Orchard Project: Supplementary

#### Colin Brislawn, PNNL, Spring 2017

#### (Includes both 16S and 18S data)

#### About this document

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

This project makes use of many packages, especially Phyloseq <https://joey711.github.io/phyloseq/>.

The goal of this document it to provide a comprehensive overview of all methods used in the paper for visualizing amplicon data.

This document is directly based on the methods used in the Founder Species project, also a collaboration with Hans Bernstein in 2016-2017.

### Setup:

library("checkpoint")

##   
## checkpoint: Part of the Reproducible R Toolkit from Microsoft  
## https://mran.microsoft.com/documents/rro/reproducibility/

library("knitr")  
checkpoint("2016-10-01", use.knitr = T)

## Scanning for packages used in this project

## Warning in deps.R(tempfile): No file at path '/var/folders/xx/  
## 9tr91yb978sfrvlckym07qj1qn7b14/T//Rtmp3MUb6T/file13d8f6089f6c2.Rmd'.

## - Discovered 0 packages

## Unable to parse 1 files:

## - OrchardSup.Rmd

## No packages found to install

## checkpoint process complete

## ---

library("ggplot2")  
library("phyloseq")  
library("RColorBrewer")  
library("viridis")  
library("scales")  
library("cowplot")

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggplot2':  
##   
## ggsave

library("vegan")

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.4-1

library("dplyr")

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library("lubridate")

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library("reshape2")  
library("ggrepel")  
#library("car")  
#library("multcomp")  
#library("lme4")  
  
packageVersion('phyloseq')  
theme\_set(theme\_bw())  
set.seed(711)  
opts\_chunk$set(cache=TRUE, dev = c('png'), warning = F)

### Import data

Our work here will focus on two amplicon data sets. One from the 16S gene, the other from the 18S gene. Because there amplicons came from different primers and are expected to target different genes (of different evolutionary history, length, complexity, etc) they will be analyzed separately. (This also makes the stats simpler.)

no\_meta <- import\_biom(file.path('../data/16S\_otus\_vsearch/otu\_table\_w\_tax.biom')  
 #,file.path('../16S/otus\_vsearch/rep\_set.tre')  
 )  
meta <- import\_qiime\_sample\_data(file.path('../metadata/generated-meta.tsv'))  
full16s <- merge\_phyloseq(meta, no\_meta)  
full16s

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 3152 taxa and 99 samples ]  
## sample\_data() Sample Data: [ 99 samples by 32 sample variables ]  
## tax\_table() Taxonomy Table: [ 3152 taxa by 7 taxonomic ranks ]

no\_meta\_18s <- import\_biom(file.path('../data/18S\_otus\_vsearch/otu\_table\_w\_tax.biom'))  
full18s <- merge\_phyloseq(meta, no\_meta\_18s)  
full18s

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 2838 taxa and 98 samples ]  
## sample\_data() Sample Data: [ 98 samples by 32 sample variables ]  
## tax\_table() Taxonomy Table: [ 2838 taxa by 7 taxonomic ranks ]

### Explore metadata

Let's take a look at columns in the metadata file.

**Make major edits to the metadata using the generate-metadata.Rmd file.** This file focuses on downstream analysis, while metadata wrangling should have be completed already.

# Let's use 16S, as it's our main focus, and has more samples.  
  
meta <- sample\_data(full16s)  
meta.n\_unique <- rapply(meta, function(x) length(table(x)))  
# The pairwise interesting factors  
meta.n\_unique[meta.n\_unique == 2]  
# Other potentially interesting factors  
meta.n\_unique[meta.n\_unique > 2 & meta.n\_unique < max(meta.n\_unique)]  
  
# Main study design:  
# 4 Substrates every 6 hours for 48 hours  
table(meta$Time..H., meta$Substrate)  
  
meta$rep. <- factor(meta$rep.)  
  
# Reorder levels of Substrate  
levels(meta$Substrate)  
meta$Substrate <- factor(meta$Substrate, levels = c("13C DIC", "13C Acetate", "13C Glucose", "Control", "Native Mat"))  
levels(meta$Substrate)  
  
# Also, rename levels of Substrate  
levels(meta$Substrate) <- c("DIC", "Acetate", "Glucose", "Control", "Native Mat")  
levels(meta$Substrate)  
  
# Fix import of standard timestamp format  
meta$collection\_timestamp <- ymd\_hms(meta$collection\_timestamp)  
  
# Let's keep Time..H. numeric  
#meta$Time..H. <- factor(meta$Time..H.)  
  
sample\_data(full16s) <- meta  
sample\_data(full18s) <- meta

## Environmental Metadata

This study includes measurements of several environmental variables over 48 hours. Understanding how these vary over time, let us focus on interesting changes.

table(meta$Time..H., meta$Substrate)  
  
head(meta)  
#meta.melt <- meta  
meta.melt = melt(meta, measure.vars = c("SKNT.mph", "SOLR.W.m.m", "PREC.in", "STEN.centibar", "TMP..C.", "fin\_diff\_Ave\_d13C.organic"))  
head(meta.melt)  
  
meta.gg <- ggplot(meta.melt, aes(Time..H., value, color = Time..H.))  
meta.gg <- meta.gg +  
 #geom\_boxplot(color = "gray") + #, outlier.size = 0) +  
 geom\_jitter(width = 0.1) +  
 facet\_grid(facets = variable~., scales = "free\_y") +  
 labs(x = "Time (h)", title = "Environmental factors") +  
 #v +  
 theme(legend.position = "none",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank())  
meta.gg

Only substrate, light, and time would realistically effect the sample.

Let's check out how the measured isotopes changed...

# Sunlight colormap  
pl.c <- scale\_color\_viridis(option = "C", discrete = F, begin = .1, end = 1)  
  
head(meta)  
table(meta$Substrate)  
meta.melt2 = melt(meta, measure.vars = c("Ave\_d13C", "Ave\_.C", "Ave\_.C.organic", "Ave\_d13C.organic", "diff\_Ave\_d13C.organic", "fin\_diff\_Ave\_d13C.organic"))  
head(meta.melt2)  
meta.gg2 <- ggplot(meta.melt2, aes(Time..H., value, color = SOLR.W.m.m))  
meta.gg2 <- meta.gg2 +   
 #geom\_boxplot(color = "gray") + #, outlier.size = 0) +  
 geom\_jitter(width = 0.1) +  
 facet\_grid(facets = variable~Substrate, scales = "free\_y", switch = "y") +  
 labs(x = "Time (h)", title = "Isotopes and Primary Productivity") +  
 pl.c +  
 theme(legend.position = "right",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 #, plot.margin=unit(c(5,5,-25,5), units = "pt")  
 )  
meta.gg2

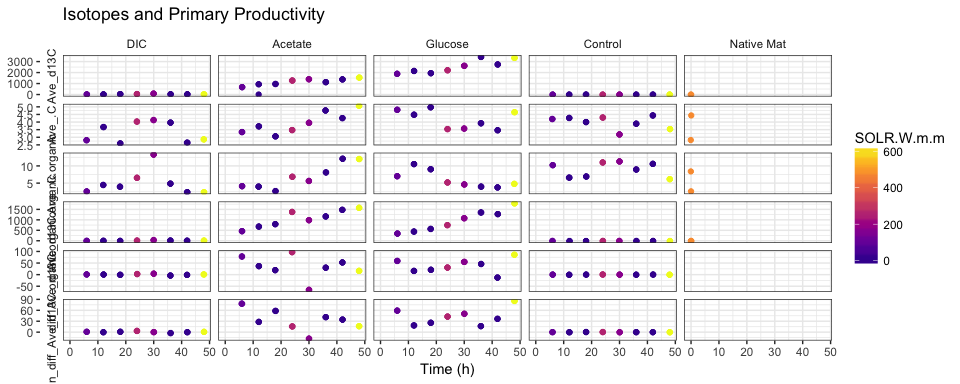


fig1.a <- ggplot(meta, aes(Time..H., SOLR.W.m.m, color = SOLR.W.m.m))  
fig1.a <- fig1.a + geom\_line(alpha = 1, color="gray80") + geom\_point(size = 1.5) +  
 #labs(y = "Ii (Wm^-2)") +  
 facet\_grid(PREC.in~.) + # This is a single 'dummy' facet so plot\_grid() will align it  
 scale\_color\_viridis(option = "C", end = .90) +  
 labs(x = "Time (h)", y =   
 #expression(atop("Irradiance", "("~ Wm^{-2}~ ")")) # two lines  
 expression("Irradiance ("~ Wm^{-2}~ ")") # one line  
 ) +  
 scale\_x\_continuous(breaks = c(0, 24, 48), limits = c(0, 48)) +  
 theme(  
 #axis.title.y = element\_text(angle = 0, vjust = .5),  
 legend.position = "none",  
 strip.text = element\_blank(), strip.background = element\_blank() # remove 'dummy' facet  
 )  
fig1.a

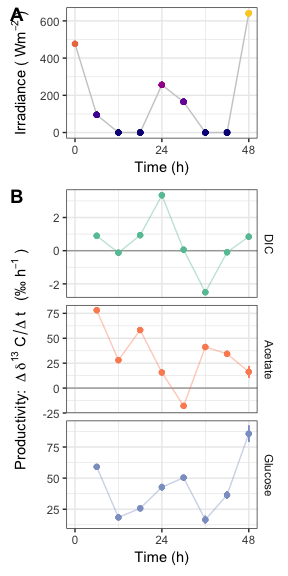
# only use key Substrates  
meta.small <- subset(meta, Substrate %in% c("Glucose", "Acetate", "DIC"))  
  
fig1.b <- ggplot(meta.small, aes(Time..H., fin\_diff\_Ave\_d13C.organic, color = Substrate))  
fig1.b <- fig1.b +  
 geom\_abline(slope = 0, color = "gray66") +  
 geom\_line(alpha = .4) +  
 geom\_point() +  
 geom\_errorbar(aes(ymax = (fin\_diff\_Ave\_d13C.organic + fin\_diff\_Stdev\_d13C.organic), ymin = (fin\_diff\_Ave\_d13C.organic - fin\_diff\_Stdev\_d13C.organic)), width = 0) +  
 # Note that we have two error bars. diff\_Stdev and diff\_Stdev\_hourly  
 scale\_color\_brewer(palette = "Set2") +  
 facet\_grid(Substrate~., scales = "free\_y") +  
 labs(x = "Time (h)", y =   
 expression("Productivity: " ~ Delta~delta^13~C / Delta ~t ~" (‰"~ h^-1~")") # note this uses the ‰ 'Per\_mille' sign  
 ) +  
 scale\_x\_continuous(breaks = c(0, 24, 48), limits = c(0, 48)) +  
 #geom\_text(aes(5, -20, label=Substrate), hjust = 0) +  
 theme(strip.background = element\_blank(),  
 legend.position = "none")  
fig1.b

#### Figure 1, a b

This figure could be rendered at 5.5 vs 2.4, to fit figure without scaling.

Here, I'm exporting at 6x3, with the expectation that it will be scaled down when composing the figure.

plot\_grid(fig1.a, fig1.b, align = "v", rel\_heights = c(1.4,3), ncol = 1, labels = c("A", "B"))

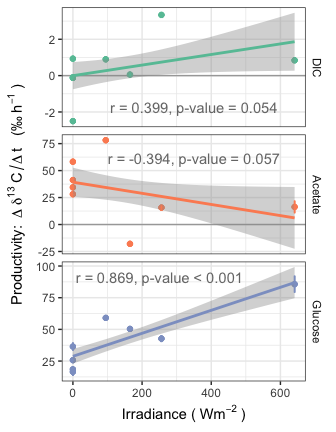


# this should match the height of fig1-cd and their widths should add to 183  
ggsave('fig1-ab.pdf', path = "./figures/", units = "mm", width = 61, height = 122, device = cairo\_pdf, scale = 1.2)

#### New Figure: productivity vs irradiance

Notes to Hans: This graph does not get a second panel for 16S and 18S because only a single measurement of carbon uptake was made. This shows one of the limitations of this measure.

# Let's see if we can add slopes and p-values to these graphs  
# See https://stackoverflow.com/questions/17022553/adding-r2-on-graph-with-facets  
  
  
# Example data for testing function  
df <- data.frame(meta.small[meta.small$Substrate == "Glucose", ])  
head(df)  
dim(df)  
  
lm\_eqn\_lighvsc = function(df){  
 pearson <- cor.test(df$SOLR.W.m.m, df$fin\_diff\_Ave\_d13C.organic, method = "pearson", conf.level = 0.95)  
 pearson  
 pearson$p.value # this p-value matches the one from summary(lm())$coefficients  
 pearson$estimate  
   
 eq <- substitute(  
 # This text parsing here is poor.  
 #atop("slope ="~slope, "p-value ="~pv), # Two lines  
 #~~R^2~"="~r2~", p-value "~pv, # with R^2  
 "r ="~r~"p-value"~pv, # One line  
 list(  
 r = paste(signif(pearson$estimate, digits = 3), ",", sep = ""),  
 pv = if(pearson$p.value <= 0.001) pv = "< 0.001" else pv = paste("= ", sprintf(fmt = "%.3f", pearson$p.value), sep = "")  
 )  
 )  
 as.character(as.expression(eq))   
}  
  
# Make a matching data frame with these calculated values  
# This is based on the stackoverflow answer, and on the Bile Acids graphs  
# See Lactobacillus acidophilus disrupts multispecies synthesis of a secondary bile acid ursocholate  
# Hans C. Bernstein, PNNL, 2017  
  
eqns <- by(data.frame(meta.small), INDICES = list(meta.small$Substrate), lm\_eqn\_lighvsc)  
#eqns  
  
df2 <- data.frame(eq = c(eqns),  
 Substrate = c("DIC", "Acetate", "Glucose"),  
 graphx = c(350, 350, 250),   
 graphy = c(-1.8, 60, 90))  
df2  
  
  
fig.pvi <- ggplot(meta.small, aes(SOLR.W.m.m, fin\_diff\_Ave\_d13C.organic, color = Substrate))  
fig.pvi <- fig.pvi +  
 geom\_abline(slope = 0, color = "gray66") +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 geom\_errorbar(aes(ymax = (fin\_diff\_Ave\_d13C.organic + fin\_diff\_Stdev\_d13C.organic), ymin = (fin\_diff\_Ave\_d13C.organic - fin\_diff\_Stdev\_d13C.organic)), width = 0) +  
 # Note that we have two error bars. diff\_Stdev and diff\_Stdev\_hourly  
 scale\_color\_brewer(palette = "Set2") + # use with color = Substrate  
 #scale\_color\_viridis(option = "C", end = .90) + # use with color = SOLR.W.m.m  
 facet\_grid(Substrate~., scales = "free\_y") +  
 geom\_text(data = df2, aes(x = graphx, y = graphy, label = eq), color = '#777777', parse = TRUE) +  
 labs(x = expression("Irradiance ("~ Wm^{-2}~ ")")  
 ,y=expression("Productivity: " ~ Delta~delta^13~C / Delta ~t ~" (‰"~ h^-1~")")  
 ) +  
 #geom\_text(aes(5, -20, label=Substrate), hjust = 0) +  
 theme(strip.background = element\_blank(),  
 legend.position = "none")  
fig.pvi



ggsave('fig-sup.pdf', path = "./figures/", units = "mm", width = 89, height = 120, device = cairo\_pdf, scale = 0.9)

## Preprocess

Remove all non-bacteria microbes, along with OTUs annotated as chloroplasts or mitochondria.

Inspect sample sizes and rarefy....

Also create cohort of Time > 0 samples. We may not want to include the initial three samples.

#### 16S

filtered16s <- subset\_taxa(full16s, Rank1 == "k\_\_Bacteria")  
ntaxa(filtered16s) / ntaxa(full16s)  
sum(taxa\_sums(filtered16s)) / sum(taxa\_sums(full16s))  
  
# Why are we losing so many reads?  
  
nonbacteria.16s <- subset\_taxa(full16s, Rank1 != "k\_\_Bacteria")  
plot\_bar(nonbacteria.16s, fill = "Rank1")

# Not NOT Archaea. It's mostly unnown microbes...  
  
plot(sort(taxa\_sums(nonbacteria.16s)))

head(sort(taxa\_sums(nonbacteria.16s), T))  
# ... Two unknown microbes: OTU\_4 and OTU\_5  
  
# After aligning out reads to Hot Lake data from Jen, we got <70% match for OTU\_5, but  
# 100% match for OTU\_4 with this taxonomy  
#CA\_scaffold\_065971 unbinned 83.2353 366.4339484 216.1214765 190.3818837 165.2089407 189.8963216 Bacteria;Chloroflexi;Anaerolineae;Anaerolineales;Anaerolineaceae;uncultured;  
# Let's fix it, with a greengenes-like name  
tax\_table(full16s)["OTU\_4",]  
# tax\_table(full16s)["OTU\_4",] <- c("Unassigned", NA, NA, NA, NA, NA, NA) # Value from taxonomy assignment  
tax\_table(full16s)["OTU\_4",] <- c("k\_\_Bacteria", "p\_\_Chloroflexi", "c\_\_Anaerolineae",  
 "o\_\_Anaerolineales", "f\_\_Anaerolineaceae", "g\_\_", "s\_\_")  
#k\_\_Bacteria; p\_\_Chloroflexi; c\_\_Anaerolineae; o\_\_Anaerolineales;  
  
# Now that OTU\_4 has been fixed, jump back into filtering  
filtered16s <- subset\_taxa(full16s, Rank1 == "k\_\_Bacteria")  
sum(taxa\_sums(filtered16s)) / sum(taxa\_sums(full16s))  
# Looks like OTU\_5 still causes us to lose 7% of reads, but that's much better  
  
filtered16s <- subset\_taxa(filtered16s, !(Rank5 %in% c("f\_\_mitochondria")))  
filtered16s <- subset\_taxa(filtered16s, !(Rank3 %in% c("c\_\_Chloroplast")))  
  
ntaxa(filtered16s) / ntaxa(full16s)  
sum(taxa\_sums(filtered16s)) / sum(taxa\_sums(full16s))  
# Very few reads are from mitochondria and chloroplast  
  
# Now throw away failed samples and normalize  
plot(sort(sample\_sums(filtered16s)))

ggplot(sample\_data(filtered16s), aes(sample\_sums(filtered16s))) + geom\_histogram(binwidth = 10000, aes(fill=Date))

ggplot(sample\_data(filtered16s), aes(sample\_sums(filtered16s))) + geom\_histogram(binwidth = 10000, aes(fill=Substrate))

sort(sample\_sums(filtered16s))[1:30]  
  
rar.16s <- rarefy\_even\_depth(filtered16s, sample.size = 22000)

## You set `rngseed` to FALSE. Make sure you've set & recorded  
## the random seed of your session for reproducibility.  
## See `?set.seed`

## ...

## 4 samples removedbecause they contained fewer reads than `sample.size`.

## Up to first five removed samples are:

## OE.42OE.40OE.37OE.38

## ...

## 393OTUs were removed because they are no longer   
## present in any sample after random subsampling

## ...

rar.16s  
  
rar.16s.cohort <- subset\_samples(rar.16s, Time..H. != 0 & Substrate %in% c("DIC", "Acetate", "Glucose"))  
rar.16s.cohort

#### 18S

18S is handled similarly, although no taxonomic filtering is used.

plot(sort(sample\_sums(full18s)))

ggplot(sample\_data(full18s), aes(sample\_sums(full18s))) + geom\_histogram(binwidth = 10000, aes(fill=Date))

ggplot(sample\_data(full18s), aes(sample\_sums(full18s))) + geom\_histogram(binwidth = 10000, aes(fill=Substrate))

sort(sample\_sums(full18s))[1:30]  
  
rar.18s <- rarefy\_even\_depth(full18s, sample.size = 22000)

## You set `rngseed` to FALSE. Make sure you've set & recorded  
## the random seed of your session for reproducibility.  
## See `?set.seed`

## ...

## 7 samples removedbecause they contained fewer reads than `sample.size`.

## Up to first five removed samples are:

## OE.75OE.47OE.45OE.52OE.25

## ...

## 353OTUs were removed because they are no longer   
## present in any sample after random subsampling

## ...

rar.18s  
  
rar.18s.cohort <- subset\_samples(rar.18s, Time..H. != 0 & Substrate %in% c("DIC", "Acetate", "Glucose"))  
rar.18s.cohort

#### Cohorts used in downstream analysis

These would look good as little heat maps...

rar.16s.cohort

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 2074 taxa and 68 samples ]  
## sample\_data() Sample Data: [ 68 samples by 32 sample variables ]  
## tax\_table() Taxonomy Table: [ 2074 taxa by 7 taxonomic ranks ]

rar.18s.cohort

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 2485 taxa and 67 samples ]  
## sample\_data() Sample Data: [ 67 samples by 32 sample variables ]  
## tax\_table() Taxonomy Table: [ 2485 taxa by 7 taxonomic ranks ]

The main feature is Substrate, varying over hours.

Looks like we lost samples throughout, but have at least one rep for each block of the study design.

Don't merge now, but maybe merge later for clear graphs.

rar.16s.cohort %>% taxa\_sums %>% sort %>% subset(. > 0) %>% length # number of OTUs

## [1] 1915

rar.16s.cohort %>% tax\_table %>% data.frame %>% select(Rank2) %>% unique %>% dim # number of Phyla

## [1] 45 1

rar.18s.cohort %>% taxa\_sums %>% sort %>% subset(. > 0) %>% length

## [1] 2322

rar.18s.cohort %>% tax\_table %>% data.frame %>% select(Rank2) %>% unique %>% dim # number of Phyla

## [1] 12 1

# Plots

I'll use rarefied data, for consistency.

## Abundance plots

There will be a matching set of relative abundance plots, for each amplicon type.

Taxa abundances will be used twice. First for stacked bar plots in figure 1, then again for input to bi-plot. The difference is that we will be merging reps before graphing taxa in the barplots in Figure 1.

# make a numeric time factor  
sample\_data(rar.16s.cohort)$Time..H. <- as.numeric(sample\_data(rar.16s.cohort)$Time..H.)  
sample\_data(rar.18s.cohort)$Time..H. <- as.numeric(sample\_data(rar.18s.cohort)$Time..H.)  
  
# Change rep name  
sample\_data(rar.16s.cohort)$newrep <- paste(sample\_data(rar.16s.cohort)$Substrate, sample\_data(rar.16s.cohort)$timepoint)  
sample\_data(rar.18s.cohort)$newrep <- paste(sample\_data(rar.18s.cohort)$Substrate, sample\_data(rar.18s.cohort)$timepoint)  
  
# Merge Samples,  
rar.16s.cohort.merged <- merge\_samples(rar.16s.cohort, c("newrep"))  
rar.18s.cohort.merged <- merge\_samples(rar.18s.cohort, c("newrep"))  
  
# Fix names after merge  
sample\_data(rar.16s.cohort.merged)$Substrate <-  
 factor((sample\_data(rar.16s.cohort.merged)$Substrate), labels = c("DIC", "Acetate", "Glucose"))  
sample\_data(rar.18s.cohort.merged)$Substrate <-  
 factor((sample\_data(rar.18s.cohort.merged)$Substrate), labels = c("DIC", "Acetate", "Glucose"))  
# Check that this works  
#head(sample\_data(rar.16s.cohort.merged))  
  
#merge OTUs, then transform to RA  
rar.16s.cohort.merged5 <- tax\_glom(rar.16s.cohort.merged, taxrank = "Rank5")  
rar.16s.cohort.merged5 <- transform\_sample\_counts(rar.16s.cohort.merged5, function(x) x / sum(x))  
rar.18s.cohort.merged5 <- tax\_glom(rar.18s.cohort.merged, taxrank = "Rank5")  
rar.18s.cohort.merged5 <- transform\_sample\_counts(rar.18s.cohort.merged5, function(x) x / sum(x))  
  
rar.16s.cohort.merged4 <- tax\_glom(rar.16s.cohort.merged5, taxrank = "Rank4")  
rar.16s.cohort.merged4 <- transform\_sample\_counts(rar.16s.cohort.merged4, function(x) x / sum(x))  
rar.18s.cohort.merged4 <- tax\_glom(rar.18s.cohort.merged5, taxrank = "Rank4")  
rar.18s.cohort.merged4 <- transform\_sample\_counts(rar.18s.cohort.merged4, function(x) x / sum(x))  
  
rar.16s.cohort.merged3 <- tax\_glom(rar.16s.cohort.merged4, taxrank = "Rank3")  
rar.16s.cohort.merged3 <- transform\_sample\_counts(rar.16s.cohort.merged3, function(x) x / sum(x))  
rar.18s.cohort.merged3 <- tax\_glom(rar.18s.cohort.merged4, taxrank = "Rank3")  
rar.18s.cohort.merged3 <- transform\_sample\_counts(rar.18s.cohort.merged3, function(x) x / sum(x))  
  
# Improve taxa names  
tax\_table(rar.16s.cohort.merged5)[,"Rank5"] <- gsub("f\_\_", "", tax\_table(rar.16s.cohort.merged5)[,"Rank5"], fixed = T)  
tax\_table(rar.18s.cohort.merged5)[,"Rank5"] <- gsub("D.\*\_", "", tax\_table(rar.18s.cohort.merged5)[,"Rank5"])  
  
tax\_table(rar.16s.cohort.merged4)[,"Rank4"] <- gsub("o\_\_", "", tax\_table(rar.16s.cohort.merged4)[,"Rank4"], fixed = T)  
tax\_table(rar.18s.cohort.merged4)[,"Rank4"] <- gsub("D\_3\_\_", "", tax\_table(rar.18s.cohort.merged4)[,"Rank4"], fixed = T)  
  
tax\_table(rar.16s.cohort.merged3)[,"Rank3"] <- gsub("c\_\_", "", tax\_table(rar.16s.cohort.merged3)[,"Rank3"], fixed = T)  
tax\_table(rar.18s.cohort.merged3)[,"Rank3"] <- gsub("D\_2\_\_", "", tax\_table(rar.18s.cohort.merged3)[,"Rank3"], fixed = T)

# Pick level to use:  
rar.16s.cohort.merged <- rar.16s.cohort.merged3  
rar.18s.cohort.merged <- rar.18s.cohort.merged3  
fillrank <- "Rank3"  
  
# Also remove [brackets] from Greengenes candidate taxa  
tax\_table(rar.16s.cohort.merged)[,"Rank3"] <- gsub("[][]", "", tax\_table(rar.16s.cohort.merged)[,"Rank3"], fixed = F)  
  
  
# Take top taxa  
rar.16s.cohort.merged.top <- prune\_taxa(names(sort(taxa\_sums(rar.16s.cohort.merged), TRUE))[0:10], rar.16s.cohort.merged)  
rar.16s.cohort.merged.top  
sum(taxa\_sums(rar.16s.cohort.merged.top)) / sum(taxa\_sums(rar.16s.cohort.merged))  
  
# Which 3 taxa do we lose if we take the top 7 taxa  
#sort(taxa\_sums(rar.16s.cohort.merged.top))/nsamples(rar.16s.cohort.merged.top)  
#tax\_table(rar.16s.cohort.merged.top)[c("OTU\_20", "OTU\_12", "OTU\_13")]  
  
fig1.c <- plot\_bar(rar.16s.cohort.merged.top, x = "Time..H.", fill = fillrank, facet\_grid = ~Substrate)  
fig1.c <- fig1.c +  
 scale\_fill\_viridis(discrete = T, begin = 0, end = .9, name = "16S Taxa") +  
 labs(x = "Time (h)") +  
 scale\_x\_continuous(breaks = c(6, 24, 48)) +  
 theme(axis.text.x = element\_text(angle = 0, hjust = .5),  
 legend.justification = c("left", "top"),  
 strip.background = element\_blank())  
fig1.c

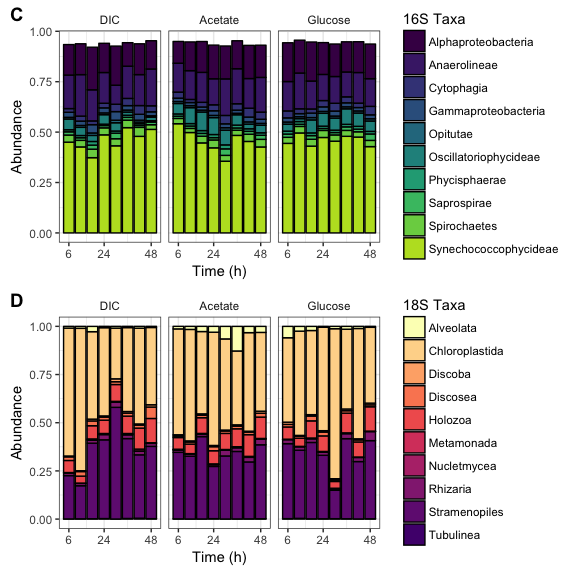
# Take top taxa  
rar.18s.cohort.merged.top <- prune\_taxa(names(sort(taxa\_sums(rar.18s.cohort.merged), TRUE))[0:10], rar.18s.cohort.merged)  
rar.18s.cohort.merged.top  
sum(taxa\_sums(rar.18s.cohort.merged.top)) / sum(taxa\_sums(rar.18s.cohort.merged))  
  
# Which 3 taxa do we lose if we take the top 7 taxa  
#sort(taxa\_sums(rar.18s.cohort.merged.top))/nsamples(rar.18s.cohort.merged.top)  
#tax\_table(rar.18s.cohort.merged.top)[c("OTU\_77", "OTU\_57", "OTU\_42")]  
  
fig1.d <- plot\_bar(rar.18s.cohort.merged.top, x = "Time..H.", fill = fillrank, facet\_grid = ~Substrate)  
fig1.d <- fig1.d +  
 scale\_fill\_viridis(discrete = T, option = "A", begin = .25, direction = -1, name = "18S Taxa") +  
 labs(x = "Time (h)") +  
 scale\_x\_continuous(breaks = c(6, 24, 48)) +  
 theme(axis.text.x = element\_text(angle = 0, hjust = .5),  
 legend.justification = c("left", "top"),  
 strip.background = element\_blank())  
fig1.d

#### Figure 1, c d

This figure could be rendered at 5.5 vs 4.8, to fit figure without scaling.

Here, I'm exporting at 6x6, with the expectation that it will be scaled down when composing the figure.

plot\_grid(fig1.c, fig1.d, align = "v", ncol = 1, labels = c("C", "D"))



# hight should match fig1-ab  
# width of both should add up to 183  
ggsave('fig1-cd.pdf', path = "./figures/", units = "mm", width = 122, height = 122, device = cairo\_pdf, scale = 1.3)

## Beta Diversity

Look for microbes that covary with carbon uptake (using the bi-plot)

Because I'm not confident of the intricacies of building ML trees from any MSA made from the 18S amplicon, I'll use Bray-Curtis dissimilarities for everything.

## Alpha diversity

This function extends phyloseq::estimate\_richness() function by implementing two evenness metrics. See this PR <https://github.com/joey711/phyloseq/pull/575>

estimate\_richness\_mod <- function(physeq, split=TRUE, measures=NULL){  
  
 if( !any(otu\_table(physeq)==1) ){  
 # Check for singletons, and then warning if they are missing.  
 # These metrics only really meaningful if singletons are included.  
 warning(  
 "The data you have provided does not have\n",  
 "any singletons. This is highly suspicious. Results of richness\n",  
 "estimates (for example) are probably unreliable, or wrong, if you have already\n",  
 "trimmed low-abundance taxa from the data.\n",  
 "\n",  
 "We recommended that you find the un-trimmed data and retry."  
 )  
 }  
   
 # If we are not splitting sample-wise, sum the species. Else, enforce orientation.  
 if( !split ){  
 OTU <- taxa\_sums(physeq)   
 } else if( split ){  
 OTU <- as(otu\_table(physeq), "matrix")  
 if( taxa\_are\_rows(physeq) ){ OTU <- t(OTU) }  
 }  
   
 # Define renaming vector:  
 renamevec = c("Observed", "Chao1", "ACE", "Shannon", "Pielou", "Simpson", "InvSimpson", "SimpsonE", "Fisher")  
 names(renamevec) <- c("S.obs", "S.chao1", "S.ACE", "shannon", "pielou", "simpson", "invsimpson", "simpsone", "fisher")  
 # If measures was not explicitly provided (is NULL), set to all supported methods  
 if( is.null(measures) ){  
 measures = as.character(renamevec)  
 }  
 # Rename measures if they are in the old-style  
 if( any(measures %in% names(renamevec)) ){  
 measures[measures %in% names(renamevec)] <- renamevec[names(renamevec) %in% measures]  
 }  
   
 # Stop with error if no measures are supported  
 if( !any(measures %in% renamevec) ){  
 stop("None of the `measures` you provided are supported. Try default `NULL` instead.")  
 }  
   
 # Initialize to NULL  
 outlist = vector("list")  
 # Some standard diversity indices  
 estimRmeas = c("Chao1", "Observed", "ACE")  
 if( any(estimRmeas %in% measures) ){   
 outlist <- c(outlist, list(t(data.frame(estimateR(OTU)))))  
 }  
 if( "Shannon" %in% measures ){  
 outlist <- c(outlist, list(shannon = diversity(OTU, index="shannon")))  
 }  
 if( "Pielou" %in% measures){  
 #print("Starting Pielou")  
 outlist <- c(outlist, list(pielou = diversity(OTU, index = "shannon")/log(estimateR(OTU)["S.obs",])))  
 }  
 if( "Simpson" %in% measures ){  
 outlist <- c(outlist, list(simpson = diversity(OTU, index="simpson")))  
 }  
 if( "InvSimpson" %in% measures ){  
 outlist <- c(outlist, list(invsimpson = diversity(OTU, index="invsimpson")))  
 }  
 if( "SimpsonE" %in% measures ){  
 #print("Starting SimpsonE")  
 outlist <- c(outlist, list(simpsone = diversity(OTU, index="invsimpson")/estimateR(OTU)["S.obs",]))  
 }  
 if( "Fisher" %in% measures ){  
 fisher = tryCatch(fisher.alpha(OTU, se=TRUE),   
 warning=function(w){  
 warning("phyloseq::estimate\_richness: Warning in fisher.alpha(). See `?fisher.fit` or ?`fisher.alpha`. Treat fisher results with caution")  
 suppressWarnings(fisher.alpha(OTU, se=TRUE)[, c("alpha", "se")])  
 }  
 )  
 if(!is.null(dim(fisher))){  
 colnames(fisher)[1:2] <- c("Fisher", "se.fisher")  
 outlist <- c(outlist, list(fisher))  
 } else {  
 outlist <- c(outlist, Fisher=list(fisher))  
 }  
 }  
 out = do.call("cbind", outlist)  
 # Rename columns per renamevec  
 namechange = intersect(colnames(out), names(renamevec))  
 colnames(out)[colnames(out) %in% namechange] <- renamevec[namechange]  
 # Final prune to just those columns related to "measures". Use grep.  
 colkeep = sapply(paste0("(se\\.){0,}", measures), grep, colnames(out), ignore.case=TRUE)  
 out = out[, sort(unique(unlist(colkeep))), drop=FALSE]  
 # Make sure that you return a data.frame for reliable performance.  
 out <- as.data.frame(out)  
 return(out)  
}

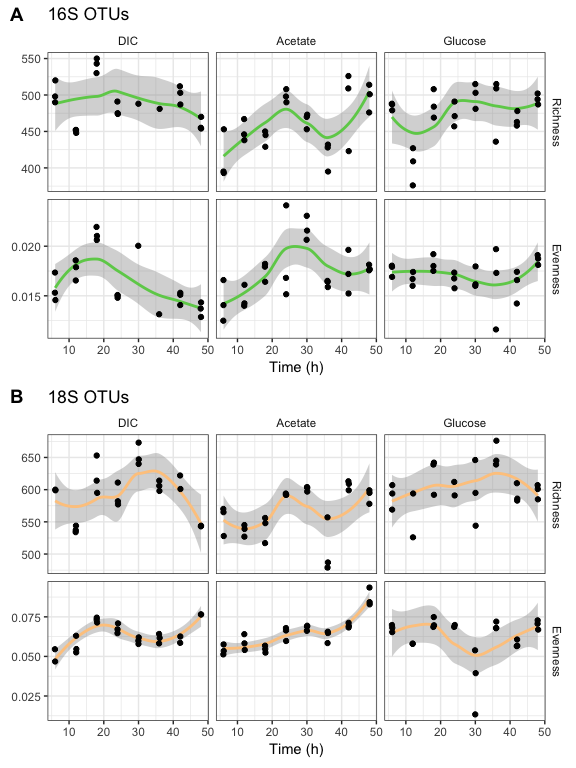
Now, let's use our modified richness function.

metrics <- c("Observed", "SimpsonE")  
rich.16s <- estimate\_richness\_mod(rar.16s.cohort, measures = metrics)  
rich.18s <- estimate\_richness\_mod(rar.18s.cohort, measures = metrics)  
  
# Rename for graphing  
graphnames <- c("Richness", "Evenness")  
names(rich.16s) <- graphnames  
names(rich.18s) <- graphnames  
  
  
# merge richness with metadata  
DF.16s <- merge(rich.16s, sample\_data(rar.16s.cohort), by = 0)  
DF.18s <- merge(rich.18s, sample\_data(rar.18s.cohort), by = 0)  
  
# melt for graphing  
reshapevars <- c("Richness", "Evenness")  
mdf.16s = melt(DF.16s, measure.vars = graphnames)  
mdf.18s = melt(DF.18s, measure.vars = graphnames)  
#head(mdf.16s)  
  
mdf.16s$Time..H.num <- as.numeric(mdf.16s$Time..H.)  
mdf.18s$Time..H.num <- as.numeric(mdf.18s$Time..H.)

### Figure 2: Alpha over time

alpha.16s.alt <- ggplot(mdf.16s, aes(Time..H., value))  
alpha.16s.alt <- alpha.16s.alt +   
 geom\_smooth(color = viridis(n = 10)[8]) +  
 geom\_jitter(width = 0.1) +  
 facet\_grid(facets = variable~Substrate, scales = "free\_y") +  
 labs(x = "Time (h)", title = "16S OTUs") +  
 theme(legend.position = "none",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 #, plot.margin=unit(c(5,5,-25,5), units = "pt")  
 )  
#alpha.16s.alt  
  
alpha.18s.alt <- ggplot(mdf.18s, aes(Time..H., value))  
alpha.18s.alt <- alpha.18s.alt +   
 geom\_smooth(color = magma(n = 10)[9]) +  
 geom\_jitter(width = 0.1) +  
 facet\_grid(facets = variable~Substrate, scales = "free\_y") +  
 labs(x = "Time (h)", title = "18S OTUs") +  
 theme(legend.position = "none",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 #axis.text.y = element\_blank()  
 #, plot.margin=unit(c(5,5,-25,5), units = "pt")  
 )  
#alpha.18s.alt  
  
plot\_grid(alpha.16s.alt, alpha.18s.alt,  
 labels = c("A", "B"),  
 align = "v", ncol = 1)

## `geom\_smooth()` using method = 'loess'  
## `geom\_smooth()` using method = 'loess'



ggsave('fig2-1col.pdf', path = "./figures/", units = "mm", width = 89, height = 120, device = cairo\_pdf, scale = 1.6)

#### Alpha over time stat test

Because n within each group is small (and sometimes missing reps), let's test for equal variance.

Then, use a classic anova() test on each combination of Substrate and alpha.

# For testing   
df <- filter(mdf.16s, Substrate == "Glucose", variable == "Evenness")  
#table(df$timepoint)  
  
runLevene <- function(df){  
 t <- leveneTest(value ~ timepoint, data = df) # hardcoded to my data  
 t  
 return(t$`Pr(>F)`[1])  
}  
  
#by(mdf.16s, INDICES = list(mdf.16s$Substrate, mdf.16s$variable), runLevene)  
#by(mdf.18s, INDICES = list(mdf.18s$Substrate, mdf.18s$variable), runLevene)  
  
# Smallest number if for 18S, Glucose:Evenness with p = 0.126, so we are good to go!  
  
runANOVA <- function(df){  
 res <- lm(value ~ timepoint, data = df) # hardcoded to my data  
 a <- anova(res)  
 a  
 a$`Pr(>F)`[1]  
 #summary(res)  
 #TukeyHSD(aov(res), "timepoint") # we don't need to preform a post-hoc test  
 return(a$`Pr(>F)`[1])  
}  
  
# Outputs  
  
kable(  
 tibble(  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 `ANOVA p-value` = c(  
 by(mdf.16s, INDICES = list(mdf.16s$Substrate, mdf.16s$variable), runANOVA)  
 )  
)  
)

|  |  |  |
| --- | --- | --- |
| Substrate | variable | ANOVA p-value |
| DIC | Richness | 0.0000086 |
| Acetate | Richness | 0.0041506 |
| Glucose | Richness | 0.0016277 |
| DIC | Evenness | 0.0000028 |
| Acetate | Evenness | 0.0152349 |
| Glucose | Evenness | 0.5163856 |

kable(  
 tibble(  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 `ANOVA p-value` = c(  
 by(mdf.18s, INDICES = list(mdf.18s$Substrate, mdf.18s$variable), runANOVA)  
 )  
)  
)

|  |  |  |
| --- | --- | --- |
| Substrate | variable | ANOVA p-value |
| DIC | Richness | 0.0000250 |
| Acetate | Richness | 0.0000647 |
| Glucose | Richness | 0.0791450 |
| DIC | Evenness | 0.0000304 |
| Acetate | Evenness | 0.0000003 |
| Glucose | Evenness | 0.0014949 |

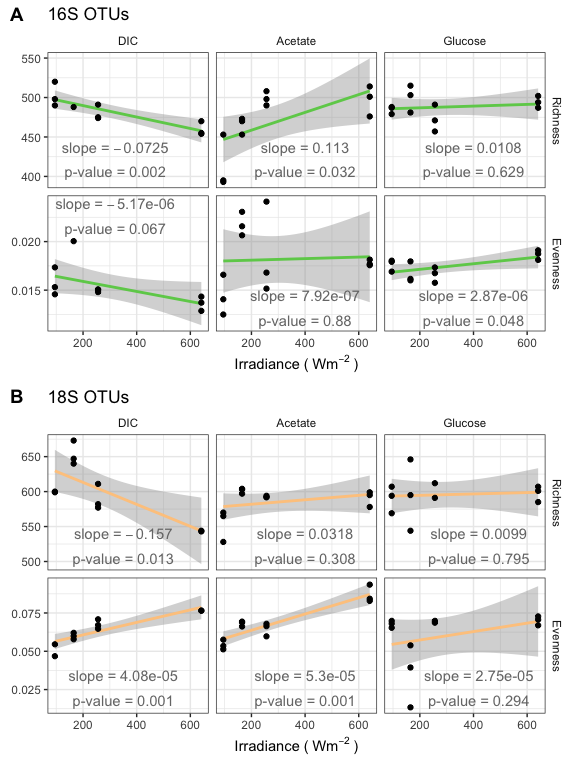
### Figure 3: Alpha over Irradiance

This set of graph shows alpha ~ Light only for during the day where there is light. (Many samples were taking during night, and these have been dropped.)

# color code by time, SOLR on the x axis  
  
# See above  
lm\_eqn\_solar = function(df){  
 m = lm(value ~ SOLR.W.m.m, df) # Hardcoded to my data  
 m  
 summary(m)$coefficients[8]  
 if(summary(m)$coefficients[8] < 0.001) {  
 outputp <- "0.001"  
 }else{  
 outputp <- round(summary(m)$coefficients[8], digits = 3)  
 }  
 eq <- substitute(  
 atop("slope ="~slope, "p-value ="~pr), # Two lines  
 #"slope ="~slope~","~~R^2~"="~r2, # One line  
 list(slope = signif(summary(m)$coefficients[2], digits = 3),  
 pr = outputp)  
 )  
 as.character(as.expression(eq))   
}  
  
  
mdf.16s.light <- subset(mdf.16s, SOLR.W.m.m > 0)  
mdf.18s.light <- subset(mdf.18s, SOLR.W.m.m > 0)  
  
eqns <- by(mdf.16s.light, INDICES = list(mdf.16s.light$Substrate, mdf.16s.light$variable), lm\_eqn\_solar)  
eqns  
  
df2 <- data.frame(eq = c(eqns),  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 graphx = rep(c(320, 400, 400), 2),   
 graphy = c(420, 420, 420, .0225, .013, .013))  
df2  
  
  
alpha.16s.light.only <- ggplot(mdf.16s.light, aes(SOLR.W.m.m, value))  
alpha.16s.light.only <- alpha.16s.light.only +   
 #geom\_boxplot(color = "gray", outlier.size = 0) +  
 geom\_smooth(method = "lm", color = viridis(n = 10)[8]) +  
 geom\_jitter(width = 0) +  
 facet\_grid(facets = variable~Substrate, scales = "free\_y") + #, switch = "y") +  
 labs(title = "16S OTUs", x = expression("Irradiance ("~ Wm^{-2}~ ")")) +  
 geom\_text(data = df2, aes(x = graphx, y = graphy, label = eq), color = '#777777', parse = TRUE) +  
 #v.c + # also add color = Time..H.num to aes() to use this  
 theme(legend.position = "right",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 #, plot.margin=unit(c(5,5,-25,5), units = "pt")  
 #strip.text.y = element\_text(hjust = 2, vjust = .5) # for aligning the facet labs()  
 )  
alpha.16s.light.only

eqns <- by(mdf.18s.light, INDICES = list(mdf.18s.light$Substrate, mdf.18s.light$variable), lm\_eqn\_solar)  
eqns  
  
df2 <- data.frame(eq = c(eqns),  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 graphx = rep(c(350, 400, 420), 2),   
 graphy = c(520, 520, 520, .025, .025, .025))  
df2  
  
alpha.18s.light.only <- ggplot(mdf.18s.light, aes(SOLR.W.m.m, value))  
alpha.18s.light.only <- alpha.18s.light.only +   
 geom\_smooth(method = "lm", color = magma(n = 10)[9]) +  
 geom\_jitter(width = 0) +  
 facet\_grid(facets = variable~Substrate, scales = "free\_y") +  
 labs(title = "18S OTUs", x = expression("Irradiance ("~ Wm^{-2}~ ")")) +  
 geom\_text(data = df2, aes(x = graphx, y = graphy, label = eq), color = '#777777', parse = TRUE) +  
 theme(legend.position = "right",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 )  
alpha.18s.light.only

plot\_grid(alpha.16s.light.only, alpha.18s.light.only,  
 labels = c("A", "B"),  
 align = "v", ncol = 1)



#ggsave('fig3-1.5col.pdf', path = "./figures/", units = "mm", width = 120, height = 160, device = cairo\_pdf, scale = 1.3)  
ggsave('fig3-1col.pdf', path = "./figures/", units = "mm", width = 89, height = 120, device = cairo\_pdf, scale = 1.6)

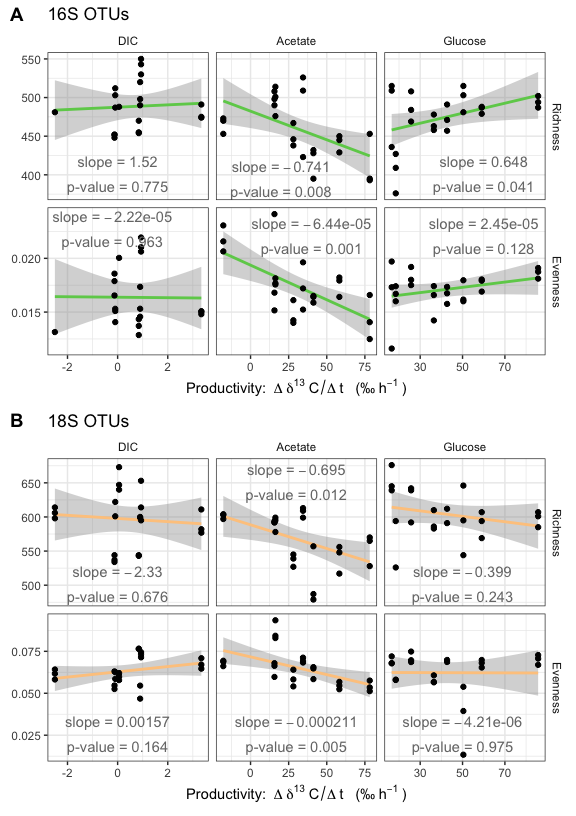
### Figure 4: Alpha over productivity

# Let's see if we can add slopes and p-values to these graphs  
# See https://stackoverflow.com/questions/17022553/adding-r2-on-graph-with-facets  
# Example data for testing function  
df <- mdf.16s[mdf.16s$Substrate == "Acetate" & mdf.16s$variable == "Richness", ]  
dim(df)  
lm\_eqn\_deltac = function(df){  
 m = lm(value ~ fin\_diff\_Ave\_d13C.organic, df) # Hardcoded to my data  
 m  
 summary(m)$coefficients[2]  
 eq <- substitute(  
 atop("slope ="~slope, "p-value ="~pr), # Two lines  
 #"slope ="~slope~","~~R^2~"="~r2, # One line  
 list(slope = signif(summary(m)$coefficients[2], digits = 3),  
 pr = round(summary(m)$coefficients[8], digits = 3))  
 )  
 as.character(as.expression(eq))   
}  
  
# Make a matching data frame with these calculated values  
# This is based on the stackoverflow answer, and on the Bile Acids graphs  
# See Lactobacillus acidophilus disrupts multispecies synthesis of a secondary bile acid ursocholate  
# Hans C. Bernstein, PNNL, 2017  
  
eqns <- by(mdf.16s, INDICES = list(mdf.16s$Substrate, mdf.16s$variable), lm\_eqn\_deltac)  
#eqns  
  
df2 <- data.frame(eq = c(eqns),  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 graphx = c(0, 20, 60, -0.21, 40, 60),   
 graphy = c(400, 393, 400, .0226, .022, .022))  
#df2  
  
alpha.16s.1 <- ggplot(mdf.16s, aes(fin\_diff\_Ave\_d13C.organic, value))  
alpha.16s.1 <- alpha.16s.1 +   
 #geom\_boxplot(color = "gray", outlier.size = 0) +  
 geom\_smooth(method = "lm", color = viridis(n = 10)[8]) +  
 geom\_jitter(width = 0) +  
 facet\_grid(facets = variable~Substrate, scales = "free") +  
 labs(title = "16S OTUs", x = expression("Productivity: " ~ Delta~delta^13~C / Delta ~t ~" (‰"~ h^-1~")")) +  
 #v.c + # also add color = Time..H.num to aes() to use this  
 geom\_text(data = df2, aes(x = graphx, y = graphy, label = eq), color = '#777777', parse = TRUE) +  
 theme(legend.position = "right",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 #, plot.margin=unit(c(5,5,-25,5), units = "pt")  
 #strip.text.y = element\_text(hjust = 2, vjust = .5) # for aligning the facet labs()  
 )  
alpha.16s.1

eqns <- by(mdf.18s, INDICES = list(mdf.18s$Substrate, mdf.18s$variable), lm\_eqn\_deltac)  
#eqns  
  
df2 <- data.frame(eq = c(eqns),  
 Substrate = c("DIC", "Acetate", "Glucose",  
 "DIC", "Acetate", "Glucose"),  
 variable = c("Richness", "Richness", "Richness",  
 "Evenness", "Evenness", "Evenness"),  
 graphx = rep(c(0, 30, 50), 2),   
 graphy = c(500, 650, 500, .025, .025, .025))  
#df2  
  
  
alpha.18s.1 <- ggplot(mdf.18s, aes(fin\_diff\_Ave\_d13C.organic, value))  
alpha.18s.1 <- alpha.18s.1 +   
 geom\_smooth(method = "lm", color = magma(n = 10)[9]) +  
 geom\_jitter(width = 0) +  
 facet\_grid(facets = variable~Substrate, scales = "free") +  
 labs(title = "18S OTUs", x = expression("Productivity: " ~ Delta~delta^13~C / Delta ~t ~" (‰"~ h^-1~")")) +  
 #v.c + # also add color = Time..H.num to aes() to use this  
 geom\_text(data = df2, aes(x = graphx, y = graphy, label = eq), color = '#777777', parse = TRUE) +  
 theme(legend.position = "right",  
 strip.background = element\_blank(),  
 axis.title.y = element\_blank()  
 )  
alpha.18s.1

#alpha.18s.1 + geom\_smooth(method = "lm", aes(color = Substrate)) + scale\_color\_brewer(palette = "Set2")

plot\_grid(alpha.16s.1, alpha.18s.1,  
 labels = c("A", "B"),  
 align = "v", ncol = 1)



#ggsave('fig4-1.5col.pdf', path = "./figures/", units = "mm", width = 120, height = 160, device = cairo\_pdf, scale = 1.3)  
ggsave('fig4-1col.pdf', path = "./figures/", units = "mm", width = 89, height = 120, device = cairo\_pdf, scale = 1.6)

##### Other alpha graphs..

...where made, but are not included here.

## RA graphs for bioplot

The same RA graphs used in the barplots will be used *before* the merging of samples. Here, we regenerate these from the rarefied cohorts.

# Glom OTUs and transform to RA  
glom.16s <- tax\_glom(rar.16s.cohort, taxrank = "Rank5")  
glom.16s <- transform\_sample\_counts(glom.16s, function(x) x / sum(x))  
  
# Take top taxa  
glom.16s.top <- prune\_taxa(names(sort(taxa\_sums(glom.16s), TRUE))[0:10], glom.16s)  
glom.16s.top  
sum(taxa\_sums(glom.16s.top)) / sum(taxa\_sums(glom.16s))  
  
# plot!  
plot\_bar(glom.16s.top, x = "Time..H.", fill = "Rank5", facet\_grid = ~Substrate)

tax\_table(glom.16s.top)  
  
  
## 18S  
glom.18s <- tax\_glom(rar.18s.cohort, taxrank = "Rank5")  
glom.18s <- transform\_sample\_counts(glom.18s, function(x) x / sum(x))  
  
glom.18s.top <- prune\_taxa(names(sort(taxa\_sums(glom.18s), TRUE))[0:10], glom.18s)  
glom.18s.top  
sum(taxa\_sums(glom.18s.top)) / sum(taxa\_sums(glom.18s))  
  
glom.18s.top  
sum(taxa\_sums(glom.18s.top)) / sum(taxa\_sums(glom.18s))  
  
plot\_bar(glom.18s.top, x = "Time..H.", fill = "Rank5", facet\_grid = ~Substrate)

Goal: for each sample (row) of the metadata table, I want to add columns with the abundance of microbes at a given taxonomy level.

# 16S  
glom.16s.top.table <- as.data.frame(otu\_table(glom.16s.top))  
row.names(glom.16s.top.table) <- as.data.frame(tax\_table(glom.16s.top))$Rank5  
  
glom.16s.top.meta <- cbind((sample\_data(glom.16s.top)), t(glom.16s.top.table))  
head(glom.16s.top.meta)  
  
# Add these taxa back into a new phyloseq object   
rar.16s.w\_tax <- rar.16s.cohort  
sample\_data(rar.16s.w\_tax) <- glom.16s.top.meta  
  
  
# 18S  
glom.18s.top.table <- as.data.frame(otu\_table(glom.18s.top))  
row.names(glom.18s.top.table) <- as.data.frame(tax\_table(glom.18s.top))$Rank5  
  
glom.18s.top.meta <- cbind((sample\_data(glom.18s.top)), t(glom.18s.top.table))  
head(glom.18s.top.meta)  
  
rar.18s.w\_tax <- rar.18s.cohort  
sample\_data(rar.18s.w\_tax) <- glom.18s.top.meta  
  
# Test graphs  
ggplot(glom.16s.top.meta, aes(Time..H., f\_\_Pseudanabaenaceae)) + geom\_point() + facet\_grid(~Substrate)

ggplot(glom.18s.top.meta, aes(Time..H., D\_4\_\_Vampyrellidae)) + geom\_point() + facet\_grid(~Substrate)

Use the phyloseq wrapper of vegan::capscale()

dist.16s <- phyloseq::distance(rar.16s.w\_tax, method = "bray", type = "samples")  
dist.18s <- phyloseq::distance(rar.18s.w\_tax, method = "bray", type = "samples")  
  
# basic plot to make sure this works:  
# Test 16s names  
plot(capscale(dist.16s ~ Time..H. + SOLR.W.m.m +   
 Substrate   
 #+ Ave\_d13C.organic + fin\_diff\_Ave\_d13C.organic +  
 # f\_\_Rhodospirillaceae + f\_\_Phormidiaceae + f\_\_Spirochaetaceae + f\_\_Saprospiraceae + f\_\_A4b + f\_\_Rhodobacteraceae + f\_\_Hyphomonadaceae + f\_\_Pseudanabaenaceae + f\_\_Cyclobacteriaceae + f\_\_Anaerolineaceae  
 ,  
 data.frame(sample\_data(rar.16s.w\_tax))))

# Names work great! Let's test new productivity vectors  
plot(capscale(dist.16s ~ Time..H. + SOLR.W.m.m +   
 pp + hpa + hpg,  
 data.frame(sample\_data(rar.16s.w\_tax))), col=sample\_data(rar.16s.w\_tax)$Substrate)

plot(capscale(dist.16s ~ Time..H. + SOLR.W.m.m +   
 fin\_diff\_Ave\_d13C.organic + Substrate,  
 data.frame(sample\_data(rar.16s.w\_tax))),  
 #col=c("red","blue","green")[sample\_data(rar.16s.w\_tax)$Substrate]  
 col="blue"  
 )

# Ordinate  
#cap.16s = ordinate(rar.16s.w\_tax, "CAP", dist.16s,  
# ~ Time..H. + SOLR.W.m.m +   
# Substrate + fin\_diff\_Ave\_d13C.organic +  
# f\_\_Rhodospirillaceae + f\_\_Phormidiaceae + f\_\_Spirochaetaceae + f\_\_Saprospiraceae + f\_\_A4b + f\_\_Rhodobacteraceae + f\_\_Hyphomonadaceae + f\_\_Pseudanabaenaceae + f\_\_Cyclobacteriaceae + f\_\_Anaerolineaceae)  
  
cap.16s = ordinate(rar.16s.w\_tax, "CAP", dist.16s,  
 ~ Time..H. + SOLR.W.m.m +   
 pp + hpa + hpg +   
 f\_\_Rhodospirillaceae + f\_\_Phormidiaceae + f\_\_Spirochaetaceae + f\_\_Saprospiraceae + f\_\_A4b + f\_\_Rhodobacteraceae + f\_\_Hyphomonadaceae + f\_\_Pseudanabaenaceae + f\_\_Cyclobacteriaceae + f\_\_Anaerolineaceae)  
  
# Two alternative ordinations. These show that choise of carbon variable does not change ordination much  
#cap.16s = ordinate(rar.16s.w\_tax, "CAP", dist.16s,  
# ~ Time..H. + SOLR.W.m.m +   
# fin\_diff\_Ave\_d13C.organic +   
# f\_\_Rhodospirillaceae + f\_\_Phormidiaceae + f\_\_Spirochaetaceae + f\_\_Saprospiraceae + f\_\_A4b + f\_\_Rhodobacteraceae + f\_\_Hyphomonadaceae + f\_\_Pseudanabaenaceae + f\_\_Cyclobacteriaceae + f\_\_Anaerolineaceae)  
  
#cap.16s = ordinate(rar.16s.w\_tax, "CAP", dist.16s,  
# ~ Time..H. + SOLR.W.m.m +   
# #fin\_diff\_Ave\_d13C.organic +   
# f\_\_Rhodospirillaceae + f\_\_Phormidiaceae + f\_\_Spirochaetaceae + f\_\_Saprospiraceae + f\_\_A4b + f\_\_Rhodobacteraceae + f\_\_Hyphomonadaceae + f\_\_Pseudanabaenaceae + f\_\_Cyclobacteriaceae + f\_\_Anaerolineaceae)  
  
# 18S  
cap.18s = ordinate(rar.18s.w\_tax, "CAP", dist.18s,  
 ~ Time..H. + SOLR.W.m.m +  
 pp + hpa + hpg +   
 D\_4\_\_Chrysophyceae + D\_4\_\_Thraustochytriaceae + D\_5\_\_Codonosigidae + D\_4\_\_Blastocystis + D\_9\_\_Podocopa + D\_4\_\_uncultured.eukaryote + D\_6\_\_Bacillariophyceae + D\_10\_\_Solanales + D\_11\_\_Alismatales + D\_4\_\_Vampyrellidae)  
  
# Add length vector  
cap16s.v <- data.frame(cap.16s$CCA$biplot)  
# The find the length (magnitude) of the vbectors in the first two dimentions  
cap16s.v$length1and2 <-sqrt(cap16s.v$CAP1^2+cap16s.v$CAP2^2)  
#cap16s.v[order(cap16s.v$length1and2, decreasing = T),]  
  
cap18s.v <- data.frame(cap.18s$CCA$biplot)  
cap18s.v$length1and2 <-sqrt(cap18s.v$CAP1^2+cap18s.v$CAP2^2)  
  
#cap16s.v <- cap16s.v[order(cap16s.v$length1and2),]   
#cap18s.v <- cap18s.v[order(length1and2),]   
  
  
# This is a great place to improve names.  
row.names(cap16s.v) <- gsub("f\_\_", "", row.names(cap16s.v), fixed = T)  
row.names(cap16s.v)[1:2] <- c("Time (h)", "Irradiance")  
row.names(cap16s.v)[3:5] <- c("NPP", "NHPa", "NHPg")  
# Improve lable for Anaerolineae   
row.names(cap16s.v)[10] <- "Anaerolineae A4b"  
# c\_\_Anaerolineae o\_\_SBR1031 f\_\_A4b  
  
row.names(cap18s.v) <- gsub("D.\*\_", "", row.names(cap18s.v))  
row.names(cap18s.v)[1:2] <- c("Time (h)", "Irradiance")  
row.names(cap18s.v)[3:5] <- c("NPP", "NHPa", "NHPg")  
# Fix name of Solanales (of order of flowering plants). The taxa database (SILVA ) places this in Chloroplastida,  
# so we will report this at the class level, instead of the overly specific family level.   
row.names(cap18s.v)[13] <- "Chloroplastida"  
  
# Add vector categories   
tmp.vector.categories <- c("Environment", "Environment",  
 "Productivity", "Productivity", "Productivity",  
 rep("Taxa", nrow(cap16s.v)-5))  
length(tmp.vector.categories)  
cap16s.v$category <- tmp.vector.categories  
cap18s.v$category <- tmp.vector.categories  
  
  
# compare relative lengths of vectors  
ggplot(cap16s.v, aes(x = category, y = length1and2, color = category)) + geom\_boxplot() + geom\_point()

ggplot(cap18s.v, aes(x = category, y = length1and2, color = category)) + geom\_boxplot() + geom\_point()

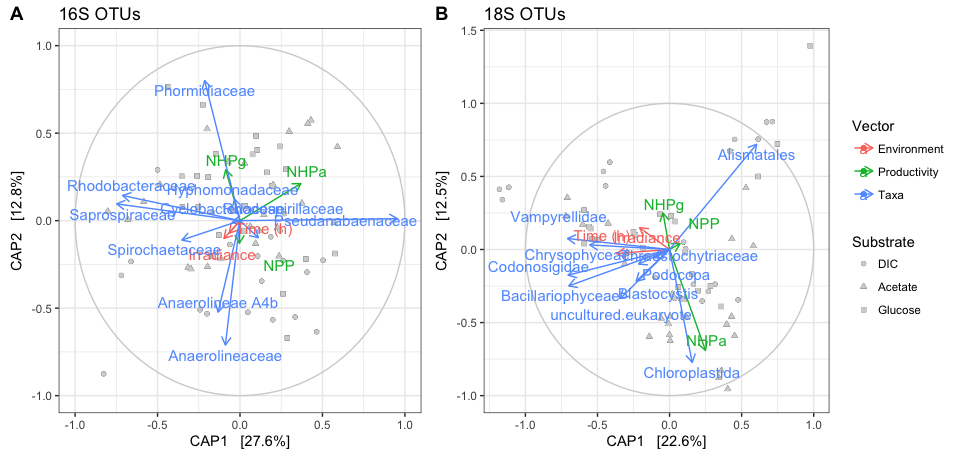
# environment is one of the least important factors  
  
# People like circles around the db-RDA vectors. So, here, have a circle:  
# From http://stackoverflow.com/questions/6862742/draw-a-circle-with-ggplot2   
circleFun <- function(center = c(0,0), radius = 1, npoints = 100){  
 r = radius  
 tt <- seq(0, 2\*pi, length.out = npoints)  
 xx <- center[1] + r \* cos(tt)  
 yy <- center[2] + r \* sin(tt)  
 return(data.frame(x = xx, y = yy))  
}  
#geom\_path will do open circles, geom\_polygon will do filled circles  
#plot + geom\_path(aes(x = x, y=y), data = circleFun(c(0,0), 1, 100), color="grey")

The bi-plot:

# Without vectors, showing off substrate clustering  
plot <- plot\_ordination(rar.16s.w\_tax, cap.16s, "samples", shape="Substrate", color = "Substrate")  
#plot +  
# geom\_path(aes(x = x, y = y, shape = NULL), data = circleFun(c(0,0), 1, 100), color="lightgrey") +  
# scale\_color\_brewer(palette = "Set2")  
  
  
plot <- plot\_ordination(rar.16s.w\_tax, cap.16s, "samples", shape = "Substrate")  
plot <- plot + geom\_point(color = "lightgrey") +  
 geom\_path(aes(x = x, y=y, shape = NULL), data = circleFun(c(0,0), 1, 100), color="lightgrey") +   
 geom\_segment(data = cap16s.v, aes(x = 0, xend = CAP1, y = 0, yend = CAP2, color = category, shape = NULL),  
 arrow = arrow(length = unit(0.25, "cm"))) +  
 geom\_text\_repel(segment.size = 0, max.iter = 2000, box.padding = unit(0.2, "lines"),  
 data = cap16s.v, aes(x=CAP1, y=CAP2, label=row.names(cap16s.v), color = category, shape = NULL), size = 4  
 #,color = cap16s.v$color, color="black"  
 ) +  
 theme(legend.position = "none") +  
 labs(title = "16S OTUs")  
plot

plot.18s <- plot\_ordination(rar.18s.w\_tax, cap.18s, "samples", shape = "Substrate")  
plot.18s <- plot.18s + geom\_point(color = "lightgrey") +  
 geom\_path(aes(x = x, y=y, shape = NULL), data = circleFun(c(0,0), 1, 100), color="lightgrey") +   
 geom\_segment(data = cap18s.v, aes(x = 0, xend = CAP1, y = 0, yend = CAP2, color = category, shape = NULL),  
 arrow = arrow(length = unit(0.25, "cm"))) +  
 geom\_text\_repel(segment.size = 0, max.iter = 2000, box.padding = unit(0.2, "lines"), data = cap18s.v,   
 aes(x=CAP1, y=CAP2, label=row.names(cap18s.v), color = category, shape = NULL), size = 4  
 #,color = cap16s.v$color, color="black"  
 ) +  
 labs(title = "18S OTUs", color = "Vector")  
plot.18s

plot\_grid(plot, plot.18s,  
 labels = c("A", "B"),  
 align = "h", nrow = 1, rel\_widths = c(1, 1.25))



ggsave('fig5-2col.pdf', path = "./figures/", units = "mm", width = 183, height = 84, device = cairo\_pdf, scale = 1.3)

# Save and Knit the document early.  
knitr::knit\_exit(append = F)

FALSE