

Palmer Station 16S Analysis

Codey Phoun

4/27/2021

Introduction

The marine microbial ecosystem of the waters off the Western Antarctic Peninsula (WAP) in the Southern Ocean is inextricably tied to the season. Each spring brings forth a phytoplankton bloom and changes in the community composition of the bacteria and other microbes in the water. These microbes are the key players in the processes of biogeochemical cycling and carbon sequestration. The Southern Ocean has experienced dramatic warming due to the effects of anthropogenic climate change and the impact of climate change on the marine bacterial community is not fully understood. The Palmer Long Term Ecological Research (PAL LTER) project has been collecting ecological data on the WAP since 1990, but detailed molecular data on this ecosystem is under sampled. Baseline measurements on the taxonomic makeup of the microbial community are needed for future research. Water samples from the surface water of the Western Antarctic Peninsula were collected as part of Dr. Shellie Bench's project at Palmer Station, over the 2012-2013, 2013-2014, and 2014-2015 austral summer seasons. This project is focused on the 15 different 16S rRNA V4 amplicon libraries generated from the sequencing of microbes in these water samples. With this data, a metagenomic analysis can help reveal the pattern of changes in the community composition of bacteria that are associated to the phytoplankton blooms off the coast of the Western Antarctic Peninsula

These 15 samples were previously processed on the SJSU CoS HPC. Quality filtering and primer and adapter removal was performed with Cutadapt. The samples were then imported into QIIME2 for further processing. The 16s libraries were run through DADA2 in QIIME2 to create amplicon sequence variants (ASVs) by denoising and dereplicating paired-end sequences before filtering for chimeras. ASVs were then taxonomically classified with VSEARCH and the SILVA-138 database. A phylogenetic tree was then created by IQ-TREE in QIIME2 for downstream analysis methods that required phylogenetic information.

Setup

Import libraries

```
library(tidyverse) # Data processing
library(ggplot2) # Plot figures
library(qiime2R) # QIIME2 artifacts to phyloseq object
library(vegan) # Ecology analysis
library(phyloseq) # Base microbiome data structure
library(microbiome) # Microbiome data analysis and visualization
library(phylosmith) # Microbiome data analysis and visualization
library(microbiomeutilities) # Microbiome data analysis and visualization
library(ggordiplots) # Vegan ordination plots for ggplot2
library(lubridate) # Date formatting
library(readxl) # Read in excel .xlsx
library(ggrepel) # Prevent overlapping text in figures
library(eulerr) # Venn diagram visualization for core microbiome
```

```
library(treemap) # Tree map visualization
library(ggpubr) # Data visualization - wrapper
library(rstatix) # Tidy statistical tests
library(RColorBrewer) # Color selection for graphics
library(tidytext) # Text repel on graphics
library(dendextend) # Dendrogram plotting
```

Load phyloseq-extended functions

```
source("https://raw.githubusercontent.com/mahendra-mariadassou/phyloseq-extended/master/load-extra-functions.R")
```

R Environment Setup

```
theme_set(theme_bw()) # set ggplot2 default theme
set.seed(100) # set seed for reproducibility with RNG functions
```

Import Data from QIIME2

Import QIIME2 artifacts produced by the 16s_full_pipeline.sh script to a Phyloseq object called physeq

```
physeq <- qza_to_phyloseq(
  features = "./qiime2/16S_libraries_feature_table_clean.qza", # ASV/OTU table
  tree = "./phylogeny/16S_libraries_iqtree_rooted.qza", # Phylogenetic tree
  taxonomy = "./qiime2/16S_libraries_vsearch_taxonomy.qza", # Taxonomy file
  metadata = "./metadata/16S_metadata.tsv" # Sample metadata
)
```

Initial structure of the Phyloseq object physeq

```
physeq
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2430 taxa and 15 samples ]
## sample_data() Sample Data: [ 15 samples by 18 sample variables ]
## tax_table() Taxonomy Table: [ 2430 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 2430 tips and 2429 internal nodes ]
```

Samples were split into three different summer stages for comparisons requiring categorical variables.

The austral summer in Antarctica lasts from November through March

Early Summer - Late November through Mid January

Mid Summer - Mid January through Mid February

Late Summer - Mid February through March

Show sample dates and summer stages for each sample

```
Sample_Name_Date <- tibble("Sample Name" = sample_names(physeq),
                           "Sample Date" = physeq@sam_data$lib_date,
                           "Summer Stage" = physeq@sam_data$Summer_Stage)

Sample_Name_Date %>%
  print(n = 15)
```

```
## # A tibble: 15 x 3
##   'Sample Name' 'Sample Date' 'Summer Stage'
##   <chr>         <fct>         <fct>
## 1 S1L13        11/27/2012    Early Summer
## 2 S1L14        2/8/2013      Mid Summer
## 3 S2L05        12/27/2013    Early Summer
## 4 S2L06        1/23/2014     Mid Summer
## 5 S2L07        2/3/2014      Mid Summer
## 6 S2L08        2/10/2014     Mid Summer
## 7 S2L09        2/28/2014     Late Summer
## 8 S2L10        3/4/2014      Late Summer
## 9 S3L03        12/1/2014     Early Summer
## 10 S3L04       12/11/2014    Early Summer
## 11 S3L05       1/12/2015     Mid Summer
## 12 S3L06       1/19/2015     Mid Summer
## 13 S3L07       2/9/2015      Mid Summer
## 14 S3L08       2/23/2015     Late Summer
## 15 S3L09       3/9/2015      Late Summer
```

Summer stages

```
table(physeq@sam_data$Summer_Stage)
```

```
##
## Early Summer Late Summer Mid Summer
##           4           4           7
```

Library seasons

```
table(physeq@sam_data$lib_season)
```

```
##
## 12-13 13-14 14-15
##      2      6      7
```

View taxa ranks

```
rank_names(physeq)
```

```
## [1] "Kingdom" "Phylum" "Class" "Order" "Family" "Genus" "Species"
```

Process and Modify the Phyloseq Object

Add total read counts to each library's sample data

```
sample_data(physeq)$total_reads <- sample_sums(physeq)
```

Convert "Summer_Stage" from character to an R factor

```
physeq@sam_data$Summer_Stage <- factor(physeq@sam_data$Summer_Stage,  
                                         levels = c("Early Summer", "Mid Summer", "Late Summer"))
```

Remove d__ prefix from taxa rank Kingdom

```
tax_table(physeq)[,1] <- gsub( "d__", "", tax_table(physeq)[,1])
```

Scale and center environmental metadata

```
sample_data(physeq)[,6:14] <- scale(sample_data(physeq)[,6:14]) # columns 6:14 contain numerical env da
```

Make vector of the sample collection dates for downstream labeling of libraries

```
lib_dates <- as.Date(paste(c(sample_data(physeq)$year),  
                           c(sample_data(physeq)$month),  
                           c(sample_data(physeq)$day),  
                           sep = "-"))
```

Agglomerate taxa on Genus level

Species labels are unreliable and many taxa are labeled as "uncultured" at species level.

```
physeq <- physeq %>%  
tax_glom(taxrank = "Genus", NArm = TRUE) # agglomerate on Genus level  
tax_table(physeq) <- tax_table(physeq)[,1:6] # drop taxa rank Species from tax_table
```

View physeq after agglomeration

```
physeq
```

```
## phyloseq-class experiment-level object  
## otu_table() OTU Table: [ 384 taxa and 15 samples ]  
## sample_data() Sample Data: [ 15 samples by 19 sample variables ]  
## tax_table() Taxonomy Table: [ 384 taxa by 6 taxonomic ranks ]  
## phy_tree() Phylogenetic Tree: [ 384 tips and 383 internal nodes ]
```

2430 species agglomerated to 384 unique genera

Rarefy to the smallest library size

Rarefaction normalizes for the differences in library sizes/sequencing depth. Historically, it has been widely used but the usefulness is under debate in the literature.

```
sample_sums(physeq)
```

```
## S1L13 S1L14 S2L05 S2L06 S2L07 S2L08 S2L09 S2L10 S3L03 S3L04 S3L05
## 324962 362535 329636 501253 548000 532506 593411 494396 221245 274018 192952
## S3L06 S3L07 S3L08 S3L09
## 377212 289685 241639 235339
```

```
min(sample_sums(physeq))
```

```
## [1] 192952
```

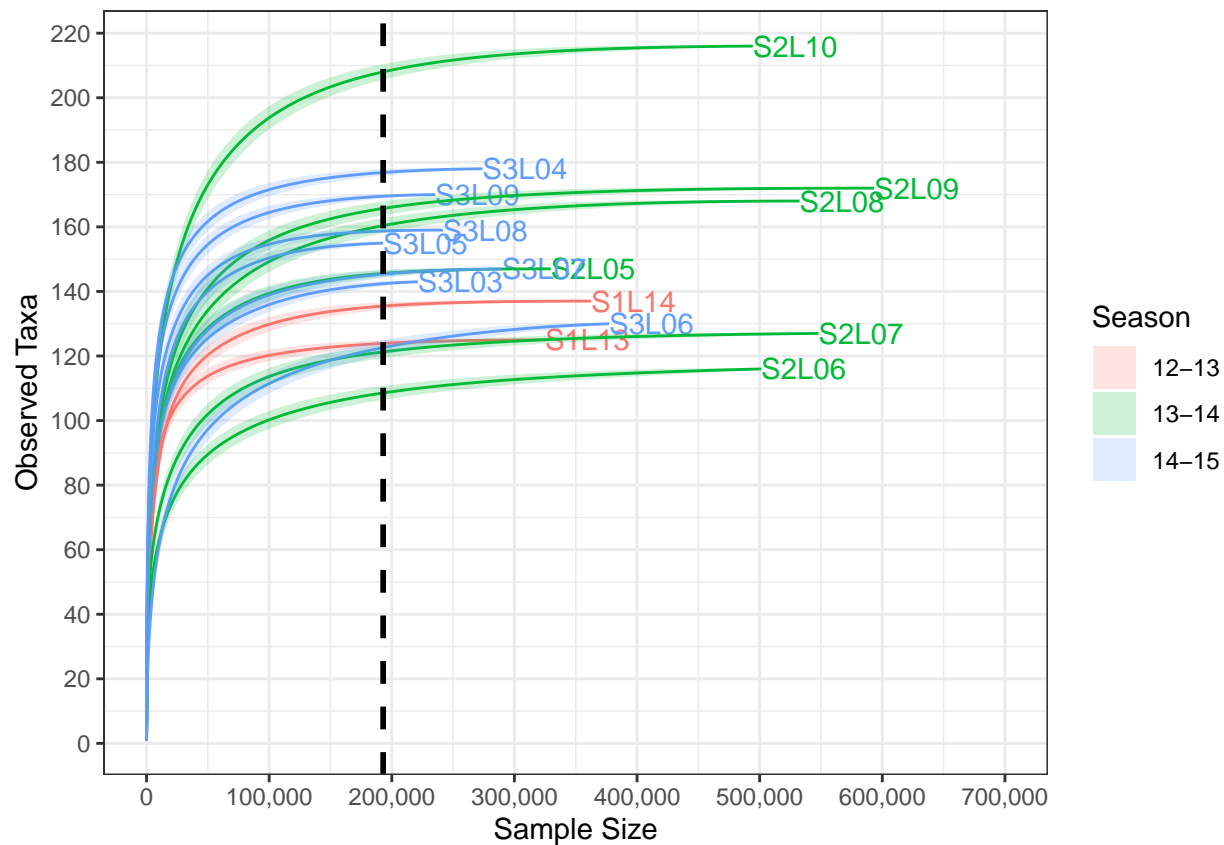
S3L05 has the lowest read count at 192,952 reads. All samples will be rarefied to 192,952.

Rarefy without replacement

```
physeq_rarefy <- rarefy_even_depth(physeq, sample.size = min(sample_sums(physeq)),
                                   rngseed = FALSE, replace = FALSE, trimOTUs = TRUE, verbose = FALSE)
```

Plot a rarefaction curve

```
rare_plot <- ggrare(physeq, step = 500, label = sample_names(physeq), # from phyloseq-extended function
                   color = "lib_season", parallel = TRUE) +
  geom_vline(xintercept = min(sample_sums(physeq)), # create vertical line on smallest sample
            linetype="dashed", size = 1) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 7),
                    labels = scales::comma,
                    limits = c(0,700000)) +
  ylab("Observed Taxa") +
  guides(color = FALSE) +
  labs(fill = "Season")
rare_plot
```



Rarefying to the smallest sample size looks to be OK because there is not a large increase in observed taxa (low slope) for each library after rarefying to 192,952.

Total number of reads before rarefying

```
sum(sample_sums(physeq))
```

```
## [1] 5518789
```

Total number of reads after rarefying

```
sum(sample_sums(physeq_rarefy))
```

```
## [1] 2894280
```

Proportion of rarefied reads to original number of reads

```
2894280/5518789
```

```
## [1] 0.5244411
```

About 52% of the original 5.5 million reads are left after rarefying. The rarefied library has almost 2.9 million reads.

Import the Environmental Metadata

Environmental data collected from Palmer Station Sample Site B at a depth of about 10 meters

<http://pal.lternet.edu/data>

Helper function to transform the data

```
data_transform <- function(df) {
  df %>%
    separate("Date", into = c("Year", "Month", "Day"), # separate and create new columns for dates
             remove = FALSE, convert = TRUE) %>%
    mutate(year_plot = ifelse(Month >= 10, 2000, # for plotting multiple years
                              ifelse(Month < 10, 2001, NA))) %>% # onto the same scale
    mutate(plot_date = make_date(year_plot, Month, Day)) %>%
    mutate(lib_season = case_when( # add library season to the dataframe
      (Year == 2012) ~ "12-13",
      (Year <= 2013 & Month <= 3) ~ "12-13",
      (Year == 2013 & Month > 3) ~ "13-14",
      (Year == 2014 & Month <= 3) ~ "13-14",
      (Year == 2014 & Month > 3) ~ "14-15",
      Year == 2015 ~ "14-15",
      TRUE ~ "none")) %>%
    mutate(Date = as.Date(Date)) %>%
    mutate(sample_16s = Date %in% lib_dates) %>% # TRUE/FALSE if date is a 16s library date
    mutate(month_day = paste(Month, Day, sep = "-"))
}
```

Read in Bacterial Abundance and Bacterial Production data

```
bacterial_abundance <- read_excel("./Palmer_Station_Metadata/Bacteria_B.xlsx", na = "-999") %>%
  data_transform()

head(bacterial_abundance)
```

```
## # A tibble: 6 x 12
##   studyName Date      Year Month Day 'Abundance (num/L~ 'Leucine Incorp. (p~
##   <chr>      <date>    <int> <int> <int>      <dbl>          <dbl>
## 1 PAL1213 2012-10-31 2012 10 31      357076923.      7.24
## 2 PAL1213 2012-11-07 2012 11 7      306538462.     10.9
## 3 PAL1213 2012-11-10 2012 11 10     212076923.     14.1
## 4 PAL1213 2012-11-14 2012 11 14     248384615.     14.1
## 5 PAL1213 2012-11-16 2012 11 16     244769231.     15.4
## 6 PAL1213 2012-11-19 2012 11 19     231615385.     11.3
## # ... with 5 more variables: year_plot <dbl>, plot_date <date>,
## #   lib_season <chr>, sample_16s <lgl>, month_day <chr>
```

Read in Chlorophyll a data

```
chlorophyll <- read_excel("./Palmer_Station_Metadata/Chlorophyll_B.xlsx", na = "-999") %>%
  data_transform()

head(chlorophyll)
```

```
## # A tibble: 6 x 12
##   studyName Date       Year Month Day 'Chlorophyll (mg/m~ 'Phaeopigment (mg/~
##   <chr>      <date>      <int> <int> <int>      <dbl>      <dbl>
## 1 PAL1213    2012-10-31    2012   10   31         1.34        -0.0398
## 2 PAL1213    2012-11-07    2012   11    7         5.39         0.130
## 3 PAL1213    2012-11-10    2012   11   10         5.14         0.260
## 4 PAL1213    2012-11-14    2012   11   14         4.88        -0.0764
## 5 PAL1213    2012-11-16    2012   11   16         2.60         0.307
## 6 PAL1213    2012-11-19    2012   11   19         6.90         0.345
## # ... with 5 more variables: year_plot <dbl>, plot_date <date>,
## #   lib_season <chr>, sample_16s <lgl>, month_day <chr>
```

Read in Primary Production data

```
primary_production <- read_excel("./Palmer_Station_Metadata/Primary_Production_B.xlsx",
                                na = "-999" ) %>%
  data_transform()

head(primary_production)
```

```
## # A tibble: 6 x 13
##   studyName Date       Year Month Day 'Primary Prod. ~ 'Prim Prod STD ~ Notes
##   <chr>      <date>      <int> <int> <int>      <dbl>      <dbl> <chr>
## 1 PAL1213    2012-10-31    2012   10   31         56.9        NA   <NA>
## 2 PAL1213    2012-11-07    2012   11    7        243.        77.8 <NA>
## 3 PAL1213    2012-11-10    2012   11   10        266.       148. <NA>
## 4 PAL1213    2012-11-14    2012   11   14         62.3        32.1 <NA>
## 5 PAL1213    2012-11-16    2012   11   16         46.0         7.23 <NA>
## 6 PAL1213    2012-11-19    2012   11   19        117.        49.4 <NA>
## # ... with 5 more variables: year_plot <dbl>, plot_date <date>,
## #   lib_season <chr>, sample_16s <lgl>, month_day <chr>
```

Read in Temperature and Salinity data

```
CTD_B <- read_excel("./Palmer_Station_Metadata/CTD_B_downcast_12-15_clean.xlsx") %>%
  data_transform()

CTD_B
```

```
## # A tibble: 99 x 17
##   file      Date       Year Month Day 'Temperature (°C)' 'Conductivity (S/~
##   <chr>      <date>      <int> <int> <int>      <dbl>      <dbl>
## 1 x121031B.~ 2012-10-31    2012   10   31        -1.53         2.70
## 2 x121107B.~ 2012-11-07    2012   11    7        -1.49         2.70
## 3 x121110B.~ 2012-11-10    2012   11   10        -1.10         2.73
## 4 x121114B.~ 2012-11-14    2012   11   14        -1.33         2.71
## 5 x121116B.~ 2012-11-16    2012   11   16        -1.29         2.71
## 6 x121119B.~ 2012-11-19    2012   11   19        -1.29         2.70
## 7 x121122B.~ 2012-11-22    2012   11   22        -1.10         2.73
## 8 x121127B.~ 2012-11-27    2012   11   27        -0.445        2.77
## 9 x121130B.~ 2012-11-30    2012   11   30        -0.378        2.78
## 10 x121207B.~ 2012-12-07    2012   12    7        -0.317        2.79
## # ... with 89 more rows, and 10 more variables: Pressure (dbar) <dbl>,
```



```
## # Fluorescence (mg/m³) <dbl>, Salinity <dbl>, Depth (m) <dbl>,
## # Density (kg/m³) <dbl>, year_plot <dbl>, plot_date <date>, lib_season <chr>,
## # sample_16s <lgl>, month_day <chr>
```

Read in Inorganic Nutrients - Phosphate, Silicate, and the Nitrite and Nitrate data

```
inorganic_nutrients <- read_excel("./Palmer_Station_Metadata/Dissolved_Inorganic_Nutrients_B.xlsx",
                                   na = "-999" ) %>%
  data_transform()
head(inorganic_nutrients)
```

```
## # A tibble: 6 x 13
##   studyName Date       Year Month Day 'Phosphate (µmol/L)' 'Silicate (µmol/L~
##   <chr>      <date>    <int> <int> <int> <dbl>                <dbl>
## 1 PAL1213  2012-10-31  2012    10   31      2.01                65.7
## 2 PAL1213  2012-11-07  2012    11    7      1.81                62.6
## 3 PAL1213  2012-11-10  2012    11   10      0.66                64.3
## 4 PAL1213  2012-11-14  2012    11   14      1.75                61.0
## 5 PAL1213  2012-11-16  2012    11   16      1.88                61.8
## 6 PAL1213  2012-11-19  2012    11   19      1.50                58.1
## # ... with 6 more variables: Nitrite and Nitrate (µmol/L) <dbl>,
## #   year_plot <dbl>, plot_date <date>, lib_season <chr>, sample_16s <lgl>,
## #   month_day <chr>
```

Join some of the environmental data to a single data frame

```
palmer_env <- full_join(chlorophyll,
                        primary_production,
                        by = "Date")
palmer_env <- full_join(palmer_env,
                        bacterial_abundance,
                        by = "Date") %>%
  data_transform()
palmer_env$Notes <- NULL
```

Plot and Explore the Environmental Metadata

Chlorophyll a and Bacterial Production (Leucine Incorporation)

```
env_12_13 <- ggplot(na.omit(subset(palmer_env, lib_season %in% c("12-13"))), # subset to 2012-2013 seas
                    aes(x = Date)) +
  geom_line(aes(y = `Chlorophyll (mg/m³)`,
                 color = "Chlorophyll (mg/m³)",
                 size = 1.05, alpha = 0.75,) +
  geom_line(aes(y = `Leucine Incorp. (pmol/L/hr)`,
                 color = "Leucine Incorp. (pmol/L/hr)",
                 size = 1.05, alpha = 0.75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
```

```

        limits = as.Date(c('2012-10-31','2013-03-21')), expand = c(0,0),
        sec.axis = sec_axis(~., breaks = lib_dates[1:2], # add sample dates as a sec
                           labels = scales::date_format("%m/%d"))) +
labs(title = "2012-2013") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 5), limits = c(0,36),
                   sec.axis = sec_axis(~ .*5, name = "Leucine Incorp. (pmol/L/hr)", # add
                                       breaks = scales::pretty_breaks(n = 5) )) +
theme(axis.title.y = element_blank(), # remove X and Y axis title labels
      axis.title.x = element_blank(),
      legend.title = element_blank()) +
geom_vline(xintercept = as.numeric(lib_dates[1:2]), linetype = 4) + # draw vertical line
scale_color_manual(values = c("#00BA38", "#619CFF"))

```

```

env_13_14 <- ggplot(na.omit(subset(palmer_env, lib_season %in% c("13-14"))),
  aes(x = Date)) +
  geom_line(aes(y = `Chlorophyll (mg/m³)`,
               color = "Chlorophyll (mg/m³)",
               size = 1.05, alpha = 0.75,) +
  geom_line(aes(y = `Leucine Incorp. (pmol/L/hr)`/5,
               color = "Leucine Incorp. (pmol/L/hr)",
               size = 1.05, alpha = 0.75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
               limits = as.Date(c('2013-10-31','2014-03-21')), expand = c(0,0),
               sec.axis = sec_axis(~., breaks = lib_dates[3:8],
                                   labels = scales::date_format("%m/%d"))) +
  labs(title = "2013-2014") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5), limits = c(0,36),
                    sec.axis = sec_axis(~ .*5, name = "Leucine Incorp. (pmol/L/hr)",
                                        breaks = scales::pretty_breaks(n = 5) )) +
  theme(axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        legend.title = element_blank()) +
  geom_vline(xintercept=as.numeric(lib_dates[3:8]), linetype = 4) +
  scale_color_manual(values = c("#00BA38", "#619CFF"))

```

```

env_14_15 <- ggplot(na.omit(subset(palmer_env, lib_season %in% c("14-15"))),
  aes(x = Date)) +
  geom_line(aes(y = `Chlorophyll (mg/m³)`,
               color = "Chlorophyll (mg/m³)",
               size = 1.05, alpha = 0.75,) +
  geom_line(aes(y = `Leucine Incorp. (pmol/L/hr)`/5,
               color = "Leucine Incorp. (pmol/L/hr)",
               size = 1.05, alpha = 0.75) +
  scale_x_date(date_breaks="months", date_labels="%b",
               limits = as.Date(c('2014-10-31','2015-03-21')), expand = c(0,0),
               sec.axis = sec_axis(~., breaks = lib_dates[9:15],
                                   labels = scales::date_format("%m/%d"))) +
  labs(title = "2014-2015") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5), limits = c(0,36),
                    sec.axis = sec_axis(~ .*5, name = "Leucine Incorp. (pmol/L/hr)",
                                        breaks = scales::pretty_breaks(n = 5) )) +
  theme(axis.title.y = element_blank(),
        axis.title.x = element_blank(),

```

```

legend.title = element_blank() +
geom_vline(xintercept=as.numeric(lib_dates[9:15]), linetype=4) +
scale_color_manual(values = c("#00BA38", "#619CFF"))

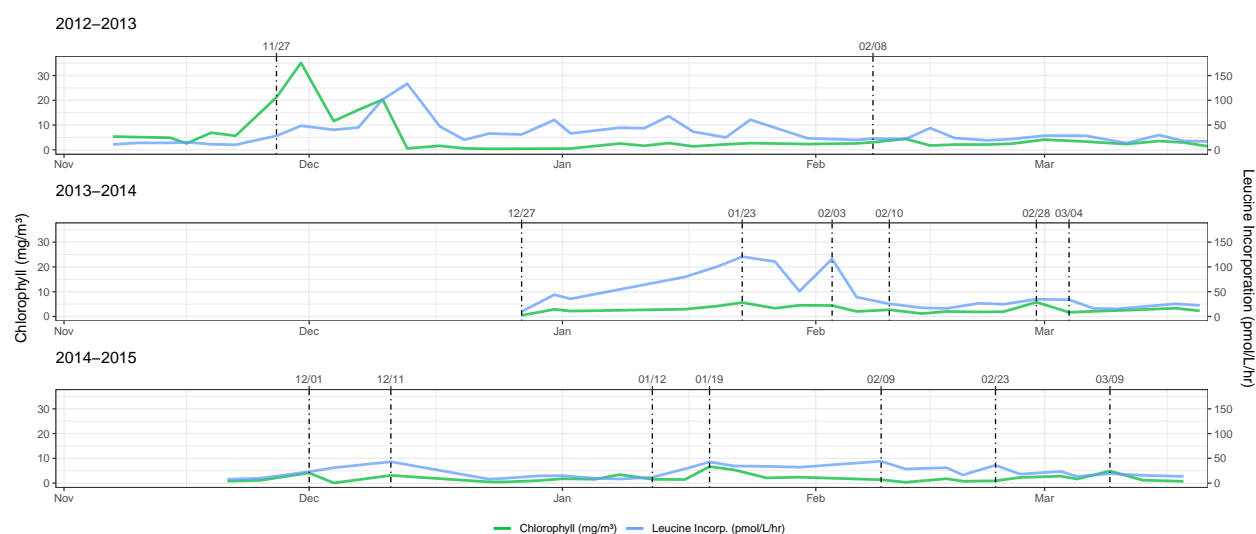
```

Dotted vertical lines with dates indicates a 16S sampling date

```

env_fig <- ggarrange(env_12_13, env_13_14, env_14_15,
  ncol = 1,
  common.legend = TRUE,
  label.y = "Chlorophyll (mg/m³)",
  legend = "bottom")
annotate_figure(env_fig,
  left = text_grob("Chlorophyll (mg/m³)", rot = 90),
  right = text_grob("Leucine Incorporation (pmol/L/hr)", rot = 270))

```



Bacterial Abundance

```

bac_abund <- ggplot(bacterial_abundance,
  aes(x = plot_date, y = `Abundance (num/L)`,
  group = factor(lib_season), color = factor(lib_season))) + # plot each season separately
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31', '2001-03-25')), # overlay multiple years on
    expand = c(0,0)) +
  labs(title = "Bacterial Abundance", x = "Month", color = "Season") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.y = element_text(margin = margin(r = 15)),
    axis.title.x = element_blank()) +
  geom_point(data = filter(bacterial_abundance, sample_16s == TRUE),
    aes(x = plot_date, y = `Abundance (num/L)`),
    pch = 18, size = 3.5, alpha = .85)

```

Bacterial Production

```

bac_prod <- ggplot(bacterial_abundance,
  aes(x = plot_date, y = `Leucine Incorp. (pmol/L/hr)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31', '2001-03-25')),
    expand = c(0,0)) +
  labs(title = "Bacterial Production", x = "Month", color = "Season") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.y = element_text(margin = margin(r = 15)),
    axis.title.x = element_blank()) +
  geom_point(data = filter(bacterial_abundance, sample_16s == TRUE),
    aes(x = plot_date, y = `Leucine Incorp. (pmol/L/hr)`,
      pch = 18, size = 3.5, alpha = .85)

```

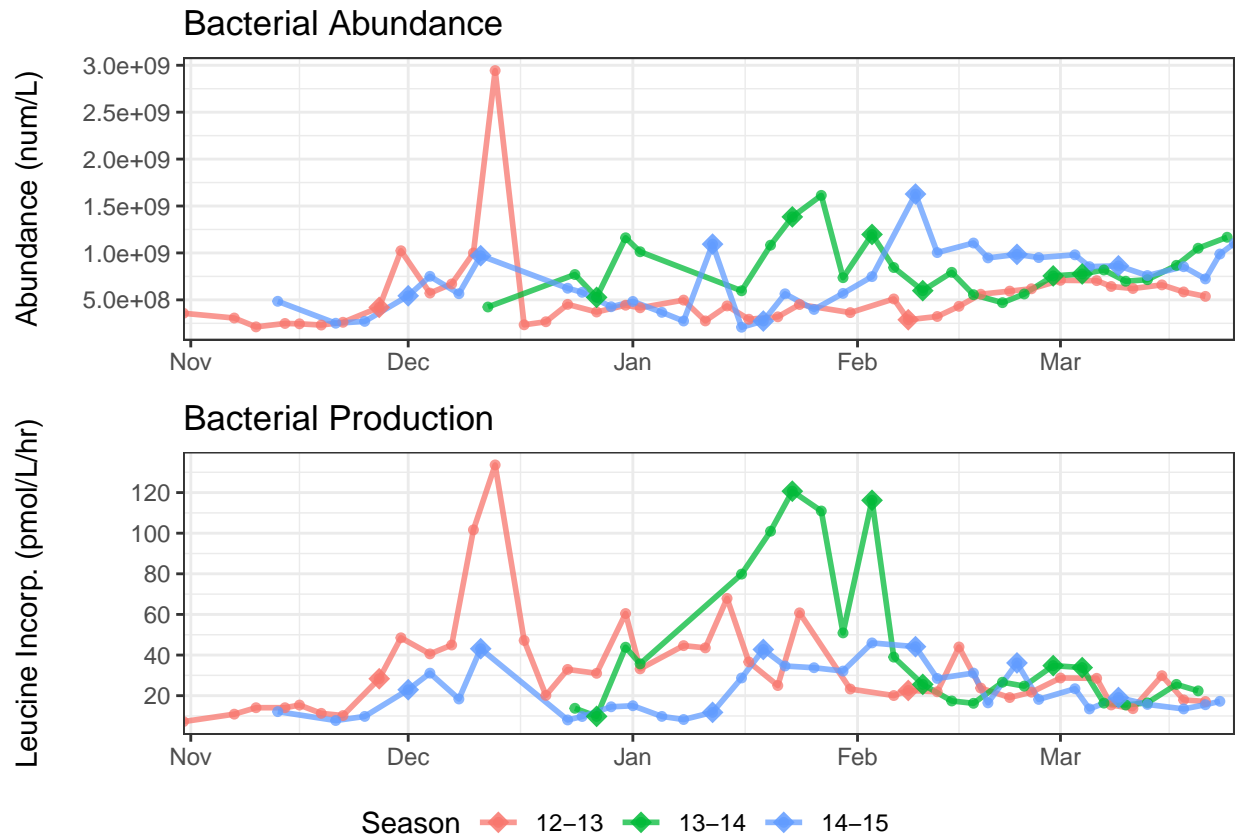
Bacterial Abundance and Production

Diamonds indicates a 16s sample date

```

ggarrange(bac_abund,
  bac_prod,
  ncol = 1,
  common.legend = TRUE,
  legend = "bottom",
  align = "v")

```



Chlorophyll

```
chlor <- ggplot(chlorophyll,
  aes(x = plot_date, y = `Chlorophyll (mg/m³)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31', '2001-03-25')),
    expand = c(0,0)) +
  labs(title = "Chlorophyll a", x = "Month", color = "Season") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  theme(axis.title.y = element_text(margin = margin(r = 15)),
    axis.title.x = element_blank()) +
  geom_point(data = filter(chlorophyll, sample_16s == TRUE),
    aes(x = plot_date, y = `Chlorophyll (mg/m³)`),
    pch = 18, size = 3.5, alpha = .85)
```

Primary Production

```
prim_prod <- ggplot(primary_production,
  aes(x = plot_date, y = `Primary Prod. (mg/m³/day)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
```

```

limits = as.Date(c('2000-10-31', '2001-03-25')),
expand = c(0,0)) +
labs(title = "Primary Production", x = "Month", color = "Season") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
theme(axis.title.y = element_text(margin = margin(r = 15)),
axis.title.x = element_blank()) +
geom_point(data = filter(primary_production, sample_16s == TRUE),
aes(x = plot_date, y = `Primary Prod. (mg/m³/day)`),
pch = 18, size = 3.5, alpha = .85)

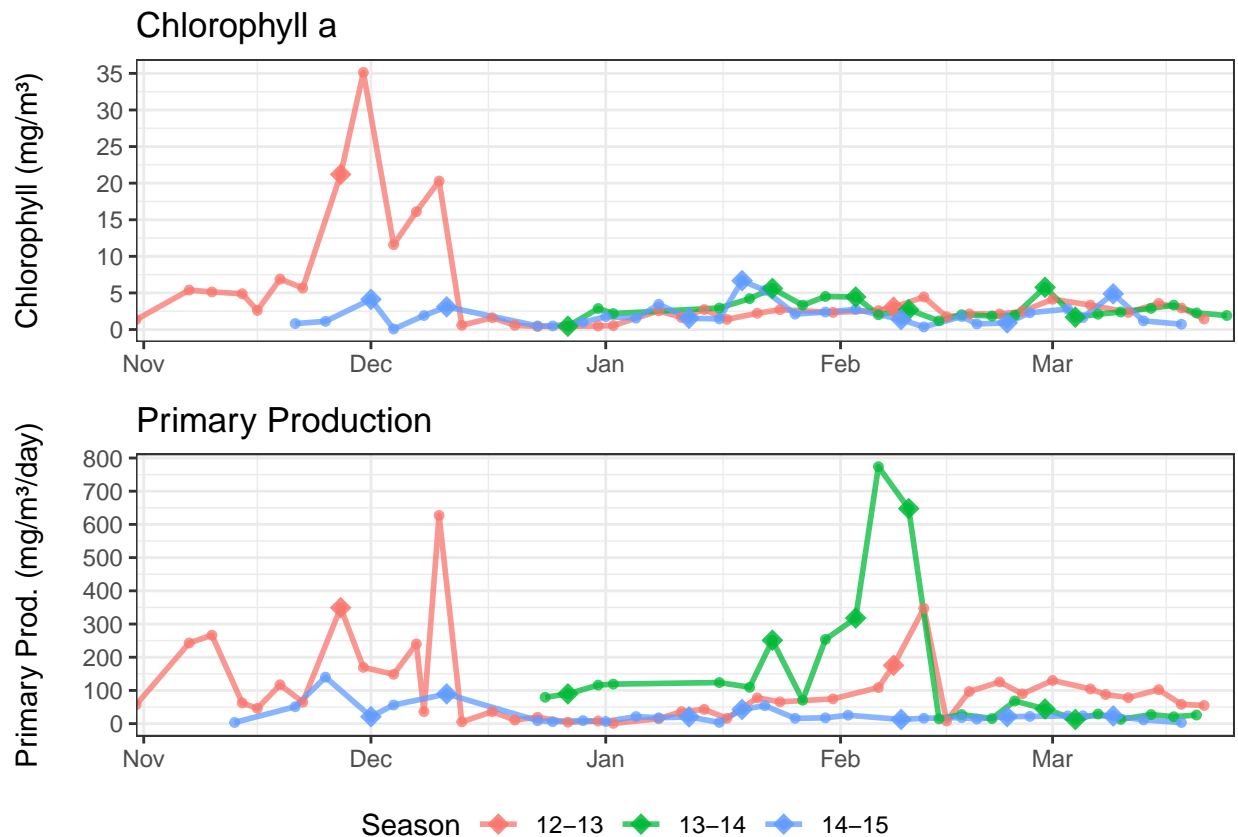
```

Chlorophyll a and Primary Production

```

ggarrange(chlor,
prim_prod,
ncol = 1,
common.legend = TRUE,
legend = "bottom",
align = "v")

```



Water Temperature

```

temp <- ggplot(CTD_B,
aes(x = plot_date, y = `Temperature (°C)`),

```

```

group = factor(lib_season), color = factor(lib_season))) +
geom_line(size = 1, alpha = .75) +
geom_point(size = 1.25, alpha = .75) +
scale_x_date(date_breaks = "months", date_labels = "%b",
              limits = as.Date(c('2000-10-31', '2001-03-25')),
              expand = c(0,0)) +
labs(title = "Water Temperature", x = "Month", color = "Season") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(-2, 2.5)) +
theme(axis.title.y = element_text(margin = margin(r = 15)),
      axis.title.x = element_blank()) +
geom_point(data = filter(CTD_B, sample_16s == TRUE),
          aes(x = plot_date, y = `Temperature (°C)`),
          pch = 18, size = 3.5, alpha = .85)

```

Water Salinity

```

sal <- ggplot(CTD_B,
  aes(x = plot_date, y = `Salinity`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31', '2001-03-25')),
    expand = c(0,0)) +
  labs(title = "Water Salinity", x = "Month", color = "Season") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10), limits = c(32.5, 34.5)) +
  theme(axis.title.y = element_text(margin = margin(r = 15)),
    axis.title.x = element_blank()) +
  geom_point(data = filter(CTD_B, sample_16s == TRUE),
    aes(x = plot_date, y = `Salinity`),
    pch = 18, size = 3.5, alpha = .85)

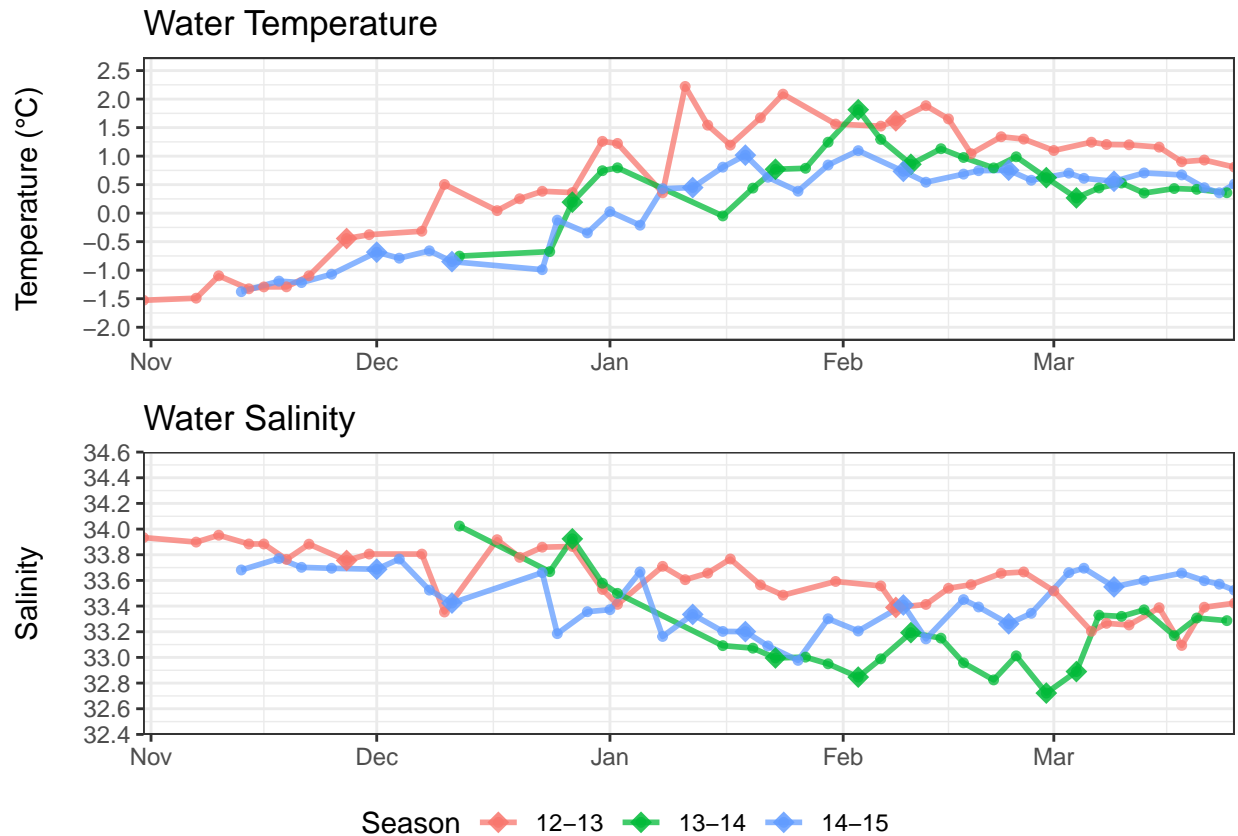
```

Water Temperature and Salinity from CTD

```

ggarrange(temp,
  sal,
  ncol = 1,
  common.legend = TRUE,
  legend = "bottom",
  align = "v")

```



Phosphate

```
phos <- ggplot(na.omit(inorganic_nutrients),
  aes(x = plot_date, y = `Phosphate (µmol/L)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31', '2001-03-25')),
    expand = c(0,0)) +
  labs(title = "Inorganic Nutrients - Phosphate", x = "Month", color = "Season") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.y = element_text(margin = margin(r = 15)),
    axis.title.x = element_blank()) +
  geom_point(data = filter(inorganic_nutrients, sample_16s == TRUE),
    aes(x = plot_date, y = `Phosphate (µmol/L)`),
    pch = 18, size = 3.5, alpha = .85)
```

Silicate

```
sil <- ggplot(inorganic_nutrients,
  aes(x = plot_date, y = `Silicate (µmol/L)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
```



```

        limits = as.Date(c('2000-10-31','2001-03-25')),
        expand = c(0,0)) +
labs(title = "Inorganic Nutrients - Silicate", x = "Month", color = "Season") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
theme(axis.title.y = element_text(margin = margin(r = 15)),
      axis.title.x = element_blank()) +
geom_point(data = filter(inorganic_nutrients, sample_16s == TRUE),
          aes(x = plot_date, y = `Silicate (μmol/L)`),
          pch = 18, size = 3.5, alpha = .85)

```

Nitrite and Nitrate

```

nn <- ggplot(inorganic_nutrients,
  aes(x = plot_date, y = `Nitrite and Nitrate (μmol/L)`,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b",
    limits = as.Date(c('2000-10-31','2001-03-25')),
    expand = c(0,0)) +
labs(title = "Inorganic Nutrients - Nitrite and Nitrate", x = "Month", color = "Season") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
theme(axis.title.y = element_text(margin = margin(r = 15)),
      axis.title.x = element_blank()) +
geom_point(data = filter(inorganic_nutrients, sample_16s == TRUE),
  aes(x = plot_date, y = `Nitrite and Nitrate (μmol/L)`),
  pch = 18, size = 3.5, alpha = .85)

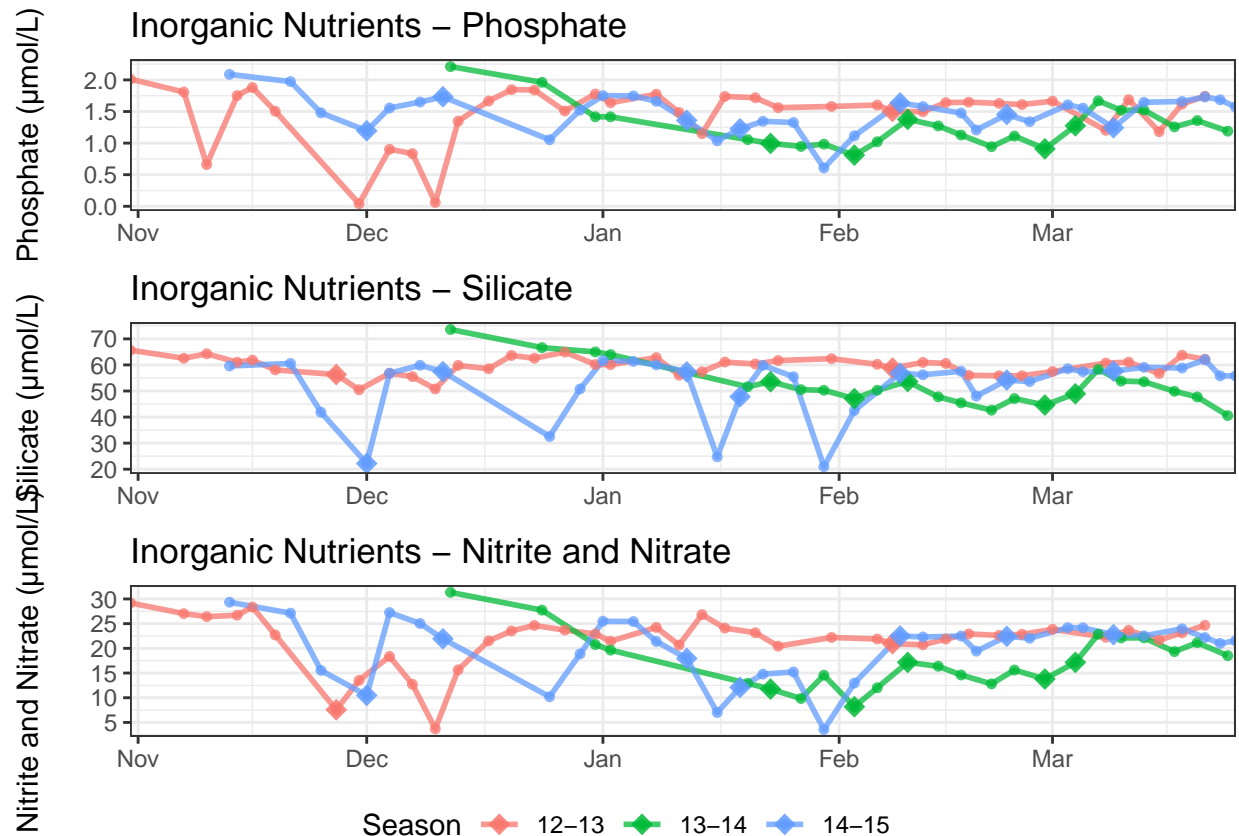
```

Inorganic Nutrients: Phosphate, Silicate, and Nitrite and Nitrate

```

ggarrange(phos,
  sil,
  nn,
  ncol = 1,
  common.legend = TRUE,
  legend = "bottom",
  align = "v")

```



Alpha Diversity: Within Sample Richness and Evenness

Observed Taxa:

Number of observed taxa at genus level (richness estimate)

Chao1 Index:

Predicted number of taxa in a sample by extrapolating out the number of rare organisms that may have been missed due to undersampling (richness estimate)

Shannon Diversity:

Estimator of species richness and species evenness: more weight on species richness

Measures the average degree of uncertainty in predicting where individual species chosen at random will belong

Inverse Simpson:

Estimator of species richness and species evenness: more weight on species evenness

Takes into account both species richness, and an evenness of abundance among the species present

Measures the probability that two individuals randomly selected from an area will belong to the same species

Alpha diversity descriptions from:

Kim BR, Shin J, Guevarra R, Lee JH, Kim DW, Seol KH, Lee JH, Kim HB, Isaacson R. Deciphering Diversity Indices for a Better Understanding of Microbial Communities. J Microbiol Biotechnol. 2017 Dec 28;27(12):2089-2093. doi: 10.4014/jmb.1709.09027. PMID: 29032640.

Prepare a data frame with the samples and their alpha diversity measures

```
# Calculate alpha diversity metrics
physeq_alpha_div <- microbiome::alpha(physeq_rarefy, index = "all")

# separate out the metadata from physeq
physeq_meta <- meta(physeq_rarefy)

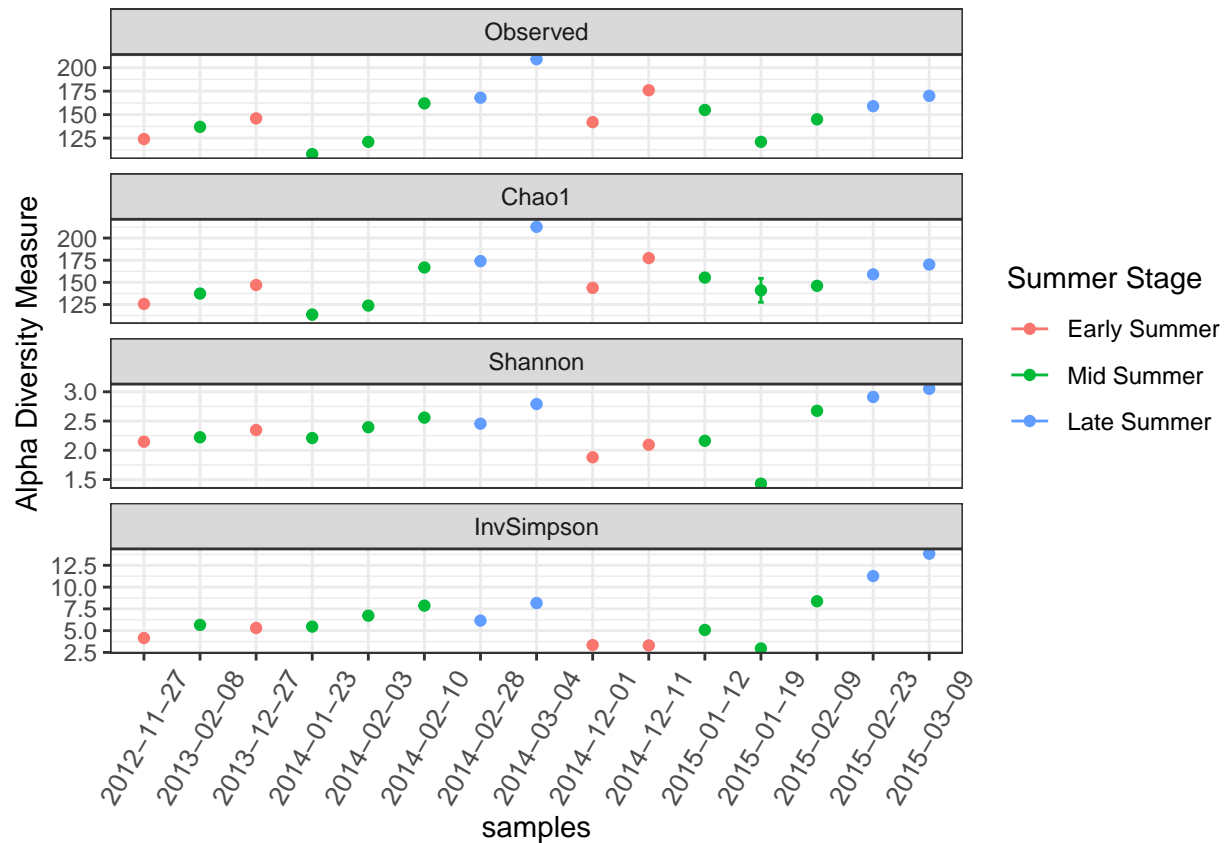
# add the sample names for merging
physeq_meta$sample_name <- rownames(physeq_meta)

# add the sample names to diversity results
physeq_alpha_div$sample_name <- rownames(physeq_alpha_div)

# merge
physeq_alpha_div <- merge(physeq_alpha_div, physeq_meta, by = "sample_name")
```

Plot observed taxa, Chao1, Shannon, and Inverse Simpson measures for each sample

```
plot_richness(physeq_rarefy, measures = c("Observed", "Chao1", "Shannon", "InvSimpson"),
             nrow = 4, color = "Summer_Stage") +
theme(axis.text.x = element_text(angle = 60, face = "plain", hjust = .5, vjust = .5, size = 10,)) +
scale_x_discrete(labels = lib_dates) +
labs(color = "Summer Stage")
```



The different alpha diversity measures shows (mostly) the same pattern across the samples.

Statistical Testing

Group Comparison:

Kruskal-Wallis rank sum test is a nonparametric alternative to ANOVA which checks the null hypothesis of whether all groups come from populations with the same median

Pairwise Comparison:

Wilcoxon rank sum test is a nonparametric alternative to two sample t-test which checks the null hypothesis of whether the two groups come from populations with the same median

Observed Taxa

```
# perform Wilcoxon test for boxplot
# p.adjust.method = "fdr" is the p-value adjustment by Benjamini & Hochberg (1995)
stat_test <- physeq_alpha_div %>%
  wilcox_test(observed ~ Summer_Stage, p.adjust.method = "fdr") %>%
  add_xy_position(x = "Summer_Stage")

# create boxplot for graphing later
alpha_obs <- ggboxplot(physeq_alpha_div,
  x = "Summer_Stage",
  y = "observed",
```

```

color = "Summer_Stage",
palette = c("#F8766D", "#00BA38", "#619CFF"),
legend = "none",
add = "jitter") +
stat_compare_means(label.y = 250, label.x = .8) + # add Kruskal-Wallis test to boxplot
stat_pvalue_manual(stat_test, label = "p.adj", step.increase = .075) + # add Wilcoxon test to boxplot
ylab("Observed Taxa") +
xlab("") +
labs(color = "Summer Stage")

```

Perform Kruskal-Wallis rank sum test on the observed taxa by the different summer stages

```
kruskal.test(observed ~ Summer_Stage, data = physeq_alpha_div)
```

```

##
## Kruskal-Wallis rank sum test
##
## data: observed by Summer_Stage
## Kruskal-Wallis chi-squared = 6.3757, df = 2, p-value = 0.04126

```

```

pairwise.wilcox.test(physeq_alpha_div$observed, physeq_alpha_div$Summer_Stage,
p.adjust.method = "fdr")

```

```

##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: physeq_alpha_div$observed and physeq_alpha_div$Summer_Stage
##
##           Early Summer Mid Summer
## Mid Summer 0.394          -
## Late Summer 0.300          0.054
##
## P value adjustment method: fdr

```

Kruskal-Wallis results indicate a difference in the median observed taxa between the 3 groups. Wilcox test shows strong evidence for difference in median observed taxa in Mid Summer and Late Summer.

Chao1 Index

```

# create boxplot for graphing later
stat_test <- physeq_alpha_div %>%
  wilcox_test(chao1 ~ Summer_Stage, p.adjust.method = "fdr") %>%
  add_xy_position(x = "Summer_Stage")

alpha_Chao1 <- ggboxplot(physeq_alpha_div,
  x = "Summer_Stage",
  y = "chao1",
  color = "Summer_Stage",
  palette = c("#F8766D", "#00BA38", "#619CFF"),
  legend = "none",

```

```

add = "jitter") +
stat_compare_means(label.y = 250, label.x = .8) +
stat_pvalue_manual(stat_test, label = "p.adj", step.increase = .075) +
ylab("Chao1 Index") +
xlab("") +
labs(color = "Summer Stage")

```

```

kruskal.test(chao1 ~ Summer_Stage, data = physeq_alpha_div)

```

```

##
## Kruskal-Wallis rank sum test
##
## data: chao1 by Summer_Stage
## Kruskal-Wallis chi-squared = 5.9286, df = 2, p-value = 0.0516

```

```

pairwise.wilcox.test(physeq_alpha_div$chao1, physeq_alpha_div$Summer_Stage,
p.adjust.method = "fdr")

```

```

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: physeq_alpha_div$chao1 and physeq_alpha_div$Summer_Stage
##
##           Early Summer Mid Summer
## Mid Summer 0.648      -
## Late Summer 0.300      0.036
##
## P value adjustment method: fdr

```

Kruskal-Wallis results indicate a difference in the median Chao1 index between the 3 groups. Wilcox test shows strong evidence for difference in median Chao1 index in Mid Summer and Late Summer.

Shannon Diversity

```

# create boxplot for graphing later
stat_test <- physeq_alpha_div %>%
  wilcox_test(diversity_shannon ~ Summer_Stage, p.adjust.method = "fdr") %>%
  add_xy_position(x = "Summer_Stage")

alpha_shannon <- ggboxplot(physeq_alpha_div,
  x = "Summer_Stage",
  y = "diversity_shannon",
  color = "Summer_Stage",
  palette = c("#F8766D", "#00BA38", "#619CFF"),
  legend = "none",
  add = "jitter") +
stat_compare_means(label.y = 4, label.x = .8) +
stat_pvalue_manual(stat_test, label = "p.adj", step.increase = .05) +
ylab("Shannon Diversity") +
xlab("") +
labs(color = "Summer Stage")

```

```
kruskal.test(diversity_shannon ~ Summer_Stage, data = physeq_alpha_div)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: diversity_shannon by Summer_Stage
## Kruskal-Wallis chi-squared = 7.9911, df = 2, p-value = 0.0184
```

```
pairwise.wilcox.test(physeq_alpha_div$diversity_shannon, physeq_alpha_div$Summer_Stage,
                     p.adjust.method = "fdr")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: physeq_alpha_div$diversity_shannon and physeq_alpha_div$Summer_Stage
##
##           Early Summer Mid Summer
## Mid Summer 0.230          -
## Late Summer 0.043          0.043
##
## P value adjustment method: fdr
```

Kruskal-Wallis on Simpson Diversity shows that there is a significant difference between the summer groups. Pairwise comparison with Wilcoxon Rank Sum Test after FDR correction shows both Early Summer - Late Summer and Mid Summer - Late Summer comparisons with with p-values under .05.

Inverse Simpson Diversity

```
# create boxplot for graphing later
stat_test <- physeq_alpha_div %>%
  wilcox_test(diversity_inverse_simpson ~ Summer_Stage, p.adjust.method = "fdr") %>%
  add_xy_position(x = "Summer_Stage")
```

```
alpha_simpson <- ggboxplot(physeq_alpha_div,
  x = "Summer_Stage",
  y = "diversity_inverse_simpson",
  color = "Summer_Stage",
  palette = c("#F8766D", "#00BA38", "#619CFF"),
  legend = "none",
  add = "jitter") +
  stat_compare_means(label.y = 20, label.x = .8) +
  stat_pvalue_manual(stat_test, label = "p.adj") +
  ylab("Inverse Simpson Diversity") +
  xlab("") +
  labs(color = "Summer Stage")
```

```
kruskal.test(diversity_inverse_simpson ~ Summer_Stage, data = physeq_alpha_div)
```

```
##
## Kruskal-Wallis rank sum test
```

```
##
## data:  diversity_inverse_simpson by Summer_Stage
## Kruskal-Wallis chi-squared = 7.6696, df = 2, p-value = 0.02161

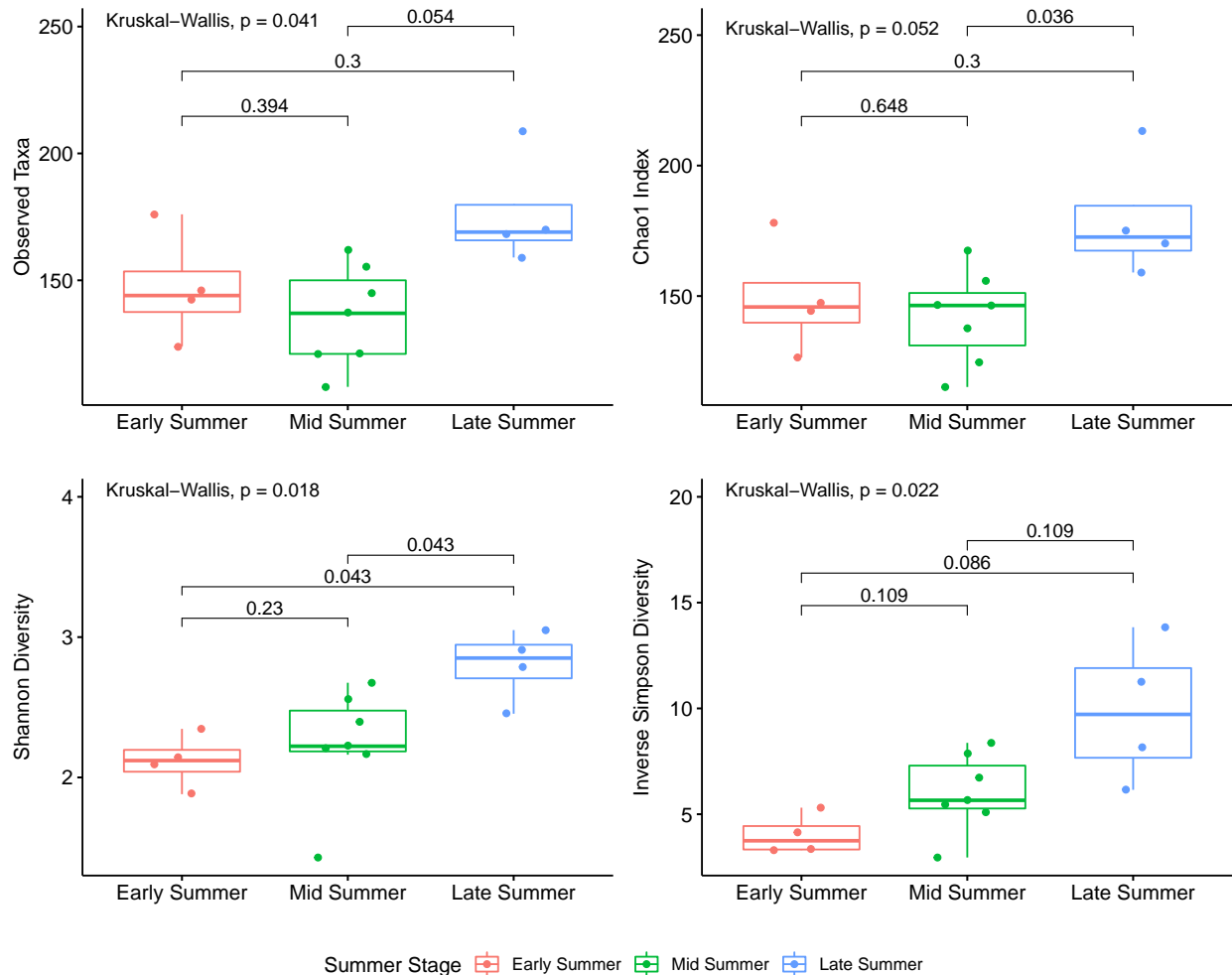
pairwise.wilcox.test(physeq_alpha_div$diversity_inverse_simpson, physeq_alpha_div$Summer_Stage,
                     p.adjust.method = "fdr")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  physeq_alpha_div$diversity_inverse_simpson and physeq_alpha_div$Summer_Stage
##
##           Early Summer Mid Summer
## Mid Summer  0.109      -
## Late Summer 0.086      0.109
##
## P value adjustment method: fdr
```

Kruskal-Wallis on Inverse Simpson Diversity shows that there is a significant difference between the summer groups. Pairwise comparison with Wilcoxon Rank Sum Test after FDR correction does not show a p-value under .05 for any pairs.

Graph all alpha diversity boxplots with their associated Kruskal-Wallis and Wilcoxon P-values

```
ggarrange(alpha_obs, alpha_Chao1, alpha_shannon, alpha_simpson,
          common.legend = TRUE, legend = "bottom", align = "hv")
```

The test on richness estimators, observed taxa and Chao1 index, shows strong evidence for difference in median richness among the 3 summer stages. Mid Summer to Late Summer have significant differences in median values.

Richness and evenness measures, Shannon diversity and Inverse Simpson diversity, also have significant differences in median diversity measures among the three summer stages. Only tests with Shannon diversity shows p-values under 0.05 for differences between Early Summer - Late Summer and Mid Summer - Late Summer.

None of the tests show significant differences between Early Summer and Mid Summer. Throughout the Antarctic summer, alpha diversity indices increase over time.

Taxa Composition

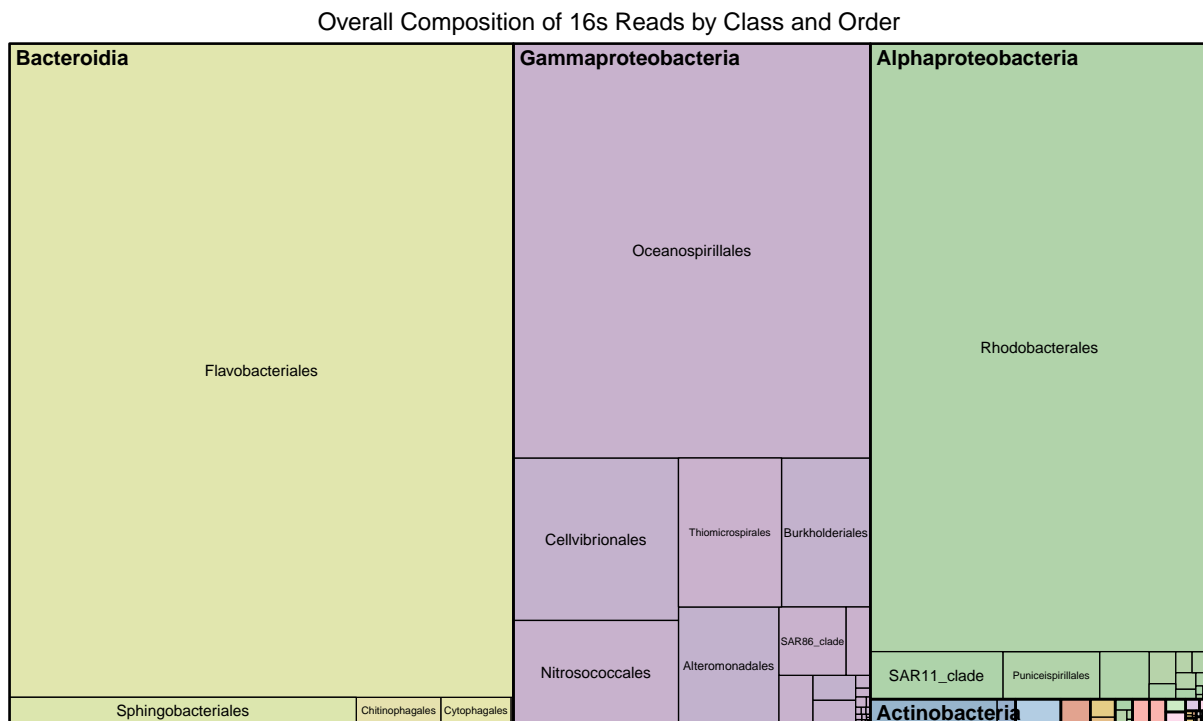
Tree Map

Create a tree map of the data at Class and Order levels

```
physeq@sam_data$merge <- "merge" # create a new column "merge" shared between all samples
physeq_order_other <- physeq %>%
  merge_samples("merge") %>% # merge all samples together
```

```
tax_glom(taxrank = "Order") %>% # agglomerate on Order level
transform_sample_counts(function(x) {100 * x/sum(x)}) %>% # transform to relative abundance
psmelt() # convert to long data frame
```

```
treemap(physeq_order_other,
  index = c("Class", "Order"),
  vSize = "Abundance",
  align.labels=list(c("left", "top"), c("center", "center")),
  fontsize.labels=c(12, 9),
  fontcolor.labels=c("black"),
  bg.labels = "transparent",
  lowerbound.cex.labels = .5,
  border.lwds = c(2, 1),
  palette = "Pastel1",
  title = "Overall Composition of 16s Reads by Class and Order"
)
```



Reads are largely dominated by a few number of classes (Bacteroidia, Gammaproteobacteria, and Alphaproteobacteria) and orders within the class (Flavobacteriales, Oceanospirillales, and Rhodobacterales)

Bar Charts

Relative Frequency by Phylum and Class

```
physeq_other <- physeq %>%
tax_glom(taxrank = "Class") %>% # agglomerate on Class level
transform_sample_counts(function(x) {100 * x/sum(x)}) %>% # transform to relative abundance
```

```

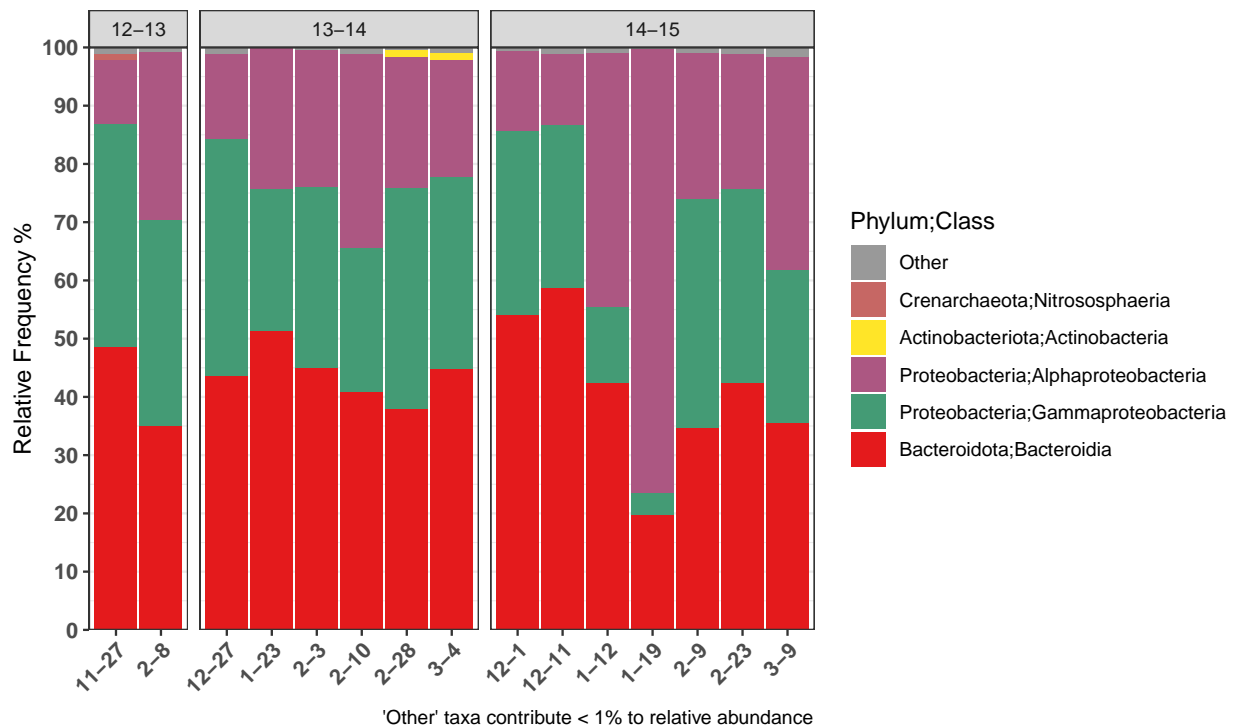
psmelt() %>% # convert from phyloseq object to a long data frame
unite(Taxa_Order, Phylum:Class, sep = ";") %>% # create Phylum;Class column
mutate(Taxa_Order = replace(Taxa_Order, Abundance < 1, "Other")) %>%
mutate(Taxa_Order = factor(Taxa_Order)) %>%
unite("month_day", month:day, sep = "-")

colourCount = length(unique(physeq_other$Taxa_Order)) # obtain number of colors needed
getPalette = colorRampPalette(brewer.pal(9, "Set1")) # create palette with Set1

ggplot(data=physeq_other,
       aes(x=reorder_within(month_day, as.Date(as.character(lib_date), "%m/%d/%Y"), lib_season), # order
           y=Abundance,
           fill=fct_reorder(Taxa_Order, Abundance, .desc = TRUE))) + # order taxa within bars by abundance
geom_bar(stat="identity",
         position = position_stack(reverse = TRUE), # order taxa in reverse
         width = 0.95) +
scale_fill_manual(values = getPalette(colourCount), # assign legend colors
                  guide = guide_legend(reverse = TRUE)) + # make legend match order of taxa in barplot
scale_y_continuous(breaks = scales::pretty_breaks(n = 10), # Y axis by 10% increments
                  expand = c(0,0)) + # remove whitespace in plot
scale_x_reordered() + # needed for reorder_within() function to order dates within each season
facet_grid(~lib_season, # draw facets by library seasons
           scales = "free_x",
           space = "free_x") +
theme(legend.direction="vertical",
      axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1, face = "bold", size = 10),
      axis.title.x=element_blank(),
      axis.text.y = element_text(face = "bold", size = 10)) +
labs(title = "Relative Frequency of Taxa by Phylum and Class for Each Season",
     caption = "'Other' taxa contribute < 1% to relative abundance",
     y = "Relative Frequency %",
     fill = "Phylum;Class")

```

Relative Frequency of Taxa by Phylum and Class for Each Season



Relative Frequency by Class and Order

```
physeq_other <- physeq %>%
  tax_glom(taxrank = "Order") %>%
  transform_sample_counts(function(x) {100 * x/sum(x)}) %>%
  psmelt() %>%
  unite(Taxa_Order, Class:Order, sep = ";") %>%
  mutate(Taxa_Order = replace(Taxa_Order, Abundance < 1, "Other")) %>%
  mutate(Taxa_Order = factor(Taxa_Order)) %>%
  unite("month_day", month:day, sep = "-")

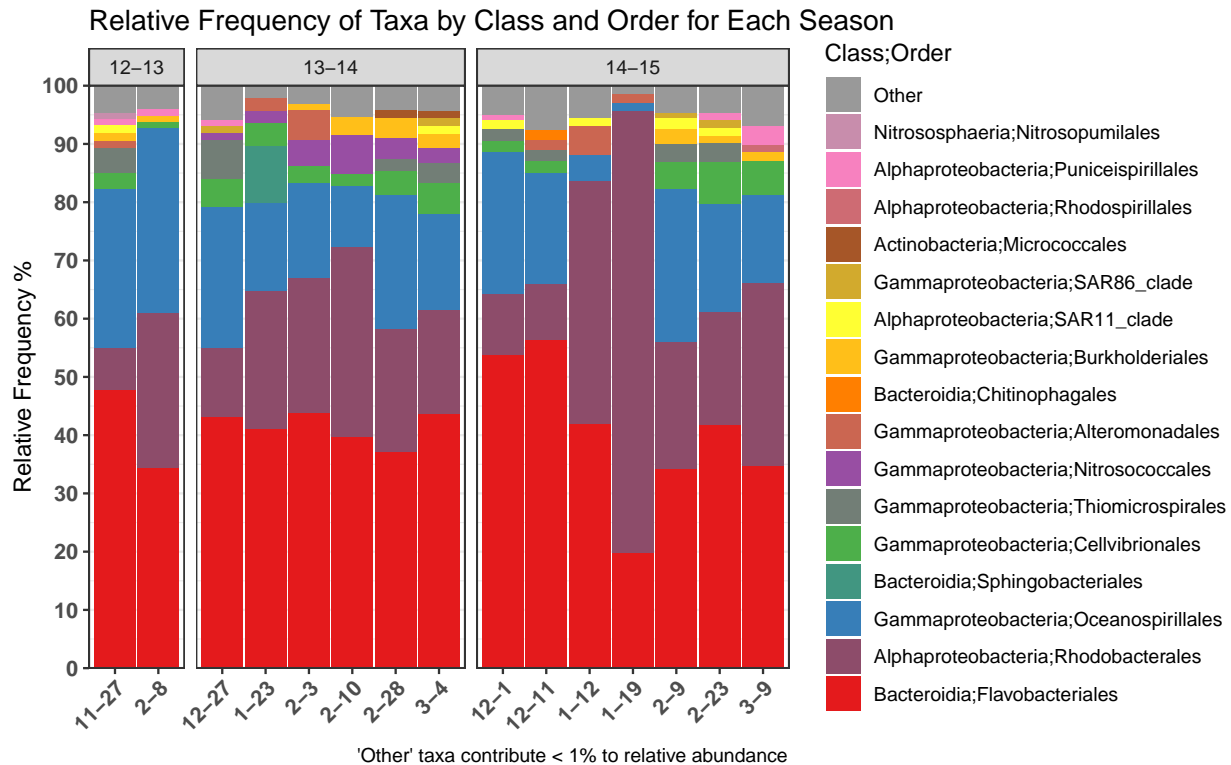
colourCount = length(unique(physeq_other$Taxa_Order))
getPalette = colorRampPalette(brewer.pal(9, "Set1"))

ggplot(data=physeq_other,
  aes(x=reorder_within(month_day, as.Date(as.character(lib_date), "%m/%d/%Y"), lib_season),
    y=Abundance,
    fill=fct_reorder(Taxa_Order, Abundance, .desc = TRUE))) +
  geom_bar(stat="identity",
    position = position_stack(reverse = TRUE),
    width = 0.95) +
  scale_fill_manual(values=getPalette(colourCount),
    guide = guide_legend(reverse=TRUE)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
    expand = c(0,0)) +
  scale_x_reordered() +
  facet_grid(~lib_season,
```

```

scales = "free_x",
space = "free_x") +
theme(legend.direction="vertical",
axis.text.x = element_text(angle = 45, hjust = 1, vjust=1, face = "bold", size = 10),
axis.title.x=element_blank(),
axis.text.y = element_text(face = "bold", size = 10)) +
labs(title = "Relative Frequency of Taxa by Class and Order for Each Season",
caption = "'Other' taxa contribute < 1% to relative abundance",
y = "Relative Frequency %",
fill = "Class;Order")

```



Relative Frequency by Family

```

physeq_other <- physeq %>%
  tax_glom(taxrank = "Family") %>%
  transform_sample_counts(function(x) {100 * x/sum(x)}) %>%
  psmelt() %>%
  #unite(Taxa_Order, Order:Family, sep = ";") %>%
  mutate(Taxa_Order = Family) %>%
  mutate(Taxa_Order = replace(Taxa_Order, Abundance < 1, "Other")) %>%
  mutate(Taxa_Order = factor(Taxa_Order)) %>%
  unite("month_day", month:day, sep = "-")

colourCount = length(unique(physeq_other$Taxa_Order))
getPalette = colorRampPalette(brewer.pal(9, "Set1"))

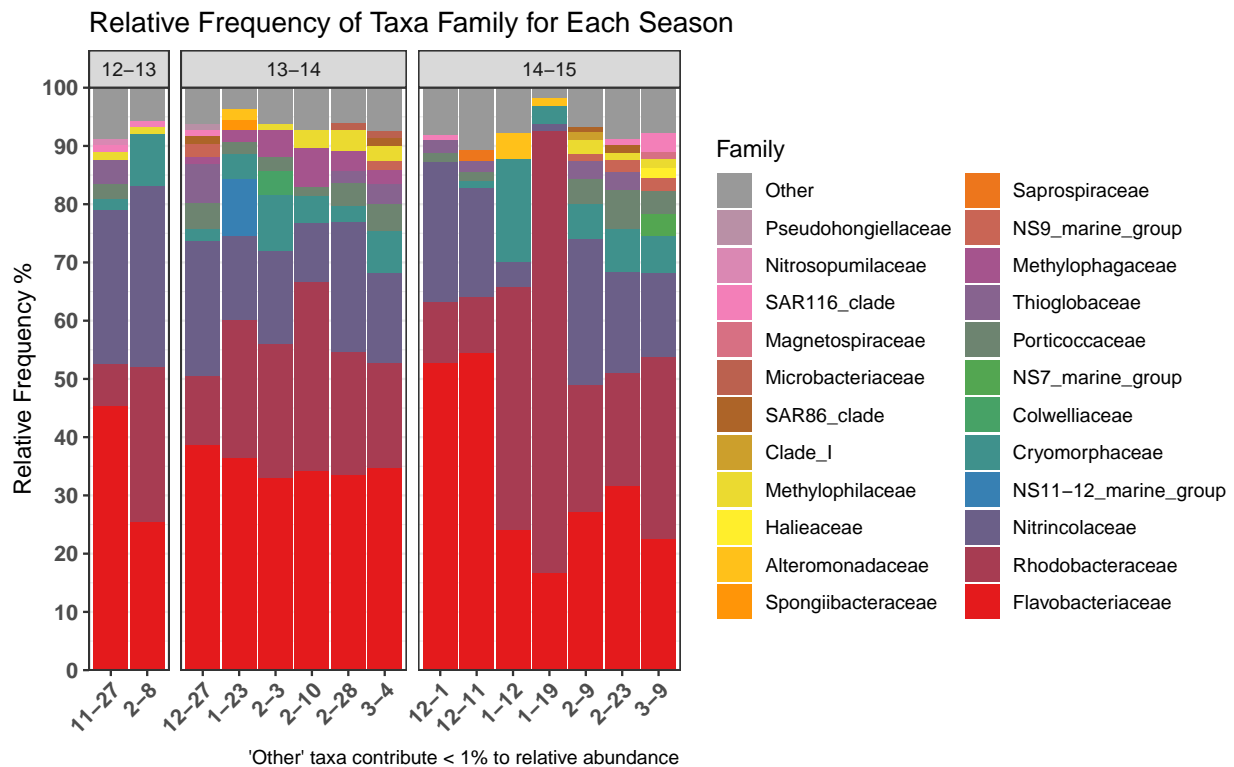
ggplot(data=physeq_other,

```

```

aes(x=reorder_within(month_day, as.Date(as.character(lib_date), "%m/%d/%Y"), lib_season),
    y=Abundance,
    fill=fct_reorder(Taxa_Order, Abundance, .desc = TRUE))) +
geom_bar(stat="identity",
    position = position_stack(reverse = TRUE),
    width = 0.95) +
scale_fill_manual(values=getPalette(colourCount),
    guide = guide_legend(reverse=TRUE)) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
    expand = c(0,0)) +
scale_x_reordered() +
facet_grid(~lib_season,
    scales = "free_x",
    space = "free_x") +
theme(legend.direction="vertical",
    axis.text.x = element_text(angle = 45, hjust = 1, vjust=1, face = "bold", size = 10),
    axis.title.x=element_blank(),
    axis.text.y = element_text(face = "bold", size = 10)) +
labs(title = "Relative Frequency of Taxa Family for Each Season",
    caption = "'Other' taxa contribute < 1% to relative abundance",
    y = "Relative Frequency %",
    fill = "Family")

```



Relative Frequency by Genus

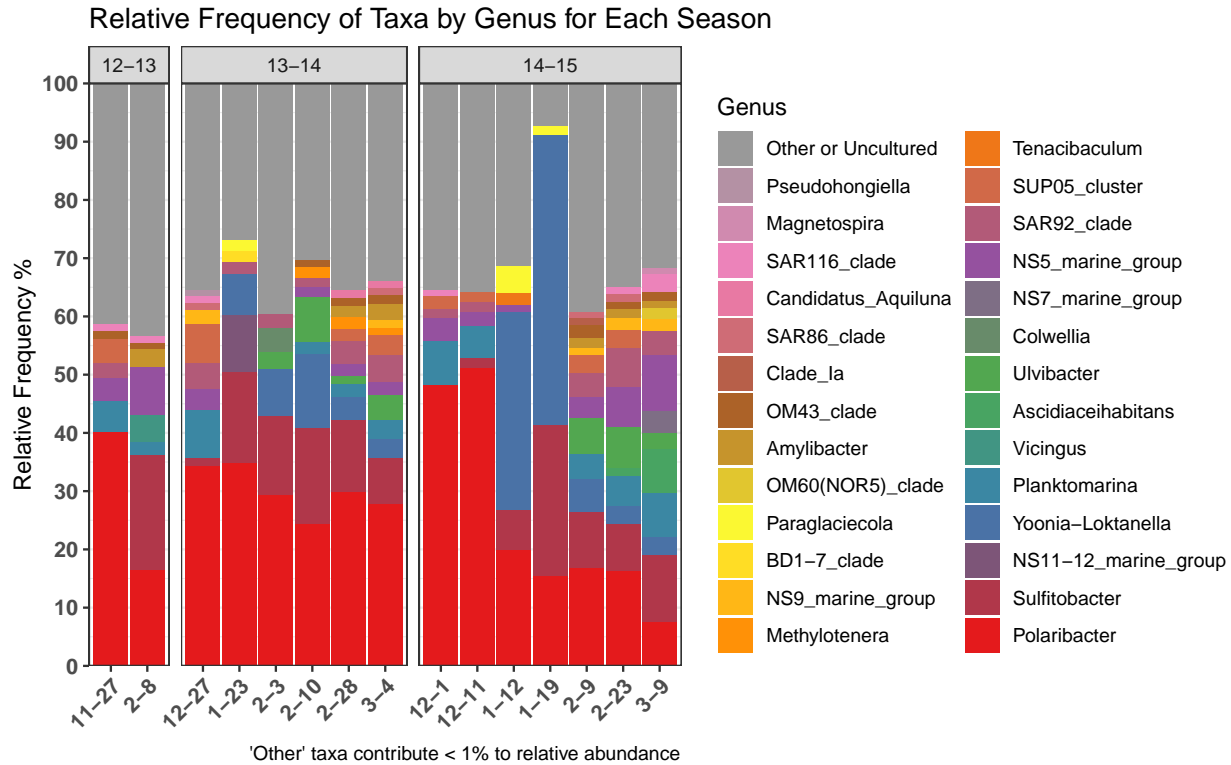
```

physeq_other <- physeq %>%
  transform_sample_counts(function(x) {100 * x/sum(x)}) %>%
  psmelt() %>%
  mutate(Taxa_Order = Genus) %>%
  mutate(Taxa_Order = replace(Taxa_Order, Abundance < 1, "Other or Uncultured")) %>%
  mutate(Taxa_Order = replace(Taxa_Order, Taxa_Order == "uncultured", "Other or Uncultured")) %>%
  mutate(Taxa_Order = factor(Taxa_Order)) %>%
  unite("month_day", month:day, sep = "-")

colourCount = length(unique(physeq_other$Taxa_Order))
getPalette = colorRampPalette(brewer.pal(9, "Set1"))

ggplot(data=physeq_other,
  aes(x=reorder_within(month_day, as.Date(as.character(lib_date), "%m/%d/%Y"), lib_season),
    y=Abundance,
    fill=fct_reorder(Taxa_Order, Abundance, .desc = TRUE))) +
geom_bar(stat="identity",
  position = position_stack(reverse = TRUE),
  width = 0.95) +
scale_fill_manual(values=getPalette(colourCount),
  guide = guide_legend(reverse=TRUE)) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 10),
  expand = c(0,0)) +
scale_x_reordered() +
facet_grid(~lib_season,
  scales = "free_x",
  space = "free_x") +
theme(legend.direction="vertical",
  axis.text.x = element_text(angle = 45, hjust = 1, vjust=1, face = "bold", size = 10),
  axis.title.x=element_blank(),
  axis.text.y = element_text(face = "bold", size = 10)) +
labs(title = "Relative Frequency of Taxa by Genus for Each Season",
  caption = "'Other' taxa contribute < 1% to relative abundance",
  y = "Relative Frequency %",
  fill = "Genus")

```



Beta Diversity, Clustering, and Ordination

Beta diversity is the between-sample diversity. Several beta diversity metrics exist but not all measures incorporate abundance information. Since the samples are dominated by only a few taxa, presence/absence measures like Jaccard or Unweighted Unifrac may not be appropriate.

Weighted Unifrac can be used to calculate the phylogenetically-weighted distance between samples. It accounts for the relative abundance of taxa shared between samples and utilizes presence/absence data.

Hierarchical Clustering

Create Weighted Unifrac distance matrix

```
physeq_wunifrac <- distance(physeq_rarefy, method = "wunifrac" )
```

Perform hierarchical clustering

```
par(mar = c(3,4,3,6)) # set margins of plot

physeq_hclust <- hclust(physeq_wunifrac, method="complete") # hierarchical cluster with complete linkage
hclust_dend <- as.dendrogram(physeq_hclust) # convert to dendrogram
mycols <- c(`Early Summer` = "#F8766D", `Mid Summer` = "#00BA38", `Late Summer` = "#619CFF") # ggplot2 default
# add dates as labels and color by library season
labels_colors(hclust_dend) <- mycols[physeq@sam_data$Summer_Stage][order.dendrogram(hclust_dend)]
```

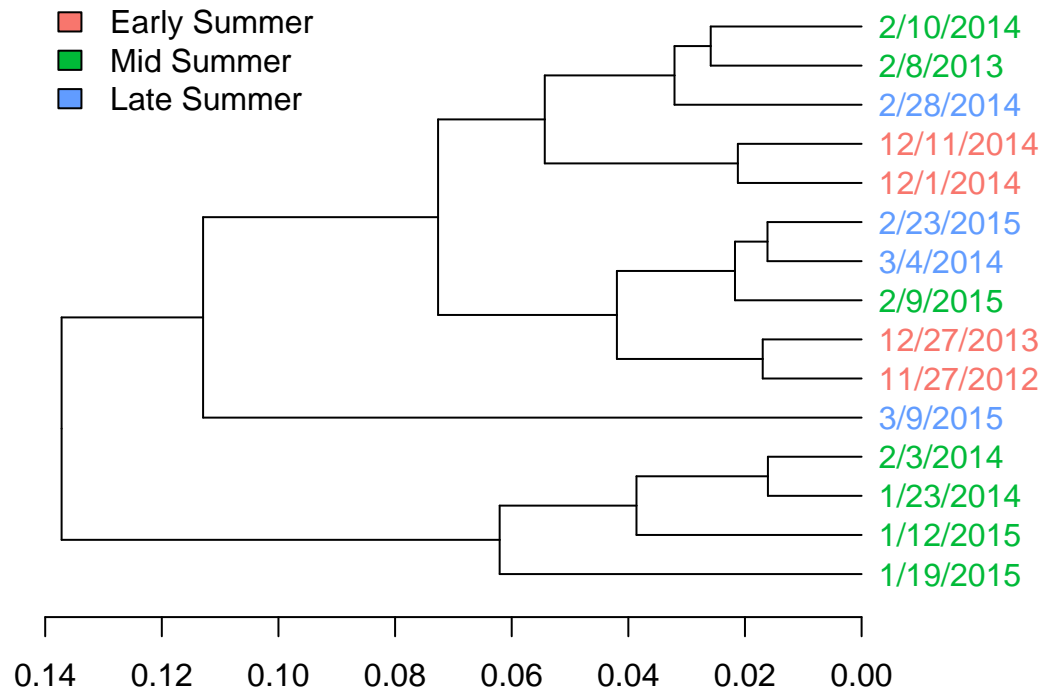


```

labels(hclust_dend) <- physeq@sam_data$lib_date[order.dendrogram(hclust_dend)]
plot(hclust_dend, main = "Clustering on Weighted Unifrac Distance with Complete Linkage",
     horiz = TRUE)
legend("topleft", legend = levels(physeq@sam_data$Summer_Stage), fill = mycols,
     box.lwd = 0, box.col = "white", bg = "white")

```

Clustering on Weighted Unifrac Distance with Complete Linkage



Several mid summer samples cluster together pretty closely. 3/9/2015 is the most distant to the rest of the samples. The rest of the samples create fairly intermixed clusters. Early summer samples do not cluster with late summer samples at low distances.

Ordination

Helper functions modified from:

<https://jacobrprice.github.io/2017/08/26/phyloseq-to-vegan-and-back.html>

<http://joey711.github.io/phyloseq-demo/phyloseq-demo.html>

```

# convert the all sample_data() within a phyloseq object to a vegan compatible data object
physeq_to_vegan_sd <- function(physeq_sd) {
  sd <- data.frame(sample_data(physeq_sd))
  return(sd)
}

```

```
# convert the otu_table() within a phyloseq object to a vegan compatible data object
physeq_to_vegan_otu <- function(physeq_otu) {
  OTU <- otu_table(physeq_otu)
  if (taxa_are_rows(OTU)) {
    OTU <- t(OTU)
  }
  return(as(OTU, "matrix"))
}
```

Perform Hellinger transformation on rarefied data. Euclidean distance is required for linear methods like principal component analysis (PCA) and redundancy analysis (RDA).

```
physeq_hel <- transform(physeq_rarefy, transform = "hellinger")
physeq_hel_distance <- distance(physeq_hel, method = "euclidean")
```

Create data structures for analysis in vegan

```
# rarefied data to vegan
veg_physeq_rarefy_sd <- physeq_to_vegan_sd(physeq_rarefy)[,6:14]
veg_physeq_rarefy_otu <- physeq_to_vegan_otu(physeq_rarefy)

# Hellinger transformed data to vegan
veg_physeq_hel_sd <- physeq_to_vegan_sd(physeq_hel)[,6:14]
veg_physeq_hel_otu <- physeq_to_vegan_otu(physeq_hel)
```

Redundancy analysis (RDA)

Perform PCA ordination (same as unconstrained RDA)

```
RDA_hel_uncon <- ordinate(physeq_hel, # perform PCA/RDA ordination
                          method = "RDA")

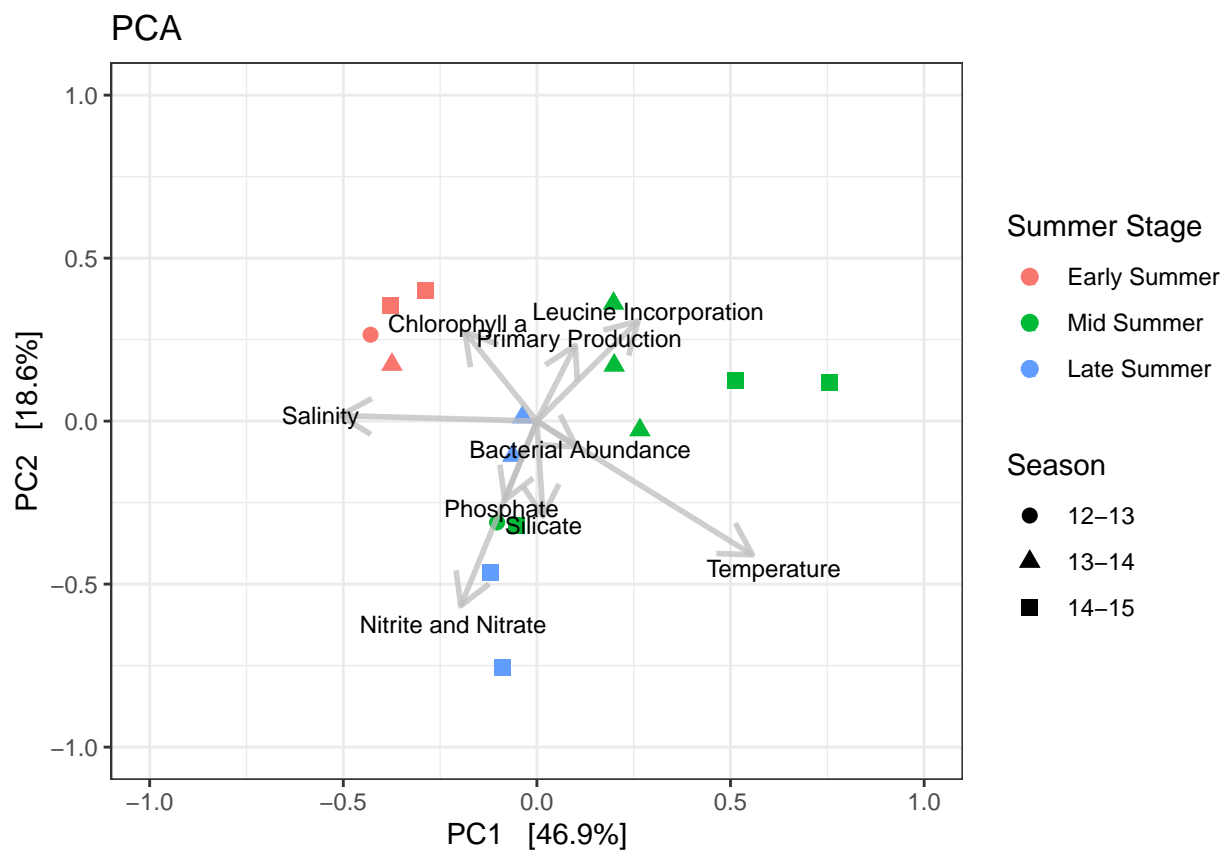
RDA_hel_fit <- gg_envfit(RDA_hel_uncon, veg_physeq_hel_sd, # fit env variables to the ordination
                       alpha = 1,
                       groups = physeq@sam_data$Summer_Stage,
                       scaling = 2,
                       perm = 100000, plot = FALSE)

p = plot_ordination(
  physeq_hel,
  ordination = RDA_hel_uncon,
  type = "samples",
  color = "Summer_Stage",
  shape = "lib_season") +
  ggtitle("PCA") +
  geom_point(size = 2.5) +
  #geom_text_repel(aes(label = as.character(lib_dates)), size = 3) + # too cluttered with all sam
  labs(color = "Summer Stage", shape = "Season") +
  scale_x_continuous(limits = c(-1,1)) +
  scale_y_continuous(limits = c(-1,1))
```

```

# Add the environmental variables as arrows
arrowmat = RDA_hel_fit$df_arrows[,c("x","y")]
# Add labels, make a data.frame
rownames(arrowmat) <- c("Bacterial Abundance", "Leucine Incorporation", "Chlorophyll a", "Phosphate", "Silicate", "Nitrite and Nitrate", "Temperature", "Salinity")
arrowdf <- data.frame(labels = rownames(arrowmat), arrowmat)
# Define the arrow aesthetic mapping
arrow_map = aes(xend = x, yend = y, x = 0, y = 0, shape = NULL, color = NULL)
label_map = aes(x = x*1.1, y = y*1.1, shape = NULL, color = NULL, label = labels)
arrowhead = arrow(length = unit(0.05, "npc"))
p1 = p + geom_segment(arrow_map, size = 1, data = arrowdf, color = "gray", arrow = arrowhead, alpha = 0.5) +
  geom_text(label_map, size = 3, data = arrowdf) #+
  #stat_ellipse(aes(group = physeq_hel$sam_data$Summer_Stage), linetype = 2, level = .95)
p1

```



Determine what environmental variables best fit with PCA axis 1 and 2

```

fit_RDA <- envfit(RDA_hel_uncon, # ordination object
  veg_physeq_hel_sd, # environmental variables
  permutations = 10000)
fit_RDA

```

```

##
## ***VECTORS
##
##          PC1      PC2      r2  Pr(>r)

```

```
## Bacterial.Abandance      0.78976 -0.61342 0.0170 0.90051
## Leucine.Incorporation    0.64828  0.76140 0.1662 0.31287
## Chlorophyll.a           -0.56154  0.82745 0.1122 0.50585
## Phosphate                -0.32364 -0.94618 0.0673 0.65813
## Silicate                 0.05354 -0.99857 0.0873 0.59304
## Nitrite.and.Nitrate      -0.32803 -0.94467 0.3680 0.06909 .
## Temperature              0.80503 -0.59323 0.4888 0.01990 *
## Salinity                 -0.99944  0.03343 0.2643 0.16308
## Primary.Production       0.39319  0.91946 0.0653 0.66963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 10000
```

Temperature and Nitrite and Nitrate have the highest correlation and lowest P values. Only temperature has a p-value under 0.05.

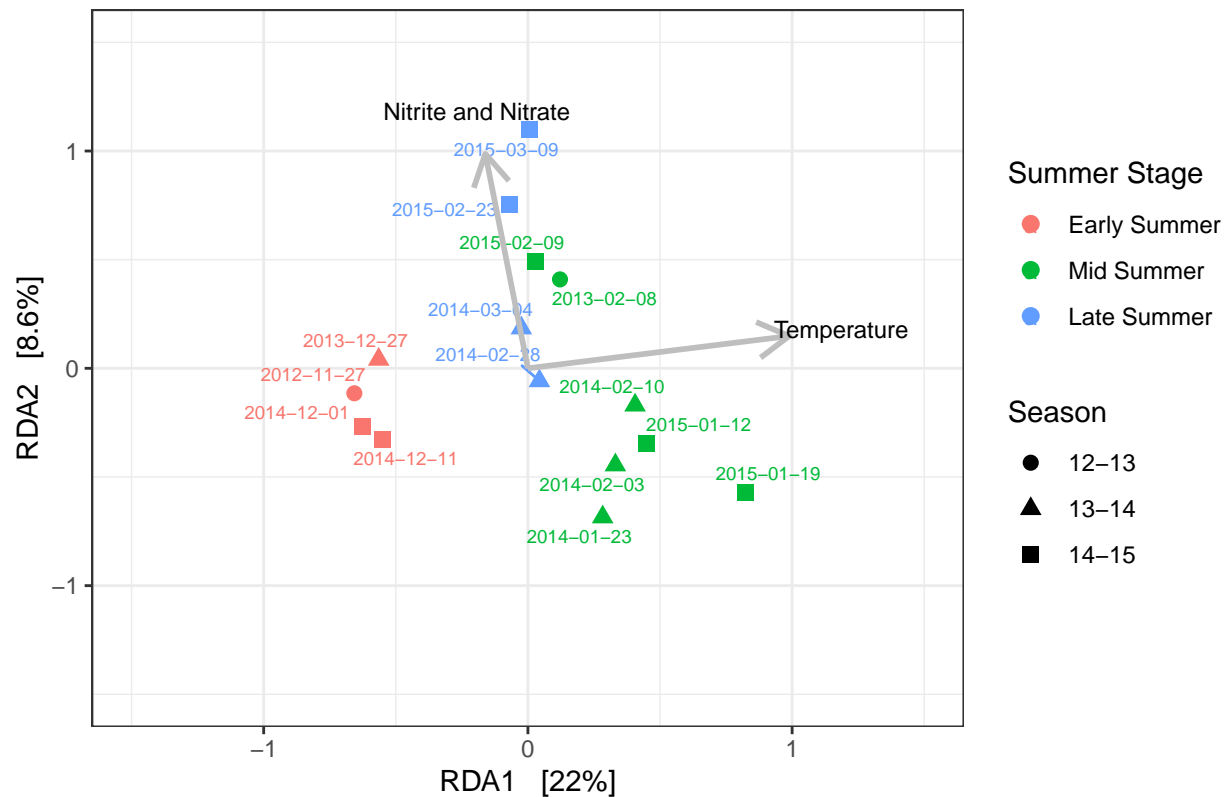
Perform RDA constrained on Temperature and Nitrite and Nitrate

```
# modified from https://github.com/joey711/phyloseq/issues/274#issuecomment-30553161
# at least 2 variables required in formula
RDA_hel <- ordinate(physeq_hel,
                    method = "RDA",
                    formula = ~ Temperature + Nitrite.and.Nitrate)

p = plot_ordination(
  physeq_hel,
  ordination = RDA_hel,
  type = "samples",
  color = "Summer_Stage",
  shape = "lib_season") +
  ggtitle("RDA - Constrained on Temperature and Nitrite and Nitrate") +
  geom_point(size = 2.5) +
  geom_text_repel(aes(label = as.character(lib_dates)), size = 2.5) +
  labs(color = "Summer Stage", shape = "Season") +
  scale_x_continuous(limits = c(-1.5,1.5)) +
  scale_y_continuous(limits = c(-1.5,1.5))

# Add the environmental variables as arrows
arrowmat = vegan::scores(RDA_hel, display = "bp")
# Add labels, make a data.frame
row.names(arrowmat) <- c("Temperature", "Nitrite and Nitrate")
arrowdf <- data.frame(labels = rownames(arrowmat), arrowmat)
# Define the arrow aesthetic mapping
arrow_map = aes(xend = RDA1, yend = RDA2, x = 0, y = 0, shape = NULL, color = NULL)
label_map = aes(x = RDA1*1.2, y = RDA2*1.2, shape = NULL, color = NULL, label = labels)
arrowhead = arrow(length = unit(0.05, "npc"))
p1 = p + geom_segment(arrow_map, size = 1, data = arrowdf, color = "gray", arrow = arrowhead) +
  geom_text(label_map, size = 3, data = arrowdf) #+
  #stat_ellipse(aes(group = physeq_hel@sam_data$Summer_Stage), linetype = 2, level = .95)
p1
```

RDA – Constrained on Temperature and Nitrite and Nitrate



RDA_hel

```
## Call: rda(formula = OTU ~ Temperature + Nitrite.and.Nitrate, data =
## data)
##
##              Inertia Proportion Rank
## Total          0.17652    1.00000
## Constrained    0.05388    0.30524    2
## Unconstrained  0.12264    0.69476   12
## Inertia is variance
##
## Eigenvalues for constrained axes:
##   RDA1   RDA2
## 0.03877 0.01511
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9   PC10
## 0.05874 0.01841 0.01445 0.00939 0.00728 0.00478 0.00375 0.00215 0.00155 0.00104
##   PC11   PC12
## 0.00068 0.00041
```

About 30% of the variance can be explained by Temperature and Nitrite and Nitrate

Perform RDA constrained on Temperature, Nitrite and Nitrate, and Salinity

```

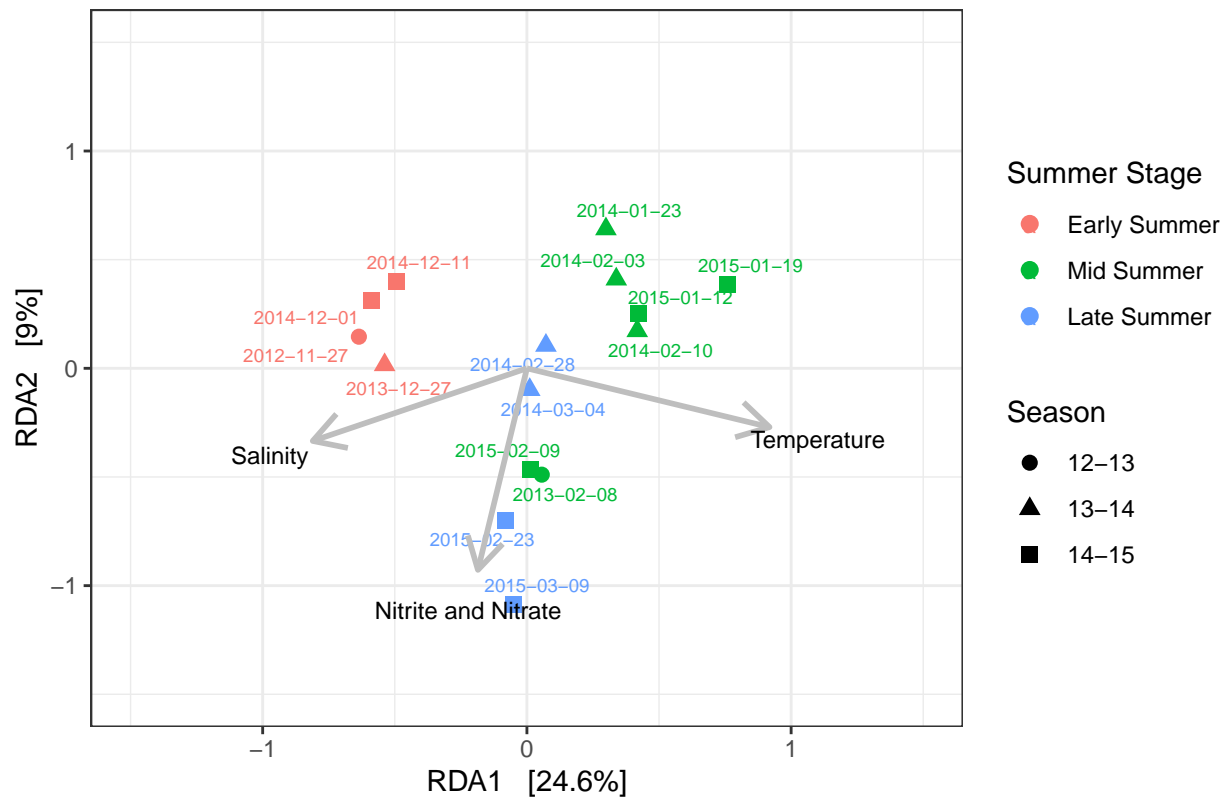
RDA_hel <- ordinate(physeq_hel,
                    method = "RDA",
                    formula = ~ Temperature + Nitrite.and.Nitrate + Salinity)

p = plot_ordination(
  physeq_hel,
  ordination = RDA_hel,
  type = "samples",
  color = "Summer_Stage",
  shape = "lib_season") +
  ggtitle("RDA - Constrained on Temperature, Nitrite and Nitrate, and Salinity") +
  geom_point(size = 2.5) +
  geom_text_repel(aes(label = as.character(lib_dates)), size = 2.5) +
  labs(color = "Summer Stage", shape = "Season") +
  scale_x_continuous(limits = c(-1.5,1.5)) +
  scale_y_continuous(limits = c(-1.5,1.5))

# Add the environmental variables as arrows
arrowmat = vegan::scores(RDA_hel, display = "bp")
# Add labels, make a data.frame
row.names(arrowmat) <- c("Temperature", "Nitrite and Nitrate", "Salinity")
arrowdf <- data.frame(labels = rownames(arrowmat), arrowmat)
# Define the arrow aesthetic mapping
arrow_map = aes(xend = RDA1, yend = RDA2, x = 0, y = 0, shape = NULL, color = NULL)
label_map = aes(x = RDA1*1.2, y = RDA2*1.2, shape = NULL, color = NULL, label = labels)
arrowhead = arrow(length = unit(0.05, "npc"))
p1 = p + geom_segment(arrow_map, size = 1, data = arrowdf, color = "gray", arrow = arrowhead) +
  geom_text(label_map, size = 3, data = arrowdf) #+
  #stat_ellipse(aes(group = physeq_hel$sam_data$Summer_Stage), linetype = 2, level = .95)
p1

```

RDA – Constrained on Temperature, Nitrite and Nitrate, and Salinity



RDA_hel

```
## Call: rda(formula = OTU ~ Temperature + Nitrite.and.Nitrate + Salinity,
## data = data)
##
##              Inertia Proportion Rank
## Total          0.17652      1.00000
## Constrained    0.06511      0.36885      3
## Unconstrained  0.11141      0.63115     11
## Inertia is variance
##
## Eigenvalues for constrained axes:
##   RDA1   RDA2   RDA3
## 0.04340 0.01583 0.00588
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9   PC10
## 0.05690 0.01765 0.01119 0.00778 0.00649 0.00376 0.00337 0.00209 0.00106 0.00068
##   PC11
## 0.00043
```

About 36% of the variance can be explained by Temperature, Nitrite and Nitrate, and Salinity

Ordinate with NMDS and plot significant environmental variables

```

nmds_ord <- ordinate(physeq_rarefy,
                     method = "NMDS",
                     distance = "wunifrac")

```

```

## Run 0 stress 0.03360338
## Run 1 stress 0.03360357
## ... Procrustes: rmse 0.000163893  max resid 0.0004370262
## ... Similar to previous best
## Run 2 stress 0.0336036
## ... Procrustes: rmse 0.0001973234  max resid 0.0005255106
## ... Similar to previous best
## Run 3 stress 0.0336036
## ... Procrustes: rmse 0.0003645399  max resid 0.0009714519
## ... Similar to previous best
## Run 4 stress 0.2085195
## Run 5 stress 0.03360356
## ... Procrustes: rmse 0.0002817317  max resid 0.0007506994
## ... Similar to previous best
## Run 6 stress 0.0336035
## ... Procrustes: rmse 0.0002966991  max resid 0.0007910278
## ... Similar to previous best
## Run 7 stress 0.03360336
## ... New best solution
## ... Procrustes: rmse 7.586772e-05  max resid 0.0001960527
## ... Similar to previous best
## Run 8 stress 0.03360338
## ... Procrustes: rmse 9.505284e-05  max resid 0.0002592863
## ... Similar to previous best
## Run 9 stress 0.03360343
## ... Procrustes: rmse 0.0001439295  max resid 0.0003777741
## ... Similar to previous best
## Run 10 stress 0.03360349
## ... Procrustes: rmse 0.0002117249  max resid 0.0005700761
## ... Similar to previous best
## Run 11 stress 0.03360338
## ... Procrustes: rmse 8.408475e-05  max resid 0.0002181166
## ... Similar to previous best
## Run 12 stress 0.03360337
## ... Procrustes: rmse 8.334017e-05  max resid 0.0002279406
## ... Similar to previous best
## Run 13 stress 0.03360355
## ... Procrustes: rmse 0.0002572385  max resid 0.0006912111
## ... Similar to previous best
## Run 14 stress 0.0336036
## ... Procrustes: rmse 0.0002725238  max resid 0.0007210382
## ... Similar to previous best
## Run 15 stress 0.03360346
## ... Procrustes: rmse 0.0001933655  max resid 0.00052088
## ... Similar to previous best
## Run 16 stress 0.03360339
## ... Procrustes: rmse 0.0001178539  max resid 0.000319796
## ... Similar to previous best
## Run 17 stress 0.03360343

```



```
## ... Procrustes: rmse 0.0001399476  max resid 0.0003659977
## ... Similar to previous best
## Run 18 stress 0.03360341
## ... Procrustes: rmse 0.0001258135  max resid 0.0003293863
## ... Similar to previous best
## Run 19 stress 0.0336035
## ... Procrustes: rmse 0.0002190795  max resid 0.0005893779
## ... Similar to previous best
## Run 20 stress 0.03360356
## ... Procrustes: rmse 0.0002645846  max resid 0.0007109157
## ... Similar to previous best
## *** Solution reached
```

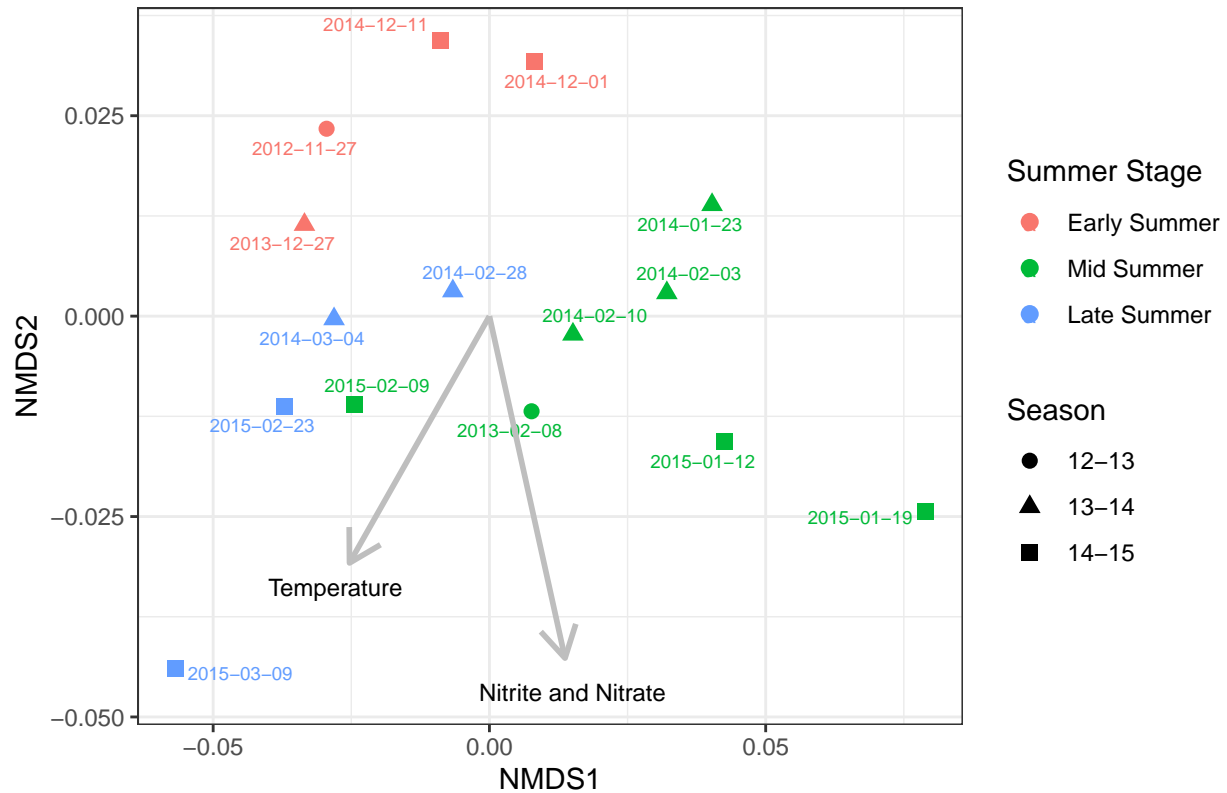
```
nmnds_fit <- gg_envfit(nmnds_ord, veg_physeq_rarefy_sd,
  alpha = .05, # minimum P-value for environmental var
  groups = physeq@sam_data$Summer_Stage,
  scaling = 2,
  perm = 100000, plot = FALSE)

names(nmnds_fit$df_arrows)[names(nmnds_fit$df_arrows) == "x"] <- "NMDS1"
names(nmnds_fit$df_arrows)[names(nmnds_fit$df_arrows) == "y"] <- "NMDS2"

p = plot_ordination(
  physeq_rarefy,
  ordination = nmnds_ord,
  type = "samples",
  color = "Summer_Stage",
  shape = "lib_season") +
  ggtitle("NMDS on Weighed Unifrac Distance Fit with Environmental Variables") +
  geom_point(size = 2.5) +
  geom_text_repel(aes(label = as.character(lib_dates)), size = 2.5) +
  labs(color = "Summer Stage", shape = "Season")

# Add the environmental variables as arrows
arrowmat = nmnds_fit$df_arrows[,c("NMDS1", "NMDS2")]
# Add labels, make a data.frame
row.names(arrowmat) <- c("Temperature", "Nitrite and Nitrate")
arrowdf <- data.frame(labels = rownames(arrowmat), arrowmat)
# Define the arrow aesthetic mapping
arrow_map = aes(xend = NMDS1, yend = NMDS2, x = 0, y = 0, shape = NULL, color = NULL)
label_map = aes(x = NMDS1*1.1, y = NMDS2*1.1, shape = NULL, color = NULL, label = labels)
arrowhead = arrow(length = unit(0.05, "npc"))
p1 = p + geom_segment(arrow_map, size = 1, data = arrowdf, color = "gray", arrow = arrowhead) +
  geom_text(label_map, size = 3, data = arrowdf) # +
  #stat_ellipse(aes(group = physeq_hel@sam_data$Summer_Stage), linetype = 2, level = .95)
p1
```

NMDS on Weighed Unifrac Distance Fit with Environmental Variables



```
fit_NMDS <- envfit(nmds_ord, veg_physeq_rarefy_sd, permutations = 100000)
fit_NMDS
```

```
##
## ***VECTORS
##
##          NMDS1    NMDS2    r2 Pr(>r)
## Bacterial.Abandance -0.68675 -0.72690 0.0051 0.9698
## Leucine.Incorporation 0.78304 0.62197 0.1981 0.2588
## Chlorophyll.a -0.03274 0.99946 0.0654 0.6407
## Phosphate -0.67517 -0.73766 0.0524 0.7257
## Silicate -0.44712 -0.89447 0.1689 0.3142
## Nitrite.and.Nitrate -0.63525 -0.77231 0.3898 0.0499 *
## Temperature 0.30538 -0.95223 0.4923 0.0167 *
## Salinity -0.79656 0.60456 0.1635 0.3451
## Primary.Production 0.48017 0.87718 0.0833 0.5771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 1e+05
```

NMDS results are similar to the RDA results. Stress value of ~ 0.0336 indicates a high goodness of fit, or that the NMDS is a good representation, in reduced dimensions, of the original distance matrix.

PERMANOVA test on categorical/group variables Summer Stage and Library Season

```
adonis2(physeq_hel_distance ~ Summer_Stage,
        data = physeq_to_vegan_sd(physeq_hel),
        permutations = 10000)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## adonis2(formula = physeq_hel_distance ~ Summer_Stage, data = physeq_to_vegan_sd(physeq_hel), permuta
##           Df SumOfSqs      R2      F Pr(>F)
## Summer_Stage  2    1.0639 0.43051 4.5358 3e-04 ***
## Residual     12    1.4074 0.56949
## Total        14    2.4713 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Significant p-value for Summer Stage groups indicates that either the centroid and/or the dispersion of between the groups is significantly different.

```
adonis2(physeq_hel_distance ~ lib_season,
        data = physeq_to_vegan_sd(physeq_hel),
        permutations = 10000)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## adonis2(formula = physeq_hel_distance ~ lib_season, data = physeq_to_vegan_sd(physeq_hel), permutati
##           Df SumOfSqs      R2      F Pr(>F)
## lib_season  2    0.43015 0.17406 1.2644 0.2476
## Residual    12    2.04114 0.82594
## Total       14    2.47128 1.00000
```

Library seasons does not have a significant p-value in the PERMANOVA test.

Test for heteroscedasticity (PERMDSIP = Permutation test + Multivariate homogeneity of groups dispersions (variances))

```
permutest(betadisper(physeq_hel_distance, physeq@sam_data$Summer_Stage), permutations = 10000)

##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 10000
##
## Response: Distances
```

```
##           Df Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups      2 0.12956 0.064781 5.1349 10000 0.0264 *
## Residuals 12 0.15139 0.012616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(betadisper(physeq_hel_distance, physeq@sam_data$lib_season), permutations = 10000)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 10000
##
## Response: Distances
##           Df Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups      2 0.071072 0.035536 1.5294 10000 0.2694
## Residuals 12 0.278827 0.023236
```

Summer Stage fails assumption of homoscedasticity for adonis/PERMANOVA test. Library Season does not fail the assumption of homoscedasticity for adonis/PERMANOVA test

Both PERMANOVA and PERMDISP tests are significant with Summer Stage and the groups are unbalanced. Therefore, you can't tell if the PERMANOVA is significant due to difference in a group's centroid or dispersion.

There is no strong evidence for Library Season groups to have different centroids or dispersions.

Core Microbiome

Core taxa must be detected and present in all samples of a given group.

Venn Diagrams

Modified from https://microbiome.github.io/tutorials/core_venn.html

```
physeq_comp <- transform(physeq, "compositional") # transform to compositional, instead of absolute counts
```

Core microbiome by Summer Stage at Order level

```
summer_stages <- unique(as.character(meta(physeq_comp)$Summer_Stage)) # vector of summer stages

physeq_comp_order <- physeq_comp %>%
  tax_glom(taxrank = "Order") # agglomerate on Order level

core_summer_stage_order <- c() # empty vector to store core microbiome results

for (n in summer_stages){
```

```

ps_sub <- subset_samples(physeq_comp_order, Summer_Stage == n)
core_m <- core_members(ps_sub,
  detection = 1/10000000, # minimum detection rate
  prevalence = 100/100, # must be present in all samples
  include.lowest = TRUE)

print(paste0("No. of core Order taxa in ", n, " : ", length(core_m)))
core_summer_stage_order[[n]] <- core_m # add core microbiome to results
}

```

```

## [1] "No. of core Order taxa in Early Summer : 43"
## [1] "No. of core Order taxa in Mid Summer : 35"
## [1] "No. of core Order taxa in Late Summer : 52"

```

Core microbiome by Summer Stage at Family level

```

physeq_comp_family <- physeq_comp %>%
tax_glom(taxrank = "Family")

core_summer_stage_family <- c()

for (n in summer_stages){
  ps_sub <- subset_samples(physeq_comp_family, Summer_Stage == n)
  core_m <- core_members(ps_sub,
    detection = 1/10000000,
    prevalence = 100/100,
    include.lowest = TRUE)

  print(paste0("No. of core Family taxa in ", n, " : ", length(core_m)))
  core_summer_stage_family[[n]] <- core_m
}

```

```

## [1] "No. of core Family taxa in Early Summer : 61"
## [1] "No. of core Family taxa in Mid Summer : 49"
## [1] "No. of core Family taxa in Late Summer : 74"

```

Core microbiome by Summer Stage at Genus level

```

core_summer_stage_genus <- c()

for (n in summer_stages){
  ps_sub <- subset_samples(physeq_comp, Summer_Stage == n)
  core_m <- core_members(ps_sub,
    detection = 1/10000000,
    prevalence = 100/100,
    include.lowest = TRUE)
  print(paste0("No. of core Genus taxa in ", n, " : ", length(core_m)))
  core_summer_stage_genus[[n]] <- core_m
}

```

```
## [1] "No. of core Genus taxa in Early Summer : 93"
## [1] "No. of core Genus taxa in Mid Summer : 66"
## [1] "No. of core Genus taxa in Late Summer : 111"
```

Core microbiome by Library Season at Genus level

```
lib_season <- unique(as.character(meta(physeq_comp)$lib_season))
core_lib_season_genus <- c() # an empty object to store information

for (n in lib_season){
  ps_sub <- subset_samples(physeq_comp, lib_season == n)
  core_m <- core_members(ps_sub,
                        detection = 1/10000000,
                        prevalence = 100/100,
                        include.lowest = TRUE)
  print(paste0("No. of core Genus taxa in ", n, " : ", length(core_m)))
  core_lib_season_genus[[n]] <- core_m
}
```

```
## [1] "No. of core Genus taxa in 12-13 : 85"
## [1] "No. of core Genus taxa in 13-14 : 85"
## [1] "No. of core Genus taxa in 14-15 : 86"
```

Plot Venn diagrams of core microbiome at Order, Family, and Genus levels

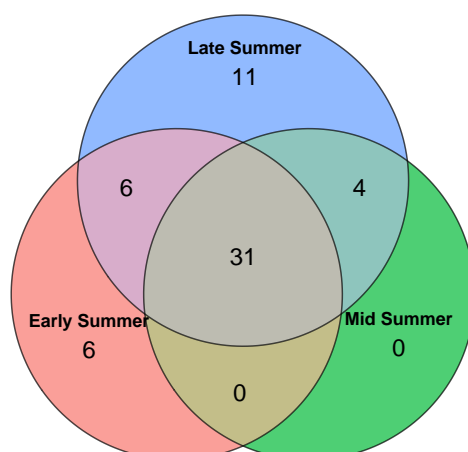
```
# create venn diagram plots
v1 <- plot(venn(core_summer_stage_order),
  fills = mycols,
  main = "Core Microbiome of Summer Stages at Order Level",
  alpha = .7,
  labels = list(fontsize = 9))

v2 <- plot(venn(core_summer_stage_family),
  fills = mycols,
  main = "Core Microbiome of Summer Stages at Family Level",
  alpha = .7,
  labels = list(fontsize = 9))

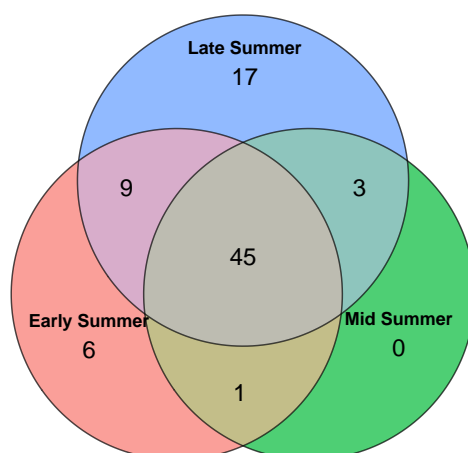
v3 <- plot(venn(core_summer_stage_genus),
  fills = mycols,
  main = "Core Microbiome of Summer Stages at Genus Level",
  cex.main = 5,
  alpha = .7,
  labels = list(fontsize = 9))

# arrange plots and add spacing between plots
ggarrange(v1, NULL, v2, NULL, v3,
  nrow = 5,
  heights = c(1, 0.2, 1, 0.2, 1))
```

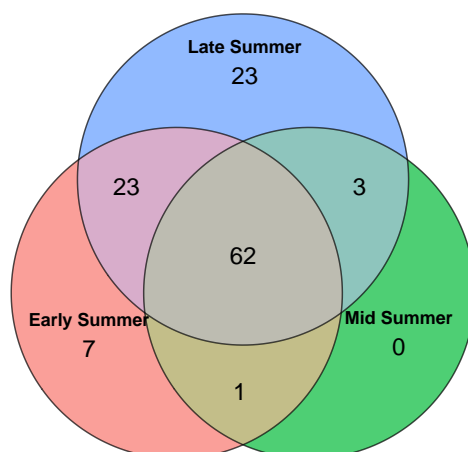
Core Microbiome of Summer Stages at Order Level



Core Microbiome of Summer Stages at Family Level



Core Microbiome of Summer Stages at Genus Level

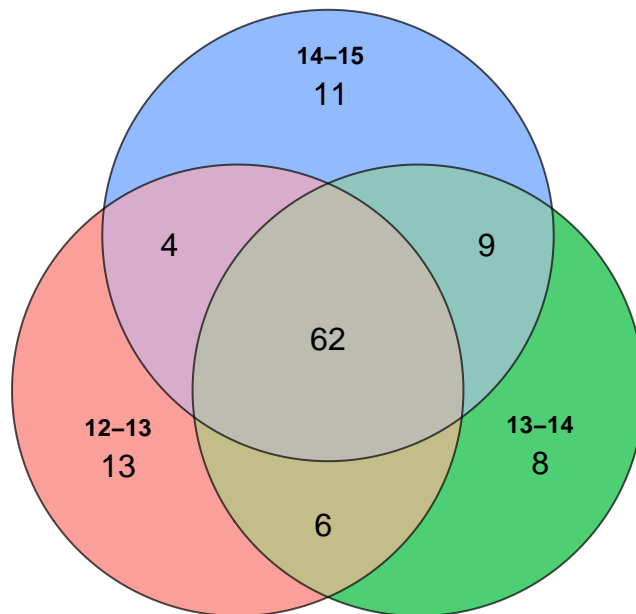


Mid summer samples did not have any taxa unique solely to the mid summer.

Plot core microbiome based on library seasons instead of summer stages

```
plot(venn(core_lib_season_genus),  
     fills = mycols,  
     main = "Core Microbiome of Library Seasons at Genus Level",  
     alpha = .7,  
     labels = list(fontsize = 9))
```

Core Microbiome of Library Seasons at Genus Level



Both summer stages and library seasons share a core of 62 taxa at Genus level.

At Order level, 31 taxa are shared between all groups.

Note, if two different ASVs are considered

Core taxa names in Summer Stages at Order level

```
early_summer_core_order <- subset_samples(  
  physeq_comp_order, Summer_Stage == "Early Summer") %>%  
  core(detection = 1/10000000,  
       prevalence = 100/100,  
       include.lowest = TRUE) %>%  
  tax_table()  
early_summer_core_order <- as.vector(early_summer_core_order[, "Order"])  
  
mid_summer_core_order <- subset_samples(  
  physeq_comp_order, Summer_Stage == "Mid Summer") %>%  
  core(detection = 1/10000000,  
       prevalence = 100/100,
```



```

                                include.lowest = TRUE) %>%
                                tax_table()
mid_summer_core_order <- as.vector(mid_summer_core_order[, "Order"])

late_summer_core_order <- subset_samples(
  physeq_comp_order, Summer_Stage == "Late Summer") %>%
  core(detection = 1/10000000,
       prevalence = 100/100,
       include.lowest = TRUE) %>%
  tax_table()
late_summer_core_order <- as.vector(late_summer_core_order[, "Order"])

core_summer_stages_order_taxa <- c()

for (n in summer_stages){
  core_m <- subset_samples(physeq_comp_order, Summer_Stage == n) %>%
  core(detection = 1/10000000,
       prevalence = 100/100,
       include.lowest = TRUE) %>%
  tax_table()
  core_m <- as.vector(core_m[, "Order"])
  core_summer_stages_order_taxa[[n]] <- core_m
}

```

31 core taxa at Order level

```

Reduce(intersect, list(core_summer_stages_order_taxa$`Early Summer`,
                       core_summer_stages_order_taxa$`Mid Summer`,
                       core_summer_stages_order_taxa$`Late Summer`))

```

```

## [1] "SAR86_clade" "Burkholderiales"
## [3] "Defluviicoccales" "Parvibaculales"
## [5] "Rhodospirillales" "Thalassobaculales"
## [7] "Puniceispirillales" "Fusobacteriales"
## [9] "Clostridiales" "Campylobacteriales"
## [11] "PeM15" "Verrucomicrobiales"
## [13] "Cytophagales" "SAR324_clade(Marine_group_B)"
## [15] "Chitinophagales" "Nitrosopumilales"
## [17] "Sphingobacteriales" "Flavobacteriales"
## [19] "Caulobacteriales" "Rhodobacteriales"
## [21] "SAR11_clade" "Microtrichales"
## [23] "Granulosicoccales" "Thiomicrospirales"
## [25] "Alteromonadales" "Cellvibrionales"
## [27] "Oceanospirillales" "Nitrosococcales"
## [29] "Thiotrichales" "Arenicellales"
## [31] "OM182_clade"

```

Core taxa unique to Early Summer and Late Summer only

```

setdiff(intersect(core_summer_stages_order_taxa$`Early Summer`, core_summer_stages_order_taxa$`Late Summer`),
        core_summer_stages_order_taxa$`Early Summer`)

## [1] "Marinimicrobia_(SAR406_clade)" "Marine_Group_II"

```

```
## [3] "JGI_0000069-P22"          "Kordiimonadales"
## [5] "JTB23"                    "KI89A_clade"
```

Core taxa unique to Mid Summer and Late Summer only

```
setdiff(intersect(core_summer_stages_order_taxa$`Mid Summer`, core_summer_stages_order_taxa$`Late Summer`
```

```
## [1] "Sphingomonadales" "Micrococcales"    "Rhizobiales"      "Pseudomonadales"
```

Core taxa unique to Early Summer only

```
setdiff(setdiff(core_summer_stages_order_taxa$`Early Summer`, core_summer_stages_order_taxa$`Late Summer`
```

```
## [1] "uncultured"          "AT-s3-44"          "Tenderiales"
## [4] "Steroidobacterales"  "Pirellulales"      "UBA10353_marine_group"
```

Core taxa unique to Late Summer only

```
setdiff(setdiff(core_summer_stages_order_taxa$`Late Summer`, core_summer_stages_order_taxa$`Early Summer`
```

```
## [1] "Methylococcales"      "Micavibrionales"
## [3] "Desulfohalobiales"    "Peptostreptococcales-Tissierellales"
## [5] "Lachnospirales"      "Bdellovibrionales"
## [7] "Bacteroidales"       "Rickettsiales"
## [9] "Ectothiorhodospirales" "Deinococcales"
## [11] "Vibrionales"
```

Core taxa names in Library Seasons at Genus level

```
core_lib_season_genus_taxa <- c()

for (n in lib_season){
  ps_sub <- subset_samples(physeq_comp, lib_season == n)
  core_m <- core(ps_sub,
                 detection = 1/10000000,
                 prevalence = 100/100,
                 include.lowest = TRUE) %>%
    tax_table()
  core_m <- as.vector(core_m[, "Genus"])
  core_lib_season_genus_taxa[[n]] <- core_m
}
```

```
core_lib_season <- Reduce(intersect, list(core_lib_season_genus_taxa$`12-13`,
                                           core_lib_season_genus_taxa$`13-14`,
                                           core_lib_season_genus_taxa$`14-15`))
core_lib_season
```

```
## [1] "SAR86_clade"          "OM43_clade"
## [3] "uncultured"          "AEGEAN-169_marine_group"
## [5] "OCS116_clade"        "Magnetospira"
```

```
## [7] "OM75_clade" "SAR116_clade"
## [9] "Psychrilyobacter" "Clostridium_sensu_stricto_1"
## [11] "PeM15" "Clade_II"
## [13] "Clade_III" "Rubritalea"
## [15] "Marinoscillum" "SAR324_clade(Marine_group_B)"
## [17] "NS9_marine_group" "Portibacter"
## [19] "Lewinella" "Candidatus_Nitrosopumilus"
## [21] "Crocinitomix" "NS11-12_marine_group"
## [23] "Vicingus" "Polaribacter"
## [25] "NS5_marine_group" "NS4_marine_group"
## [27] "Ulvibacter" "Litorimonas"
## [29] "Fretibacter" "Hellea"
## [31] "Robiginitomaculum" "Sulfitobacter"
## [33] "Yoonia-Loktanella" "Ascidiaaceihabitans"
## [35] "Clade_Ia" "Clade_IV"
## [37] "Amylibacter" "Planktomarina"
## [39] "Brevundimonas" "Sva0996_marine_group"
## [41] "Granulosicoccus" "SUP05_cluster"
## [43] "Paraglaciecola" "Psychromonas"
## [45] "Colwellia" "SAR92_clade"
## [47] "Cocleimonas" "Leucothrix"
## [49] "Arenicella" "BD1-7_clade"
## [51] "OM60(NOR5)_clade" "Pseudohongiella"
## [53] "OM182_clade"
```

```
core_summer_stages_genus_taxa <- c()

for (n in summer_stages){
  ps_sub <- subset_samples(physeq_comp, Summer_Stage == n)
  core_m <- core(ps_sub,
    detection = 1/10000000,
    prevalence = 100/100,
    include.lowest = TRUE) %>%
    tax_table()
  core_m <- as.vector(core_m[, "Genus"])
  core_summer_stages_genus_taxa[[n]] <- core_m
}
```

```
core_summer_stage <- Reduce(intersect, list(core_summer_stages_genus_taxa$`Early Summer`,
  core_summer_stages_genus_taxa$`Mid Summer`,
  core_summer_stages_genus_taxa$`Late Summer`))

core_summer_stage
```

```
## [1] "SAR86_clade" "OM43_clade"
## [3] "uncultured" "AEGEAN-169_marine_group"
## [5] "OCS116_clade" "Magnetospira"
## [7] "OM75_clade" "SAR116_clade"
## [9] "Psychrilyobacter" "Clostridium_sensu_stricto_1"
## [11] "PeM15" "Clade_II"
## [13] "Clade_III" "Rubritalea"
## [15] "Marinoscillum" "SAR324_clade(Marine_group_B)"
## [17] "NS9_marine_group" "Portibacter"
## [19] "Lewinella" "Candidatus_Nitrosopumilus"
```

```
## [21] "Crocinitomix"           "NS11-12_marine_group"
## [23] "Vicingus"              "Polaribacter"
## [25] "NS5_marine_group"      "NS4_marine_group"
## [27] "Ulvibacter"            "Litorimonas"
## [29] "Fretibacter"           "Hellea"
## [31] "Robiginitomaculum"     "Sulfitobacter"
## [33] "Yoonia-Loktanella"     "Ascidiaaceihabitans"
## [35] "Clade_Ia"              "Clade_IV"
## [37] "Amylibacter"           "Planktomarina"
## [39] "Brevundimonas"         "Sva0996_marine_group"
## [41] "Granulosicoccus"       "SUP05_cluster"
## [43] "Paraglaciecola"        "Psychromonas"
## [45] "Colwellia"             "SAR92_clade"
## [47] "Cocleimonas"           "Leucothrix"
## [49] "Arenicella"            "BD1-7_clade"
## [51] "OM60(NOR5)_clade"      "Pseudohongiella"
## [53] "OM182_clade"
```

At genus level, the core taxa in all summer stages is the same as the core taxa in all library seasons.

Note: There are 62 unique ASVs at the genus level shown in the Venn diagrams, but only 53 unique taxa names are shown with the intersection between all summer stages or library seasons. This is due to the presence of “uncultured” taxa being counted together as one.

Top 10 taxa at Order, Family, and Genus levels by Proportion

Top 10 taxa at Order level by proportion

```
# create phyloseq object of the core microbiome
physeq_core_counts <- core(physeq,
                           detection = 1,
                           prevalence = 100/100,
                           include.lowest = TRUE)
```

```
core_order_prop <- as_tibble(taxa_proportions(physeq_core_counts, "Order")) %>%
  arrange(desc(Proportion))
```

```
top_10_core_order <- core_order_prop[1:10,]
top_10_core_order
```

```
## # A tibble: 10 x 2
##   Order          Proportion
##   <chr>          <dbl>
## 1 Flavobacteriales 0.408
## 2 Rhodobacterales 0.257
## 3 Oceanospirillales 0.185
## 4 Cellvibrionales 0.033
## 5 Nitrosococcales 0.022
## 6 Thiomicrospirales 0.019
## 7 Alteromonadales 0.015
## 8 Sphingobacteriales 0.012
```

```
## 9 Burkholderiales      0.01
## 10 SAR11_clade          0.008
```

```
sum(top_10_core_order[,2])
```

```
## [1] 0.969
```

Top 10 Order taxa sum to 96.9% of the total proportion of reads

Top 10 taxa at Family level by proportion

```
core_family_prop <- as_tibble(taxa_proportions(physeq_core_counts, "Family")) %>%
  arrange(desc(Proportion))
```

```
top_10_core_family <- core_family_prop[1:10,]
top_10_core_family
```

```
## # A tibble: 10 x 2
##   Family      Proportion
##   <chr>      <dbl>
## 1 Flavobacteriaceae    0.344
## 2 Rhodobacteraceae    0.257
## 3 Nitrincolaceae      0.179
## 4 Cryomorphaceae      0.054
## 5 Porticoccaceae      0.028
## 6 Methylophagaceae    0.022
## 7 Thioglobaceae       0.019
## 8 NS11-12_marine_group 0.012
## 9 Methylophilaceae    0.01
## 10 NS9_marine_group    0.009
```

```
sum(top_10_core_family[,2])
```

```
## [1] 0.934
```

Top 10 Family taxa sum to 93.4% of the total proportion of reads

Top 10 taxa at Genus level by proportion

```
core_genus_prop <- as_tibble(taxa_proportions(physeq_core_counts, "Genus")) %>%
  arrange(desc(Proportion))
```

```
top_10_core_genus <- core_genus_prop[1:10,]
top_10_core_genus
```

```
## # A tibble: 10 x 2
##   Genus      Proportion
##   <chr>      <dbl>
## 1 Polaribacter    0.286
```

```
## 2 uncultured 0.254
## 3 Sulfitobacter 0.116
## 4 Yoonia-Loktanelia 0.089
## 5 Planktomarina 0.033
## 6 NS5_marine_group 0.03
## 7 SAR92_clade 0.028
## 8 Ulvibacter 0.026
## 9 SUP05_cluster 0.019
## 10 NS11-12_marine_group 0.012
```

```
sum(top_10_core_genus[,2])
```

```
## [1] 0.893
```

Many taxa are uncultured at Genus level

Top 10 Taxa by Abundance at Order Level

```
physeq_order <- physeq %>%
  tax_glom(taxrank = "Order") %>%
  transform_sample_counts(function(x) {100 * x/sum(x)}) %>%
  psmelt() %>%
  mutate(year_plot = ifelse(month >= 10, 2000, # for plotting multiple years
    ifelse(month < 10, 2001, NA))) %>% # onto the same scale
  mutate(plot_date = make_date(year_plot, month, day))
```

Flavobacteriales

```
Flavobacteriales <- physeq_order %>%
  filter(Order == "Flavobacteriales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Flavobacteriales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Rhodobacterales

```
Rhodobacterales <- physeq_order %>%
  filter(Order == "Rhodobacterales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Rhodobacterales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Oceanospirillales

```
Oceanospirillales <- physeq_order %>%
  filter(Order == "Oceanospirillales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Oceanospirillales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 4)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Cellvibrionales

```
Cellvibrionales <- physeq_order %>%
  filter(Order == "Cellvibrionales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Cellvibrionales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Nitrosococcales

```
Nitrosococcales <- physeq_order %>%
  filter(Order == "Nitrosococcales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Nitrosococcales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Thiomicrospirales

```
Thiomicrospirales <- physeq_order %>%
  filter(Order == "Thiomicrospirales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Thiomicrospirales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Alteromonadales

```

Alteromonadales <- physeq_order %>%
  filter(Order == "Alteromonadales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Alteromonadales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Sphingobacteriales

```

Sphingobacteriales <- physeq_order %>%
  filter(Order == "Sphingobacteriales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Sphingobacteriales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Burkholderiales

```

Burkholderiales <- physeq_order %>%
  filter(Order == "Burkholderiales") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Burkholderiales", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

SAR11

```

SAR11 <- physeq_order %>%
  filter(Order == "SAR11_clade") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "SAR11", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```



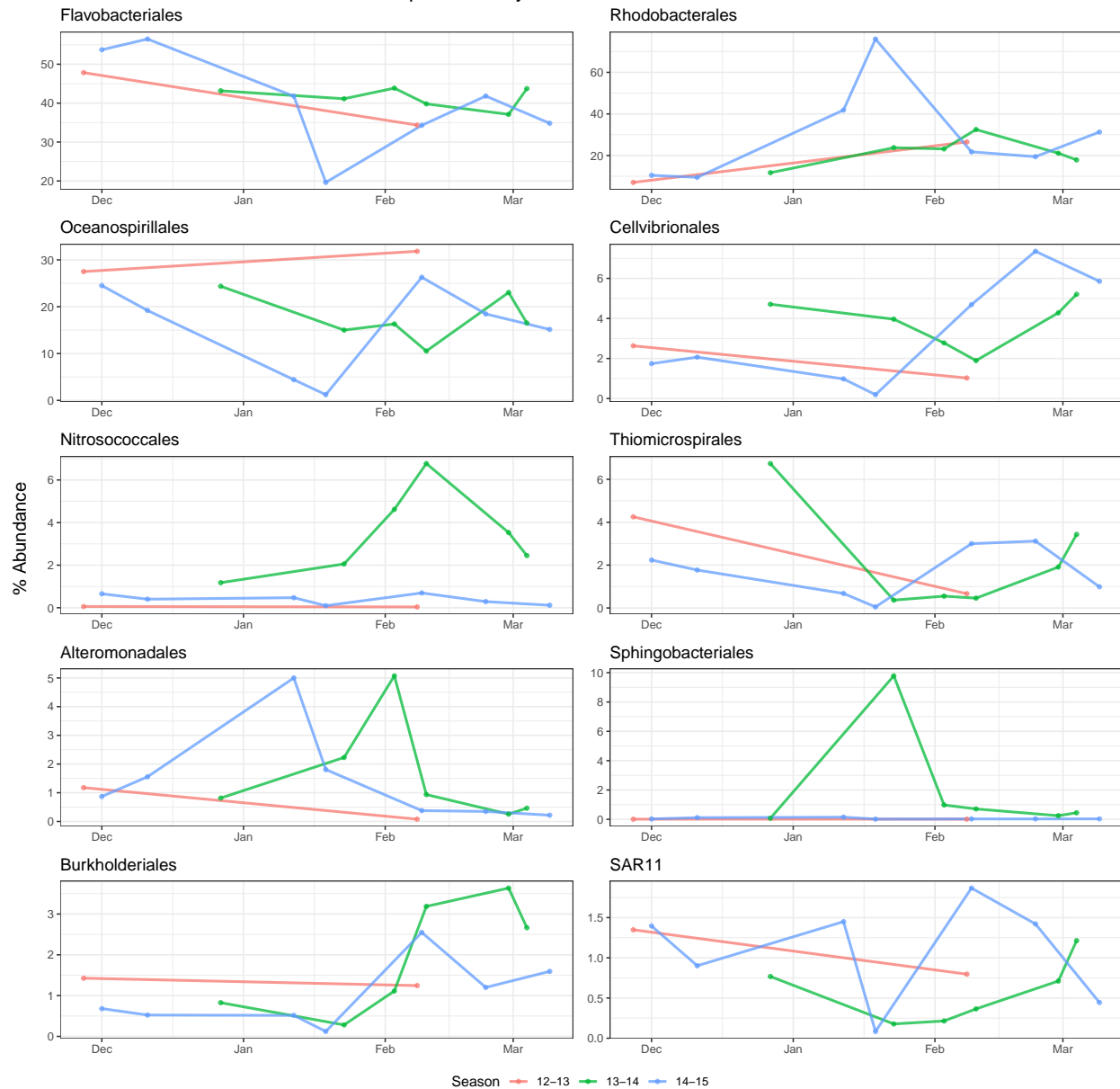
```

top_10_order_plot <- ggarrange(
  Flavobacteriales,
  Rhodobacterales,
  Oceanospirillales,
  Cellvibrionales,
  Nitrosococcales,
  Thiomicrospirales,
  Alteromonadales,
  Sphingobacteriales,
  Burkholderiales,
  SAR11,
  nrow = 5,
  ncol = 2,
  common.legend = TRUE,
  legend = "bottom",
  align = "hv")

annotate_figure(top_10_order_plot,
  left = text_grob("% Abundance", rot = 90, size = 14),
  top = text_grob("Top 10 Taxa by Abundance at Order Level", size = 16))

```

Top 10 Taxa by Abundance at Order Level



Top 10 Taxa by Abundance at Family Level

```

physeq_family <- physeq %>%
  tax_glom(taxrank = "Family") %>%
  transform_sample_counts(function(x) {100 * x/sum(x)}) %>%
  psmelt() %>%
  mutate(year_plot = ifelse(month >= 10, 2000, # for plotting multiple years
    ifelse(month < 10, 2001, NA))) %>% # onto the same scale
  mutate(plot_date = make_date(year_plot, month, day))

```

Flavobacteriaceae

```

Flavobacteriaceae <- physeq_family %>%
  filter(Family == "Flavobacteriaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Flavobacteriaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Rhodobacteraceae

```

Rhodobacteraceae <- physeq_family %>%
  filter(Family == "Rhodobacteraceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Rhodobacteraceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Nitrincolaceae

```

Nitrincolaceae <- physeq_family %>%
  filter(Family == "Nitrincolaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Nitrincolaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 4)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Cryomorphaceae

```

Cryomorphaceae <- physeq_family %>%
  filter(Family == "Cryomorphaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Cryomorphaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

Porticoccaceae

```
Porticoccaceae <- physeq_family %>%
  filter(Family == "Porticoccaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Porticoccaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Methylophagaceae

```
Methylophagaceae <- physeq_family %>%
  filter(Family == "Methylophagaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Methylophagaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Thioglobaceae

```
Thioglobaceae <- physeq_family %>%
  filter(Family == "Thioglobaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Thioglobaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

NS11-12_marine_group

```
NS11_12_marine_group <- physeq_family %>%
  filter(Family == "NS11-12_marine_group") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "NS11-12 Marine Group", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

Methylophilaceae

```

Methylophilaceae <- physeq_family %>%
  filter(Family == "Methylophilaceae") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "Methylophilaceae", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

NS9_marine_group

```

NS9_marine_group <- physeq_family %>%
  filter(Family == "NS9_marine_group") %>%
  ggplot(aes(x = plot_date, y = Abundance,
    group = factor(lib_season), color = factor(lib_season))) +
  geom_line(size = 1, alpha = .75) +
  geom_point(size = 1.25, alpha = .75) +
  scale_x_date(date_breaks = "months", date_labels = "%b") +
  labs(title = "NS9 Marine Group", color = "Season", y = "% Abundance") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())

```

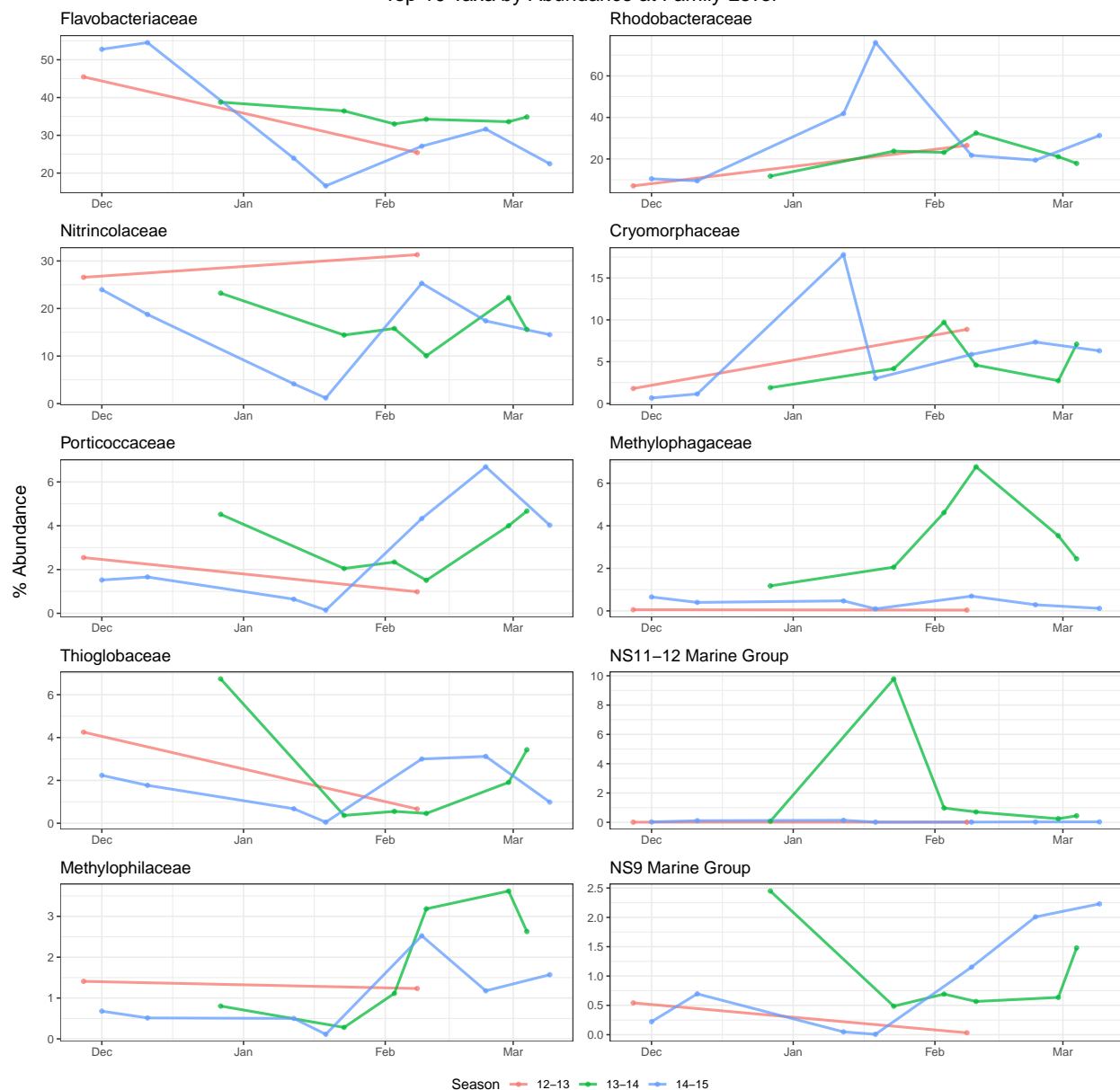
```

top_10_family_plot <- ggarrange(
  Flavobacteriaceae,
  Rhodobacteraceae,
  Nitrincolaceae,
  Cryomorphaceae,
  Porticoccaceae,
  Methylophagaceae,
  Thioglobaceae,
  NS11_12_marine_group,
  Methylophilaceae,
  NS9_marine_group,
  nrow = 5,
  ncol = 2,
  common.legend = TRUE,
  legend = "bottom",
  align = "hv")

annotate_figure(top_10_family_plot,
  left = text_grob("% Abundance", rot = 90, size = 14),
  top = text_grob("Top 10 Taxa by Abundance at Family Level", size = 16))

```

Top 10 Taxa by Abundance at Family Level



sessionInfo()

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
```

```

## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel stats      graphics grDevices utils      datasets methods
## [8] base
##
## other attached packages:
## [1] biomformat_1.18.0      scales_1.1.1
## [3] gridExtra_2.3          ape_5.4-1
## [5] reshape2_1.4.4         dendextend_1.14.0
## [7] tidytext_0.3.0         RColorBrewer_1.1-2
## [9] rstatix_0.7.0          ggpubr_0.4.0
## [11] treemap_2.4-2          eulerr_6.1.0
## [13] ggrepel_0.9.1.9999     readxl_1.3.1
## [15] lubridate_1.7.10       ggordiplots_0.4.0
## [17] glue_1.4.2             microbiomeutilities_1.00.15
## [19] phylosmith_1.0.5       microbiome_1.12.0
## [21] phyloseq_1.34.0        vegan_2.5-7
## [23] lattice_0.20-41        permute_0.9-5
## [25] qiime2R_0.99.4         forcats_0.5.1
## [27] stringr_1.4.0          dplyr_1.0.5
## [29] purrr_0.3.4            readr_1.4.0
## [31] tidyr_1.1.3            tibble_3.1.0
## [33] ggplot2_3.3.3          tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
## [1] backports_1.2.1      Hmisc_4.5-0          plyr_1.8.6
## [4] igraph_1.2.6         polylabelr_0.2.0     splines_4.0.4
## [7] SnowballC_0.7.0      gridBase_0.4-7       digest_0.6.27
## [10] foreach_1.5.1        htmltools_0.5.1.1    viridis_0.5.1
## [13] fansi_0.4.2          magrittr_2.0.1       checkmate_2.0.0
## [16] cluster_2.1.1        openxlsx_4.2.3       Biostrings_2.58.0
## [19] graphlayouts_0.7.1   modelr_0.1.8         prettyunits_1.1.1
## [22] jpeg_0.1-8.1         colorspace_2.0-0     rvest_1.0.0
## [25] haven_2.3.1          xfun_0.22            crayon_1.4.1
## [28] jsonlite_1.7.2       survival_3.2-7       iterators_1.0.13
## [31] polyclip_1.10-0      gtable_0.3.0         zlibbioc_1.36.0
## [34] XVector_0.30.0       car_3.0-10           Rhdf5lib_1.12.1
## [37] BiocGenerics_0.36.0  abind_1.4-5          pheatmap_1.0.12
## [40] DBI_1.1.1            Rcpp_1.0.6           xtable_1.8-4
## [43] viridisLite_0.3.0    progress_1.2.2       htmlTable_2.1.0
## [46] units_0.7-1          foreign_0.8-81        proxy_0.4-25
## [49] Formula_1.2-4        stats4_4.0.4         DT_0.17
## [52] truncnorm_1.0-8      htmlwidgets_1.5.3    httr_1.4.2
## [55] ellipsis_0.3.1       pkgconfig_2.0.3      NADA_1.6-1.1
## [58] farver_2.1.0         nnet_7.3-15          dbplyr_2.1.0
## [61] utf8_1.2.1           labeling_0.4.2       later_1.1.0.1
## [64] tidyselect_1.1.0     rlang_0.4.10         munsell_0.5.0
## [67] cellranger_1.1.0     tools_4.0.4          cli_2.3.1
## [70] generics_0.1.0       ade4_1.7-16          broom_0.7.5
## [73] fastmap_1.1.0        evaluate_0.14        yaml_2.2.1
## [76] knitr_1.31           fs_1.5.0             tidygraph_1.2.0
## [79] zip_2.1.1            ggraph_2.0.5         nlme_3.1-152
## [82] mime_0.10            xml2_1.3.2           tokenizers_0.2.1

```

```

## [85] compiler_4.0.4      rstudioapi_0.13      curl_4.3
## [88] png_0.1-7           ggsignif_0.6.1       e1071_1.7-6
## [91] zCompositions_1.3.4 reprex_1.0.0          tweenr_1.0.2
## [94] stringi_1.5.3       highr_0.8            Matrix_1.3-2
## [97] classInt_0.4-3      multtest_2.46.0      vctrs_0.3.7
## [100] pillar_1.5.1        lifecycle_1.0.0      rhdf5filters_1.2.0
## [103] cowplot_1.1.1       data.table_1.14.0    httpuv_1.5.5
## [106] R6_2.5.0            latticeExtra_0.6-29 promises_1.2.0.1
## [109] rio_0.5.26          KernSmooth_2.23-18 janeaustenr_0.1.5
## [112] IRanges_2.24.1      codetools_0.2-18    MASS_7.3-53.1
## [115] assertthat_0.2.1    rhdf5_2.34.0         withr_2.4.1
## [118] S4Vectors_0.28.1    mgcv_1.8-34          hms_1.0.0
## [121] grid_4.0.4          rpart_4.1-15         gghalves_0.1.1
## [124] class_7.3-18         rmarkdown_2.7        carData_3.0-4
## [127] Rtsne_0.15          sf_0.9-8             ggforce_0.3.3
## [130] shiny_1.6.0         Biobase_2.50.0       base64enc_0.1-3

```