

Selected MAGs

```
library(tidyverse)
library(distillR)
library(ape)
library(vegan)
library(ggrepel)
library(broom)
```

```
gene_annotations <- read_tsv("data/gene_annotations.tsv.xz") %>%
  filter(gene!="gene") %>%
  mutate(ec=str_c("[EC:",ec,"]")) %>%
  mutate(genome=substr(gene, 1, 7))
```

```
genome_metadata <- read_csv("data/genome_metadata.csv") %>%
  rename(genome=MAG)
```

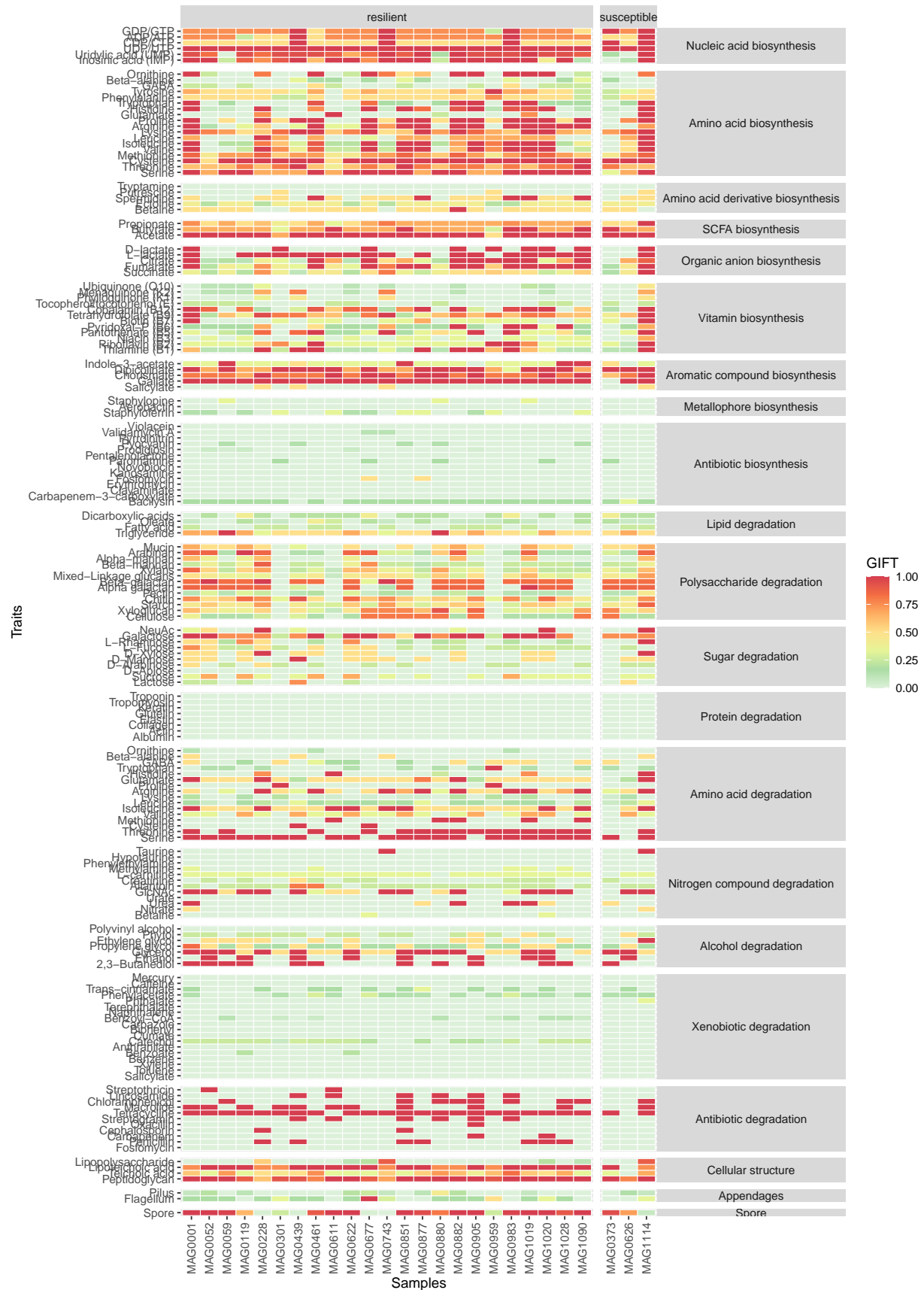
```
gene_distillation <- distill(gene_annotations,GIFT_db, genomecol = 14, annotcol = c(5,6), verbosity=F)
```

```
##
## Identifiers in the annotation table: 2294
## Identifiers in the database: 1547
## Identifiers in both: 204
## Percentage of annotation table identifiers used for distillation: 8.89%
## Percentage of database identifiers used for distillation: 13.19%
```

```
gift_elements <- to.elements(gene_distillation,GIFT_db)
```

```
gift_elements %>%
  as.data.frame() %>%
  rownames_to_column(var="genome") %>%
  pivot_longer(!genome,names_to="trait",values_to="gift") %>%
  mutate(
    status = case_when(
      genome %in% c("MAG1114", "MAG0373", "MAG0626") ~ "susceptible",
      TRUE ~ "resilient"
    )
  ) %>%
  mutate(functionid = substr(trait, 1, 3)) %>%
  mutate(trait = case_when(
    trait %in% GIFT_db$Code_element ~ GIFT_db$Element[match(trait, GIFT_db$Code_element)],
    TRUE ~ trait
  )) %>%
  mutate(functionid = case_when(
    functionid %in% GIFT_db$Code_function ~ GIFT_db$Function[match(functionid, GIFT_db$Code_function)],
    TRUE ~ functionid
  )) %>%
  mutate(trait=factor(trait,levels=unique(GIFT_db$Element))) %>%
  mutate(functionid=factor(functionid,levels=unique(GIFT_db$Function))) %>%
  ggplot(aes(x=genome,y=trait,fill=gift)) +
    geom_tile(colour="white", linewidth=0.2)+
    scale_fill_gradientn(colours=rev(c("#d53e4f", "#f46d43", "#fdae61", "#fee08b", "#e6f598", "#abd9e9")))
    facet_grid(functionid ~ status, scales="free",space="free") +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
          strip.text.y = element_text(angle = 0)) +
```

```
labs(y="Traits",x="Samples",fill="GIFT")
```



Functional differences

```
functional_differences <- gift_elements %>%
  as.data.frame() %>%
  rownames_to_column("genome") %>%
  mutate(
    status = case_when(
      genome %in% c("MAG1114", "MAG0373", "MAG0626") ~ "susceptible",
      TRUE ~ "resilient"
    )
  ) %>%
  pivot_longer(-c(genome, status), names_to = "trait", values_to = "value") %>%
  nest(data = -trait) %>%
  mutate(
    fit = map(data, ~lm(value ~ status, data = .x)),
    tid = map(fit, tidy)
  ) %>%
  unnest(tid) %>%
  filter(term == "statussusceptible") %>%
  mutate(p_value_adj = p.adjust(p.value, method = "bonferroni")) %>%
  select(id = trait,
         estimate,
         p.value,
         p_value_adj) %>%
  arrange(p.value)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `tid = map(fit, tidy)`.
## Caused by warning in `summary.lm()`:
## ! essentially perfect fit: summary may be unreliable
```

```
functional_differences %>%
  filter(p.value<0.05)
```

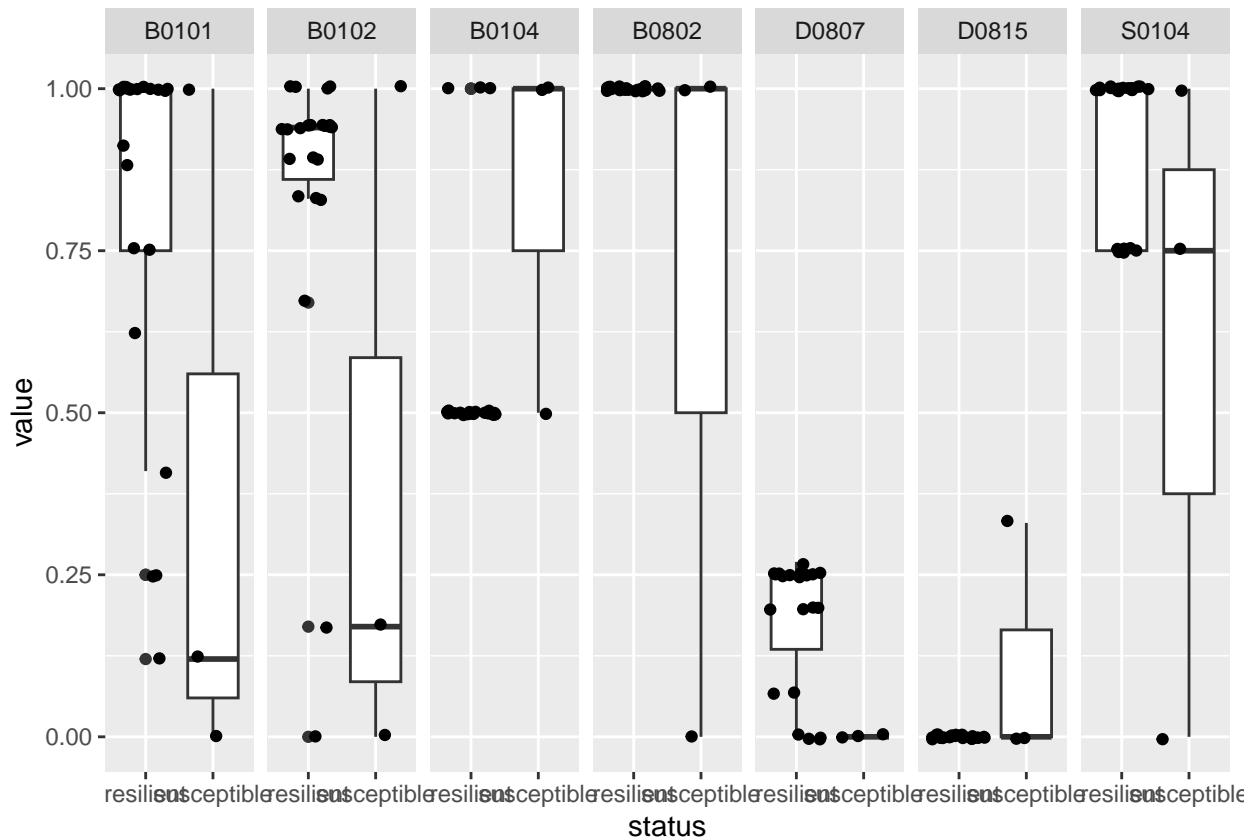
```
## # A tibble: 7 x 4
##   id      estimate p.value p_value_adj
##   <chr>      <dbl>   <dbl>      <dbl>
## 1 B0802    -0.333 0.00334      0.470
## 2 D0815     0.110 0.00334      0.470
## 3 D0807    -0.183 0.00489      0.689
## 4 S0104    -0.341 0.00690      0.973
## 5 B0102    -0.453 0.0164        1
## 6 B0104     0.268 0.0265        1
## 7 B0101    -0.450 0.0298        1
```

```
gift_elements %>%
  as.data.frame() %>%
  select(functional_differences %>% filter(p.value<0.05) %>% pull(id)) %>%
  rownames_to_column(var="genome") %>%
  pivot_longer(!genome, names_to = "trait", values_to = "value") %>%
  mutate(
    status = case_when(
      genome %in% c("MAG1114", "MAG0373", "MAG0626") ~ "susceptible",
      TRUE ~ "resilient"
    )
  )
```

```

) %>%
ggplot(aes(x=status, y=value, group=status))+
  geom_boxplot() +
  geom_jitter() +
  facet_grid(. ~ trait, scales="free",space="free")

```



Functional ordination

```

gift_pcoa <- gift_elements %>%
  as.data.frame() %>%
  vegdist(method="euclidean") %>%
  pcoa()

gift_pcoa_rel_eigen <- gift_pcoa$values$Relative_eig[1:10]

# Get genome positions
gift_pcoa_vectors <- gift_pcoa$vectors %>% #extract vectors
  as.data.frame() %>%
  select(Axis.1,Axis.2) # keep the first 2 axes

gift_pcoa_eigenvalues <- gift_pcoa$values$Eigenvalues[c(1,2)]

gift_pcoa_gifts <- cov(gift_elements, scale(gift_pcoa_vectors)) %*% diag((gift_pcoa_eigenvalues/(nrow(g
  as.data.frame() %>%

```

```

rename(Axis.1=1,Axis.2=2) %>%
rownames_to_column(var="label") %>%
#get function summary vectors
mutate(func=substr(label,1,3)) %>%
group_by(func) %>%
summarise(Axis.1=mean(Axis.1),
          Axis.2=mean(Axis.2)) %>%
rename(label=func) %>%
filter(!label %in% c("S01","S02","S03"))

scale <- 15 # scale for vector loadings
gift_pcoa_vectors %>%
rownames_to_column(var="genome") %>%
left_join(genome_metadata, by="genome") %>%
mutate(
  status = case_when(
    genome %in% c("MAG1114", "MAG0373", "MAG0626") ~ "susceptible",
    TRUE ~ "resilient"
  )
) %>%
ggplot() +
  #genome positions
  #scale_color_manual(values=order_colors)+
  geom_point(aes(x=Axis.1,y=Axis.2, color=status),
             alpha=0.9, shape=16, size=4) +
  #scale_color_manual(values=phylum_colors) +
  scale_size_continuous(range = c(0.1,5)) +
  #loading positions
  geom_segment(data=gift_pcoa_gifts,
              aes(x=0, y=0, xend=Axis.1 * scale, yend=Axis.2 * scale),
              arrow = arrow(length = unit(0.3, "cm"),
                             type = "open",
                             angle = 25),
              linewidth = 0.5,
              color = "black") +
  #Primary and secondary scale adjustments
  scale_x_continuous(name = paste0("PCoA1 (",round(gift_pcoa_rel_eigen[1]*100, digits = 2), " %)" ),
                    sec.axis = sec_axis(~ . / scale, name = "Loadings on PCoA1")
  ) +
  scale_y_continuous(name = paste0("PCoA2 (",round(gift_pcoa_rel_eigen[2]*100, digits = 2), " %)" ),
                    sec.axis = sec_axis(~ . / scale, name = "Loadings on PCoA2")
  ) +
  geom_label_repel(data = gift_pcoa_gifts,
                  aes(label = label, x = Axis.1 * scale, y = Axis.2 * scale),
                  segment.color = 'transparent') +
  theme_minimal() +
  theme(legend.position = "none")

```

