4_HLB<-read.csv("V4_HLB.csv", header = TRUE, check.names = FALSE)
V2_HLB<-read.csv("V2_HLB.csv", header = TRUE, check.names = FALSE)
```

Format data for plotting

```
otus_16s_meta<-otus_16s[,1]
otus_16s_counts<-otus_16s[,2:29]
otus_16s_prop<-as.data.frame(prop.table(as.matrix(otus_16s_counts),2)*100)
otus_16s_mouse_prop<-cbind(otus_16s_meta,otus_16s_prop)

colnames(metag)[1]<-c("Bin")
colnames(metag_hqmq)[1]<-c("Bin")

V2_meta<-V2_HLB[,1:2]
V2_counts<-V2_HLB[,3:30]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mouse_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_HLB[,1:2]
V4_counts<-V4_HLB[,3:30]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mouse_prop<-cbind(V4_meta,V4_prop)
```

Create columns with number of MAGs at each threshold for mWGS data

```
metag_test<- metag %>% melt() %>% group_by(Bin,variable)%>% add_tally(value>0, name = "above_0") %>%
             add_tally(value>=0.01, name = "above_.01") %>%
             add_tally(value>=0.001,name = "above_.001") %>% add_tally(value>=0.1,name = "above.1") %>%
             group_by(variable) %>% mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
             mutate(BinCount_.001=sum(above_.001)) %>%
             mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="mWGS") %>%
             select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set)%>%
             distinct()
```

Create columns with number of MAGs at each threshold for 16S data

```
otus_16s_test<- otus_16s_mouse_prop %>% melt() %>% group_by(otus_16s_meta,variable)%>% add_tally(value>0, na
me =
                "above_0") %>%
                add_tally(value>=0.01, name = "above_.01") %>%
                add_tally(value>=0.001,name = "above_.001") %>%
                add_tally(value>=0.1,name = "above.1") %>%
                group_by(variable) %>% mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))
%>%
                mutate(BinCount_.001=sum(above_.001)) %>%
                mutate(BinCount.1=sum(above.1)) %>% ungroup %>%mutate(set="16S")%>%
                select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set)%>%
                distinct()
```

Create columns with number of MAGs at each threshold for MA-GenTA data

```
test_melt_V4<-melt(V4_mouse_prop) %>% as_tibble
probe_bin_counts_V4<- test_melt_V4 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V4<- probe_bin_counts_V4 %>% filter(probes_per_bin>=10) %>%
              group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V4<- filtered_V4 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0") %>%
                     add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="JAX") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()

test_melt_V2<-melt(V2_mouse_prop) %>% as_tibble
probe_bin_counts_V2<- test_melt_V2 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V2<- probe_bin_counts_V2 %>% filter(probes_per_bin>=10) %>% group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V2<- filtered_V2 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0")
                     %>%add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="Allegro") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()
```

Plot the graph

```
combined<-rbind(otus_16s_test,metag_test, allegro_cutoffs_V2, allegro_cutoffs_V4)
combined_melt<-melt(combined, id.vars = c("variable","set"))
colnames(combined_melt)<-c("Sample","set","BinCount","value")
combined_melt$set<-factor(combined_melt$set, levels = c("16S","Allegro", "JAX","mWGS"))

levels(combined_melt$BinCount)<-list("No\nThreshold"="BinCount_0",
                                     "0.001%"="BinCount_.001",
                                     "0.01%"="BinCount_.01",
                                     "0.1%"="BinCount.1")

combined_melt$BinCount<-factor(combined_melt$BinCount, levels =
                        c("0.1%","0.01%","0.001%","No\nThreshold"))

theme_set(theme_bw())
ggplot(combined_melt, aes(x=BinCount, y=value, fill=set, color=set))+
  geom_dotplot(binaxis='y', stackdir='center', position="dodge", binwidth = 9, dotsize = 1.3,
  stackratio = .7)+
  scale_fill_manual(values=c("#972426","#597387","#94ae3f","#bfc0bd","#A9845C"))+
  scale_color_manual(values=c("#5A1517","#364450","#536222","#818181","#6D583F"))+
  xlab("Abundance Threshold")+
  ylab("Number of MAGs")
```

Supplementary figure 2

Import and format the hqmq mWGS data

```
metag_hqmq<-read.csv("hqmq_mapping.csv", header = TRUE, check.names = FALSE)

hqmq_test<- metag_hqmq %>% melt() %>% group_by(Bin,variable)%>%
            add_tally(value>0, name = "above_0") %>%
            add_tally(value>=0.01, name = "above_.01") %>%
            add_tally(value>=0.001,name = "above_.001") %>%
            add_tally(value>=0.1,name = "above.1") %>% group_by(variable) %>%
            mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
            mutate(BinCount_.001=sum(above_.001)) %>%
            mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="hqmq-mWGS") %>%
            select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set)%>%
            distinct()
```

Combine and plot the hqmq data with the rest of data from 3e

```
combined<-rbind(otus_16s_test,metag_test, allegro_cutoffs_V2, allegro_cutoffs_V4,hqmq_test)
combined_melt<-melt(combined, id.vars = c("variable","set"))
colnames(combined_melt)<-c("Sample","set","BinCount","value")
combined_melt$set<-factor(combined_melt$set, levels = c("16S","Allegro", "JAX","mWGS","hqmq-mWGS"))

Bincount_names<-c('BinCount_0'=">0%",
                  'BinCount_.001'=">=0.001%",
                  'BinCount_.01'=">=0.01%",
                  'BinCount.1'=">=0.1%")
combined_melt$BinCount<-factor(combined_melt$BinCount, levels =
                        c("BinCount.1","BinCount_.01","BinCount_.001","BinCount_0"))

theme_set(theme_bw())
ggplot(combined_melt, aes(x=BinCount, y=value, fill=set, color=set))+
      geom_dotplot(binaxis='y', stackdir='center', position="dodge", binwidth = 9, dotsize = 1.3,
      stackratio = .7)+
      scale_fill_manual(values=c("#972426","#597387","#94ae3f","#bfc0bd","#A9845C"))+
      scale_color_manual(values=c("#5A1517","#364450","#536222","#818181","#6D583F"))+
      xlab("Abundance Threshold")+
      ylab("Number of MAGs")
```

Figure 3f

Import data tables

```
metag<-read.csv("CCF_new_mapping.csv", header = TRUE, check.names = FALSE)
V4_HLB<-read.csv("V4_CCF.csv", header = TRUE, check.names = FALSE)
V2_HLB<-read.csv("V2_CCF.csv", header = TRUE, check.names = FALSE)
```

Format data tables and convert MA-GenTA tables to percent abundance

```
colnames(metag)[1]<-c("Bin")
colnames(metag_hqmq)[1]<-c("Bin")

V2_meta<-V2_HLB[,1:2]
V2_counts<-V2_HLB[,3:31]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mouse_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_HLB[,1:2]
V4_counts<-V4_HLB[,3:31]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mouse_prop<-cbind(V4_meta,V4_prop)
```

Create columns with number of MAGs at each threshold for mWGS data

```
metag_test<- metag %>% melt() %>% group_by(Bin,variable)%>% add_tally(value>0, name = "above_0") %>%
             add_tally(value>=0.01, name = "above_.01") %>%
             add_tally(value>=0.001,name = "above_.001") %>% add_tally(value>=0.1,name = "above.1") %>%
             group_by(variable) %>% mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
             mutate(BinCount_.001=sum(above_.001)) %>%
             mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="mWGS") %>%
             select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set)%>%
             distinct()
```

Create columns with number of MAGs at each threshold for MA-GenTA data

```
test_melt_V4<-melt(V4_mouse_prop) %>% as_tibble
probe_bin_counts_V4<- test_melt_V4 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V4<- probe_bin_counts_V4 %>% filter(probes_per_bin>=10) %>% group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V4<- filtered_V4 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0") %>%
                     add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="JAX") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()

test_melt_V2<-melt(V2_mouse_prop) %>% as_tibble
probe_bin_counts_V2<- test_melt_V2 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V2<- probe_bin_counts_V2 %>% filter(probes_per_bin>=10) %>% group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V2<- filtered_V2 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0") %>%
                     add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0)) %>% mutate(BinCount_.01=sum(above_.01)) %>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="Allegro") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()
```

Combine and plot the data

```
combined<-rbind(metag_test, allegro_cutoffs_V2, allegro_cutoffs_V4)
combined_melt<-melt(combined, id.vars = c("variable","set"))
colnames(combined_melt)<-c("Sample","set","BinCount","value")
combined_melt$set<-factor(combined_melt$set, levels = c("Allegro", "JAX","mWGS"))

levels(combined_melt$BinCount)<-list("No\nThreshold"="BinCount_0",
                                     "0.001%"="BinCount_.001",
                                     "0.01%"="BinCount_.01",
                                     "0.1%"="BinCount.1")

combined_melt$BinCount<-factor(combined_melt$BinCount, levels =
                        c("0.1%","0.01%","0.001%","No\nThreshold"))

theme_set(theme_bw())
ggplot(combined_melt, aes(x=BinCount, y=value, fill=set, color=set))+
      geom_dotplot(binaxis='y', stackdir='center', position="dodge", binwidth = 9, dotsize = 1.3,
      stackratio = .7)+
      scale_fill_manual(values=c("#597387","#94ae3f","#bfc0bd","#A9845C"))+
      scale_color_manual(values=c("#364450","#536222","#818181","#6D583F"))+
      xlab("Abundance Threshold")+
      ylab("Number of MAGs")
```

Figure 3g

Import the data tables

```
otus_16s<-read.csv("otu_stool>4.csv", header = TRUE, check.names = FALSE)
V4_HLB<-read.csv("V4_VNDR.csv", header = TRUE, check.names = FALSE)
V2_HLB<-read.csv("V2_VNDR.csv", header = TRUE, check.names = FALSE)
```

Convert to percent abundances

```
otus_16s_meta<-otus_16s[,1]
otus_16s_counts<-otus_16s[,2:4]
otus_16s_prop<-as.data.frame(prop.table(as.matrix(otus_16s_counts),2)*100)
otus_16s_mouse_prop<-cbind(otus_16s_meta,otus_16s_prop)

V2_meta<-V2_HLB[,1:2]
V2_counts<-V2_HLB[,3:5]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mouse_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_HLB[,1:2]
V4_counts<-V4_HLB[,3:5]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mouse_prop<-cbind(V4_meta,V4_prop)
```

Create columns with number of MAGs at each threshold for 16S data

```
otus_16s_test<- otus_16s_mouse_prop %>% melt() %>% group_by(otus_16s_meta,variable)%>%
                add_tally(value>0, name = "above_0") %>%
                add_tally(value>=0.01, name = "above_.01") %>%
                add_tally(value>=0.001,name = "above_.001") %>%
                add_tally(value>=0.1,name = "above.1") %>%
                group_by(variable) %>% mutate(BinCount_0=sum(above_0)) %>%
                mutate(BinCount_.01=sum(above_.01)) %>%
                mutate(BinCount_.001=sum(above_.001)) %>%
                mutate(BinCount.1=sum(above.1)) %>% ungroup %>%mutate(set="16S")%>%
                select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set)%>%
                distinct()
```

Create columns with number of MAGs at each threshold for MA-GenTA data

```
test_melt_V4<-melt(V4_mouse_prop) %>% as_tibble
probe_bin_counts_V4<- test_melt_V4 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V4<- probe_bin_counts_V4 %>% filter(probes_per_bin>=10) %>% group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V4<- filtered_V4 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0") %>%
                     add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="JAX") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()

test_melt_V2<-melt(V2_mouse_prop) %>% as_tibble
probe_bin_counts_V2<- test_melt_V2 %>% group_by(Bin,variable) %>%
                      add_tally(value>0,name = "probes_per_bin") %>%
                      mutate(Bin_abund=sum(value))

filtered_V2<- probe_bin_counts_V2 %>% filter(probes_per_bin>=10) %>% group_by(Bin,variable) %>%
              summarize(abundance=mean(Bin_abund))

allegro_cutoffs_V2<- filtered_V2 %>% group_by(Bin,variable) %>%
                     add_tally(abundance>=0, name = "above_0") %>%
                     add_tally(abundance>=0.01, name="above_.01") %>%
                     add_tally(abundance>=0.001,name = "above_.001") %>%
                     add_tally(abundance>=0.1,name="above.1") %>%
                     ungroup %>% group_by(variable) %>%
                     mutate(BinCount_0=sum(above_0))%>%mutate(BinCount_.01=sum(above_.01))%>%
                     mutate(BinCount_.001=sum(above_.001)) %>%
                     mutate(BinCount.1=sum(above.1)) %>% ungroup %>% mutate(set="Allegro") %>%
                     select(variable,BinCount.1,BinCount_.01,BinCount_.001,BinCount_0,set) %>%
                     distinct()
```

Combine data and plot

```
combined<-rbind(otus_16s_test, allegro_cutoffs_V2, allegro_cutoffs_V4)
combined_melt<-melt(combined, id.vars = c("variable","set"))
colnames(combined_melt)<-c("Sample","set","BinCount","value")
levels(combined_melt$BinCount)<-list("No\nThreshold"="BinCount_0",
                                     "0.001%"="BinCount_.001",
                                     "0.01%"="BinCount_.01",
                                     "0.1%"="BinCount.1")

combined_melt$BinCount<-factor(combined_melt$BinCount, levels =
                        c("0.1%","0.01%","0.001%","No\nThreshold"))
combined_melt$set<-factor(combined_melt$set, levels = c("16S","Allegro", "JAX"))

theme_set(theme_bw())
ggplot(combined_melt, aes(x=BinCount, y=value, fill=set, color=set))+
      geom_boxplot(position = position_dodge(0.8))+
      geom_dotplot(binaxis='y', stackdir='center', position="dodge", binwidth = 9, dotsize = .5,
      stackratio = .7)+
      scale_fill_manual(values=c("#972426","#597387","#94ae3f","#bfc0bd","#A9845C"))+
      scale_color_manual(values=c("#5A1517","#364450","#536222","#818181","#6D583F"))+
      xlab("Abundance Threshold")+
      ylab("Number of MAGs")
```

Figure 4

Figure 4a

Import MAG and OTU read data

```
V2_otu=as.matrix(read.table("V2_sum_bin.csv", header = TRUE, row.names = 1, sep = ",",
                            check.names = FALSE))
V4_otu=as.matrix(read.table("V4_sum_bin.csv", header = TRUE, row.names = 1, sep = ",",
                            check.names = FALSE))
A16S_otu=as.matrix(read.table("16S_for_pcoa.csv", header = TRUE, row.names = 1, sep = ",",
                              check.names = FALSE))
meta_otu=as.matrix(read.table("HLB_new_mapping_for_pcoa.csv", header = TRUE, row.names = 1, sep = ",",
                              check.names = FALSE))
```

Import taxon tables

```
V2_tax=as.matrix(read.table("V2_tax.csv", header = TRUE, row.names = 1, sep = ",",
                            check.names = FALSE))
V4_tax=as.matrix(read.table("V4_tax.csv", header = TRUE, row.names = 1, sep = ",",
                            check.names = FALSE))
A16S_tax=as.matrix(read.table("16s_tax.csv", header = TRUE, row.names = 1, sep = ",",
                              check.names = FALSE))
meta_tax=as.matrix(read.table("meta_tax.csv", header = TRUE, row.names = 1, sep = ",",
                              check.names = FALSE))
```

Create Phyloseq objects

```
TAX_V2=tax_table(V2_tax)
TAX_V4=tax_table(V4_tax)
TAX_16S=tax_table(A16S_tax)
TAX_meta=tax_table(meta_tax)

OTU_V2=otu_table(V2_otu,taxa_are_rows=TRUE)
OTU_V4=otu_table(V4_otu,taxa_are_rows=TRUE)
OTU_16S=otu_table(A16S_otu,taxa_are_rows=TRUE)
OTU_meta=otu_table(meta_otu,taxa_are_rows=TRUE)
```

Merge OTU and TAX

```
TAX_OTU_V2=phyloseq(OTU_V2,TAX_V2)
TAX_OTU_V4=phyloseq(OTU_V4,TAX_V4)
TAX_OTU_16S=phyloseq(OTU_16S,TAX_16S)
TAX_OTU_meta=phyloseq(OTU_meta,TAX_meta)
```

Load in metadata

```
metadata=sample_data(as.data.frame(read.csv("metadata.csv", header = TRUE,
                                            row.names = sample_names(TAX_OTU_V2))))
```

Merge OTU, TAXA and metadata

```
TAX_OTU_meta_V2=merge_phyloseq(TAX_OTU_V2,metadata)
TAX_OTU_meta_V4=merge_phyloseq(TAX_OTU_V4,metadata)
TAX_OTU_meta_16S=merge_phyloseq(TAX_OTU_16S,metadata)
TAX_OTU_meta_meta=merge_phyloseq(TAX_OTU_meta,metadata)
```

Create Bray-Curtis dissimilarity matrices

```
Bray_V2=distance(TAX_OTU_meta_V2,"bray")
Bray_V4=distance(TAX_OTU_meta_V4,"bray")
Bray_16S=distance(TAX_OTU_meta_16S,"bray")
Bray_meta=distance(TAX_OTU_meta_meta,"bray")
```

Ordinate the data using NMDS

```
TAX_OTU_V2.nmds = ordinate(TAX_OTU_meta_V2, method="NMDS", distance=Bray_V2)
TAX_OTU_V4.nmds = ordinate(TAX_OTU_meta_V4, method="NMDS", distance=Bray_V4)
TAX_OTU_16S.nmds = ordinate(TAX_OTU_meta_16S, method="NMDS", distance=Bray_16S)
TAX_OTU_meta.nmds = ordinate(TAX_OTU_meta_meta, method="NMDS", distance=Bray_meta)
```

Plot the graphs

```
theme_set(theme_bw())
V2nmds<-plot_ordination(TAX_OTU_meta_V2,TAX_OTU_V2.nmds, axes=c(1, 2), color="Diet", shape = "Strain") +
        scale_color_manual(values = c("#777777","#92C46D","#2A7D7D")) +
        geom_point(size=3)
V2nmds<-V2nmds+ scale_shape_manual(values = c(1,19))+
        ggtitle("Allegro")+ theme(plot.title = element_text(hjust = 0.5),axis.title = element_blank())
V2nmds$layers<- V2nmds$layers[-1]

V4nmds<-plot_ordination(TAX_OTU_meta_V4,TAX_OTU_V4.nmds, axes=c(1, 2), color="Diet",
                        shape = "Strain") +
        scale_color_manual(values = c("#777777","#92C46D","#2A7D7D")) +
        geom_point(size=3)
V4nmds<-V4nmds+ scale_shape_manual(values = c(1,19))+
        ggtitle("JAX")+ theme(plot.title = element_text(hjust = 0.5),axis.title = element_blank())
V4nmds$layers<- V4nmds$layers[-1]

metanmds<-plot_ordination(TAX_OTU_meta_meta,TAX_OTU_meta.nmds, axes=c(1, 2), color="Diet",
                          shape = "Strain") +
          scale_color_manual(values = c("#777777","#92C46D","#2A7D7D")) +
          geom_point(size=3)
metanmds<-metanmds+ scale_shape_manual(values = c(1,19))+
          ggtitle("mWGS")+ theme(plot.title = element_text(hjust = 0.5),axis.title = element_blank())
metanmds$layers<- metanmds$layers[-1]

a16snmds<-plot_ordination(TAX_OTU_meta_16S,TAX_OTU_16S.nmds, axes=c(1, 2), color="Diet",
                          shape = "Strain") +
          scale_color_manual(values = c("#777777","#92C46D","#2A7D7D")) +
          geom_point(size=3)
a16snmds<-a16snmds+ scale_shape_manual(values = c(1,19))+
          ggtitle("16S")+ theme(plot.title = element_text(hjust = 0.5),axis.title = element_blank())
a16snmds$layers<- a16snmds$layers[-1]

ggarrange(a16snmds,metanmds,V2nmds,V4nmds, ncol = 4, nrow = 1, legend = c("right"),
          common.legend = TRUE)
```

Perform PERMANOVA statistics (Supplementary Table 3)

```
V2_otu<-as.data.frame(t(read.csv("V2_sum_bin.csv", header=TRUE, row.names = 1)))
V2_tax<-read.csv("V2_tax.csv", header=TRUE, row.names = 1)
meta<-read.csv("metadata.csv",header=TRUE, row.names = 1)
V2.dist<-vegdist(V2_otu, distance="bray")
adonis(V2.dist~Timepoint_Strain, data=meta, permutations=1000)

V4_otu<-as.data.frame(t(read.csv("V4_sum_bin.csv", header=TRUE, row.names = 1)))
V4_tax<-read.csv("V4_tax.csv", header=TRUE, row.names = 1)
V4.dist<-vegdist(V4_otu, distance="bray")
adonis(V4.dist~Timepoint_Strain, data=meta, permutations=1000)

a16_otu<-as.data.frame(t(read.csv("16S_for_pcoa.csv", header=TRUE, row.names = 1)))
a16_tax<-read.csv("16s_tax.csv", header=TRUE, row.names = 1)
a16.dist<-vegdist(a16_otu, distance="bray")
adonis(a16.dist~Timepoint_Strain, data=meta, permutations=1000)

mwgs_otu<-as.data.frame(t(read.csv("metag_for_pcoa.csv", header=TRUE, row.names = 1)))
mwgs_tax<-read.csv("meta_tax.csv", header=TRUE, row.names = 1)
mwgs.dist<-vegdist(mwgs_otu, distance="bray")
adonis(mwgs.dist~Timepoint_Strain, data=meta, permutations=1000)
```

Figure 4b

Import the count tables and change “Bin” to “MAG”

```
JAX_counts<-read.csv("JAX_count_tables.csv", header = TRUE, check.names = FALSE)
colnames(JAX_counts)[1]<-c("MAG")

Allegro_counts<-read.csv("Allegro_count_table.csv", header = TRUE, check.names=FALSE)
colnames(Allegro_counts)[1]<-c("MAG")
```

Import MAG_ORF_KO table generated from Eggnog mapper

```
MAG_ORF_KO<-read.table("bin_pathway_split_no4_5")

#Change column names to MAG, ORF, and KO
colnames(MAG_ORF_KO)[1:3]<-c("ORF","MAG","KO")

#Remove the ORF column
MAG_KO<- subset(MAG_ORF_KO, select = c("MAG","KO"))
```

Join the count table and MAG_KO table, using MAG as the reference between the two files

```
JAX_joined<-right_join(MAG_KO, JAX_counts, by="MAG")
Allegro_joined<-right_join(MAG_KO, Allegro_counts, by="MAG")
```

Group by KO and sum the values for each sample per KO

```
JAX_KO_SUM<- JAX_joined %>% group_by(KO) %>% summarise_at(c(2:29), sum)
Allegro_KO_SUM<- Allegro_joined %>% group_by(KO) %>% summarise_at(c(2:29), sum)
```

Make relative abundances of KOs for each sample

```
JAX_meta<-JAX_KO_SUM[,1]
JAX_counts<-JAX_KO_SUM[,2:29]
JAX_prop<-as.data.frame(prop.table(as.matrix(JAX_counts),2)) #sums to 1
JAX_mapping_prop<-cbind(JAX_meta,JAX_prop)

Allegro_meta<-Allegro_KO_SUM[,1]
Allegro_counts<-Allegro_KO_SUM[,2:29]
Allegro_prop<-as.data.frame(prop.table(as.matrix(Allegro_counts),2)) #sums to 1
Allegro_mapping_prop<-cbind(Allegro_meta,Allegro_prop)
```

Export tables. Make separate tab-delimited text files for each comparison. Import tables into LEfSe on the Galaxy server and perform LDA analysis.

```
write.csv(JAX_mapping_prop,"JAX_mapping_prop.csv")
write.csv(Allegro_mapping_prop,"Allegro_mapping_prop.csv")
```

Combine tables for JAX probe set, HLB444 vs. B6 comparisons in both Chow and HF. Import table.

```
TEST<-read.csv("JAX_HLB_B6_Chow_HF_TEST.csv", header = TRUE)

ggplot(TEST, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO Pathway")+
  ylab("LDA Score (log10)")+
  facet_grid(cols=vars(diet), rows = vars(pathway_group),  scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))
```

Supplementary Figures 4-11

Import LDA results for each comparison

```
Allegro_HLB_B6_HF<-read.csv("Allegro_HLB_B6_HF_LDA.csv", header = TRUE)
Allegro_HLB_B6_Chow<-read.csv("Allegro_HLB_B6_Chow_LDA.csv", header = TRUE)
Allegro_HLB_Chow_HF<-read.csv("Allegro_HLB_Chow_HF_LDA.csv", header = TRUE)
Allegro_B6_Chow_HF<-read.csv("Allegro_B6_Chow_HF_LDA.csv", header = TRUE)

JAX_HLB_B6_HF<-read.csv("JAX_HLB_B6_HF_LDA.csv", header = TRUE)
JAX_HLB_B6_Chow<-read.csv("JAX_HLB_B6_Chow_LDA.csv", header = TRUE)
JAX_HLB_Chow_HF<-read.csv("JAX_HLB_Chow_HF_LDA.csv", header = TRUE)
JAX_B6_Chow_HF<-read.csv("JAX_B6_Chow_HF_LDA.csv", header = TRUE)
```

Plot the graphs

```
theme_set(theme_bw())
ggplot(Allegro_HLB_B6_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))


ggplot(Allegro_HLB_B6_Chow, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(Allegro_HLB_Chow_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(Allegro_B6_Chow_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(JAX_HLB_B6_Chow, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(JAX_HLB_B6_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(JAX_HLB_Chow_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))

ggplot(JAX_B6_Chow_HF, aes(reorder(pathway,-LDA), LDA, fill=Class))+
  geom_bar(stat="identity", col="black")+
  scale_fill_manual(values=c("#BEBEBE","#274b69"))+
  coord_flip()+
  xlab("KO")+
  ylab("LDA Score (log10)")+
  facet_grid(rows = vars(Pathway_class), scales = "free", space = "free_y")+
  theme(strip.text.y = element_text(angle = 0))
```

Combine files for Supplementary Table X

```
Allegro_HLB_B6_HF_combine<-read.csv("Allegro_HLB_B6_HF_combine.csv", header = TRUE)
Allegro_HLB_B6_Chow_combine<-read.csv("Allegro_HLB_B6_Chow_combine.csv", header = TRUE)
Allegro_HLB_Chow_HF_combine<-read.csv("Allegro_HLB_Chow_HF_combine.csv", header = TRUE)
Allegro_B6_Chow_HF_combine<-read.csv("Allegro_B6_Chow_HF_combine.csv", header = TRUE)

JAX_HLB_B6_HF_combine<-read.csv("JAX_HLB_B6_HF_combine.csv", header = TRUE)
JAX_HLB_B6_Chow_combine<-read.csv("JAX_HLB_B6_Chow_combine.csv", header = TRUE)
JAX_HLB_Chow_HF_combine<-read.csv("JAX_HLB_Chow_HF_combine.csv", header = TRUE)
JAX_B6_Chow_HF_combine<-read.csv("JAX_B6_Chow_HF_combine.csv", header = TRUE)

joined<-full_join(Allegro_HLB_B6_HF_combine,Allegro_HLB_B6_Chow_combine, by="KO")
joined2<-full_join(Allegro_HLB_Chow_HF_combine,Allegro_B6_Chow_HF_combine, by="KO")
Allegro_joined<-full_join(joined,joined2, by="KO")

joined3<-full_join(JAX_HLB_B6_HF_combine,JAX_HLB_B6_Chow_combine, by="KO")
joined4<-full_join(JAX_HLB_Chow_HF_combine,JAX_B6_Chow_HF_combine, by="KO")
JAX_joined<-full_join(joined3,joined4, by="KO")

Allegro_Jax_joined<-full_join(Allegro_joined,JAX_joined, by="KO")
#export joined tables to make table for paper
write.csv(Allegro_Jax_joined,"allegro_jax_joined.csv")
```

Supplementary Figure 3

Figure S3a

Import data tables

```
V2_mapping<-read.csv("V2_controls.csv",  header = TRUE)
V4_mapping<-read.csv("V4_controls.csv",  header = TRUE)
```

Convert to percent abundance

```
V2_meta<-V2_mapping[,1:2]
V2_counts<-V2_mapping[,3:12]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mapping_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4_mapping[,1:2]
V4_counts<-V4_mapping[,3:12]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mapping_prop<-cbind(V4_meta,V4_prop)
```

Select human stool samples and apply thresholds

```
V2_human<-select(V2_mapping_prop, Bin,Probe,J00YQC.human_stool_P1V2, J00YSZ.human_stool_P2V2) %>%
          melt() %>% group_by(Bin,Probe) %>%
          summarize(mean=mean(value)) %>% filter(mean>=0.001) %>% add_tally(mean>=0.001) %>%
          ungroup %>% filter(n>=10) %>% mutate(set=">0.001%, 15+ Probes") %>% group_by(Bin) %>%
          mutate(BinSum=sum(mean)) %>%
          ungroup %>% mutate(design="V2")


V4_human<-select(V4_mapping_prop, Bin,Probe,J00YP1.human_stool_P1V4,J00YRO.human_stool_P2V4) %>%
          melt() %>% group_by(Bin,Probe) %>%
          summarize(mean=mean(value)) %>% filter(mean>=0.001) %>% add_tally(mean>=0.001) %>%
          ungroup %>% filter(n>=10) %>% mutate(set=">0.001%, 15+ Probes") %>% group_by(Bin) %>%
          mutate(BinSum=sum(mean)) %>%
          ungroup %>% mutate(design="V4")

human_combined<-rbind(V2_human,V4_human)
h2_exclude<-droplevels(human_combined) #drop the levels from parent dataframe
            levels(h2_exclude$Bin) #check the levels to make sure theyre correct
h2_ordered<-h2_exclude[order(h2_exclude$BinSum,decreasing=T),] #order the Bins by the decreasing BinSum number
h2_unique<-unique(h2_ordered$Bin) #get the bin names
h2_unique #check the bin names
#force the order of the levels to be in decreasing BinSum order
h2_for_plot<- within(h2_ordered, Bin <- factor(Bin, levels = c("extra-SRR5925348.8",
                                        "extra-ERR982795.38" ,"single-China_7_110627.15",
                                        "single-China_G1-4A_111220.15", "extra-SRR3539764.35",
                                        "extra-SRR8443416.55","extra-SRR8581402.2" ,
                                        "extra-ERR1762100.9","extra-SRR7533643.1",
                                        "single-China_1_110627.11" , "single-China_43_110531.14" ,
                                        "extra-SRR8291361.67","extra-ERR1762120.34",
                                        "extra-SRR3223201.41" , "extra-ERR982833.21","iMGMC-244")))
levels(h2_for_plot$Bin) #check the levels to make sure correct
```

Plot the graph

```
theme_set(theme_bw())
ggplot(h2_for_plot, aes(x=mean, y=reorder(Bin, mean)))+
  geom_boxplot(color="black", lwd=0.3, outlier.size = 0.5, fill="gray")+
  #aes(x=reorder(Bin,-mean,sum))+
  geom_point(size=0.5)+
  #stat_summary(fun.data = give.n,geom="text")+ #vjust= -1, position="identity"
  #stat_summary(fun.y = sum, geom = "point",shape=23, size=1, color="black", fill="black")+
  theme(legend.position = "none",  axis.title.y = element_blank(), axis.ticks.y = element_blank())+
  xlab("Abundance (%)")+
  xscale("log10")+
  facet_grid(cols = vars(design), scales = "free")+#, cols = vars(variable)
  guides(fill=guide_legend(nrow = 5))+
  theme(strip.text = element_blank(), axis.text.x = element_text(angle=45, hjust=1))
```

Figure S3b

Import data tables

```
V2<-read.csv("V2_mouse.csv", header = TRUE, check.names = FALSE)
V4<-read.csv("V4_mouse.csv", header = TRUE, check.names = FALSE)
```

Convert to percent abundances

```
V2_meta<-V2[,1:2]
V2_counts<-V2[,3:72]
V2_prop<-as.data.frame(prop.table(as.matrix(V2_counts),2)*100)
V2_mapping_prop<-cbind(V2_meta,V2_prop)

V4_meta<-V4[,1:2]
V4_counts<-V4[,3:72]
V4_prop<-as.data.frame(prop.table(as.matrix(V4_counts),2)*100)
V4_mapping_prop<-cbind(V4_meta,V4_prop)
```

Select counts for the SFB MAG (iMGMC-930)

```
V2_SFB<- V2_mapping_prop %>% filter(Bin=="iMGMC-930") %>% melt() %>% group_by(Bin,variable) %>%
         add_tally(value>0, name="probes_per_bin") %>% mutate(BinSum=sum(value)) %>% ungroup %>%
         select(Bin,variable,probes_per_bin,BinSum) %>% distinct()%>% filter(!probes_per_bin==0)%>%
         filter(!probes_per_bin==1)

V4_SFB<- V4_mapping_prop %>% filter(Bin=="iMGMC-930")%>% melt() %>% group_by(Bin,variable) %>%
         add_tally(value>0, name="probes_per_bin") %>% mutate(BinSum=sum(value)) %>% ungroup %>%
         select(Bin,variable,probes_per_bin,BinSum) %>% distinct() %>% filter(!probes_per_bin==0)
```

Plot the graphs

```
theme_set(theme_bw())
ggplot(V4_SFB, aes(x=reorder(variable, -probes_per_bin), y=probes_per_bin))+
  geom_bar(stat="identity", fill="#94ae3f",color="black")+
  theme(axis.text.x = element_text(angle=45, hjust=1), axis.ticks.x = element_blank(), axis.text =
  element_text(size=8))+
  xlab("Samples")+
  ylab("Probes per Bin")+
  theme(axis.text.x = element_blank())

ggplot(V2_SFB, aes(x=reorder(variable, -probes_per_bin), y=probes_per_bin))+
  geom_bar(stat="identity", fill="#274b69",color="black")+
  theme(axis.text.x = element_text(angle=45, hjust=1), axis.ticks.x = element_blank(), axis.text =
  element_text(size=8))+
  xlab("Samples")+
  ylab("Probes per Bin")+
  theme(axis.text.x = element_blank())+
  ylim(0,20)
```