Lake Erie HABs Community Ecology Manuscript Supplement

Contents


```
library(phyloseq)
library(ggplot2)
library(dplyr)
library(reshape2)
library(scales)
library(grid)
library(gridExtra)
library(cowplot)
setwd("~/git_repos/chabs/miseq_may2015/analysis")
source("~/git_repos/MicrobeMiseq/R/miseqR.R")

theme_set(theme_bw())

station_colors = c("red", "#ffa500", "#0080ff")

order_dates <- function(df) {
  df$Date <- factor(df$Date,
    levels = c("6/16","6/30","7/8","7/14","7/21",
      "7/29","8/4","8/11","8/18","8/25","9/2","9/8","9/15",
      "9/23","9/29","10/6","10/15","10/20","10/27"))
  return(df)
}

named_list <- function(...){
    names <- as.list(substitute(list(...)))[-1L]
```

```
    result <- list(...)
    names(result) <- names
    result
}

grid_arrange_shared_legend <- function(...) {
    plots <- list(...)
    g <- ggplotGrob(plots[[1]] + theme(legend.position = "bottom"))$grobs
    legend <- g[[which(sapply(g, function(x) x$name) == "guide-box")]]
    lheight <- sum(legend$height)
    grid.arrange(
        do.call(arrangeGrob, lapply(plots, function(x)
            x + theme(legend.position = "none"))),
        legend,
        ncol = 1,
        heights = unit.c(unit(1, "npc") - lheight, lheight))
}
```

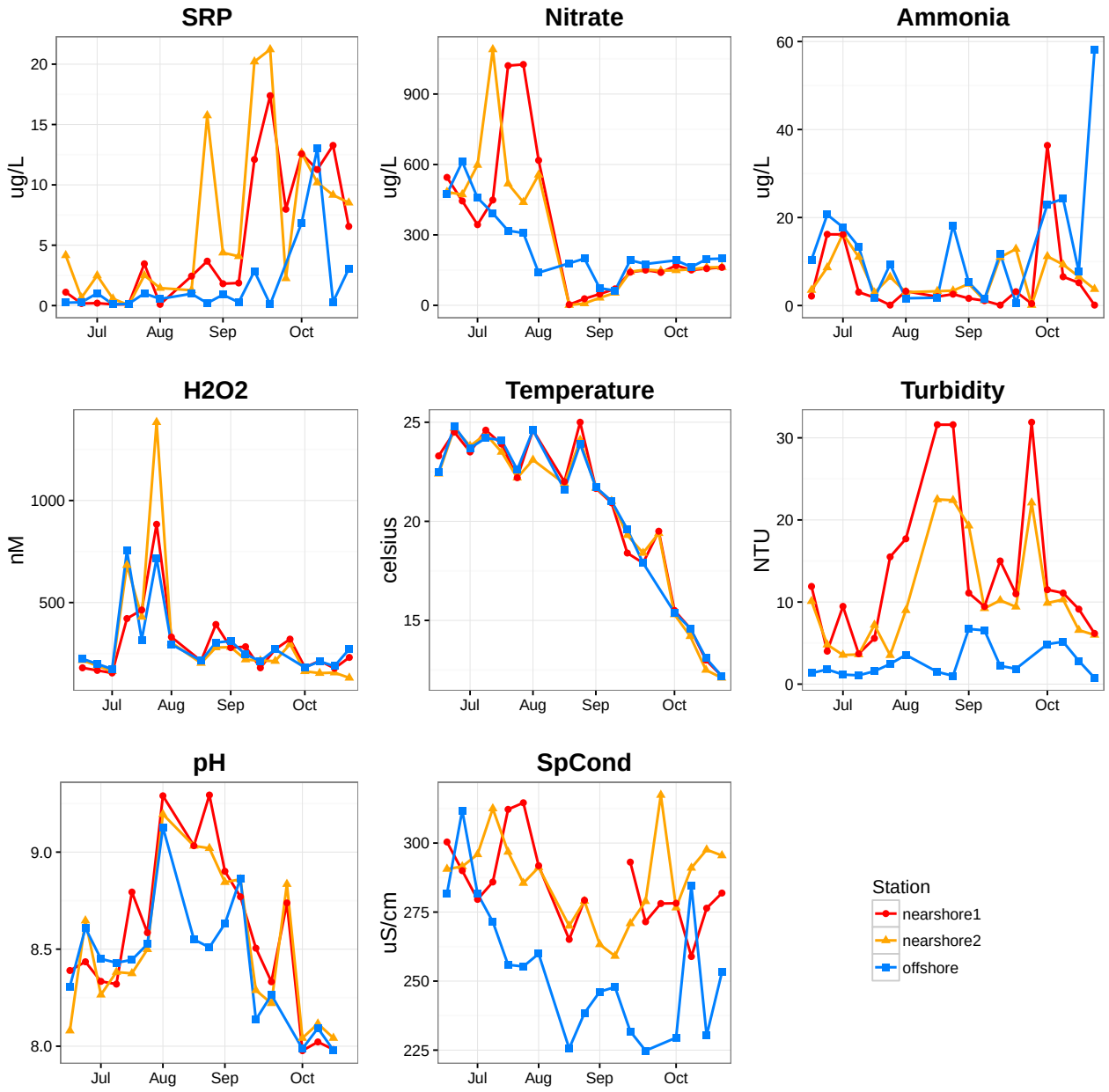
```
load("erie-data.RData")
load("supplement.RData")
```

```
# Scale reads to even depth
erie_scale <-
  erie %>%
    scale_reads(n = 15000, round = "round")
```

S1: Environmental variables

```
# Format
nutrient_df <-
  sample_data(erie) %>%
    order_dates()

plot_nutrients <- function(df, nutrient, title, ylabs) {
  ggplot(df, aes_string(
    x = "Date",
    y = nutrient,
    group = "Station",
    shape = "Station",
    color = "Station")
  ) +
    scale_x_discrete(
      breaks = c("7/8", "8/4", "9/2", "10/6"),
      labels = c("Jul", "Aug", "Sep", "Oct"),
      drop = FALSE
    ) +
    geom_line(size = 0.8) +
    geom_point(size = 1.8) +
    ggtitle(title) +
    xlab("") +
```

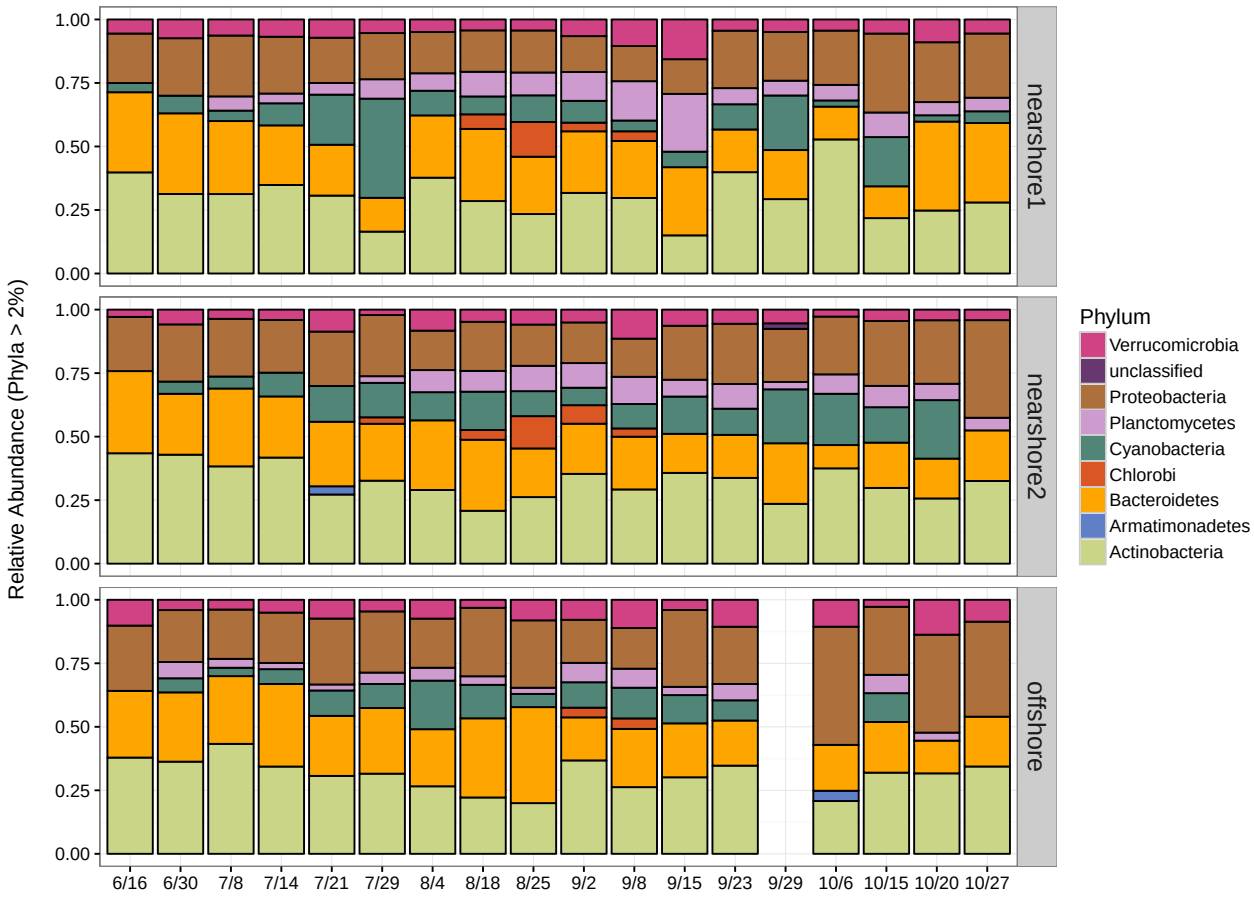
```
    ylab(ylabs) +
    scale_color_manual(values = station_colors) +
    theme(
        axis.title.y = element_text(size = 14),
        plot.title = element_text(size = 16, face = "bold")
    )
}

nutplot_vars <- c("SRP", "Nitrate", "Ammonia", "H2O2", "Temp", "Turbidity", "pH", "SpCond")
nutplot_titles <- c("SRP", "Nitrate", "Ammonia", "H2O2", "Temperature", "Turbidity", "pH", "SpCond")
nutplot_ylabs <- c(rep("ug/L", 3), "nM", "celsius", "NTU", "", "uS/cm")

nutplots <- list()

for (i in 1:length(nutplot_vars)) {
  nutplots[[i]] <- plot_nutrients(
    df = nutrient_df,
    nutrient = nutplot_vars[i],
    title = nutplot_titles[i],
    ylabs = nutplot_ylabs[i]
  )
}

g_legend <- function(a.gplot) {
    tmp <- ggplot_gtable(ggplot_build(a.gplot))
    leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
    legend <- tmp$grobs[[leg]]
    legend
}

legend <- g_legend(nutplots[[1]])

nutplots_noleg <- lapply(nutplots, function(x){ x + theme(legend.position = "none")})
nutplots_noleg[[9]] <- legend

# multiplot
do.call("grid.arrange", c(nutplots_noleg, ncol = 3))
```



S2: Phylum composition of full community across sites

```
# Subset to full community,
# Transform to long format and prune out phyla below 2% in each sample
# for easier to read stacked barplot
erie_phylum <-
  erie %>%
    transform_sample_counts(function(x) {x/sum(x)}) %>%
    tax_glom(taxrank = "Phylum") %>%
    psmelt() %>%
    group_by(Phylum) %>%
    filter(Abundance > 0.02) %>%
```

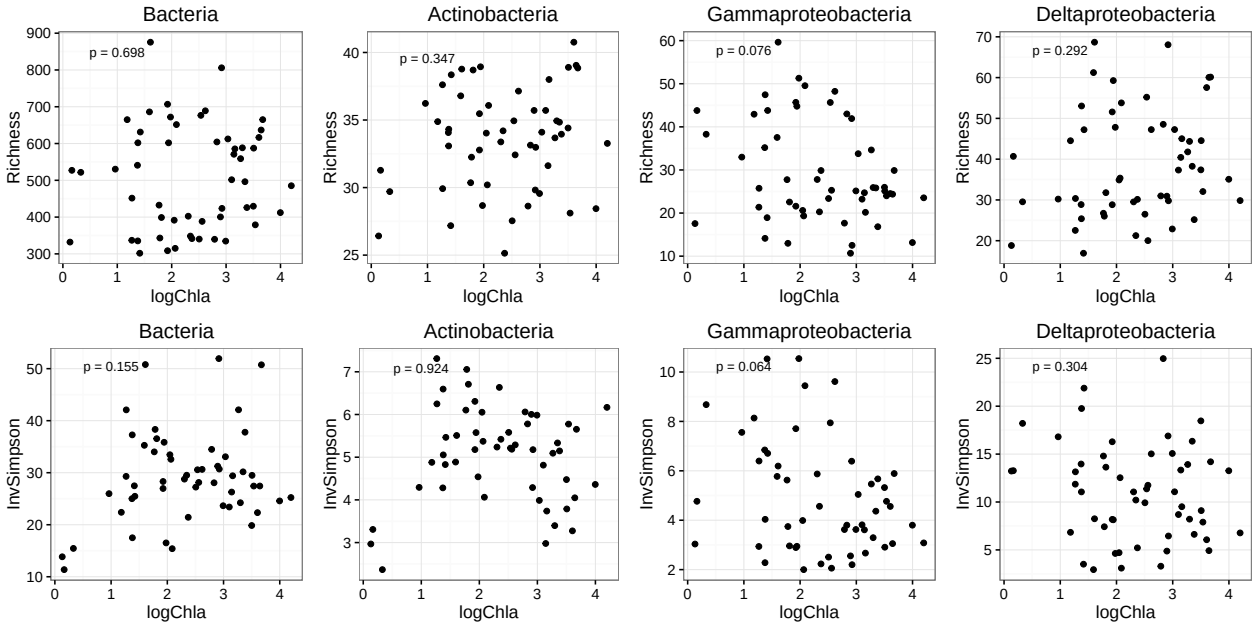
```
    order_dates() %>%
    arrange(Phylum)


# Set colors for plotting
phylum_colors <- c(
  "#CBD588", "#5F7FC7", "orange","#DA5724", "#508578", "#CD9BCD",
   "#AD6F3B", "#673770","#D14285", "#652926", "#C84248",
  "#8569D5", "#5E738F","#D1A33D", "#8A7C64", "#599861"
)


# Plot
ggplot(erie_phylum, aes(x = Date, y = Abundance, fill = Phylum)) +
  facet_grid(Station~., scales = "free_y") +
  geom_bar(stat = "identity") +
  geom_bar(
    stat = "identity",
    position = "fill",
    colour = "black",
    show.legend = FALSE
  ) +
  scale_fill_manual(values = phylum_colors) +
  guides(fill = guide_legend(reverse = TRUE, keywidth = 1, keyheight = 1)) +
  ylab("Relative Abundance (Phyla > 2%) \n") +
  theme(
    axis.title.x = element_blank(),
    strip.text = element_text(size = 14)
  )
```



S3: Alpha diversity for other taxonomic groups

We will load data from the original manuscript to avoid rerunning all this analysis

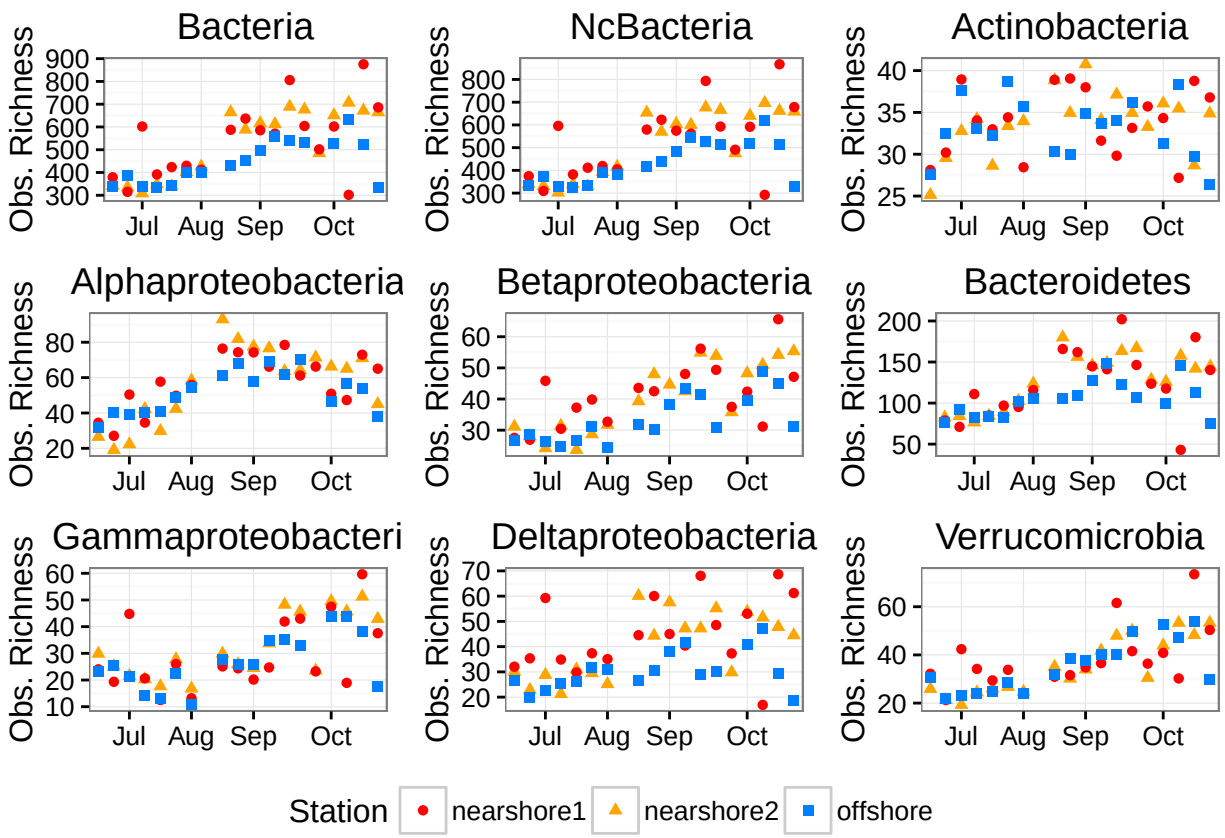
```
## Arrange plots for final figure
grid.arrange(
   obs_plots[[1]], obs_plots[[3]],  obs_plots[[7]], obs_plots[[8]],
   simp_plots[[1]], simp_plots[[3]],  simp_plots[[7]], simp_plots[[8]],
   ncol = 4
)
```



S4: Seasonal alpha diversity

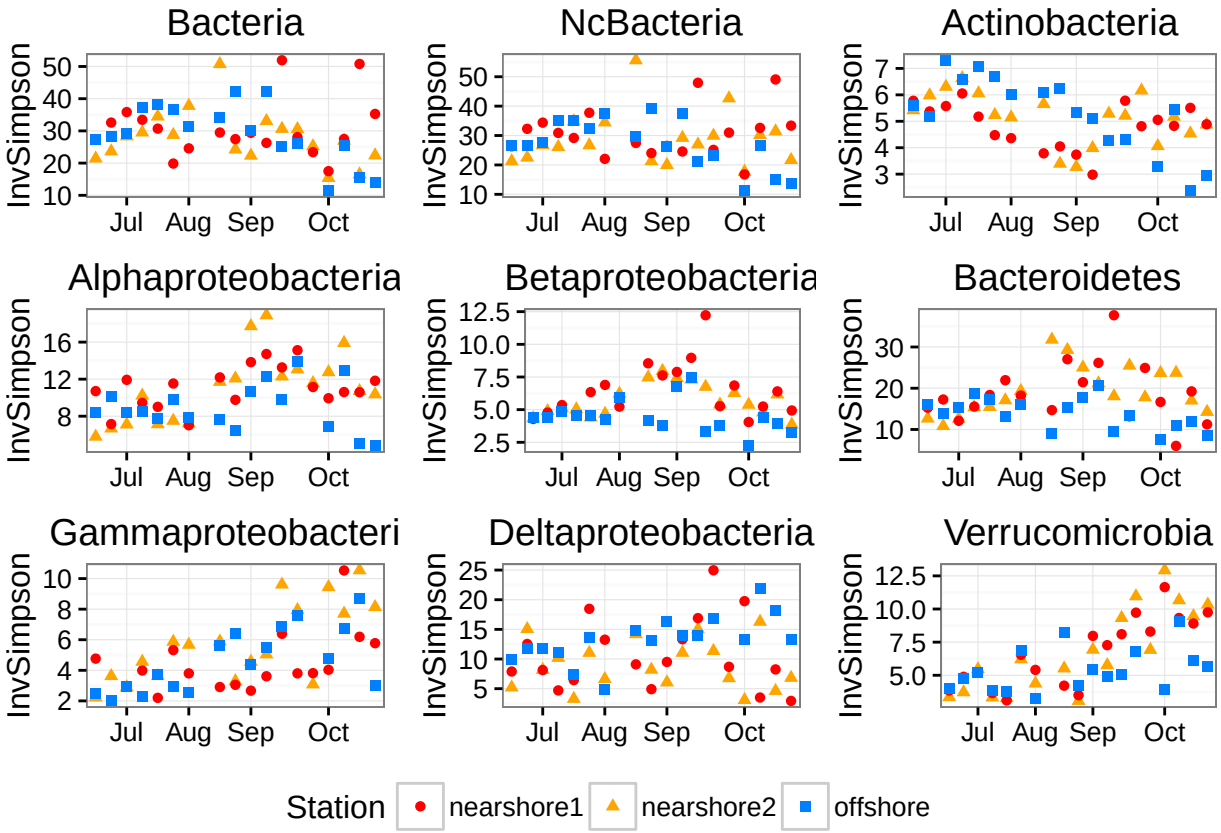
```
groups <- named_list("Bacteria", "NcBacteria", "Actinobacteria", "Alphaproteobacteria",
  "Betaproteobacteria", "Bacteroidetes", "Gammaproteobacteria", "Deltaproteobacteria",
  "Verrucomicrobia")

divs <- named_list(c("Richness", "InvSimpson"))

richness_time_plots <- lapply(groups, function(x) {
    dat <-
      alpha_comb %>%
        filter(Alphadiv == "Richness") %>%
        filter(Taxa == x) %>%
        order_dates

    g <- ggplot(dat, aes(x = Date, y = Estimate, group = Station, shape = Station, color = Station)) +
      geom_point() +
      ggtitle(x) +
      scale_color_manual(values = station_colors) +
      scale_x_discrete(
        breaks = c("7/8", "8/4", "9/2", "10/6"),
        labels = c("Jul", "Aug", "Sep", "Oct"),
        drop = FALSE
      ) +
      ylab("Obs. Richness") +
      theme(
        axis.title.x = element_blank(),
        axis.title.y = element_text(size = 12)
      )

    return(g)
```

```
})

rich <- do.call("grid_arrange_shared_legend", c(richness_time_plots))
```




```
simp_time_plots <- lapply(groups, function(x) {
    dat <-
      alpha_comb %>%
        filter(Alphadiv == "InvSimpson") %>%
        filter(Taxa == x) %>%
        order_dates

    g <- ggplot(dat, aes(x = Date, y = Estimate, group = Station, shape = Station, color = Station)) +
      geom_point() +
      ggtitle(x) +
      scale_color_manual(values = station_colors) +
      scale_x_discrete(
        breaks = c("7/8", "8/4", "9/2", "10/6"),
        labels = c("Jul", "Aug", "Sep", "Oct"),
        drop = FALSE
      ) +
      ylab("InvSimpson") +
      theme(
        axis.title.x = element_blank(),
        axis.title.y = element_text(size = 12)
      )
```

```
    return(g)
})

simp <- do.call("grid_arrange_shared_legend", c(simp_time_plots))
```



Linear mixed models

```
library(lme4)

rich_dat <-
  alpha_comb %>%
    filter(Alphadiv == "Richness") %>%
    filter(Taxa == "NcBacteria")

rich_lmm_fit <- lmer(Estimate ~ Days + 1|Station, data = rich_dat, REML = FALSE)
rich_lmm_null <- lmer(Estimate ~ 1|Station, data = rich_dat, REML = FALSE)
anova(rich_lmm_null, rich_lmm_fit)
```

```
## Data: rich_dat
## Models:
## rich_lmm_null: Estimate ~ 1 | Station
## rich_lmm_fit: Estimate ~ Days + 1 | Station
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## rich_lmm_null  3 679.97 685.88 -336.99   673.97
```

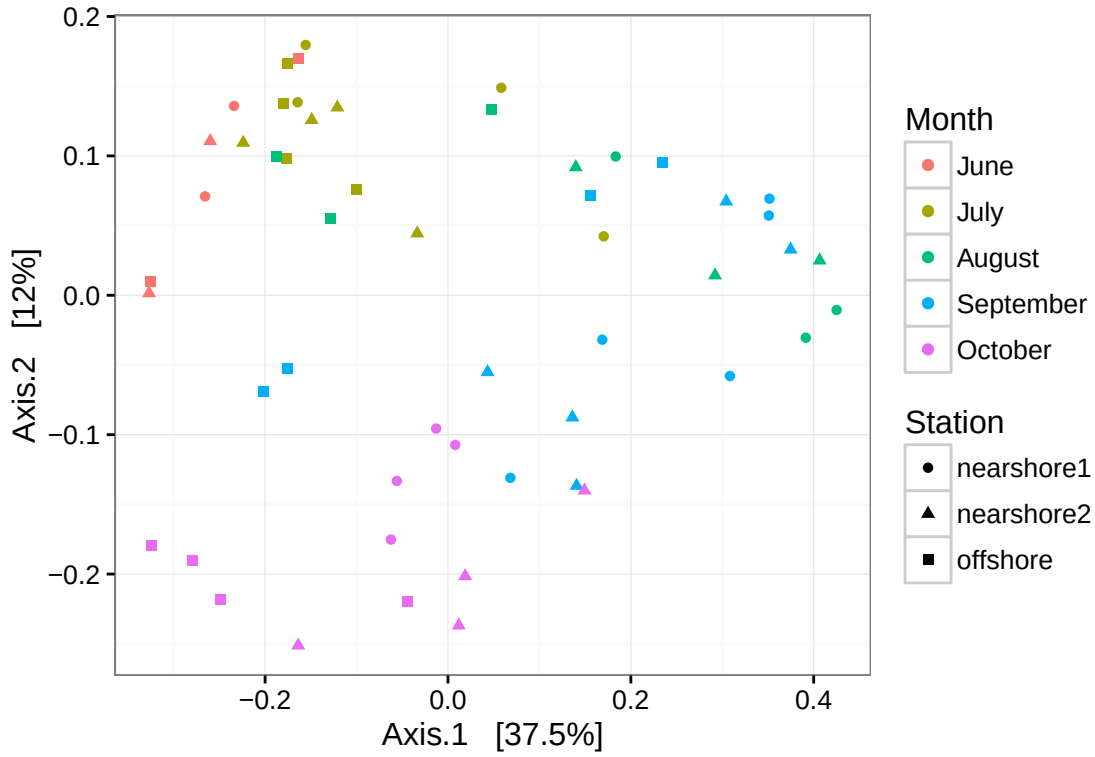
```
## rich_lmm_fit   5 657.30 667.15 -323.65   647.30 26.675      2  1.613e-06
##
## rich_lmm_null
## rich_lmm_fit  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simp_dat <-
  alpha_comb %>%
    filter(Alphadiv == "InvSimpson") %>%
    filter(Taxa == "NcBacteria")

simp_lmm_fit <- lmer(Estimate ~ Days + 1|Station, data = simp_dat, REML = FALSE)
simp_lmm_null <- lmer(Estimate ~ 1|Station, data = simp_dat, REML = FALSE)
anova(simp_lmm_null, simp_lmm_fit)
```

```
## Data: simp_dat
## Models:
## simp_lmm_null: Estimate ~ 1 | Station
## simp_lmm_fit: Estimate ~ Days + 1 | Station
##               Df   AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## simp_lmm_null  3 382.9 388.81 -188.45    376.9
## simp_lmm_fit   5 385.4 395.25 -187.70    375.4 1.4998      2     0.4724
```

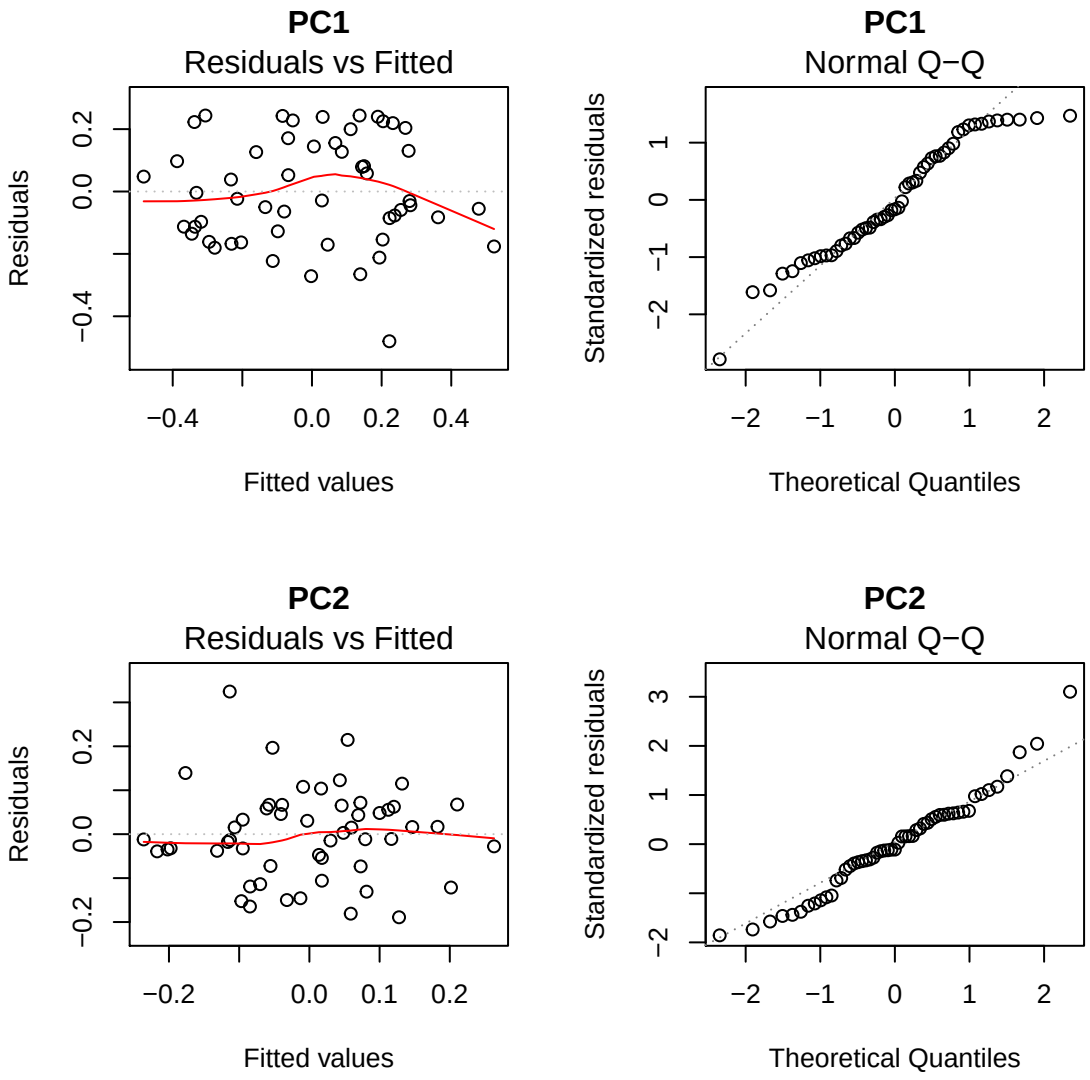
S5: PCoA full community

```
full_pcoa <- ordinate(
  physeq = erie_scale,
  method = "PCoA",
  distance = "bray"
)

sample_data(erie_scale)$Month <- factor(sample_data(erie_scale)$Month,
      levels = c("June", "July", "August", "September", "October"))

plot_ordination(
  physeq = erie_scale,
  axes = 1:3,
  ordination = full_pcoa,
  color = "Month",
  shape = "Station"
)
```



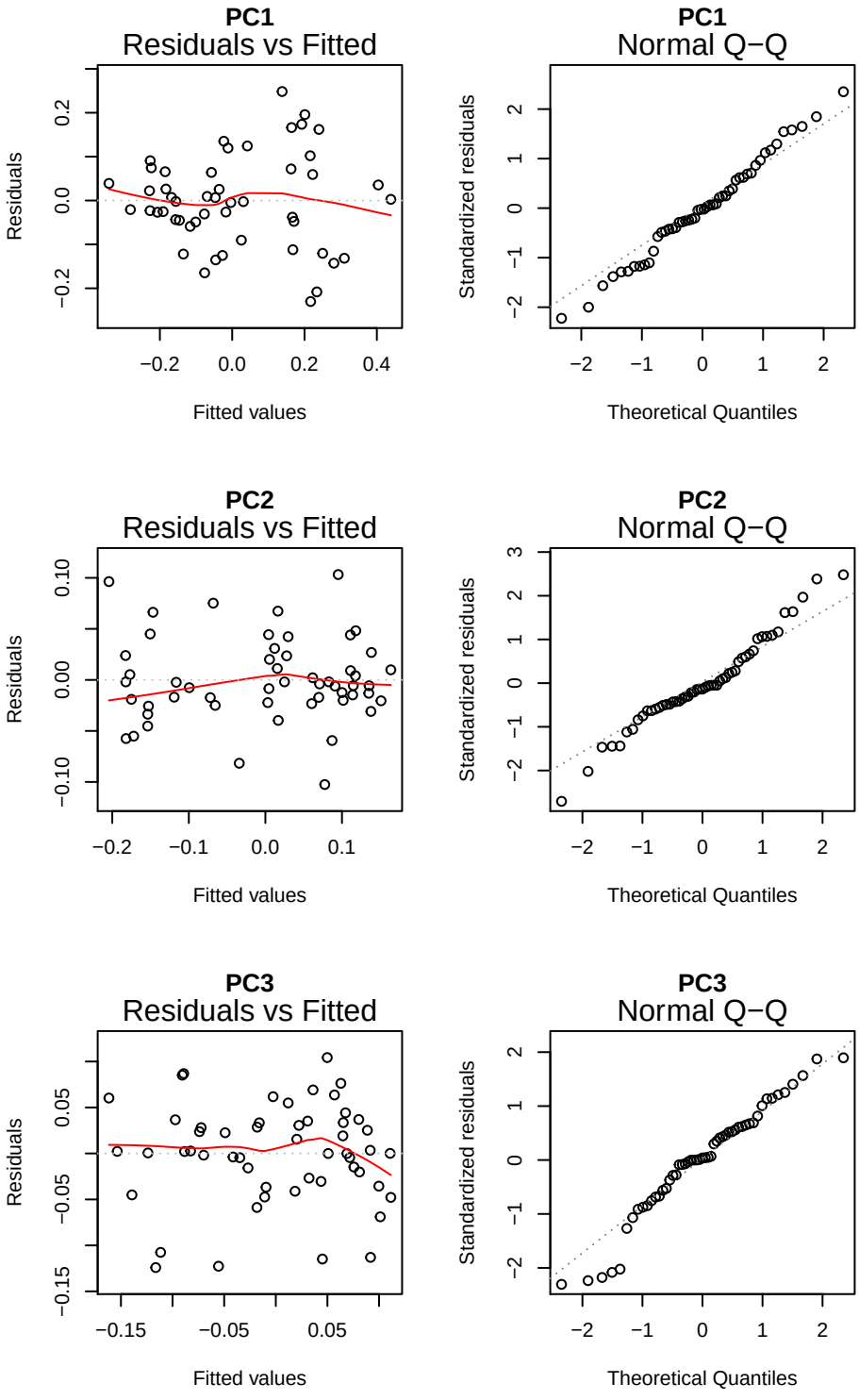
S6: Cyanobacteria linear model residuals and qqplots

```
par(mfrow = c(2,2))
plot(cyano_models$PC1, which = 1:2, main = "PC1", labels.id = NA)
plot(cyano_models$PC2, which = 1:2, main = "PC2", labels.id = NA)
```



S7: Nc-bacteria linear model residuals and qqplots

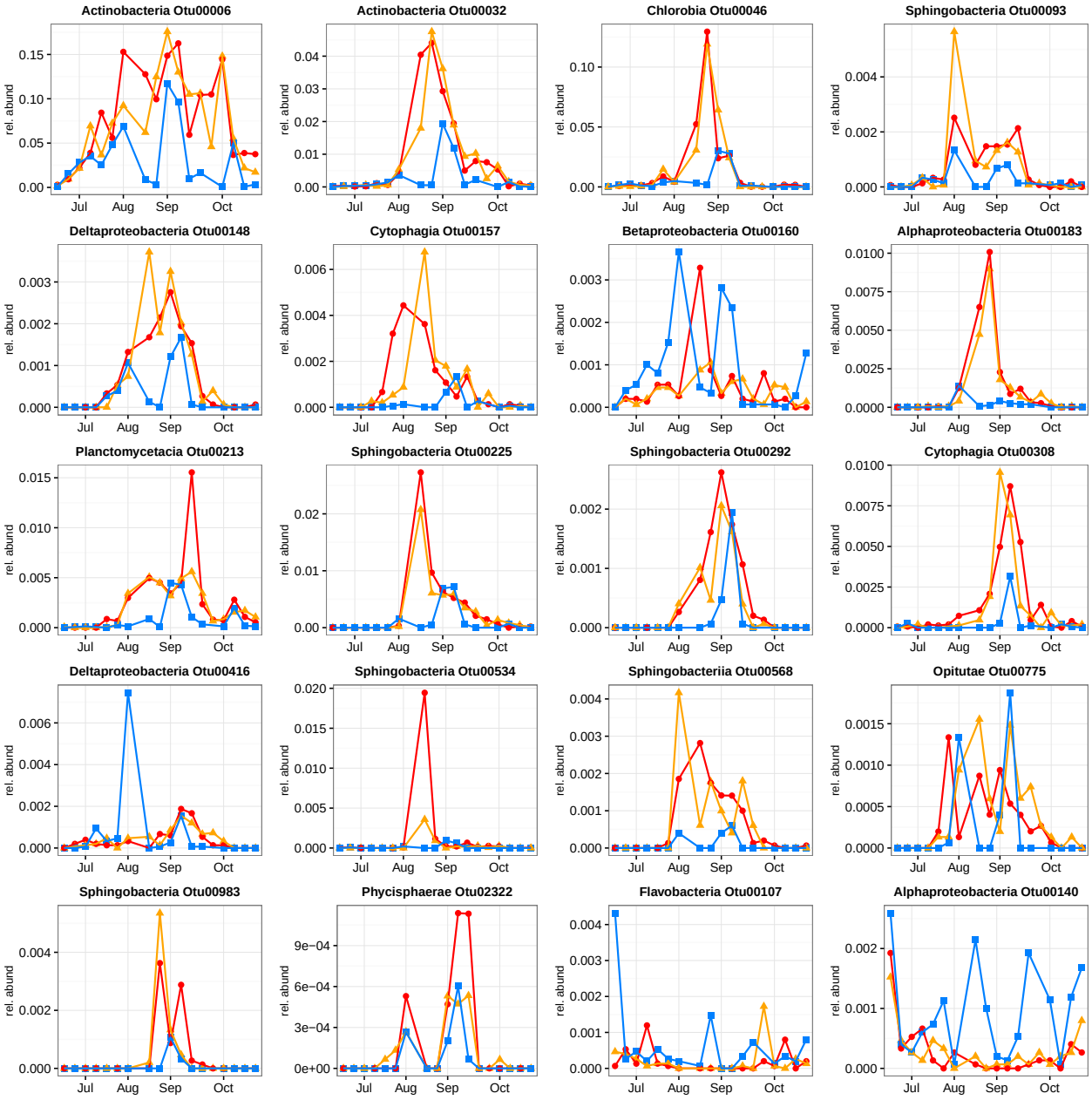
```
par(mfrow = c(3,2))
plot(non_cyano_models$PC1, which = 1:2, main = "PC1", labels.id = NA)
plot(non_cyano_models$PC2, which = 1:2, main = "PC2", labels.id = NA)
plot(non_cyano_models$PC3, which = 1:2, main = "PC3", labels.id = NA)
```



S8: Differentially abundant taxa on PC1

```
erie_long <-
  erie_scale %>%
```

```
    transform_sample_counts(function(x) {x/sum(x)}) %>%
    subset_taxa(Species %in% unlist(sig_otus)) %>%
    psmelt() %>%
    order_dates()

pc1_pos_plots <- lapply(sig_otus$pc1_pos,
  function(x) {
    df_otu <- filter(erie_long, OTU == x)
    plot_otus(df = df_otu, otu = x, taxrank = "Class")
  }
)

pc1_neg_plots <- lapply(sig_otus$pc1_neg,
  function(x) {
    df_otu <- filter(erie_long, OTU == x)
    plot_otus(df = df_otu, otu = x, taxrank = "Class")
  }
)

do.call("grid.arrange", c(pc1_pos_plots, pc1_neg_plots))
```


Actinobacteria Otu00006
Actinobacteria Otu00032
Chlorobia Otu00046
Sphingobacteria Otu00093
Deltaproteobacteria Otu00148
Cytophagia Otu00157
Betaproteobacteria Otu00160
Alphaproteobacteria Otu00183
Planctomycetacia Otu00213
Sphingobacteria Otu00225
Sphingobacteria Otu00292
Cytophagia Otu00308
Deltaproteobacteria Otu00416
Sphingobacteria Otu00534
Sphingobacteriia Otu00568
Opitutae Otu00775
Sphingobacteria Otu00983
Phycisphaerae Otu02322
Flavobacteria Otu00107
Alphaproteobacteria Otu00140
rel. abund
Jul
Aug
Sep
Oct