

Reproducible Analysis: The Two-Wing admixed structure of environmental microbial communities

Yushi Tang

2022-08-26

```
library(ggplot2)      # for generating plots
library(latex2exp)    # for plot text latex
library(cowplot)      # for merging plots
library(gridExtra)    # for gridding plots
library(grid)         # for gridding plots
library(stringr)      # for uppercase first letter
library(ggh4x)        # for grid plot with free axis
library(vegan)        # for example data in pcoa analysis
library(ggvegan)      # for ggplot vegan results
```

Preface

This document provides reproducible research records for our manuscript about the Two-Wing admixed structure of environmental microbial communities. We provide step-by-step instructions for main results in both the original article and the supplemental materials.

Main Figures

Figure 1. Observed abundance distribution

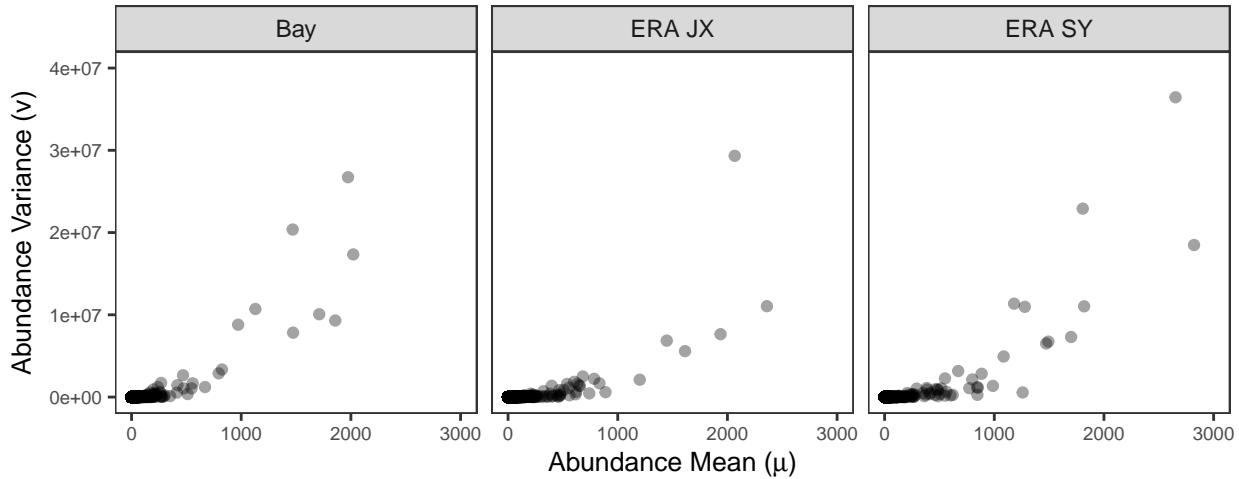
The over-dispersion pattern of microbial communities

```
load('../output/HZ/hz_mean_var.RData')
dat_plt$Region <- factor(dat_plt$Region, levels=c('Bay', 'ERA JX', 'ERA SY'))
ggplot(dat_plt, aes(x=miseqmean, y=miseqvar)) +
  geom_point(alpha=0.36, size=1.5) +
  xlab(TeX('Abundance Mean ($\mu$)')) +
  ylab(TeX('Abundance Variance ($v$)')) +
  xlim(0,3000) +
  ylim(0,4e7) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 10),
        axis.text.x = element_text(size = 7),
```

```

axis.text.y = element_text(size = 7)) +
theme(aspect.ratio=1) +
facet_grid(~Region)

```



```

load('../output/A0_ASV/ao_mean_var.RData')
dat_plt$Industry <- factor(dat_plt$Industry, levels=c('Dye','Pharmaceutical','Pesticide'))
dat_plt$Process <- factor(dat_plt$Process, levels=c('Influent','Anoxic','Oxic','Effluent'))
ggplot(dat_plt, aes(x=miseqmean, y=miseqvar)) +
  geom_point(alpha=0.36, size=1.5) +
  xlab(TeX('Abundance Mean ($\mu$)')) +
  ylab(TeX('Abundance Variance ($v$)')) +
  xlim(0,1000) +
  ylim(0,4e6) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 10),
        axis.text.x = element_text(size = 7),
        axis.text.y = element_text(size = 7)) +
  theme(aspect.ratio=1) +
  facet_grid(Industry~Process)

```

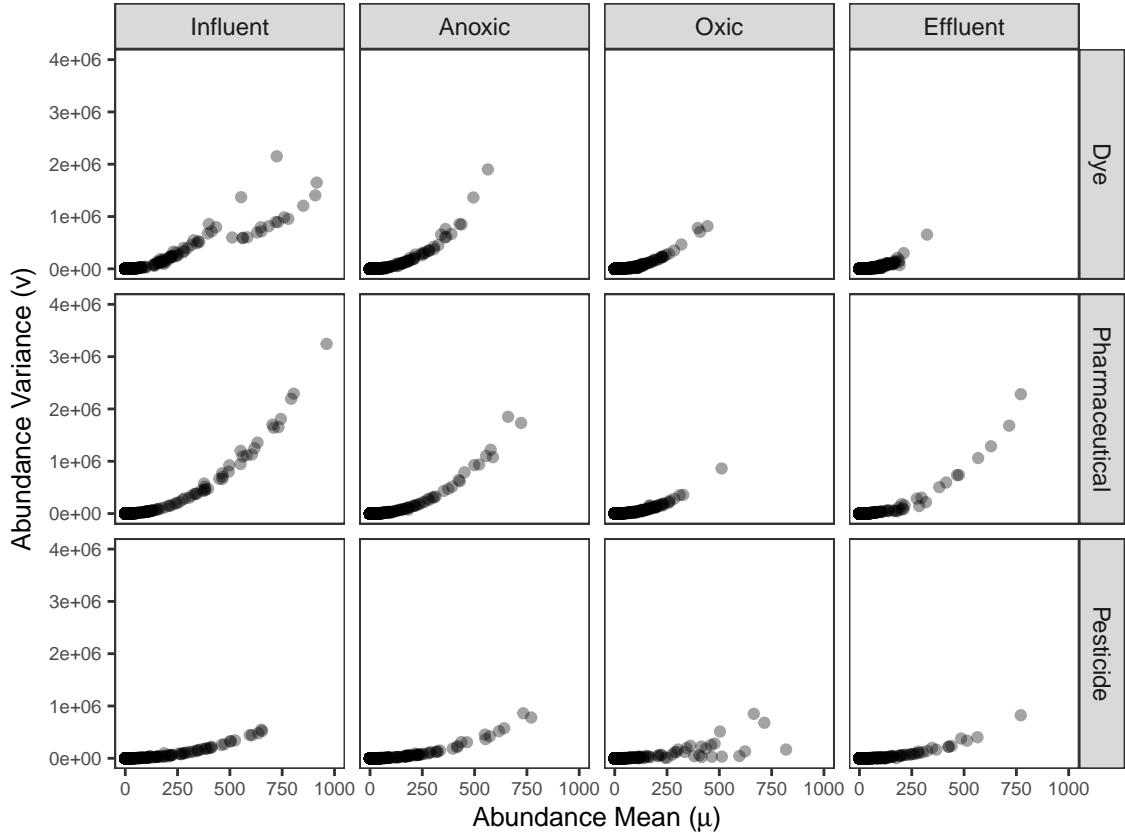


Figure 2. The Two-Wing admixed structure

Before boundary refining (Figure 2A)

```

param_plt <- data.frame()
# load AO parameter trio
vec_process <- c('influent','anoxic','oxic','effluent')
for(item in vec_process){
  file_in <- paste0('../output/AO_ASV/param_trio_asv_',item,'.RData')
  load(file_in)
  param_trio$ID <- rownames(param_trio)
  # extract poisson distributed microbes
  param_poisson <- param_trio[param_trio$k==Inf,]
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # extract zero-inflated microbes
  param_zeroinf <- param_trio[param_trio$p0!=0,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  param_plt <- rbind(param_plt, param_gpm)
}

# load HZ parameter trio
vec_process <- c('bay','era')
for(item in vec_process){

```

```

file_in <- paste0('..../output/HZ/param_trio_', item, '.RData')
load(file_in)
param_trio$ID <- rownames(param_trio)
# extract poisson distributed microbes
param_poisson <- param_trio[param_trio$k==Inf,]
# extract gamma-poisson distributed microbes
param_gpm <- param_trio[param_trio$k!=Inf,]
# extract zero-inflated microbes
param_zeroinf <- param_trio[param_trio$p0!=0,]
# annotate procedure
param_gpm$Procedure <- str_to_title(item)
param_plt <- rbind(param_plt, param_gpm)
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$Procedure <- ifelse(param_plt$Procedure=='Era', 'ERA', param_plt$Procedure)

param_plt$Procedure <- factor(param_plt$Procedure,
                               levels=c('Bay', 'Influent', 'Oxic',
                                       'ERA', 'Anoxic', 'Effluent'))

param_plt$mu_up <- 4.457417
param_plt$mu_up <- ifelse(param_plt$Procedure=='ERA',
                           4.881282, param_plt$mu_up)
param_plt$mu_up <- ifelse(param_plt$Procedure=='Influent',
                           7.142101, param_plt$mu_up)
param_plt$mu_up <- ifelse(param_plt$Procedure=='Anoxic',
                           5.209871, param_plt$mu_up)
param_plt$mu_up <- ifelse(param_plt$Procedure=='Oxic',
                           4.299989, param_plt$mu_up)
param_plt$mu_up <- ifelse(param_plt$Procedure=='Effluent',
                           4.405847, param_plt$mu_up)

param_plt$k_low <- 10.452344
param_plt$k_low <- ifelse(param_plt$Procedure=='ERA',
                           10.452344, param_plt$k_low)
param_plt$k_low <- ifelse(param_plt$Procedure=='Influent',
                           8.844581, param_plt$k_low)
param_plt$k_low <- ifelse(param_plt$Procedure=='Anoxic',
                           8.673801, param_plt$k_low)
param_plt$k_low <- ifelse(param_plt$Procedure=='Oxic',
                           8.673801, param_plt$k_low)
param_plt$k_low <- ifelse(param_plt$Procedure=='Effluent',
                           8.812485, param_plt$k_low)

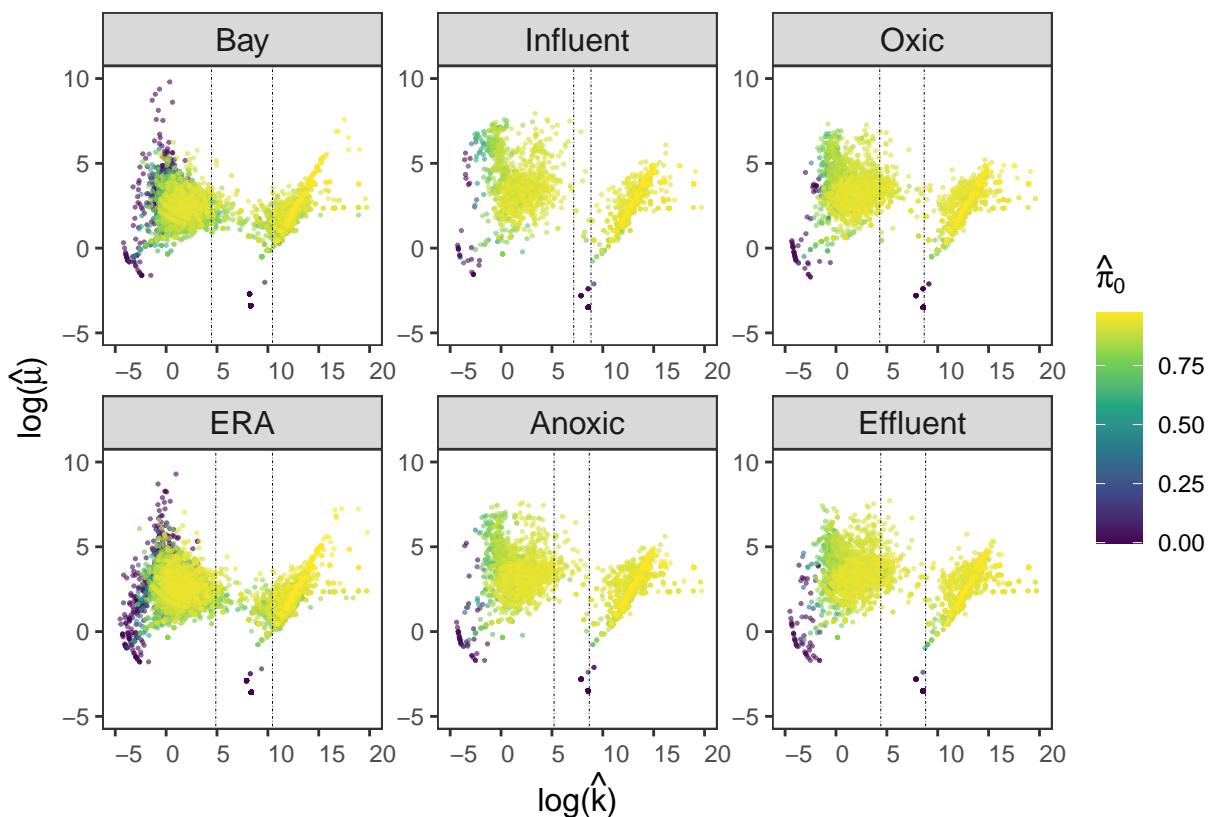
ggplot() +
  geom_point(data=param_plt,
             aes(x=log(k), y=log(mu), color=p0), alpha=0.6, size=0.24) +
  geom_vline(data=param_plt, aes(xintercept = mu_up),
             linetype="dotdash", color = "black", size=0.12) +
  geom_vline(data=param_plt, aes(xintercept = k_low),

```

```

    linetype="dotdash", color = "black", size=0.12) +
ylab(TeX(''\log{(\hat{\mu})}')) +
xlab(TeX(''\log{(\hat{k})}')) +
ggtitle(TeX('')) +
xlim(-5,20) +
ylim(-5,10) +
theme_bw() +
theme(aspect.ratio=1) +
scale_color_continuous(type = "viridis") +
facet_wrap(Procedure~., scale='free', nrow=2) +
labs(color=TeX('$\hat{\pi}_0$')) +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0),
      axis.text.y = element_text(size=9),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 12),
      legend.text = element_text(color = "black", size = 9))

```



After boundary refining (Figure 2B)

```

# load parameters mle
vec_process <- c('influent','anoxic','oxic','effluent')
param_plt <- data.frame()

```

```

for(item in vec_process){
  load(paste0('../output/AO_ASV/param_trio_asv_',item,'.RData'))
  load(paste0('../output/AO_ASV/mic_od_',item,'.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/AO_ASV/mic_id_mu_k_',item,'.RData'))
  param_gpm$Wing <- ifelse(param_gpm$ID %in% id_mu, 'mu', 'Other')
  param_gpm$Wing <- ifelse(param_gpm$ID %in% id_k, 'k', param_gpm$Wing)
  param_plt <- rbind(param_plt, param_gpm)

}

vec_process <- c('bay','era')
for(item in vec_process){
  load(paste0('../output/HZ/param_trio_',item,'.RData'))
  load(paste0('../output/HZ/mic_od_',item,'.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/HZ/mic_id_mu_k_',item,'.RData'))
  param_gpm$Wing <- ifelse(param_gpm$ID %in% id_mu, 'mu', 'Other')
  param_gpm$Wing <- ifelse(param_gpm$ID %in% id_k, 'k', param_gpm$Wing)

  param_plt <- rbind(param_plt, param_gpm)

}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$Procedure <- ifelse(param_plt$Procedure=='Era','ERA',param_plt$Procedure)

param_plt$Procedure <- factor(param_plt$Procedure,
                               levels=c('Bay','Influent','Oxic',
                                       'ERA','Anoxic','Effluent'))
param_plt$Wing <- factor(param_plt$Wing,
                         levels=c('mu','k', 'Other'))

ggplot() +
  geom_point(data=param_plt,
             aes(x=log(k), y=log(mu), color=Wing), alpha=0.36, size=0.6) +
  ylab(TeX('\\log{(\hat{\mu})}')) +
  xlab(TeX('\\log{k}')) +
  ggtitle(TeX('')) +
  xlim(-5,20) +
  ylim(-5,10) +
  theme_bw() +
  theme(aspect.ratio=1) +

```

```

facet_wrap(Procedure~, scale='free') +
scale_colour_manual(values = c('#984ea3', '#4daf4a'),
                    labels = unname(TeX(c('$\mu$', '$k$')))) +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0),
      axis.text.y = element_text(size=9),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 10),
      legend.text = element_text(color = "black", size = 9)) +
guides(color = guide_legend(override.aes = list(size = 1.8)))

```

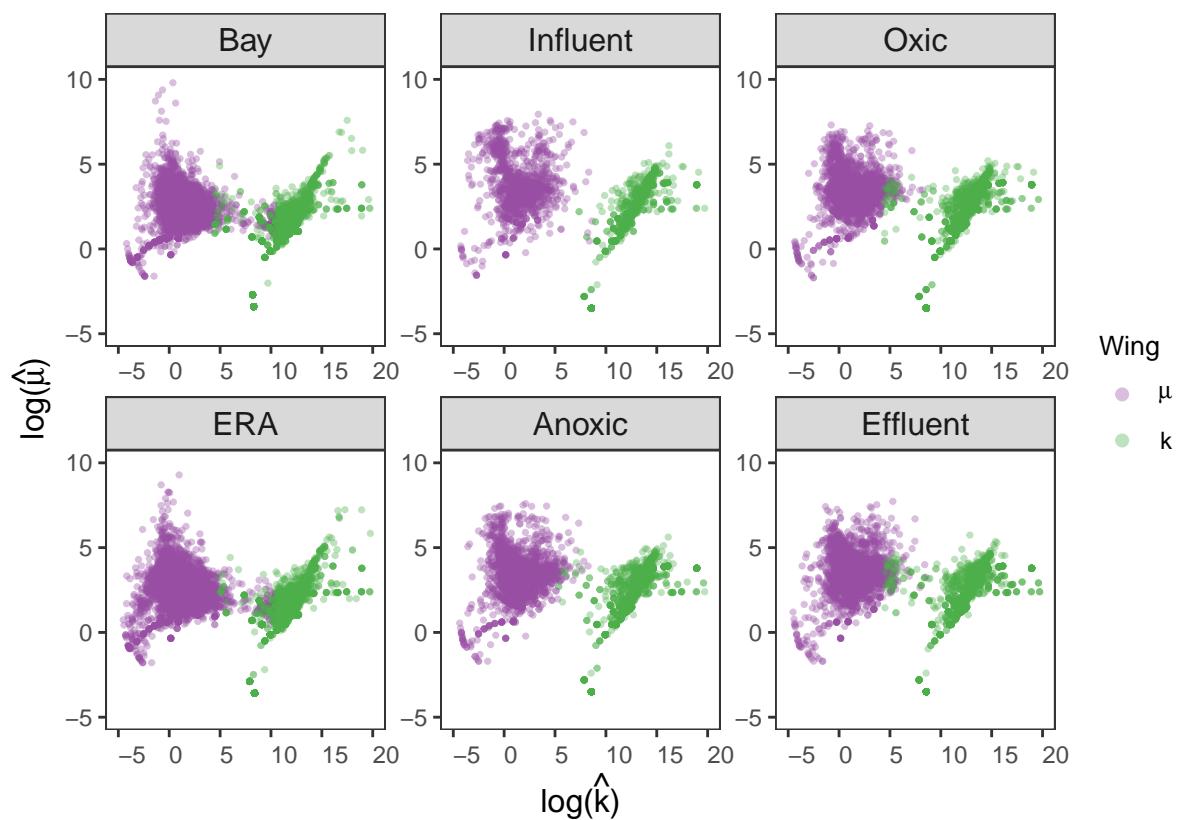


Figure 3. Dimensionality reduction analysis of μ -wing and k -wing sub-community PCoA

```

# for bay
bay_plt <- data.frame()
lab_site <- c(rep(c('HB1', 'HB2', 'HB3', 'HB4', 'HB5', 'HB6', 'HB7', 'HB8', 'HB9', 'HB10'), each=3))
load('../output/HZ/pcoa_bay.RData')

bay_plt <- rbind(bay_plt, data.frame(PCoA1=pcoa_mu$vectors[,1],
                                         PCoA2=pcoa_mu$vectors[,2],
                                         Site=lab_site,
                                         ))

```

```

                    Wing='mu-wing',
                    Region='Bay'))
bay_plt <- rbind(bay_plt, data.frame(PCoA1=pcoa_k$vectors[,1],
                                         PCoA2=pcoa_k$vectors[,2],
                                         Site=lab_site,
                                         Wing='k-wing',
                                         Region='Bay'))
bay_plt <- rbind(bay_plt, data.frame(PCoA1=pcoa_all$vectors[,1],
                                         PCoA2=pcoa_all$vectors[,2],
                                         Site=lab_site,
                                         Wing='all',
                                         Region='Bay'))
bay_plt$Site <- factor(bay_plt$Site,
                        levels=c('HB1','HB2','HB3','HB4','HB5','HB6','HB7','HB8','HB9','HB10'))
bay_plt$Wing <- factor(bay_plt$Wing,
                        levels=c('all','mu-wing','k-wing'))

g_bay <- ggplot(bay_plt, aes(x=PCoA1, y=PCoA2, color=Site)) +
  geom_point(size=0.8, alpha=0.8) +
  theme_bw() +
  xlab(TeX('PCoA1')) +
  ylab(TeX('PCoA2')) +
  ggtitle('') +
  theme(plot.title=element_text(hjust=0, size=rel(0.0)),
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 8, angle=-90, vjust=0.5, hjust=0),
        axis.text.y = element_text(size = 8),
        strip.text = element_text(size = 12),
        legend.title = element_text(color = "black", size = 12),
        legend.text = element_text(color = "black", size = 8)) +
  scale_colour_manual(values=c('#a6cee3','#1f78b4','#b2df8a','#33a02c',
                             '#fb9a99','#e31a1c','#fdbf6f','#ff7f00',
                             '#cab2d6','#6a3d9a')) +
  facet_grid2(Wing~Region, scales='free',
              independent = "all",labeler='label_parsed')

# for era
era_plt <- data.frame()
lab_site <- c(rep(c('SY1','SY2','SY3','SY4','SY5','SY6',
                     'JX1','JX2','JX3','JX4','JX5','JX6'), each=3))
load('../output/HZ/pcoa_era.RData')

era_plt <- rbind(era_plt, data.frame(PCoA1=pcoa_mu$vectors[,1],
                                         PCoA2=pcoa_mu$vectors[,2],
                                         Site=lab_site,
                                         Wing='mu-wing',
                                         Region='ERA'))
era_plt <- rbind(era_plt, data.frame(PCoA1=pcoa_k$vectors[,1],
                                         PCoA2=pcoa_k$vectors[,2],
                                         Site=lab_site,
                                         Wing='k-wing',
                                         Region='ERA'))

```

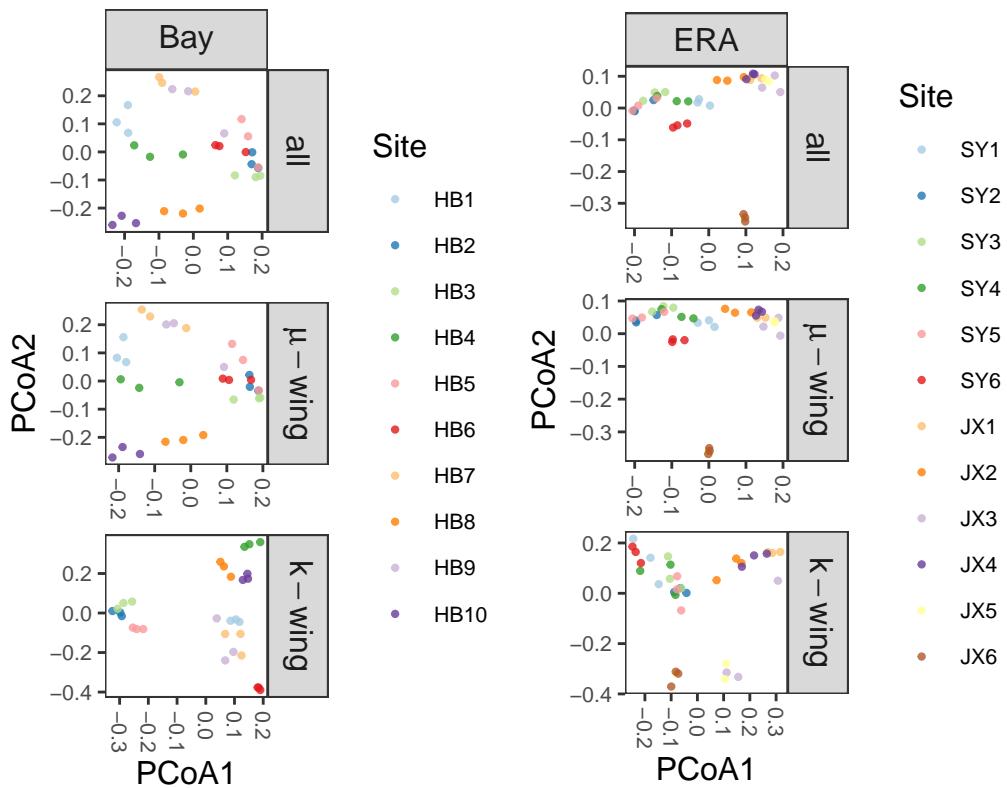
```

            Site=lab_site,
            Wing='k-wing',
            Region='ERA'))
era_plt <- rbind(era_plt, data.frame(PCoA1=pcoa_all$vectors[,1],
                                         PCoA2=pcoa_all$vectors[,2],
                                         Site=lab_site,
                                         Wing='all',
                                         Region='ERA'))
era_plt$Site <- factor(era_plt$Site,
                        levels=c('SY1','SY2','SY3','SY4','SY5','SY6','JX1','JX2','JX3','JX4','JX5','JX6'))
era_plt$Wing <- factor(era_plt$Wing,
                        levels=c('all','mu-wing','k-wing'))

g_era <- ggplot(era_plt, aes(x=PCoA1, y=PCoA2, color=Site)) +
  geom_point(size=0.8, alpha=0.8) +
  theme_bw() +
  xlab(TeX('PCoA1')) +
  ylab(TeX('PCoA2')) +
  ggtitle('') +
  theme(plot.title=element_text(hjust=0, size=rel(0.0)),
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 8, angle=-90, vjust=0.5, hjust=0),
        axis.text.y = element_text(size = 8),
        strip.text = element_text(size = 12),
        legend.title = element_text(color = "black", size = 12),
        legend.text = element_text(color = "black", size = 8)) +
  scale_colour_manual(values=c('#a6cee3','#1f78b4','#b2df8a','#33a02c',
                             '#fb9a99','#e31a1c','#fdbf6f','#ff7f00',
                             '#cab2d6','#6a3d9a','#ffff99','#b15928')) +
  facet_grid2(Wing~Region, scales='free',
              independent = "all", labeller='label_parsed')

plot_grid(NULL, g_bay, g_era, NULL,
          rel_widths = c(1,5,5,1), nrow=1, align='v')

```



```

dat_plt <- data.frame()
vec_process <- c('influent', 'anoxic', 'oxic', 'effluent')
lab_site <- c(rep(c('LS', 'SF', 'CZ', 'BA', 'YF', 'GB', 'YTSW', 'XHC', 'ZC', 'YT', 'YN'), each=3))
for(item in vec_process){
  load(paste0('../output/A0_ASV/pcoa_',item,'.RData'))
  dat_plt <- rbind(dat_plt, data.frame(PCoA1=pcoa_mu$vectors[,1],
                                         PCoA2=pcoa_mu$vectors[,2],
                                         Site=lab_site,
                                         Wing='mu-wing',
                                         Procedure=str_to_title(item)))
  dat_plt <- rbind(dat_plt, data.frame(PCoA1=pcoa_k$vectors[,1],
                                         PCoA2=pcoa_k$vectors[,2],
                                         Site=lab_site,
                                         Wing='k-wing',
                                         Procedure=str_to_title(item)))
  dat_plt <- rbind(dat_plt, data.frame(PCoA1=pcoa_all$vectors[,1],
                                         PCoA2=pcoa_all$vectors[,2],
                                         Site=lab_site,
                                         Wing='all',
                                         Procedure=str_to_title(item)))
}

dat_plt$Wing <- factor(dat_plt$Wing, levels=c('all', 'mu-wing', 'k-wing'))
dat_plt$Procedure <- factor(dat_plt$Procedure, levels=str_to_title(vec_process))

g_pre <- ggplot(dat_plt, aes(x=PCoA1, y=PCoA2, color=Site)) +
  geom_point() +

```

```

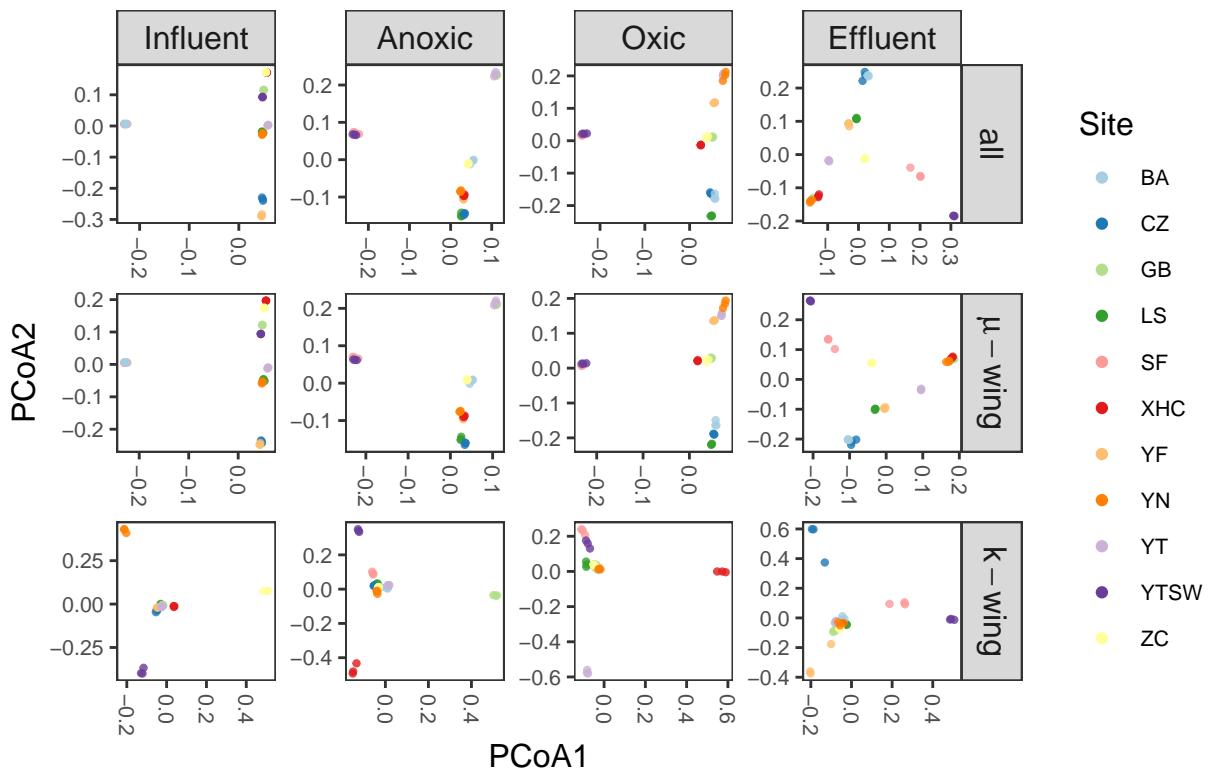
theme_bw() +
xlab(TeX('PCoA1')) +
ylab(TeX('PCoA2')) +
ggtitle('') +
theme(plot.title=element_text(hjust=0, size=rel(0.0)),
      aspect.ratio=1,
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 8, angle=-90, vjust=0.5, hjust=0),
      axis.text.y = element_text(size = 8),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 12),
      legend.text = element_text(color = "black", size = 8)) +
scale_colour_manual(values=c('#a6cee3', '#1f78b4', '#b2df8a', '#33a02c',
                           '#fb9a99', '#e31a1c', '#fdbf6f', '#ff7f00',
                           '#cab2d6', '#6a3d9a', '#ffff99')) +
facet_grid2(Wing~Procedure, scales='free',
            independent = "all", labeller='label_parsed')

g_leg <- get_legend(g_pre)

g_main <- ggplot(dat_plt, aes(x=PCoA1, y=PCoA2, color=Site)) +
geom_point(size=0.8, alpha=0.8) +
theme_bw() +
xlab(TeX('PCoA1')) +
ylab(TeX('PCoA2')) +
ggtitle('') +
theme(plot.title=element_text(hjust=0, size=rel(0.0)),
      aspect.ratio=1,
      legend.position="none",
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 8, angle=-90, vjust=0.5, hjust=0),
      axis.text.y = element_text(size = 8),
      strip.text = element_text(size = 12)) +
scale_colour_manual(values=c('#a6cee3', '#1f78b4', '#b2df8a', '#33a02c',
                           '#fb9a99', '#e31a1c', '#fdbf6f', '#ff7f00',
                           '#cab2d6', '#6a3d9a', '#ffff99')) +
facet_grid2(Wing~Procedure, scales='free',
            independent = "all", labeller='label_parsed')

plot_grid(g_main, g_leg,
          rel_widths = c(5, 1), nrow=1)

```



PCA

Follow similar algorithm as above. Specifically, run `../figures/vignettes/pca_hz.Rmd` and `../figures/vignettes/pca_ao.Rmd`

CCA

Follow similar algorithm as above. Specifically, run `../figures/vignettes/cca_ao_ori.Rmd`

Figure 4. The *Two-Wing* structure generalizes relative-abundance-based categorizing method

Using *Method 0* to identify microbial categories

```
# load parameters mle
vec_process <- c('influent', 'anoxic', 'oxic', 'effluent')
param_plt <- data.frame()
for(item in vec_process){
  load(paste0('../output/AO_ASV/param_trio_asv_', item, '.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/AO_ASV/taxa_category_', item, '.RData'))}
```

```

param_gpm$Category <- ifelse(param_gpm$ID %in% id_rt, 'RT', 'Other')
param_gpm$Category <- ifelse(param_gpm$ID %in% id_mt, 'IT', param_gpm$Category)
param_gpm$Category <- ifelse(param_gpm$ID %in% id_at, 'AT', param_gpm$Category)
param_gpm$Category <- ifelse(param_gpm$ID %in% id_crt, 'CRT', param_gpm$Category)
param_gpm$Category <- ifelse(param_gpm$ID %in% id_cat, 'CAT', param_gpm$Category)
param_gpm$Category <- ifelse(param_gpm$ID %in% id_crat, 'CRAT', param_gpm$Category)

param_plt <- rbind(param_plt, param_gpm)

}

vec_process <- c('bay','era')
for(item in vec_process){
  load(paste0('../output/HZ/param_trio_',item,'.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/HZ/taxa_category_',item,'.RData'))
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_rt, 'RT', 'Other')
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_mt, 'IT', param_gpm$Category)
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_at, 'AT', param_gpm$Category)
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_crt, 'CRT', param_gpm$Category)
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_cat, 'CAT', param_gpm$Category)
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_crat, 'CRAT', param_gpm$Category)

  param_plt <- rbind(param_plt, param_gpm)
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$Procedure <- ifelse(param_plt$Procedure=='Era','ERA',param_plt$Procedure)

param_plt$Procedure <- factor(param_plt$Procedure,
                               levels=c('Bay','Influent','Oxic',
                                       'ERA','Anoxic','Effluent'))
param_plt$Category <- factor(param_plt$Category,
                             levels=c('AT','CAT','CRAT',
                                     'IT','CRT','RT'))

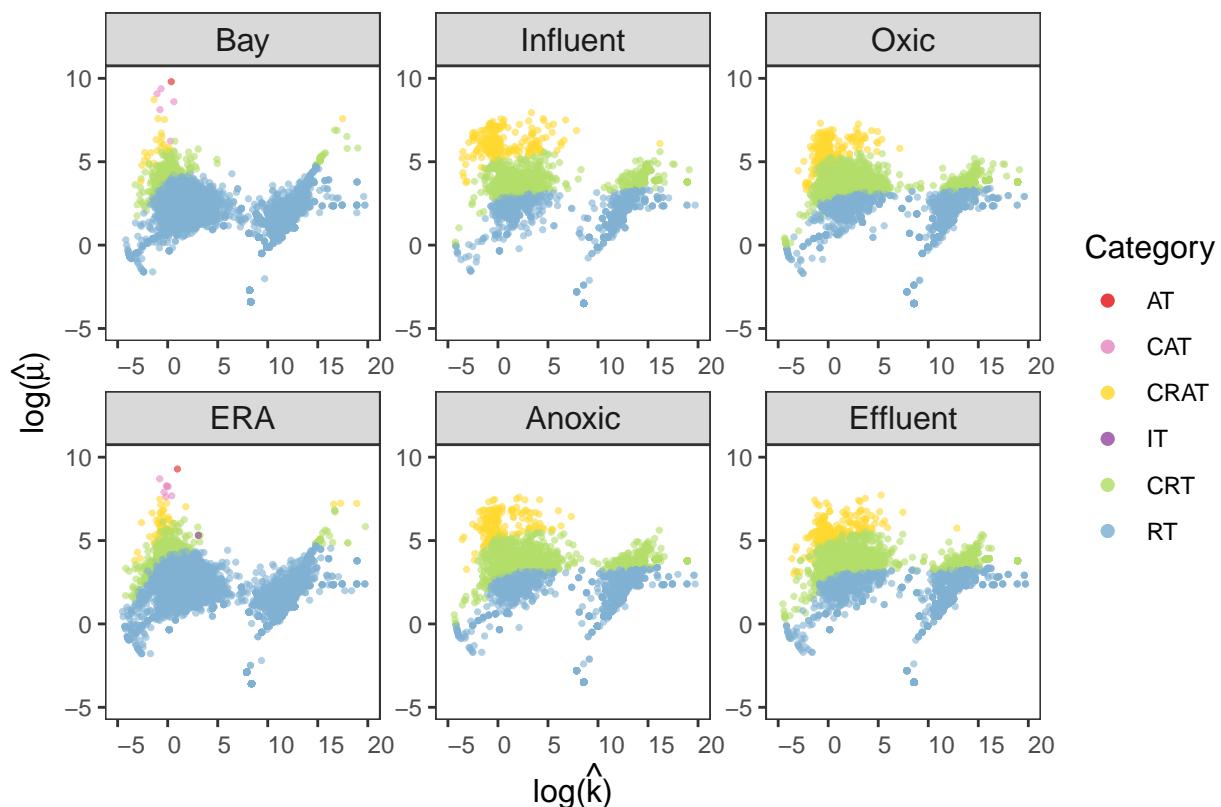
ggplot() +
  geom_point(data=param_plt,
             aes(x=log(k), y=log(mu), color=Category), alpha=0.6,size=0.6) +
  geom_point(data=param_plt[param_plt$Category=='IT',],
             aes(x=log(k), y=log(mu), color=Category), alpha=0.6,size=0.6) +
  ylab(TeX('\\log{(\hat{\mu})}')) +
  xlab(TeX('\\log{k}')) +
  ggtitle(TeX('')) +
  xlim(-5,20) +
  ylim(-5,10) +
  theme_bw()

```

```

theme(aspect.ratio=1) +
facet_wrap(Procedure~., scale='free') +
scale_colour_manual(values = c('#e41a1c', '#e78ac3', '#ffd92f',
                             '#984ea3', '#b3de69', '#80b1d3')) +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0),
      axis.text.y = element_text(size=9),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 12),
      legend.text = element_text(color = "black", size = 9)) +
guides(color = guide_legend(override.aes = list(size = 1.8)))

```



Using Method a-c to identify microbial categories

The algorithm is similar as above. Specifically, run `../figures/vignettes/main/mu_k_category_abcd.Rmd`

Figure 5. Comparing RT & CRT in μ -wing vs. k -wing

Comparing CCA results

```

sapply(list.files(pattern=".R$",
                  path="..../R/PMCosm",
                  full.names=TRUE),
      source)
# load abundance data

```

```

mic <- read.table(file='../../data/2018/resample_feature-table.txt',
                  header=TRUE, row.names=1)
# construct column id
bay_loci <- c('HB1.1', 'HB1.2', 'HB1.3', 'HB2.1', 'HB2.2', 'HB2.3',
              'HB3.1', 'HB3.2', 'HB3.3', 'HB4.1', 'HB4.2', 'HB4.3',
              'HB5.1', 'HB5.2', 'HB5.3', 'HB6.1', 'HB6.2', 'HB6.3',
              'HB7.1', 'HB7.2', 'HB7.3', 'HB8.1', 'HB8.2', 'HB8.3',
              'HB9.1', 'HB9.2', 'HB9.3', 'HB10.1', 'HB10.2', 'HB10.3')
era_sy_loci <- c('SY1.1', 'SY1.2', 'SY1.3', 'SY2.1', 'SY2.2', 'SY2.3',
                  'SY3.1', 'SY3.2', 'SY3.3', 'SY4.1', 'SY4.2', 'SY4.3',
                  'SY5.1', 'SY5.2', 'SY5.3', 'SY6.1', 'SY6.2', 'SY6.3')
era_jx_loci <- c('JX1.1', 'JX1.2', 'JX1.3', 'JX2.1', 'JX2.2', 'JX2.3',
                  'JX3.1', 'JX3.2', 'JX3.3', 'JX4.1', 'JX4.2', 'JX4.3',
                  'JX5.1', 'JX5.2', 'JX5.3', 'JX6.1', 'JX6.2', 'JX6.3')
# load environment factors
env <- read.csv('../../data/2018/env_table.csv', row.names=1)
env_rep <- env[rep(seq_len(nrow(env)), each = 3), ]
rownames(env_rep) <- c(era_sy_loci, era_jx_loci, bay_loci)

# for bay
id_col <- bay_loci
load('../output/HZ/mic_id_mu_k_bay.RData')
load('../output/HZ/taxa_category_bay.RData')
# extract abundance
mic_mu <- mic[intersect(id_mu, c(id_rt, id_crt)), id_col]
mic_k <- mic[intersect(id_k, c(id_rt, id_crt)), id_col]
# cca analysis
# original data set, species on columns
cc_mu_rt_crt <- cca(t(mic_mu), env_rep[id_col,])
cc_k_rt_crt <- cca(t(mic_k), env_rep[id_col,])
save(cc_mu_rt_crt, cc_k_rt_crt, file='../output/HZ/cca_mu_k_rt_crt_bay.RData')

# for era
id_col <- c(era_sy_loci, era_jx_loci)
load('../output/HZ/mic_id_mu_k_era.RData')
load('../output/HZ/taxa_category_era.RData')
# extract abundance
mic_mu <- mic[intersect(id_mu, c(id_rt, id_crt)), id_col]
mic_k <- mic[intersect(id_k, c(id_rt, id_crt)), id_col]
# cca analysis
# original data set, species on columns
cc_mu_rt_crt <- cca(t(mic_mu), env_rep[id_col, 1:ncol(env_rep)])
cc_k_rt_crt <- cca(t(mic_k), env_rep[id_col, 1:ncol(env_rep)])
save(cc_mu_rt_crt, cc_k_rt_crt, file='../output/HZ/cca_mu_k_rt_crt_era.RData')

load('../output/HZ/cca_mu_k_rt_crt_bay.RData')
cc_mu_bay <- cc_mu_rt_crt
cc_k_bay <- cc_k_rt_crt

load('../output/HZ/cca_mu_k_rt_crt_era.RData')
cc_mu_era <- cc_mu_rt_crt
cc_k_era <- cc_k_rt_crt

```

```

g_bay_mu <- autoplot(cc_mu_bay) +
  theme_bw() +
  xlab(TeX('CCA1')) +
  ylab(TeX('CCA2')) +
  ggtitle(TeX('Bay RT & CRT $\backslash\mu\$-wing')) +
  theme(plot.title=element_text(size = 16, hjust = 0.5, vjust = 0),
        legend.position = c(0.2,0.2),
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(color = "black", size = 14),
        legend.text = element_text(color = "black", size = 10))

g_bay_k <- autoplot(cc_k_bay) +
  theme_bw() +
  xlab(TeX('CCA1')) +
  ylab(TeX('CCA2')) +
  ggtitle(TeX('Bay RT & CRT $k\$-wing')) +
  theme(plot.title=element_text(size = 16, hjust = 0.5, vjust = 0),
        legend.position = "none",
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(color = "black", size = 14),
        legend.text = element_text(color = "black", size = 10))

g_era_mu <- autoplot(cc_mu_era) +
  theme_bw() +
  xlab(TeX('CCA1')) +
  ylab(TeX('CCA2')) +
  xlim(c(-1.5,4)) +
  ggtitle(TeX('ERA RT & CRT $\backslash\mu\$-wing')) +
  theme(plot.title=element_text(size = 16, hjust = 0.5, vjust = 0),
        legend.position = "none",
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(color = "black", size = 14),
        legend.text = element_text(color = "black", size = 10))

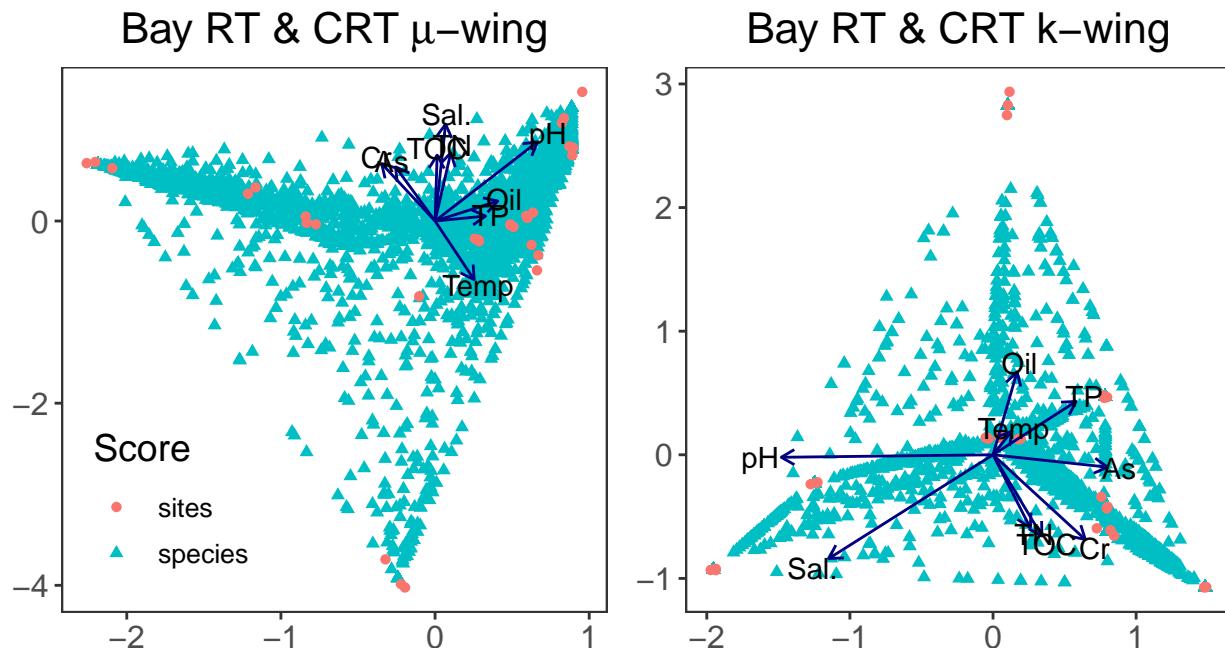
```

```

g_era_k <- autoplot(cc_k_era) +
  theme_bw() +
  xlab(TeX('CCA1')) +
  ylab(TeX('CCA2')) +
  xlim(c(-1.5,4)) +
  ggtitle(TeX('ERA RT & CRT $k\$-wing')) +
  theme(plot.title=element_text(size = 16, hjust = 0.5, vjust = 0),
        legend.position = "none",
        aspect.ratio=1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(color = "black", size = 14),
        legend.text = element_text(color = "black", size = 10))

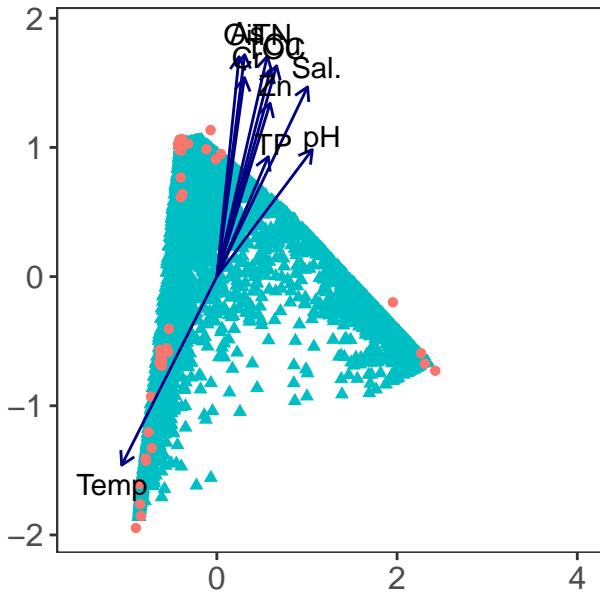
```

```
plot_grid(g_bay_mu, g_bay_k, nrow=1, align='v')
```

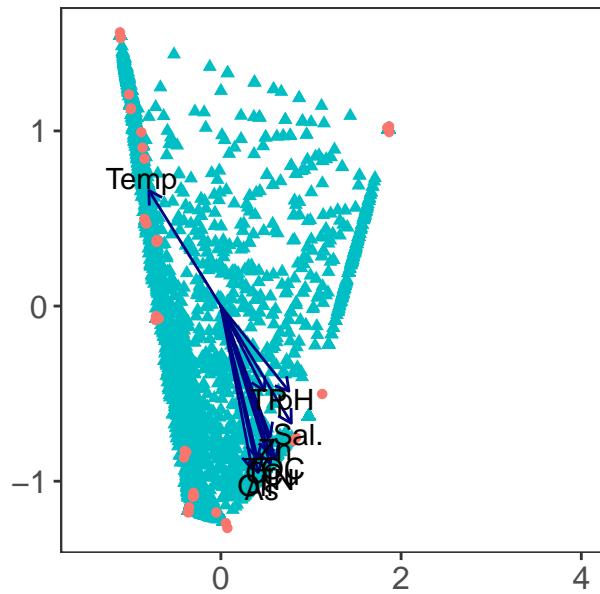


```
plot_grid(g_era_mu, g_era_k, nrow=1, align='v')
```

ERA RT & CRT μ -wing



ERA RT & CRT k-wing



Comparing Beta diversity

```
sapply(list.files(pattern=".R$", path="../R/PMCosm", full.names=TRUE), source)
# load abundance data
mic <- read.table(file='../data/2018/resample_feature-table.txt',
                  header=TRUE, row.names=1)
# construct column id
bay_loci <- c('HB1.1', 'HB1.2', 'HB1.3', 'HB2.1', 'HB2.2', 'HB2.3',
              'HB3.1', 'HB3.2', 'HB3.3', 'HB4.1', 'HB4.2', 'HB4.3',
              'HB5.1', 'HB5.2', 'HB5.3', 'HB6.1', 'HB6.2', 'HB6.3',
              'HB7.1', 'HB7.2', 'HB7.3', 'HB8.1', 'HB8.2', 'HB8.3',
              'HB9.1', 'HB9.2', 'HB9.3', 'HB10.1', 'HB10.2', 'HB10.3')
era_sy_loci <- c('SY1.1', 'SY1.2', 'SY1.3', 'SY2.1', 'SY2.2', 'SY2.3',
                  'SY3.1', 'SY3.2', 'SY3.3', 'SY4.1', 'SY4.2', 'SY4.3',
                  'SY5.1', 'SY5.2', 'SY5.3', 'SY6.1', 'SY6.2', 'SY6.3')
era_jx_loci <- c('JX1.1', 'JX1.2', 'JX1.3', 'JX2.1', 'JX2.2', 'JX2.3',
                  'JX3.1', 'JX3.2', 'JX3.3', 'JX4.1', 'JX4.2', 'JX4.3',
                  'JX5.1', 'JX5.2', 'JX5.3', 'JX6.1', 'JX6.2', 'JX6.3')

# for bay
load('../output/HZ/taxa_category_bay.RData')
load('../output/HZ/param_trio_bay.RData')
param_trio <- param_trio[param_trio$k!=Inf,]
id_mu_wing <- rownames(param_trio[param_trio$k<exp(7),])
id_k_wing <- rownames(param_trio[param_trio$k>exp(7),])
id_col <- bay_loci
mic_rt_mu <- mic[intersect(id_rt, id_mu_wing), id_col]
mic_rt_k <- mic[intersect(id_rt, id_k_wing), id_col]
mic_crt_mu <- mic[intersect(id_crt, id_mu_wing), id_col]
mic_crt_k <- mic[intersect(id_crt, id_k_wing), id_col]
mic_rt_crt_mu <- mic[intersect(c(id_rt, id_crt), id_mu_wing), id_col]
```

```

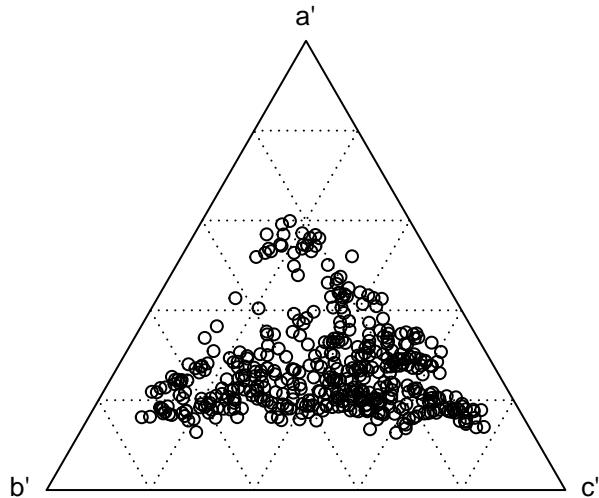
mic_rt_crt_k <- mic[intersect(c(id_rt,id_crt), id_k_wing), id_col]
# triangular plot data
bd_rt_mu <- betadiver(t(count_to_preabs(mic_rt_mu)))
bd_rt_k <- betadiver(t(count_to_preabs(mic_rt_k)))
bd_crt_mu <- betadiver(t(count_to_preabs(mic_crt_mu)))
bd_crt_k <- betadiver(t(count_to_preabs(mic_crt_k)))
bd_rt_crt_mu <- betadiver(t(count_to_preabs(mic_rt_crt_mu)))
bd_rt_crt_k <- betadiver(t(count_to_preabs(mic_rt_crt_k)))
save(bd_rt_mu, bd_rt_k, bd_crt_mu, bd_crt_k, bd_rt_crt_mu, bd_rt_crt_k,
     file='../output/HZ/bd_rt_crt_bay.RData')

# for era
load('../output/HZ/taxa_category_era.RData')
load('../output/HZ/param_trio_era.RData')
param_trio <- param_trio[param_trio$K!=Inf,]
id_mu_wing <- rownames(param_trio[param_trio$K<exp(7),])
id_k_wing <- rownames(param_trio[param_trio$K>exp(7),])
id_col <- c(era_sy_loci, era_jx_loci)
mic_rt_mu <- mic[intersect(id_rt, id_mu_wing), id_col]
mic_rt_k <- mic[intersect(id_rt, id_k_wing), id_col]
mic_crt_mu <- mic[intersect(id_crt, id_mu_wing), id_col]
mic_crt_k <- mic[intersect(id_crt, id_k_wing), id_col]
mic_rt_crt_mu <- mic[intersect(c(id_rt,id_crt), id_mu_wing), id_col]
mic_rt_crt_k <- mic[intersect(c(id_rt,id_crt), id_k_wing), id_col]
# triangular plot data
bd_rt_mu <- betadiver(t(count_to_preabs(mic_rt_mu)))
bd_rt_k <- betadiver(t(count_to_preabs(mic_rt_k)))
bd_crt_mu <- betadiver(t(count_to_preabs(mic_crt_mu)))
bd_crt_k <- betadiver(t(count_to_preabs(mic_crt_k)))
bd_rt_crt_mu <- betadiver(t(count_to_preabs(mic_rt_crt_mu)))
bd_rt_crt_k <- betadiver(t(count_to_preabs(mic_rt_crt_k)))
save(bd_rt_mu, bd_rt_k, bd_crt_mu, bd_crt_k, bd_rt_crt_mu, bd_rt_crt_k,
     file='../output/HZ/bd_rt_crt_era.RData')

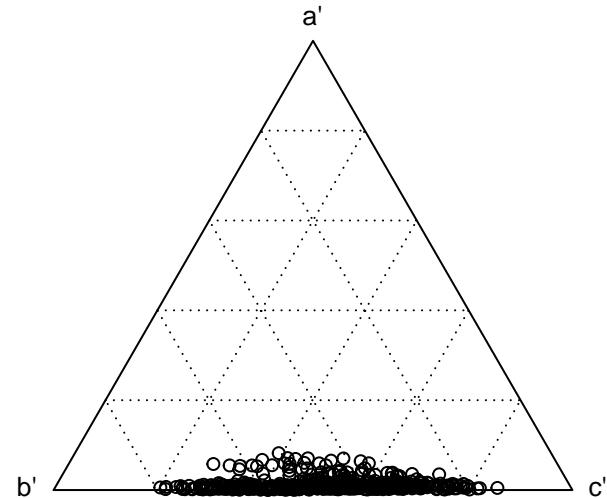
load('../output/HZ/bd_rt_crt_bay.RData')
par(mfrow=c(2,2))
par(mar=c(t=1,l=1,b=0,r=1))
layout(matrix(c(1,2,3,4),ncol=2),heights=c(1,5,1,5))
plot.new()
text(0.5,0.2,TeX("Bay RT & CRT $\backslash\mu$-wing"),cex=1.5,font=1)
plot(bd_rt_crt_mu)
plot.new()
text(0.5,0.2,TeX("Bay RT & CRT $k$-wing"),cex=1.5,font=1)
plot(bd_rt_crt_k)

```

Bay RT & CRT μ -wing

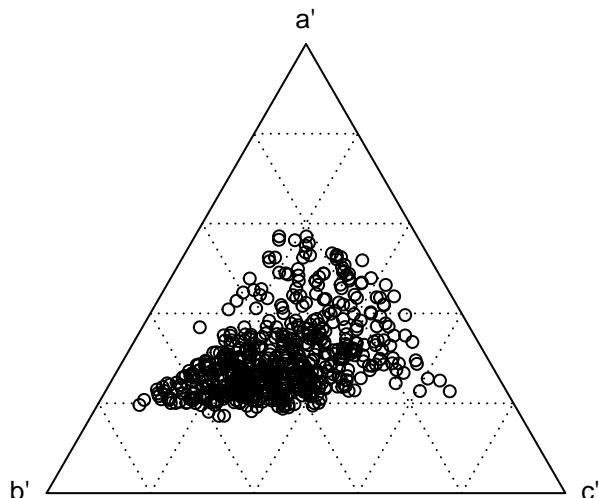


Bay RT & CRT k-wing



```
load('../output/HZ/bd_rt_crt_era.RData')
par(mfrow=c(2,2))
par(mar=c(t=1,l=1,b=0,r=1))
layout(matrix(c(1,2,3,4),ncol=2),heights=c(1,5,1,5))
plot.new()
text(0.5,0.2,TeX("ERA RT & CRT $\backslash\mu\$-wing"),cex=1.5,font=1)
plot(bd_rt_crt_mu)
plot.new()
text(0.5,0.2,TeX("ERA RT & CRT $k\$-wing"),cex=1.5,font=1)
plot(bd_rt_crt_k)
```

ERA RT & CRT μ -wing



ERA RT & CRT k-wing

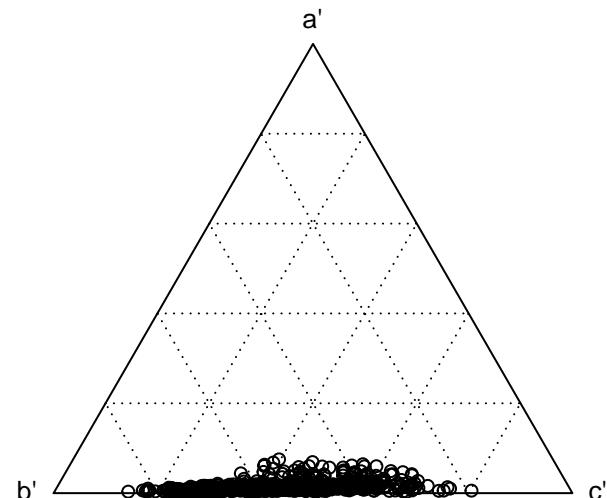


Figure 6. Comparing preferences about habitats and life strategies between μ -wing microbes and k -wing microbes

Generalists and Specialists

```

# load parameters mle
vec_process <- c('influent','anoxic','oxic','effluent')
param_plt <- data.frame()
for(item in vec_process){
  load(paste0('../output/A0_ASV/param_trio_asv_',item,'.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/A0_ASV/taxa_category_niche_',item,'.RData'))
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_g, 'Generalist', 'Other')
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_s, 'Specialist', param_gpm$Category)

  param_plt <- rbind(param_plt, param_gpm)
}

# load parameters mle
vec_region <- c('bay','era')
for(item in vec_region){
  load(paste0('../output/HZ/param_trio_',item,'.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # annotate taxa category
  load(paste0('../output/HZ/taxa_category_niche_',item,'.RData'))
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_g, 'Generalist', 'Other')
  param_gpm$Category <- ifelse(param_gpm$ID %in% id_s, 'Specialist', param_gpm$Category)

  param_plt <- rbind(param_plt, param_gpm)
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$Procedure <- ifelse(param_plt$Procedure=='Era','ERA',param_plt$Procedure)
param_plt$Procedure <- factor(param_plt$Procedure,
                               levels=c('Bay','Influent','Oxic',
                                       'ERA','Anoxic','Effluent'))
param_plt$Category <- factor(param_plt$Category,
                               levels=c('Generalist','Specialist','Other'))

g_pre <- ggplot() +
  geom_point(data=param_plt,
             aes(x=log(k), y=log(mu), color=Category), alpha=1,size=0.6) +

```

```

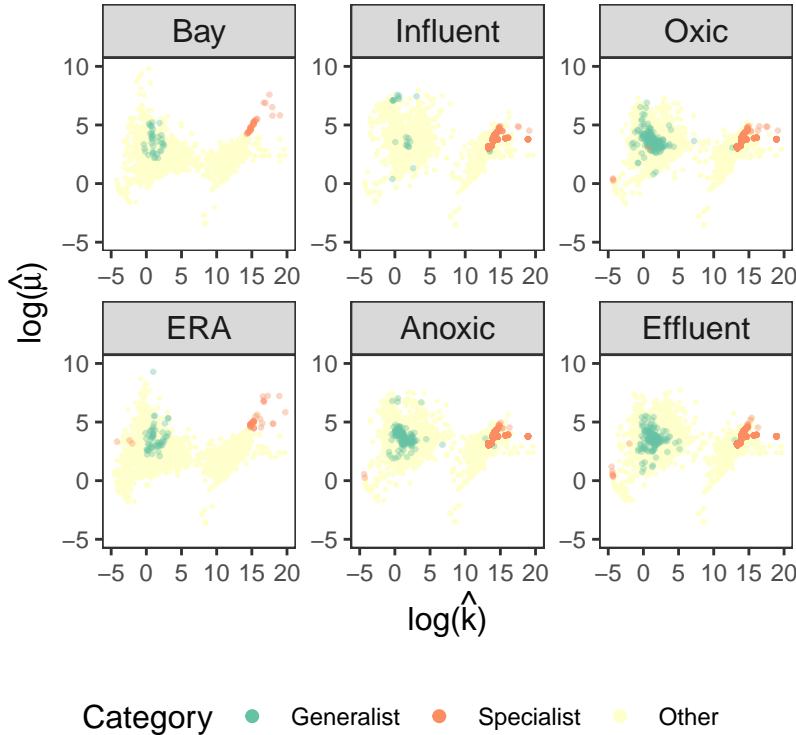
ylab(TeX('\'\log{(\hat{\mu})}')) +
xlab(TeX('\'\log{(\hat{k})}')) +
ggtitle(TeX('')) +
xlim(-5,20) +
ylim(-5,10) +
theme_bw() +
theme(aspect.ratio=1) +
facet_wrap(Procedure~., scale='free', nrow=2) +
scale_colour_manual(values = c('#66c2a5','fc8d62','#ffffcc')) +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      legend.position = "bottom",
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0.5),
      axis.text.y = element_text(size=9),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 12),
      legend.text = element_text(color = "black", size = 9)) +
guides(color = guide_legend(nrow = 1, override.aes = list(size = 1.8)))

g_leg <- get_legend(g_pre)

g1 <- ggplot() +
  geom_point(data=param_plt[param_plt$Category=='Other',],
             aes(x=log(k), y=log(mu)), color='#ffffcc', alpha=1, size=0.24) +
  geom_point(data=param_plt[param_plt$Category=='Generalist',],
             aes(x=log(k), y=log(mu)), color='#66c2a5', alpha=0.36, size=0.48) +
  geom_point(data=param_plt[param_plt$Category=='Specialist',],
             aes(x=log(k), y=log(mu)), color='fc8d62', alpha=0.36, size=0.48) +
  ylab(TeX('\'\log{(\hat{\mu})}')) +
  xlab(TeX('\'\log{(\hat{k})}')) +
  ggtitle(TeX('')) +
  xlim(-5,20) +
  ylim(-5,10) +
  theme_bw() +
  theme(aspect.ratio=1) +
  facet_wrap(Procedure~., scale='free', nrow=2) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "none",
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0.5),
        axis.text.y = element_text(size=9),
        strip.text = element_text(size = 12)) +
  guides(color = guide_legend(nrow = 1, override.aes = list(size = 1.8)))

plot_grid(g1, g_leg, NULL,
          rel_heights = c(6, 1, 0.5), ncol=1)

```



Category ● Generalist ● Specialist ● Other

r-strategists and *K*-strategists

```
sapply(list.files(pattern=".R$", path="../R/PMCosm", full.names=TRUE), source)

# HZ data
# load abundance data
mic <- read.table(file='../data/2018/resample_feature-table.txt',
                  header=TRUE, row.names=1)
# load annotations
anno <- read.csv('../data/2018/taxa_anno.csv')
# load rrn data base
rrn <- read.table(file = '../data/rrnDB/rrnDB-5.8_pantaxa_stats_NCBI.tsv', sep = '\t', header = TRUE)

mic$ID <- rownames(mic)

# construct column id
bay_loci <- c('HB1.1', 'HB1.2', 'HB1.3', 'HB2.1', 'HB2.2', 'HB2.3',
               'HB3.1', 'HB3.2', 'HB3.3', 'HB4.1', 'HB4.2', 'HB4.3',
               'HB5.1', 'HB5.2', 'HB5.3', 'HB6.1', 'HB6.2', 'HB6.3',
               'HB7.1', 'HB7.2', 'HB7.3', 'HB8.1', 'HB8.2', 'HB8.3',
               'HB9.1', 'HB9.2', 'HB9.3', 'HB10.1', 'HB10.2', 'HB10.3')
era_sy_loci <- c('SY1.1', 'SY1.2', 'SY1.3', 'SY2.1', 'SY2.2', 'SY2.3',
                  'SY3.1', 'SY3.2', 'SY3.3', 'SY4.1', 'SY4.2', 'SY4.3',
                  'SY5.1', 'SY5.2', 'SY5.3', 'SY6.1', 'SY6.2', 'SY6.3')
era_jx_loci <- c('JX1.1', 'JX1.2', 'JX1.3', 'JX2.1', 'JX2.2', 'JX2.3',
                  'JX3.1', 'JX3.2', 'JX3.3', 'JX4.1', 'JX4.2', 'JX4.3',
                  'JX5.1', 'JX5.2', 'JX5.3', 'JX6.1', 'JX6.2', 'JX6.3')
```

```

hz_rrn_k <- data.frame()
hz_rrn_mu <- data.frame()

# for bay
id_col <- bay_loci
load('../output/HZ/mic_id_mu_k_bay.RData')
param_anno_mu <- data.frame(ID = id_mu)
param_anno_k <- data.frame(ID = id_k)
param_anno_mu <- merge(param_anno_mu, anno, by='ID')
param_anno_k <- merge(param_anno_k, anno, by='ID')
# annotate rrn
rrn_mu <- rrn_anno(param_anno_mu, rrn)
rrn_k <- rrn_anno(param_anno_k, rrn)
# compute community rrn
rrn_mu <- merge(rrn_mu, mic[,c('ID',id_col)])
rrn_k <- merge(rrn_k, mic[,c('ID',id_col)])
crrn_mu <- calc_comm_rrn(rrn_mu)
crrn_k <- calc_comm_rrn(rrn_k)
hz_rrn_k <- rbind(hz_rrn_k,
                     data.frame(crrn = crrn_k,
                                mean = mean(crrn_k),
                                sd = sd(crrn_k),
                                ci_u = mean(crrn_k) + sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                                ci_l = mean(crrn_k) - sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                                Wing = 'k',
                                Wing.Data = 'k.HZ',
                                Process = 'Bay'))
hz_rrn_mu <- rbind(hz_rrn_mu,
                     data.frame(crrn = crrn_mu,
                                mean = mean(crrn_mu),
                                sd = sd(crrn_mu),
                                ci_u = mean(crrn_mu) + sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                                ci_l = mean(crrn_mu) - sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                                Wing = 'mu',
                                Wing.Data = 'mu.HZ',
                                Process = 'Bay'))

# for era
id_col <- c(era_sy_loci, era_jx_loci)
load('../output/HZ/mic_id_mu_k_era.RData')
param_anno_mu <- data.frame(ID = id_mu)
param_anno_k <- data.frame(ID = id_k)
param_anno_mu <- merge(param_anno_mu, anno, by='ID')
param_anno_k <- merge(param_anno_k, anno, by='ID')
# annotate rrn
rrn_mu <- rrn_anno(param_anno_mu, rrn)
rrn_k <- rrn_anno(param_anno_k, rrn)
# compute community rrn
rrn_mu <- merge(rrn_mu, mic[,c('ID',id_col)])
rrn_k <- merge(rrn_k, mic[,c('ID',id_col)])
crrn_mu <- calc_comm_rrn(rrn_mu)
crrn_k <- calc_comm_rrn(rrn_k)

```

```

hz_rrn_k <- rbind(hz_rrn_k,
                     data.frame(crrn = crrn_k,
                                mean = mean(crrn_k),
                                sd = sd(crrn_k),
                                ci_u = mean(crrn_k) + sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                                ci_l = mean(crrn_k) - sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                                Wing = 'k',
                                Wing.Data = 'k.HZ',
                                Process = 'ERA'))

hz_rrn_mu <- rbind(hz_rrn_mu,
                     data.frame(crrn = crrn_mu,
                                mean = mean(crrn_mu),
                                sd = sd(crrn_mu),
                                ci_u = mean(crrn_mu) + sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                                ci_l = mean(crrn_mu) - sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                                Wing = 'mu',
                                Wing.Data = 'mu.HZ',
                                Process = 'ERA'))

save(hz_rrn_k, hz_rrn_mu,
      file='../output/HZ/crrn_asv_5v8_ncbi.RData')

# AO data
# load abundance data
mic <- read.table('../data/AO/ASV_table.txt')
# load annotations
anno <- read.csv('../data/AO/ASV_Anno.csv')
# load rrn data base
rrn <- read.table(file = '../data/rrnDB/rrnDB-5.8_pantaxa_stats_NCBI.tsv', sep = '\t', header = TRUE)

mic$ID <- rownames(mic)
id_dye <- c('LS', 'SF', 'CZ', 'BA', 'YF')
id_med <- c('GB', 'YTSW', 'XHC', 'ZC')
id_pes <- c('YT', 'YN')
lis_ind <- list(id_dye, id_med, id_pes)
names(lis_ind) <- c('Dye', 'Medicine', 'Pesticide')
id_inf <- c('Inf1', 'Inf2', 'Inf3')
id_axi <- c('Ax1', 'Ax2', 'Ax3')
id_oxi <- c('Ox1', 'Ox2', 'Ox3')
id_eff <- c('Eff1', 'Eff2', 'Eff3')
lis_pro <- list(id_inf, id_axi, id_oxi, id_eff)
names(lis_pro) <- c('Influent', 'Anoxic', 'Oxic', 'Effluent')
ao_rrn_k <- data.frame()
ao_rrn_mu <- data.frame()

for(region in 1:4){
  # extract column id
  id_col <- c()
  for(i in 1:length(lis_ind)){
    id_ind <- lis_ind[[i]]
    id_pro <- lis_pro[[region]]
    id_col <- c(id_col, paste0(rep(id_ind, each=length(id_pro)), '_', rep(id_pro, length(id_ind)))))
  }
}

```

```

load(paste0('../output/AO_ASV/mic_id_mu_k_',tolower(names(lis_pro)[region]),'.RData'))
param_anno_mu <- data.frame(ID = id_mu)
param_anno_k  <- data.frame(ID = id_k)
param_anno_mu <- merge(param_anno_mu, anno, by='ID')
param_anno_k  <- merge(param_anno_k,  anno, by='ID')
# annotate rrrn
rrn_mu <- rrn_anno(param_anno_mu, rrrn)
rrn_k  <- rrn_anno(param_anno_k,  rrrn)
# compute community rrrn
rrn_mu <- merge(rrn_mu, mic[,c('ID',id_col)])
rrn_k  <- merge(rrn_k,  mic[,c('ID',id_col)])
crrn_mu <- calc_comm_rrn(rrn_mu)
crrn_k  <- calc_comm_rrn(rrn_k)
ao_rrn_k <- rbind(ao_rrn_k,
                    data.frame(crrn = crrn_k,
                               mean = mean(crrn_k),
                               sd   = sd(crrn_k),
                               ci_u = mean(crrn_k) + sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                               ci_l = mean(crrn_k) - sd(crrn_k)*1.96/sqrt(length(crrn_k)),
                               Wing = 'k',
                               Wing.Data = 'k.AO',
                               Process = names(lis_pro)[region]))
ao_rrn_mu <- rbind(ao_rrn_mu,
                     data.frame(crrn = crrn_mu,
                               mean = mean(crrn_mu),
                               sd   = sd(crrn_mu),
                               ci_u = mean(crrn_mu) + sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                               ci_l = mean(crrn_mu) - sd(crrn_mu)*1.96/sqrt(length(crrn_mu)),
                               Wing = 'mu',
                               Wing.Data = 'mu.AO',
                               Process = names(lis_pro)[region]))
}

save(ao_rrn_k, ao_rrn_mu,
      file='../output/AO_ASV/crrn_asv_5v8_ncbi.RData')

load('../output/HZ/crrn_asv_5v8.RData')
load('../output/AO_ASV/crrn_asv_5v8.RData')

rrn <- rbind(ao_rrn_k, ao_rrn_mu, hz_rrn_k, hz_rrn_mu)

rrn$Process <- factor(rrn$Process,
                       levels=c('Bay','ERA',
                               'Influent','Anoxic','Oxic','Effluent'))

rrn$Wing  <- factor(rrn$Wing,
                     levels=c('mu','k'))

rrn$Wing.Data <- factor(rrn$Wing.Data,
                        levels=c('mu.AO','k.AO','mu.HZ','k.HZ'))

ggplot() +

```

```

geom_violin(data=rrn, mapping=aes(x=Wing, y=crrn, color=Wing.Data), fill='#f2f2f2', trim=FALSE) +
  geom_point(data=rrn, mapping=aes(x=Wing, y=mean, color=Wing.Data)) +
  geom_errorbar(data=rrn, mapping=aes(x=Wing, ymin=ci_l, ymax=ci_u, color=Wing.Data), width=0.08) +
  ylab(TeX('c-rrn')) +
  xlab(TeX('Wing')) +
  facet_wrap(Process~, nrow=1) +
  scale_x_discrete(labels=unname(TeX(c('$\\mu$', '$k$')))) +
  scale_colour_manual(values = c('#4daf4a', '#984ea3', '#e41a1c', '#377eb8'),
                       labels=unname(TeX(c('$\\mu_{AO}$', '$k_{AO}$', '$\\mu_{HZ}$', '$k_{HZ}$')))) +
  theme_bw() +
  theme(aspect.ratio=5,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0),
        axis.text.y = element_text(size=9),
        strip.text = element_text(size = 12),
        legend.title = element_text(color = "black", size = 12),
        legend.text = element_text(color = "black", size = 9),
        legend.position='bottom') +
  guides(color = guide_legend(nrow = 1))

```

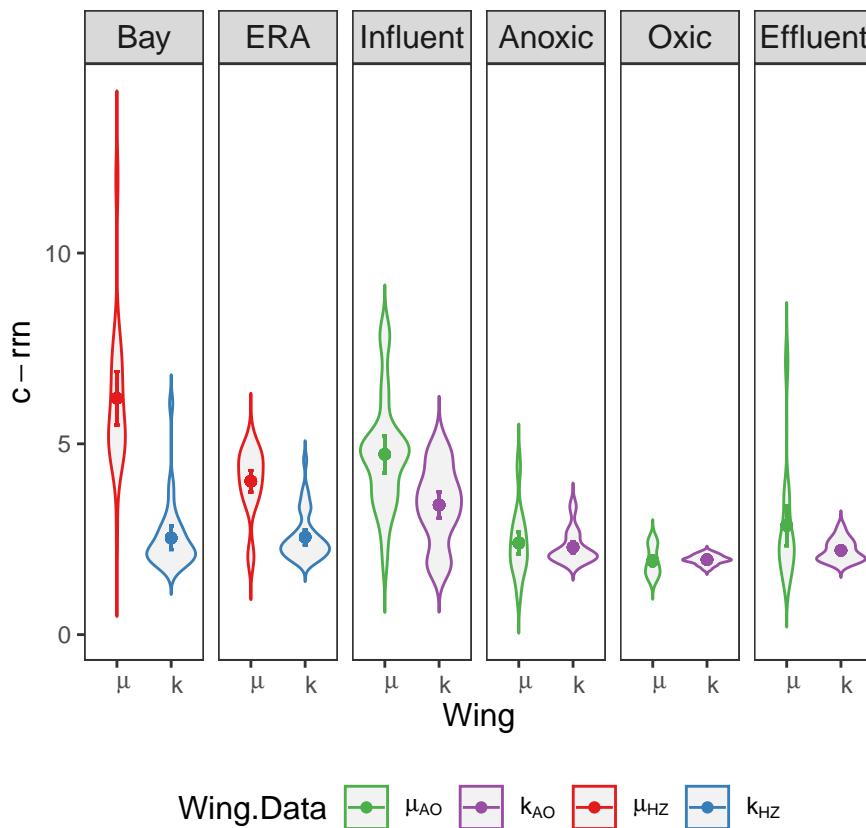


Figure 7. Identifying influential microbes

Top crucial microbes in AO systems

```

# load parameters mle
vec_process <- c('influent','anoxic','oxic','effluent')
param_plt <- data.frame()
for(item in vec_process){
  # load mle
  load(paste0('../output/AO_ASV/param_trio_asv_', item, '.RData'))
  # load taxa category
  load(paste0('../output/AO_ASV/taxa_category_', item, '.RData'))
  param_trio$ID <- rownames(param_trio)
  # extract poisson distributed microbes
  param_poisson <- param_trio[param_trio$k==Inf,]
  # extract gamma-poisson distributed microbes
  param_gpm <- param_trio[param_trio$k!=Inf,]
  # extract zero-inflated microbes
  param_zeroinf <- param_trio[param_trio$pio!=0,]
  # annotate procedure
  param_gpm$Procedure <- str_to_title(item)
  # mean log(mu) of non-rare taxa
  load(paste0('../output/AO_ASV/mic_id_mu_k_', item, '.RData'))
  param_mu <- param_gpm[id_mu,]
  param_mu_nrt <- param_mu[c(id_at,id_cat,id_crat,id_mt),]
  v_mu <- param_mu_nrt[param_mu_nrt$mu > 0, 'mu']
  v_mu <- v_mu[!is.na(v_mu)]
  param_gpm$mean_log_mu_nrt <- mean(log(v_mu))
  param_plt <- rbind(param_plt, param_gpm)
}
rownames(param_plt) <- c(1:nrow(param_plt))

param_plt$Procedure <- factor(param_plt$Procedure,
                                levels=str_to_title(vec_process))

anno <- read.csv('../data/AO/ASV_Anno.csv')
id_acinetobacter <- c('bd4b9ead01d30e8039cb21d78f7efcd5',
                      '1c5e18cc56e5efade244e7fc49c24f1d') # Acinetobacter_townieri
id_akyh767 <- c('43a98390e1117f64b1bd45a7563f09a0',
                 '59cdd305738ed8a593f27f1cbaf4c569',
                 '7a0f7d306463c549075e0e5802d7358f',
                 'ec63e696bfd76850686bd43a2cd64271')
id_bacillus <- c('fc80343bbff631d21a4cba01b5d6693a',
                  'a504d0d75935746af558b8c505e74893',
                  '179d9962a38537ec660ad4f0a50e3bb6')
id_caldisericum <- c('06bc6796c2bb295757c90547dfd29136',
                      '8c9130cbc687566413257dd89018848f',
                      'cf38b2bbcec166a97f60632b9fdbcc62',
                      '53ab803b7849cb439e5f307bc491c3b8',
                      '83efa7c67befc6feb583a49f7bea1a34')
id_geosporobacter <- c('0c53807abde2b63cb3b28b5687593385',
                        '7ed86c307b1d9c1b87ddea2da00d3f80',

```

```

        'f499bccb0988c22894b385cfbac47784',
        '50449b4900b568e6aadb1ee3849ed6ba',
        'c04a9b88e59c60ed4a8d8d6417d963b6')
id_klebsiella <- c('1422cee1cb6fe3c6a930f64376e078c5',
                     '1422cee1cb6fe3c6a930f64376e078c5',
                     'b69673f11512820d21e9ce551b7ba5db')
id_ns9_mg <- c('48284cf4b7f620f3418aac4583857a54',
                 '21063ffcb8393e675fb0b7d297739233',
                 '0925095bb87af3c33c8eb77b787fb356')
id_saccharimonadales <- c('12bea826f0b1b6ed9917829e132cf028',
                            '463462818b9654d8688d98dbea80bb48',
                            '0368fd1dacd87c90123caae160f4843d')
id_sm1a02 <- c('b76087038188beb879f9a0a203fc67f2',
                 '2d9086eebf6fb0e51d928efc315e99c',
                 '89eb994d43fb76ed4eb0189fb430b273')
id_trichococcus <- c('1573cd934361448a0202d2b6b9d2ed44',
                       '009d0f3716fd09acf1bb98efa94772f5',
                       '2443041f67a112eee280e5f9f1637826',
                       'dfbd7a749ff8a068977dbbc00ec8dc2e')
id_wolinella <- c('01e1b5a8f49619c74afcebc965e6f375',
                   '49658925bc987768981650aa54c9ef91')

param_plt$Microbe <- ifelse(param_plt$ID %in% id_acinetobacter, 'Acinetobacter', 'Other')
param_plt$Microbe <- ifelse(param_plt$ID %in% id_akyh767, 'AKYH767', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_bacillus, 'Bacillus', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_caldisericum, 'Caldisericum', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_geosporobacter, 'Geosporobacter', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_klebsiella, 'Klebsiella', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_ns9_mg, 'NS9 Marine Group', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_saccharimonadales, 'Saccharimonadales', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_sm1a02, 'SM1A02', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_trichococcus, 'Trichococcus', param_plt$Microbe)
param_plt$Microbe <- ifelse(param_plt$ID %in% id_wolinella, 'Wolinella', param_plt$Microbe)

param_plt$Microbe <- factor(param_plt$Microbe,
                             levels=c('Acinetobacter', 'AKYH767', 'Bacillus', 'Caldisericum',
                                      'Geosporobacter', 'Klebsiella', 'NS9 Marine Group',
                                      'Saccharimonadales', 'SM1A02', 'Trichococcus', 'Wolinella',
                                      'Other'))

g0 <- ggplot() +
  geom_point(data=param_plt,
             aes(x=log(k), y=log(mu), color=Microbe), alpha=1, size=0.6) +
  ylab(TeX('\\log{\\mu}')) +
  xlab(TeX('\\log{k}')) +
  ggtitle(TeX('')) +
  xlim(-5, 20) +
  ylim(-5, 10) +
  theme_bw() +
  theme(aspect.ratio=1) +
  facet_wrap(Procedure~., scale='free') +
  scale_colour_manual(values = c('#377eb8', '#4daf4a', '#ff7f00', '#a6cee3',
                                '#a65628', '#984ea3', '#252525',

```

```

        '#f781bf', '#e41a1c', '#b2df8a', '#fb9a99',
        '#ffffcc')) +
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      axis.text.x = element_text(size = 9),
      axis.text.y = element_text(size=9),
      strip.text = element_text(size = 12),
      legend.title = element_text(color = "black", size = 12),
      legend.text = element_text(color = "black", size = 9))

g_leg <- get_legend(g0)

g1 <- ggplot() +
  geom_point(data=param_plt[param_plt$Microbe=='Other',],
             aes(x=log(k), y=log(mu)), color='#ffffcc', alpha=0.6, size=0.24) +
  geom_hline(data=param_plt, aes(yintercept=mean_log_mu_nrt),
             linetype='dotdash', color='black', size=0.12) +
  geom_point(data=param_plt[param_plt$Microbe=='Acinetobacter',],
             aes(x=log(k), y=log(mu)), color='#377eb8', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='AKYH767',],
             aes(x=log(k), y=log(mu)), color='#4daf4a', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Bacillus',],
             aes(x=log(k), y=log(mu)), color='#ff7f00', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Caldisericum',],
             aes(x=log(k), y=log(mu)), color='#a6cee3', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Geosporobacter',],
             aes(x=log(k), y=log(mu)), color='#a65628', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Klebsiella',],
             aes(x=log(k), y=log(mu)), color='#984ea3', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='NS9 Marine Group',],
             aes(x=log(k), y=log(mu)), color='#252525', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Saccharimonadales',],
             aes(x=log(k), y=log(mu)), color='#f781bf', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='SM1A02',],
             aes(x=log(k), y=log(mu)), color='#e41a1c', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Trichococcus',],
             aes(x=log(k), y=log(mu)), color='#b2df8a', alpha=1, size=0.48) +
  geom_point(data=param_plt[param_plt$Microbe=='Wolinella',],
             aes(x=log(k), y=log(mu)), color='#fb9a99', alpha=1, size=0.48) +
  ylab(TeX('`\\log{(`\\hat{`\\mu`})}`')) +
  xlab(TeX('`\\log{(`\\hat{k}`})}`')) +
  ggtitle(TeX('')) +
  xlim(-5,20) +
  ylim(-5,10) +
  theme_bw() +
  theme(aspect.ratio=1) +
  facet_wrap(Procedure~., scale='free') +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "none",

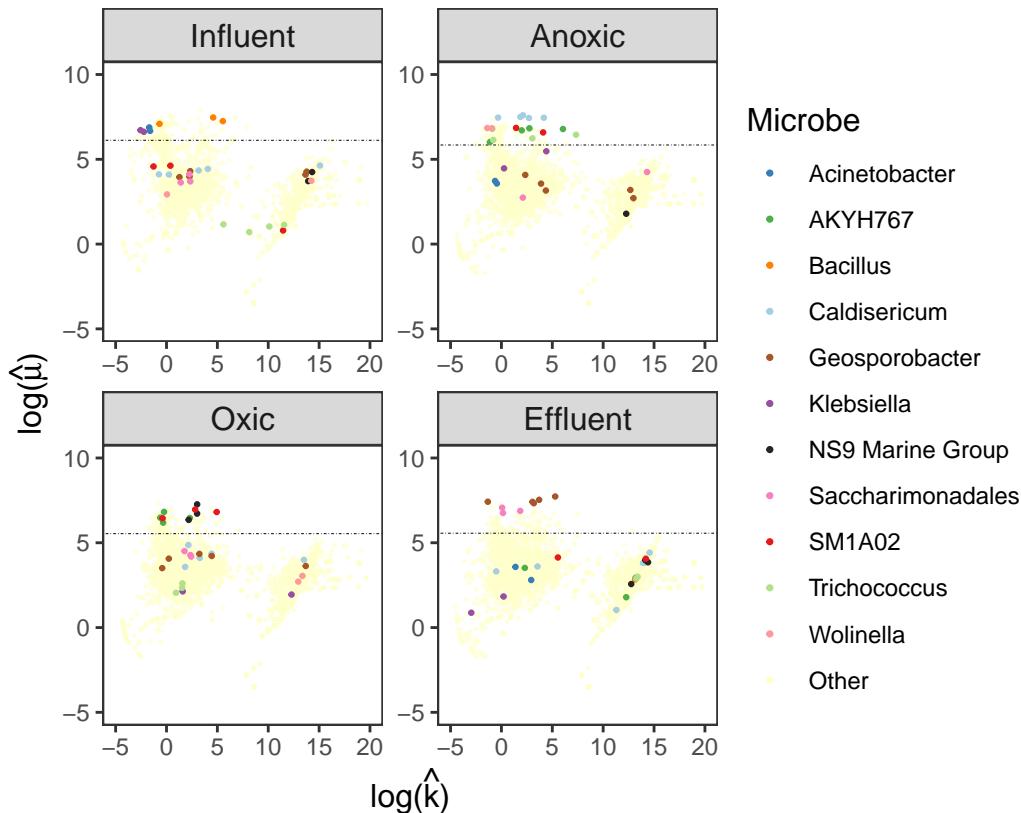
```

```

axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12),
axis.text.x = element_text(size = 9),
axis.text.y = element_text(size=9),
strip.text = element_text(size = 12))

plot_grid(g1, g_leg, NULL,
          rel_widths = c(4, 1, 1), nrow=1)

```



Estimated abundance changes along the nutrient gradient

```

# Lambda shape mic #####
id_genus <- c('Flavobacterium','Psychrobacter',
            'Woeseia','Lokiarchaeia')
id_family <- c()
vec_process_hz <- c('bay','era')
vec_process_ao <- c('influent','anoxic','oxic','effluent')
anno_hz <- read.csv('../data/2018/taxa_anno.csv')
anno_ao <- read.csv('../data/A0/ASV_Anno.csv')

param_plt <- data.frame()

for(mic in id_genus){
  for(item in vec_process_hz){
    file_in <- paste0('../output/HZ/param_trio_',item,'.RData')
    load(file_in)
  }
}

```

```

param_trio$ID <- rownames(param_trio)
load(paste0('../output/HZ/mic_id_mu_k_', item, '.RData'))
param_mu <- param_trio[id_mu,]
param_mu <- merge(param_mu, anno_hz, by='ID')
param_mic <- param_mu[param_mu$G==mic,]
if(nrow(param_mic)>0){
  mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
  mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                         mu_sd   = sd(param_mic$mu),
                         mu_up   = mu_up,
                         mu_lo   = mu_lo,
                         n_mic   = nrow(param_mic),
                         region  = ifelse(item=='bay', 'Bay', 'ERA'),
                         mic     = mic)
} else if(nrow(param_mic)==1) {
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                         mu_sd   = 0,
                         mu_up   = 0,
                         mu_lo   = 0,
                         n_mic   = 0,
                         region  = ifelse(item=='bay', 'Bay', 'ERA'),
                         mic     = mic)
} else {
  dat_out <- data.frame(mu_mean = 0,
                         mu_sd   = 0,
                         mu_up   = 0,
                         mu_lo   = 0,
                         n_mic   = 0,
                         region  = ifelse(item=='bay', 'Bay', 'ERA'),
                         mic     = mic)
}
param_plt <- rbind(param_plt, dat_out)
}

for(item in vec_process_ao){
  file_in <- paste0('../output/AO_ASV/param_trio_asv_', item, '.RData')
  load(file_in)
  param_trio$ID <- rownames(param_trio)
  load(paste0('../output/AO_ASV/mic_id_mu_k_', item, '.RData'))
  param_mu <- param_trio[id_mu,]
  param_mu <- merge(param_mu, anno_ao, by='ID')
  param_mic <- param_mu[param_mu$G==mic,]
  if(nrow(param_mic)>0){
    mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                           mu_sd   = sd(param_mic$mu),
                           mu_up   = mu_up,
                           mu_lo   = mu_lo,
                           n_mic   = nrow(param_mic),
                           region  = str_to_title(item),
                           mic     = mic)
  }
}

```

```

                    mic      = mic)
} else if(nrow(param_mic)==1) {
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                        mu_sd   = 0,
                        mu_up   = 0,
                        mu_lo   = 0,
                        n_mic   = 0,
                        region  = str_to_title(item),
                        mic     = mic)
} else {
  dat_out <- data.frame(mu_mean = 0,
                        mu_sd   = 0,
                        mu_up   = 0,
                        mu_lo   = 0,
                        n_mic   = 0,
                        region  = str_to_title(item),
                        mic     = mic)
}
param_plt <- rbind(param_plt, dat_out)
}
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$region <- factor(param_plt$region,
                            levels=c('Influent','Anoxic','Oxic',
                                    'Effluent','ERA','Bay'))
save(param_plt, file='../output/mic_lambda.RData')

# ell shape mic #####
id_genus <- c('Oligella','Klebsiella',
             'Pseudarcobacter','Acetoanaerobium',
             'Actinomycetaceae','Bacteroides','Chryseomicrombium','Fusibacter')
id_family <- c()
vec_process_hz <- c('bay','era')
vec_process_ao <- c('influent','anoxic','oxic','effluent')
anno_hz <- read.csv('../data/2018/taxa_anno.csv')
anno_ao <- read.csv('../data/AO/ASV_Anno.csv')

param_plt <- data.frame()

for(mic in id_genus){
  for(item in vec_process_hz){
    file_in <- paste0('../output/HZ/param_trio_',item,'.RData')
    load(file_in)
    param_trio$ID <- rownames(param_trio)
    load(paste0('../output/HZ/mic_id_mu_k_',item,'.RData'))
    param_mu <- param_trio[id_mu,]
    param_mu <- merge(param_mu, anno_hz, by='ID')
    param_mic <- param_mu[param_mu$G==mic,]
    if(nrow(param_mic)>0){
      mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      dat_out <- data.frame(mu_mean = mean(param_mic$mu),

```

```

        mu_sd    = sd(param_mic$mu),
        mu_up    = mu_up,
        mu_lo    = mu_lo,
        n_mic    = nrow(param_mic),
        region   = ifelse(item=='bay','Bay','ERA'),
        mic      = mic)
} else if(nrow(param_mic)==1) {
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                         mu_sd    = 0,
                         mu_up    = 0,
                         mu_lo    = 0,
                         n_mic    = 0,
                         region   = ifelse(item=='bay','Bay','ERA'),
                         mic      = mic)
} else {
  dat_out <- data.frame(mu_mean = 0,
                         mu_sd    = 0,
                         mu_up    = 0,
                         mu_lo    = 0,
                         n_mic    = 0,
                         region   = ifelse(item=='bay','Bay','ERA'),
                         mic      = mic)
}
param_plt <- rbind(param_plt, dat_out)
}

for(item in vec_process_ao){
  file_in <- paste0('../output/AO_ASV/param_trio_asv_',item,'.RData')
  load(file_in)
  param_trio$ID <- rownames(param_trio)
  load(paste0('../output/AO_ASV/mic_id_mu_k_',item,'.RData'))
  param_mu <- param_trio[id_mu,]
  param_mu <- merge(param_mu, anno_ao, by='ID')
  param_mic <- param_mu[param_mu$G==mic,]
  if(nrow(param_mic)>0){
    mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                           mu_sd    = sd(param_mic$mu),
                           mu_up    = mu_up,
                           mu_lo    = mu_lo,
                           n_mic    = nrow(param_mic),
                           region   = str_to_title(item),
                           mic      = mic)
  } else if(nrow(param_mic)==1) {
    dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                           mu_sd    = 0,
                           mu_up    = 0,
                           mu_lo    = 0,
                           n_mic    = 0,
                           region   = str_to_title(item),
                           mic      = mic)
  }
}

```

```

    } else {
      dat_out <- data.frame(mu_mean = 0,
                            mu_sd   = 0,
                            mu_up   = 0,
                            mu_lo   = 0,
                            n_mic   = 0,
                            region  = str_to_title(item),
                            mic     = mic)
    }
  param_plt <- rbind(param_plt, dat_out)
}
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$region <- factor(param_plt$region,
                           levels=c('Influent','Anoxic','Oxic',
                           'Effluent','ERA','Bay'))

save(param_plt, file='../output/mic_ell.RData')

# u shape mic #####
id_genus <- c('Pseudomonas','Bacillus')
id_family <- c('Planococcaceae')
vec_process_hz <- c('bay','era')
vec_process_ao <- c('influent','anoxic','oxic','effluent')
anno_hz <- read.csv('../data/2018/taxa_anno.csv')
anno_ao <- read.csv('../data/A0/ASV_Anno.csv')

param_plt <- data.frame()

for(mic in id_genus){
  for(item in vec_process_hz){
    file_in <- paste0('../output/HZ/param_trio_',item,'.RData')
    load(file_in)
    param_trio$ID <- rownames(param_trio)
    load(paste0('../output/HZ/mic_id_mu_k_',item,'.RData'))
    param_mu <- param_trio[id_mu,]
    param_mu <- merge(param_mu, anno_hz, by='ID')
    param_mic <- param_mu[param_mu$G==mic,]
    if(nrow(param_mic)>0){
      mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                            mu_sd   = sd(param_mic$mu),
                            mu_up   = mu_up,
                            mu_lo   = mu_lo,
                            n_mic   = nrow(param_mic),
                            region  = ifelse(item=='bay','Bay','ERA'),
                            mic     = mic)
    } else if(nrow(param_mic)==1) {
      dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                            mu_sd   = 0,

```

```

        mu_up   = 0,
        mu_lo   = 0,
        n_mic   = 0,
        region  = ifelse(item=='bay','Bay','ERA'),
        mic     = mic)
    } else {
      dat_out <- data.frame(mu_mean = 0,
                             mu_sd   = 0,
                             mu_up   = 0,
                             mu_lo   = 0,
                             n_mic   = 0,
                             region  = ifelse(item=='bay','Bay','ERA'),
                             mic     = mic)
    }
    param_plt <- rbind(param_plt, dat_out)
  }
}

for(item in vec_process_ao){
  file_in <- paste0('../output/AO_ASV/param_trio_asv_',item,'.RData')
  load(file_in)
  param_trio$ID <- rownames(param_trio)
  load(paste0('../output/AO_ASV/mic_id_mu_k_',item,'.RData'))
  param_mu <- param_trio[id_mu,]
  param_mu <- merge(param_mu, anno_ao, by='ID')
  param_mic <- param_mu[param_mu$G==mic,]
  if(nrow(param_mic)>0){
    mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
    dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                           mu_sd   = sd(param_mic$mu),
                           mu_up   = mu_up,
                           mu_lo   = mu_lo,
                           n_mic   = nrow(param_mic),
                           region  = str_to_title(item),
                           mic     = mic)
  } else if(nrow(param_mic)==1) {
    dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                           mu_sd   = 0,
                           mu_up   = 0,
                           mu_lo   = 0,
                           n_mic   = 0,
                           region  = str_to_title(item),
                           mic     = mic)
  } else {
    dat_out <- data.frame(mu_mean = 0,
                           mu_sd   = 0,
                           mu_up   = 0,
                           mu_lo   = 0,
                           n_mic   = 0,
                           region  = str_to_title(item),
                           mic     = mic)
  }
}

```

```

param_plt <- rbind(param_plt, dat_out)
}
}

for(mic in id_family){
  for(item in vec_process_hz){
    file_in <- paste0('../output/HZ/param_trio_',item,'.RData')
    load(file_in)
    param_trio$ID <- rownames(param_trio)
    load(paste0('../output/HZ/mic_id_mu_k_',item,'.RData'))
    param_mu <- param_trio[id_mu,]
    param_mu <- merge(param_mu, anno_hz, by='ID')
    param_mic <- param_mu[param_mu$F==mic,]
    if(nrow(param_mic)>0){
      mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
      dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                             mu_sd   = sd(param_mic$mu),
                             mu_up   = mu_up,
                             mu_lo   = mu_lo,
                             n_mic   = nrow(param_mic),
                             region  = ifelse(item=='bay','Bay','ERA'),
                             mic     = mic)
    } else if(nrow(param_mic)==1) {
      dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                             mu_sd   = 0,
                             mu_up   = 0,
                             mu_lo   = 0,
                             n_mic   = 0,
                             region  = ifelse(item=='bay','Bay','ERA'),
                             mic     = mic)
    } else {
      dat_out <- data.frame(mu_mean = 0,
                             mu_sd   = 0,
                             mu_up   = 0,
                             mu_lo   = 0,
                             n_mic   = 0,
                             region  = ifelse(item=='bay','Bay','ERA'),
                             mic     = mic)
    }
    param_plt <- rbind(param_plt, dat_out)
  }
}

for(item in vec_process_ao){
  file_in <- paste0('../output/AO_ASV/param_trio_asv_',item,'.RData')
  load(file_in)
  param_trio$ID <- rownames(param_trio)
  load(paste0('../output/AO_ASV/mic_id_mu_k_',item,'.RData'))
  param_mu <- param_trio[id_mu,]
  param_mu <- merge(param_mu, anno_ao, by='ID')
  param_mic <- param_mu[param_mu$F==mic,]

```

```

if(nrow(param_mic)>1){
  mu_up <- mean(param_mic$mu)+1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
  mu_lo <- mean(param_mic$mu)-1.96*sd(param_mic$mu)/sqrt(nrow(param_mic))
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                         mu_sd   = sd(param_mic$mu),
                         mu_up   = mu_up,
                         mu_lo   = mu_lo,
                         n_mic   = nrow(param_mic),
                         region  = str_to_title(item),
                         mic     = mic)
} else if(nrow(param_mic)==1) {
  dat_out <- data.frame(mu_mean = mean(param_mic$mu),
                         mu_sd   = 0,
                         mu_up   = 0,
                         mu_lo   = 0,
                         n_mic   = 0,
                         region  = str_to_title(item),
                         mic     = mic)
} else {
  dat_out <- data.frame(mu_mean = 0,
                         mu_sd   = 0,
                         mu_up   = 0,
                         mu_lo   = 0,
                         n_mic   = 0,
                         region  = str_to_title(item),
                         mic     = mic)
}
param_plt <- rbind(param_plt, dat_out)
}
}

rownames(param_plt) <- c(1:nrow(param_plt))
param_plt$region <- factor(param_plt$region,
                           levels=c('Influent','Anoxic','Oxic',
                           'Effluent','ERA','Bay'))

save(param_plt, file='../output/mic_u.RData')

dat_plt <- data.frame()

load('../output/mic_u.RData')
param_plt$Microbe <- param_plt$mic
param_plt$Shape  <- 'U-shape'
dat_plt <- rbind(dat_plt, param_plt)

load('../output/mic_ell.RData')
param_plt$Microbe <- param_plt$mic
param_plt$Shape  <- 'L-shape'
dat_plt <- rbind(dat_plt, param_plt)

load('../output/mic_lambda.RData')
param_plt$Microbe <- param_plt$mic
param_plt$Shape  <- 'Lambda-shape'
dat_plt <- rbind(dat_plt, param_plt)

```

```

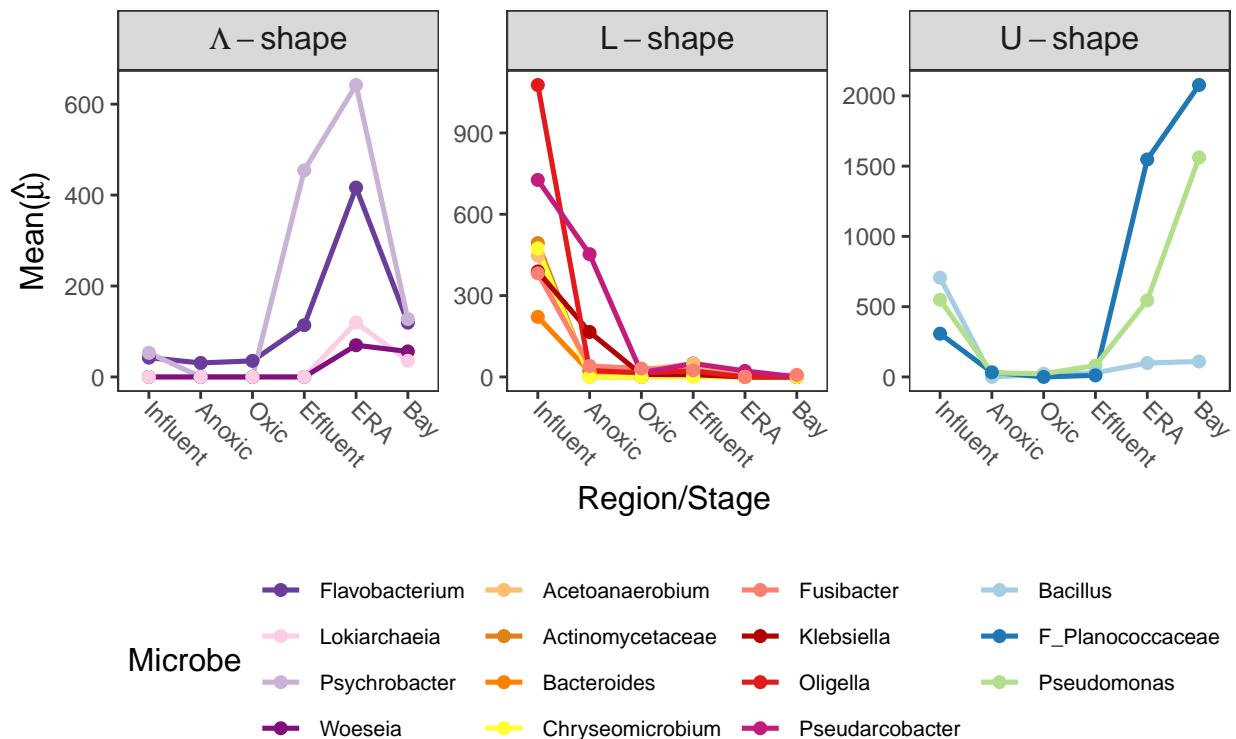
dat_plt$Shape <- factor(dat_plt$Shape,
                        levels=c('Lambda-shape','L-shape','U-shape'))

dat_plt$Microbe <- ifelse(dat_plt$Microbe=='Planococcaceae','F_Planococcaceae',dat_plt$Microbe)

dat_plt$Microbe <- factor(dat_plt$Microbe,
                           levels=c('Flavobacterium','Lokiarchaeia','Psychrobacter','Woeseia',
                           'Acetoanaerobium','Actinomycetaceae','Bacteroides',
                           'Chryseomicrobium','Fusibacter','Klebsiella',
                           'Oligella','Pseudaracobacter',
                           'Bacillus','F_Planococcaceae','Pseudomonas'))

ggplot(data=dat_plt,
       aes(x=region, y=mu_mean, group=Microbe)) +
  geom_line(size=0.9, aes(color=Microbe)) +
  geom_point(size=1.8, aes(color=Microbe)) +
  ylab(TeX('Mean(\hat{\mu}))')) +
  xlab(TeX('Region/Stage')) +
  ggtitle(TeX('')) +
  theme_bw() +
  theme(aspect.ratio=1) +
  facet_wrap(Shape~, scale='free', labeller='label_parsed') +
  scale_colour_manual(values = c('#6a3d9a','#fccde5','#cab2d6','#810f7c',
                                '#fdbf6f','#e08214','#ff7f00','#ffff33',
                                '#fb8072','#b30000','#e31a1c','#c51b7d',
                                '#a6cee3','#1f78b4','#b2df8a')) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = 'bottom',
        legend.title = element_text(size=12),
        legend.text = element_text(size=8),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 9, vjust = 0.5, hjust = 0, angle = -45),
        axis.text.y = element_text(size=9),
        strip.text = element_text(size = 12)) +
  guides(color=guide_legend(ncol=4, byrow=FALSE))

```



```
sessionInfo()

## R version 4.2.0 (2022-04-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.3
##
## Matrix products: default
## BLAS:    /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK:  /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices  utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ggvegan_0.1-0   vegan_2.6-2    lattice_0.20-45  permute_0.9-7
## [5] ggh4x_0.2.1    stringr_1.4.0   gridExtra_2.3    cowplot_1.1.1
## [9] latex2exp_0.9.4 ggplot2_3.3.6
##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.1.2  xfun_0.31        purrr_0.3.4       splines_4.2.0
## [5] colorspace_2.0-3 vctrs_0.4.1      generics_0.1.3    viridisLite_0.4.0
## [9] htmltools_0.5.3  yaml_2.3.5       mgcv_1.8-40      utf8_1.2.2
## [13] rlang_1.0.4     pillar_1.8.0     glue_1.6.2       withr_2.5.0
## [17] lifecycle_1.0.1  munsell_0.5.0   gtable_0.3.0      evaluate_0.15
## [21] labeling_0.4.2   knitr_1.39      fastmap_1.1.0    parallel_4.2.0
## [25] fansi_1.0.3     highr_0.9       Rcpp_1.0.9       scales_1.2.0
## [29] farver_2.1.1    digest_0.6.29   stringi_1.7.8    dplyr_1.0.9
```

```
## [33] ggrepel_0.9.1      cli_3.3.0        tools_4.2.0       magrittr_2.0.3
## [37] tibble_3.1.8        cluster_2.1.3    tidyverse_1.3.0   pkgconfig_2.0.3
## [41] MASS_7.3-58.1       Matrix_1.4-1     rmarkdown_2.14    rstudioapi_0.13
## [45] R6_2.5.1            nlme_3.1-158     compiler_4.2.0
```