Resuscitation of microbial seed banks

Venus Kuo and Jay T. Lennon

26 September, 2019

Objective

Analysis of plant and microbial data from soils exposed to recombinantly produced resuscitation promoting factor (Rpf)

Set working environment and load packages

```r
# Clear and set working directory
rm(list = ls())
setwd("~/GitHub/BrassicaRpf/data")

# Require and/or install packages
package.list <- c('vegan', 'nlme' ,'data.table', 'plyr', 'pander', 'reshape', 'grid', 'png', 'car', 'bbr
for (package in package.list) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package)
    library(package, character.only=T)
  } }

# Load sem function
sem <- function(x, ...){sd(x, na.rm = TRUE)/sqrt(length(na.omit(x)))}

# Load t-test custom functions #
ttest <- function(reg, coefnum, val){
  co <- coef(summary(reg))
  tstat <- (co[coefnum,1]-val)/co[coefnum,2]
  pstat <- 2 * pt(abs(tstat), reg$df.residual, lower.tail = FALSE)
  return(list = c(t = tstat, df = reg$df.residual, p =  pstat))
}
```

1) Rpf effects on Brassica rapa traits

```r
# Load plant trait data
plant.data <- read.delim("~/GitHub/BrassicaRpf/data/plantTrait.txt", sep = ",", header = TRUE)
# Remove soil sterilization treatment from dataset #
plant.data <- subset(plant.data, soil == "Live")

# Subseting data for relevant information
data.sub <- subset(plant.data, select=(c(Treatment, abiomass, bbiomass)))
# Combining the above and belowground biomass #
data.m <- melt(data.sub)
```

```
## Using Treatment as id variables
```

```
# Above-ground shoot biomass
# Selecting shoot biomass data for plotting #
shoot.rpf <- data.m[ which(data.m$Treatment == "Rpf+" & data.m$variable == "abiomass"),]
shoot.con <- data.m[ which(data.m$Treatment == "Rpf-" & data.m$variable == "abiomass"),]
# Generating shoot biomass mean and sem table #
shoot.mean <- aggregate(plant.data$abiomass ~ Treatment, plant.data, mean)
shoot.sem <- aggregate(plant.data$abiomass ~ Treatment, plant.data, sem)
shoot.sem.LL <- shoot.mean[2] + shoot.sem[2]
shoot.sem.UL <- shoot.mean[2] - shoot.sem[2]
shoot.table <- data.frame(shoot.mean[1], shoot.mean[2], shoot.sem[2],
          shoot.sem.LL[1], shoot.sem.UL[1])
colnames(shoot.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
shoot.table <- shoot.table[order(shoot.table$mean),]

# Below-ground root biomass
# Selecting root biomass data for plotting #
root.rpf <- data.m[ which(data.m$Treatment == "Rpf+" & data.m$variable == "bbiomass"),]
root.con <- data.m[ which(data.m$Treatment == "Rpf-" & data.m$variable == "bbiomass"),]
# Generating root biomass  mean and sem table #
root.mean <- aggregate(plant.data$bbiomass ~ Treatment, plant.data, mean)
root.sem <- aggregate(plant.data$bbiomass ~ Treatment, plant.data, sem)
root.sem.LL <- root.mean[2] + root.sem[2]
root.sem.UL <- root.mean[2] - root.sem[2]
root.table <- data.frame(root.mean[1], root.mean[2], root.sem[2],
          root.sem.LL[1], root.sem.UL[1])
colnames(root.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
root.table <- root.table[order(root.table$mean),]

# Flower number per plant
# Flower count data for plotting #
flower.rpf <- plant.data[ which(plant.data$Treatment == "Rpf+"),]
flower.con <- plant.data[ which(plant.data$Treatment == "Rpf-"),]
# Generate flower count data table #
flower.mean <- aggregate(plant.data$flower.count ~ Treatment, plant.data, mean)
flower.sem <- aggregate(plant.data$flower.count ~ Treatment, plant.data, sem)
flower.sem.LL <- flower.mean[2] + flower.sem[2]
flower.sem.UL <- flower.mean[2] - flower.sem[2]
flower.table <- data.frame(flower.mean[1], flower.mean[2], flower.sem[2],
          flower.sem.LL[1], flower.sem.UL[1])
colnames(flower.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
flower.table <- flower.table[order(flower.table$mean),]


# Generating 3 panel figure showing Rpf effects on plant traits #
png(filename="../figures/Figure2-PlantTraits.png",
    width = 1200, height = 800, res = 96*2)
par(oma=c(7,3,7,1), mar=c(2,3,3,3.5), mfrow=c(1,3))

# Plotting first panel for shoot biomass #
shoot.fig <- plot(jitter(rep(1, length(shoot.con$value)), amount = 0.1), shoot.con$value,
      ylim = c(0, 1.5), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(shoot.rpf$value)), amount = 0.1), shoot.rpf$value, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean value point #
points(1, mean(shoot.con$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(shoot.rpf$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y-axis label #
mtext(expression('Shoot biomass (g)'), side = 2,
      outer = FALSE, cex = 1.25, line = 2.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("0", "0.5", "1.0", "1.5"), at = c(0, 0.5, 1.0, 1.5))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at=c(0, 0.5, 1.0, 1.5), labels = F, tck = -0.05)
axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))
axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Adding SEM bars #
arrows(x0 = c(2,1), y0 = shoot.table$mean, y1 = shoot.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(2,1), y0 = shoot.table$mean, y1 = shoot.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label #
text(0.65,1.45 ,labels="A", col="black", cex=2)

# p-value label
mtext(text = expression(italic("P")~" = 0.017") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 8"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)

# Plotting second panel for root biomass #
root.fig <- plot(jitter(rep(1, length(root.con$value)), amount = 0.1), root.con$value,
      ylim = c(0, 1.5), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(root.rpf$value)), amount = 0.1), root.rpf$value, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean circle #
points(1, mean(root.con$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
```

```
points(2, mean(root.rpf$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y-axis label #
mtext(expression('Root biomass (g)'), side = 2,
      outer = FALSE, cex = 1.25, line = 2.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("0", "0.5", "1.0", "1.5"), at = c(0, 0.5, 1.0, 1.5))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at=c(0, 0.5, 1.0, 1.5), labels = F, tck = -0.05)
axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))
axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Adding SEM bars #
arrows(x0 = c(2,1), y0 = root.table$mean, y1 = root.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(2,1), y0 = root.table$mean, y1 = root.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label
text(0.65,1.45 ,labels="B", col="black", cex=2)

# p-value label
mtext(text = expression(italic("P")~" = 0.049") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 8"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)

# Plotting third panel for flower count #
flowercount.fig <- plot(jitter(rep(1, length(flower.con$flower.count)), amount = 0.1), flower.con$flowe
      ylim = c(0, 15), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(flower.rpf$flower.count)), amount = 0.1), flower.rpf$flower.count, pch = 21
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

points(1, mean(flower.con$flower.count), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(flower.rpf$flower.count), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis label #
mtext(expression('Flower number'), side = 2,
      outer = FALSE, cex = 1.25, line = 2.5, adj = 0.5)
```

```
# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("0", "5", "10", "15"), at = c(0, 5, 10, 15))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at=c(0, 5, 10, 15), labels = F, tck = -0.05)
axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))
axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Add SEM bars
arrows(x0 = c(2,1), y0 = flower.table$mean, y1 = flower.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(2,1), y0 = flower.table$mean, y1 = flower.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label #
text(0.65,14.5 ,labels="C", col="black", cex=2)

# p-value label
mtext(text = expression(italic("P")~" = 0.097") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 8"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)


# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Figure2-PlantTraits.png")
grid.raster(img)


# Statistical test: Two sample t-test of Rpf effect on plant traits
# Total plant biomass
total.ttest <- t.test(total.biomass ~ Treatment, data = plant.data)
total.ttest   # Significant: t = 2.8365, df = 9.2015, p = 0.0191
```

```
##
##  Welch Two Sample t-test
##
## data:  total.biomass by Treatment
## t = 2.8365, df = 9.2015, p-value = 0.0191
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.133024 1.163976
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##           1.942375           1.293875
```

```
# Root biomass
root.ttest <- t.test(bbiomass ~ Treatment, data = plant.data)
root.ttest    # Significant: t = 2.3017, df = 8.3705, p = 0.04895
```

```
##
##  Welch Two Sample t-test
##
## data:  bbiomass by Treatment
## t = 2.3017, df = 8.3705, p-value = 0.04895
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.002854404 0.979395596
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##           0.951000           0.459875
```

```
# Shoot biomass
shoot.ttest <- t.test(abiomass ~ Treatment, data = plant.data)
shoot.ttest   # Significant: t = 2.704, df = 14, p = 0.01712
```

```
##
##  Welch Two Sample t-test
##
## data:  abiomass by Treatment
## t = 2.704, df = 14, p-value = 0.01712
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.04596519 0.39853481
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##            1.05625            0.83400
```

```
# Flower number
flower.ttest <- t.test(flower.count ~ Treatment, data = plant.data)
flower.ttest  # Marginally significant: t = 1.7954, df = 12.288, p = 0.09721
```

```
##
##  Welch Two Sample t-test
##
## data:  flower.count by Treatment
## t = 1.7954, df = 12.288, p-value = 0.09721
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6838463  7.1838463
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##              10.00               6.75
```

```
# Seed number
seed.ttest <- t.test(seed.count ~ Treatment, data = plant.data)
seed.ttest    # Non-significant: t = 0.83156, df = 12.983, p = 0.4207
```

```
##
##  Welch Two Sample t-test
##
## data:  seed.count by Treatment
## t = 0.83156, df = 12.983, p-value = 0.4207
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -25.3735  57.1235
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##             55.250             39.375
```

```
# Shoot:Root ratio
SRratio.ttest <- t.test(shoot.root ~ Treatment, data = plant.data)
SRratio.ttest # Non-significant: t = -1.2033, df = 12.903, p = 0.2505
```

```
##
##  Welch Two Sample t-test
##
## data:  shoot.root by Treatment
## t = -1.2033, df = 12.903, p-value = 0.2505
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.4725964  0.4195267
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##           1.582071           2.108606
```

```
# Specific leaf area (SLA)
sla.ttest <- t.test(SLA ~ Treatment, data = plant.data)
sla.ttest     # Non-significant: t = 0.80287, df = 9.1288, p = 0.4424
```

```
##
##  Welch Two Sample t-test
##
## data:  SLA by Treatment
## t = 0.80287, df = 9.1288, p-value = 0.4424
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -96.05961 202.11318
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##           247.3297           194.3029
```

```
# Shoot height
height.ttest <- t.test(height ~ Treatment, data = plant.data)
height.ttest  # Non-significant: t = 1.1102, df = 13.384, p = 0.2865
```

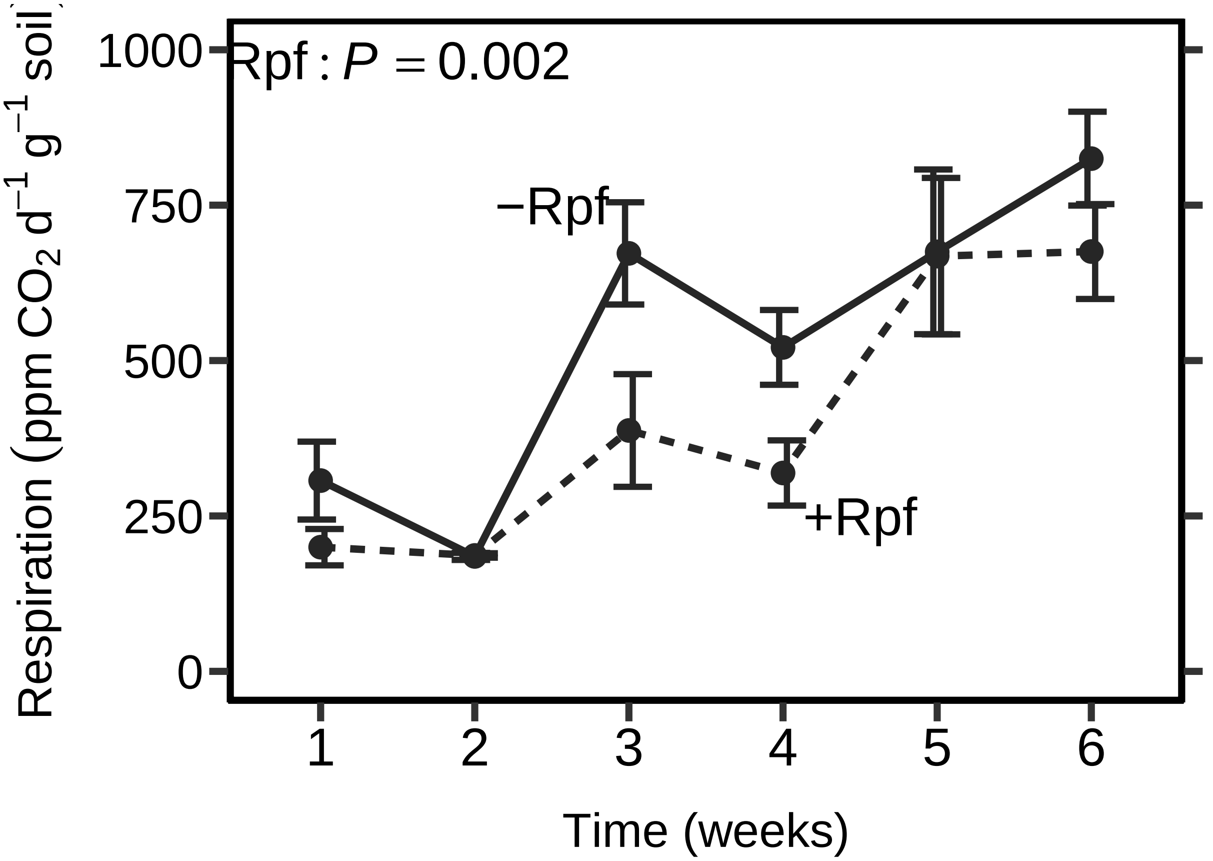
```
##
##  Welch Two Sample t-test
##
## data:  height by Treatment
## t = 1.1102, df = 13.384, p-value = 0.2865
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.643533 11.393533
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##            30.2125            26.3375
```

2a) Rpf effects on soil microbial CO2 respiration

```r
# Load soil respirtation data #
CO2 <- read.csv("~/GitHub/BrassicaRpf/data/GCH_CO2.txt", sep = ",", header = TRUE)

# Subsetting relevant data from CO2 dataset #
CO2.sub <- subset(CO2, plant=="present" & hour =="24" & soil=="live", select=c(Treatment, soil, Week.1,
# Change column names #
colnames(CO2.sub) <- c("Treatment", "soil", "1","2","3","4","5","6")
# Melt dataset into three columns of treatment, soil and weeks for anaylsis
CO2.m <- melt(CO2.sub)
```

```
## Using Treatment, soil as id variables
```

```r
# Change variable column name to week
colnames(CO2.m) <- c("Treatment", "soil", "Week", "value")
# Set week as factor
CO2.m$Week <- as.factor(CO2.m$Week)

# Standardizing the soil respiration values by 24 hours
CO2.m$StdCO2 <- (CO2.m$value)/24


# Calculating CO2 respiration mean and sem table #
CO2.means.sem <- ddply(CO2.m, c("Treatment", "Week"), summarise,
                   mean=mean(StdCO2), sem=sd(StdCO2)/sqrt(length(StdCO2)))
CO2.means.sem <- transform(CO2.means.sem, lower=mean-sem, upper=mean+sem)

# The errorbars overlapped, so use position_dodge to move them horizontally
pd <- position_dodge(0)


# Generate figure 3 for Rpf effects on soil respiration #
co <- ggplot(CO2.means.sem, aes(x=Week, y=mean, colour=Treatment, group=Treatment)) +
      geom_errorbar(aes(ymax=upper, ymin=lower), position=position_dodge(0.1),
                data=CO2.means.sem, width = 0.5, size=1.1) +
      geom_line(aes(linetype=Treatment) ,position=pd, size=1.3) +
      geom_point(aes(shape=Treatment), position=pd, size=4) +
      scale_shape_manual(values=c(16, 16)) +
      xlab("Time (weeks)") +
      ylab(expression(~Respiration~(ppm~CO[2]~d^-1~g^-1~soil)))

co + scale_y_continuous(limits = c(0, 1000), breaks = seq(0, 1000, 250),
                     sec.axis = sec_axis(~ . * 1, labels = c(" "," "," ", " ", " "))) +
     theme_classic() +
     theme(axis.text.y=element_text(colour = "black", size = 18),
          axis.text.x=element_text(colour = "black", size = 20),
          axis.ticks = element_line(size = 1.25),
          axis.ticks.length = unit(.25, "cm"),
          axis.title.y = element_text(size = 18, colour = "black", margin = margin(0,10,0,0)),
          axis.title.x = element_text(size = 18, colour = "black", margin = margin(15,10,0,10)),
          panel.border = element_rect(linetype = "solid", colour = "black", size = 2, fill = NA),
          legend.position="none") +
    scale_color_manual(values=c('gray15','gray15')) +
  annotate("text", x = 2.5, y = 750, label = "-Rpf", cex=7) +
  annotate("text", x = 4.5, y = 250, label = "+Rpf", cex=7) +
  #annotate("text", x = 1.5, y = 975, label = "Rpf: P = 0.002", cex=7) +
  annotate("text", x = 1.5, y = 975, label = "Rpf: italic(P)==0.002", parse = TRUE, cex=7) +
  #annotate("text", x = 0.925, y = 900, label = "n = 8", cex=7)



ggsave("../figures/Figure3-Soilrespiration.png", width = 20, height = 15, units = "cm")
```



```
# Statistical tests: Rpf effects on soil CO2 respiration
# RM-ANOVA
CO2.rm <- lme(StdCO2 ~ Week*Treatment, random = ~ 1 | soil,
              correlation = corAR1(form = ~1 | soil),
              data = CO2.m)
# Make ANOVA table #
set.caption("RMANOVA for soil CO2 respiration")
pander(anova(CO2.rm))
```

Table 1: RMANOVA for soil CO2 respiration

	numDF	denDF	F-value	p-value
(Intercept)	1	84	57.47	4.161e-11
Week	5	84	21.6	7.772e-14
Treatment	1	84	9.676	0.00255

	numDF	denDF	F-value	p-value
Week:Treatment	5	84	1.217	0.3084

```
    # Significant effect of week: F5,84 = 21.6, p < 0.000
    # Significant effect of Rpf: F1,84 = 9.676, p = 0.00255
    # Non-significant interaction: F5,84 = 1.217, p = 0.3084
```

2b) Effect of Rpf on soil microbial abundance

```
# Load microbial qPCR abundance data #
abundance <- read.csv("~/GitHub/BrassicaRpf/data/qPCR.txt", sep = ",", header = TRUE)
# remove week 1 data #
abundance <- subset(abundance, Week == "6")
# Reset data frame index #
rownames(abundance) <- NULL

# Standardizing gene copy abundances by soil amount #
abundance$stdGeneCopy <- (abundance$GeneCopy)/(abundance$Soil)
# Log10 transforming standardized gene copy number #
abundance$log10stdGeneCopy <- log10(abundance$stdGeneCopy)

# Splitting abundance dataset by fungi and bacteria #
data.sub.bac <- subset(abundance, Gene == "rRNA")
data.sub.fun <- subset(abundance, Gene == "ITS")

# Calculating fungi to bacteria ratios #
gca.B <- data.sub.bac$stdGeneCopy
gca.F <- data.sub.fun$stdGeneCopy
abundance$FB <- (gca.F/gca.B)
ratioFB <- abundance[1:18, ]
ratioFB <- subset(ratioFB, select=(c(Treatment, Week, FB)))

# Subsetting abundance dataset
data.sub <- subset(abundance, select=(c(Treatment, Gene, stdGeneCopy)))
# Melting the dataset
data.m <- melt(data.sub)
```

```
## Using Treatment, Gene as id variables
```

```
# Bacterial abundance data for plotting #
bac.rpf <- data.m[ which(data.m$Treatment == "Rpf+" & data.m$Gene == "rRNA"),]
bac.con <- data.m[ which(data.m$Treatment == "Rpf-" & data.m$Gene == "rRNA"),]
# Bacterial abundance mean and sem table #
bac.mean <- aggregate(data.sub.bac$stdGeneCopy ~ Treatment, data.sub.bac, mean)
bac.sem <- aggregate(data.sub.bac$stdGeneCopy ~ Treatment, data.sub.bac, sem)
bac.sem.LL <- bac.mean[2] - bac.sem[2]
bac.sem.UL <- bac.mean[2] + bac.sem[2]
bac.table <- data.frame(bac.mean[1], bac.mean[2], bac.sem[2],
          bac.sem.LL[1], bac.sem.UL[1])
colnames(bac.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
bac.table <- bac.table[order(bac.table$mean),]
```

```
# Fungal abundance data for plotting #
fun.rpf <- data.m[ which(data.m$Treatment == "Rpf+" & data.m$Gene == "ITS"),]
fun.con <- data.m[ which(data.m$Treatment == "Rpf-" & data.m$Gene == "ITS"),]
# Fungal abundance mean and sem table #
fun.mean <- aggregate(data.sub.fun$stdGeneCopy ~ Treatment, data.sub.fun, mean)
fun.sem <- aggregate(data.sub.fun$stdGeneCopy ~ Treatment, data.sub.fun, sem)
fun.sem.LL <- fun.mean[2] - fun.sem[2]
fun.sem.UL <- fun.mean[2] + fun.sem[2]
fun.table <- data.frame(fun.mean[1], fun.mean[2], fun.sem[2],
          fun.sem.LL[1], fun.sem.UL[1])
colnames(fun.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
fun.table <- fun.table[order(fun.table$mean),]

# Fungi: Bacteria ratio data for plotting #
FB.rpf <- ratioFB[ which(ratioFB$Treatment == "Rpf+"),]
FB.con <- ratioFB[ which(ratioFB$Treatment == "Rpf-"),]
# FB: Bacteria ratio mean and sem #
FB.mean <- aggregate(ratioFB$FB ~ Treatment, ratioFB, mean)
FB.sem <- aggregate(ratioFB$FB ~ Treatment, ratioFB, sem)
FB.sem.LL <- FB.mean[2] + FB.sem[2]
FB.sem.UL <- FB.mean[2] - FB.sem[2]
FB.table <- data.frame(FB.mean[1], FB.mean[2], FB.sem[2],
          FB.sem.LL[1], FB.sem.UL[1])
colnames(FB.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
FB.table <- FB.table[order(FB.table$mean),]


# Generating 3 panel figure for Rpf effects on soil microbial abundance
# Plotting microbial abundance and ratio figure #
png(filename="../figures/Figure4-MicrobialAbundanceRatio.png",
    width = 1200, height = 800, res = 96*2)
par(oma=c(7,3,7,1), mar=c(2,3,3,4), mfrow=c(1,3))

# Panel 1: Rpf effects on bacterial abundance #
abun.bac.fig <- plot(jitter(rep(1, length(bac.con$value)), amount = 0.1), bac.con$value,
      ylim = c(0, 2.4E6), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(bac.rpf$value)), amount = 0.1), bac.rpf$value, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data pointfor each treatment #
points(1, mean(bac.con$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(bac.rpf$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis labels
mtext(expression('16S rRNA gene copy/ g soil'), side = 2,
      outer = FALSE, cex = 1, line = 3.5, adj = 0.5)
```

```
# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0E0", "8.0E5", "1.6E6", "2.4E6"),
     at = c(0.0E0, 8.0E5, 1.6E6, 2.4E6))

axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0E0, 8.0E5, 1.6E6, 2.4E6), labels = F, tck = -0.05)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Adding SEM #
arrows(x0 = c(2,1), y0 = bac.table$mean, y1 = bac.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(2,1), y0 = bac.table$mean, y1 = bac.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label #
text(0.7,2.3E6 ,labels="A", col="black", cex=2)

# P-value label
mtext(text = expression(italic("P")~" = 0.016") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 9"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)

# Panel 2: Rpf effects on fungal abundance #
abun.fun.fig <- plot(jitter(rep(1, length(fun.con$value)), amount = 0.1), fun.con$value,
      ylim = c(0, 1.5E6), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(fun.rpf$value)), amount = 0.1), fun.rpf$value, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data pointfor each treatment #
points(1, mean(fun.con$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(fun.rpf$value), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis labels
mtext(expression('ITS gene copy / g soil'), side = 2,
      outer = FALSE, cex = 1, line = 3.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0E0", "5.0E5", "1.0E6", "1.5E6"),
     at = c(0.0E0, 5.0E5, 1.0E6, 1.5E6))
```

```
axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0E0, 5.0E5, 1.0E6, 1.5E6), labels = F, tck = -0.05)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Adding SEM #
arrows(x0 = c(1,2), y0 = fun.table$mean, y1 = fun.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(1,2), y0 = fun.table$mean, y1 = fun.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label #
text(0.7,1.45E6 ,labels="B", col="black", cex=2)

# p-value
mtext(text = expression(italic("P")~" = 0.007") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 9"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)

# Panel 3: Rpf effects on fungal abundance #
FB.fig <- plot(jitter(rep(1, length(FB.con$FB)), amount = 0.1), FB.con$FB,
      ylim = c(0, 1.5), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(FB.rpf$FB)), amount = 0.1), FB.rpf$FB, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data pointfor each treatment #
points(1, mean(FB.con$FB), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(FB.rpf$FB), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis labels
mtext(expression('Fungal : bacterial ratio'), side = 2,
      outer = FALSE, cex = 1, line = 2.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0", "0.5", "1.0", "1.5"),
     at = c(0.0, 0.5, 1.0, 1.5))

axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0, 0.5, 1.0, 1.5), labels = F, tck = -0.05)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
```

```r
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.05)

# Adding SEM #
arrows(x0 = c(1,2), y0 = FB.table$mean, y1 = FB.table$LCI, angle = 90,
       length = 0.15, lwd = 2)
arrows(x0 = c(1,2), y0 = FB.table$mean, y1 = FB.table$UCI, angle = 90,
       length=0.15, lwd = 2)

# Panel label #
text(0.7,1.45,labels="C", col="black", cex=2)

# p-value
mtext(text = expression(italic("P")~" = 0.018") , side =3, line = -1.2, adj = 0.925, col="black", cex=0
# Sample number label
#mtext(text = expression(italic("n")~" = 9"), side = 3, line = -2.2, adj = 0.925, col="black", cex=0.8)

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```r
graphics.off()

# Show Plot
img <- readPNG("../figures/Figure4-MicrobialAbundanceRatio.png")
grid.raster(img)


# Statistical tests: 2 sample t-test for Rpf effects on microbial abundance
# Bacterial abundance
bac.ttest <- t.test(stdGeneCopy ~ Treatment, data = data.sub.bac)
bac.ttest    # Significant: t = 2.7069, df = 14.777, p = 0.0164
```

```
##
##  Welch Two Sample t-test
##
## data:  stdGeneCopy by Treatment
## t = 2.7069, df = 14.777, p-value = 0.0164
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  104946.7 887208.7
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##            1683463            1187385
```

```r
# Fungal abundance
fun.ttest <- t.test(stdGeneCopy ~ Treatment, data = data.sub.fun)
fun.ttest    # Significant: t = -3.2719, df = 11.701, p = 0.00689
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  stdGeneCopy by Treatment
## t = -3.2719, df = 11.701, p-value = 0.00689
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -636711.6 -126817.7
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##           236624.1           618388.7
```

```
# F:B ratio
FB.ttest <- t.test(FB ~ Treatment, data = ratioFB)
FB.ttest     # Significant: t= -2.8454, df = 9.4476, p = 0.01835
```

```
##
##  Welch Two Sample t-test
##
## data:  FB by Treatment
## t = -2.8454, df = 9.4476, p-value = 0.01835
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.80465024 -0.09474171
## sample estimates:
## mean in group Rpf- mean in group Rpf+
##          0.1582796          0.6079756
```

4) Effect of Rpf and metabolic status on soil bacterial community structure

Set up work environment and load packages

```
source("~/GitHub/BrassicaRpf/bin/DiversityFunctions.R")
source("~/GitHub/BrassicaRpf/bin/MothurTools.R")
source("~/GitHub/BrassicaRpf/bin/phylodiversity2.R")

# Run All: Select if all section are to be re-run
run.all <- TRUE

# Load files
  # design = general design file for experiment
  # shared = OTU table from mothur with sequence similarity clustering
  # tax    = Taxonomy for 97% similarity OTUs

design <- "~/GitHub/BrassicaRpf/data/Brassica.design.txt"
shared <- "~/GitHub/BrassicaRpf/mothur/output/Brassica.bac.final.shared"
tax <- "~/GitHub/BrassicaRpf/mothur/output/Brassica.bac.final.0.03.taxonomy"

# Import Design
design <- read.delim(design, header=T, row.names=1)
# Import Shared Files
OTU <- read.otu(shared = shared, cutoff = "0.03") # 97% similarity
# Import Taxonomy
```

```
OTU.tax <- read.tax(taxonomy = tax, format = "rdp")
# Import Phylogenetic tree
OTU.tre <- read.tree("../phylo/Brassica.bac.rename.tree.2")
```

Soil bacteria beta and alpha diversity

```
### Sequence Coverage
# Remove OTUs with less than two occurences across all sites #
OTU <- OTU[, which(colSums(OTU) >= 2)]

# Remove mock community #
OTU <- OTU[1:20, ]

# Determine coverage of sequences #
cov.seqs <- count.groups(OTU)
cov.mean <- mean(cov.seqs) # 160,871
cov.sem <- sem(cov.seqs) # 16,095.38
cov.min <- min(cov.seqs) # 79,797
total.seqs <- sum(cov.seqs) # 3,217,419

# Good's coverage
goods.c <- function(x = ""){
              1 - (apply(OTU, 1, function(x){sum(x == 1)}) / rowSums(x))
}

goods.c.Brassica <- goods.c(OTU)
mean.good.c <- mean(goods.c.Brassica) # 0.984  Good mean coverage
min.good.c <- min(goods.c.Brassica) # 0.967  Good lowest coverage

### Alpha diversity
# Resampling code to estimate alpha diversity (used if run.all = T)
if (run.all == TRUE){
  rich <- round(richness.iter(input = OTU, size = 1000,
                              iters = 100, shared = "FALSE"), 3)
  even <- round(evenness.iter(input = OTU, size = 1000,
                              iters = 100, shared = "FALSE",
                              method = "simp_even"), 3)
  rare <- rarefy(OTU, 1000, se = FALSE, MARGIN = 1)
  # Write output to files
  write.table(rich, "../data/rich.txt", sep = "\t",
              col.names = T, row.names = T)
  write.table(even, "../data/even.txt", sep = "\t",
              col.names = T, row.names = T)
}

# Read in alpha diversity files from above
rich2 <- read.table("../data/rich.txt", sep = "\t")
even2 <- read.table("../data/even.txt", sep = "\t")

# Merge data to design and calculate mean and sem per sample
rich.data <- merge(design, rich2, by = "row.names")
row.names(rich.data) <- rich.data$Row.names
```

```
rich.data <- rich.data[sort(row.names(rich.data)), ]
rich.mean <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, mean, na.rm = TRUE),3)
rich.sem <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, sem, na.rm = TRUE), 3)

even.data <- merge(design, even2, by = "row.names")
row.names(even.data) <- even.data$Row.names
even.data <- even.data[sort(row.names(even.data)), ]
even.mean <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, mean, na.rm = TRUE),3)
even.sem <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, sem, na.rm = TRUE),4)

# Make new dataframe merging design file and mean diversity
Brassica.div <- data.frame(design[sort(row.names(design)), ], rich.mean, even.mean)

# Take averages of technial reps
rich.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, rich = mean(rich.mean))
even.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, even = mean(even.mean))

# Reshape data
rich.2 <- reshape(rich.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")
even.2 <- reshape(even.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")


## Statistical test: One-way ANOVA of Rpf and metabolic status
# Soil bacterial richness
trans <- Brassica.div[ which(Brassica.div$type == 'cDNA'), ]
gDNA <- Brassica.div[ which(Brassica.div$type == 'DNA'), ]

trans.aov <- aov(rich.mean ~ treatment, trans)
summary(trans.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1     81      81   0.032  0.862
## Residuals    8  20161    2520
```

```
gDNA.aov <- aov(rich.mean ~ treatment, gDNA)
summary(gDNA.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1   1718    1718   0.572  0.471
## Residuals    8  24038    3005
```

```
rich.anova.c <- aov(rich.mean ~ treatment*type, Brassica.div)
summary(rich.anova.c)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## treatment       1    527   526.6   0.191  0.668
## type            1      6     5.8   0.002  0.964
## treatment:type  1   1273  1272.8   0.461  0.507
## Residuals      16  44199  2762.4
```

```
# Soil bacterial evenness
even.anova.c <- aov(even.mean ~ treatment*type, Brassica.div)
summary(even.anova.c)
```

```
##                Df  Sum Sq  Mean Sq F value Pr(>F)
## treatment       1 0.00058 0.000583   0.052  0.823
## type            1 0.00792 0.007920   0.704  0.414
## treatment:type  1 0.00006 0.000058   0.005  0.944
## Residuals      16 0.18012 0.011257
```

```
# Alpha diversity: Wilcox Signed Rank Test to account for paired design
# Richness
rich.wilcox <- wilcox.test(Brassica.div$rich.mean[which(Brassica.div$treatment == 'Rpf-')], Brassica.di
rich.wilcox   # No effect of rpf
```

```
##
##  Wilcoxon signed rank test
##
## data:  Brassica.div$rich.mean[which(Brassica.div$treatment == "Rpf-")] and Brassica.div$rich.mean[wh
## V = 28, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

```
# Evenness
even.wilcox <- wilcox.test(Brassica.div$even.mean[which(Brassica.div$treatment == 'Rpf-')], Brassica.di
even.wilcox   # No effect of rpf
```

```
##
##  Wilcoxon signed rank test
##
## data:  Brassica.div$even.mean[which(Brassica.div$treatment == "Rpf-")] and Brassica.div$even.mean[wh
## V = 28, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

```
### Beta Diversity
# Make presence-absence matrix
OTU.PA <- (OTU > 0) * 1

# Make relative abundence matrix
OTU.REL <- OTU
for (i in 1:dim(OTU)[1]){
  OTU.REL[i,] <- OTU[i,]/sum(OTU[i,])
  }

# Log-transform relative abundances
OTU.REL.log <- decostand(OTU, method="log")

# Generate sample distance matrix from log-transformed relative abundance of OTU
Brassica.bc.dis <- vegdist(OTU.REL.log, method = "bray", binary = "FALSE")
Brassica.dis.mean <- mean(Brassica.bc.dis)

# Principal Coordinates Analysis (PCoA)
Brassica.PCoA <- cmdscale(Brassica.bc.dis, eig = TRUE, k = 3)
explainvar1 <- round(Brassica.PCoA$eig[1] / sum(Brassica.PCoA$eig), 3) * 100
explainvar2 <- round(Brassica.PCoA$eig[2] / sum(Brassica.PCoA$eig), 3) * 100
explainvar3 <- round(Brassica.PCoA$eig[3] / sum(Brassica.PCoA$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

# OTU Scores
otu.scores <- t(cor(Brassica.PCoA$points, OTU.REL))
otu.scores <- as.matrix(otu.scores)[,1:2]
```

```r
otu.scores <- otu.scores[abs(otu.scores[,1]) > 0.7|abs(otu.scores[,2]) > 0.7,]

# Average BC Distance Between Treatments
Brassica.bc.dis.m <- as.matrix(Brassica.bc.dis)
all.equal(row.names(Brassica.div), rownames(Brassica.bc.dis.m))
```

```
## [1] TRUE
```

```r
treatment.div <- unique(Brassica.div$treatment)
treatment.dis <- rep(NA, length(treatment.div))
for(i in 1:length(treatment.div)){
  temp <- row.names(Brassica.div[Brassica.div$treatment == treatment.div[i], ])
  treatment.dis[i] <- Brassica.bc.dis.m[temp[1], temp[2]]
}

mean(treatment.dis)
```

```
## [1] 0.4775868
```

```r
# Plot figure -- Supplement for all bacteria ordination
png(filename="../figures/Suppl.Fig4.Bacteria.png",
    width = 1800, height = 800, res = 96*2)

layout(matrix(1:3, 1, 3), widths = c(20, 9, 2.5))

par(mar = c(7, 10, 1, 0) + 0.5)

plot(Brassica.PCoA$points[ ,1], Brassica.PCoA$points[ ,2],
     ylim = c(-0.4, 0.4), xlim = c(-0.5, 0.4),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""), line = 5,
     #xlab = "", ylab = "", xaxt = "n", yaxt = "n",
     pch = 22, cex = 2.0, type = "n", cex.lab = 2.5, cex.axis = 3,
     axes = FALSE)
```

```
## Warning in plot.window(...): "line" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "line" is not a graphical parameter
```

```r
# Add Axes
axis(side = 1, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 2, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 3, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
axis(side = 4, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
abline(h = 0, v = 0, lty = 3)

box(lwd = 2)

# Subset data
all.equal(row.names(Brassica.PCoA$points), rownames(Brassica.div))
```

```
## [1] TRUE
```

```r
Brassica.points <- data.frame(Brassica.PCoA$points, Brassica.div)

# Active community
Brassica.active.rpf <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.active.no <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf-"), ]
# Total community
Brassica.total.rpf <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.total.no <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf-"), ]

# Add points
# Active community Rpf+
points(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Active community Rpf-
points(Brassica.active.no[ ,1], Brassica.active.no[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)
# Total community Rpf+
points(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Total community Rpf-
points(Brassica.total.no[ ,1], Brassica.total.no[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)

# Add ellipses
# Active Rpf+
ordiellipse(cbind(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2]), Brassica.active.rpf$treatment, ki
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Active Rpf-
ordiellipse(cbind(Brassica.active.no[ ,1], Brassica.active.no[ ,2]), Brassica.active.no$treatment, kind=
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf+
ordiellipse(cbind(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2]), Brassica.total.rpf$treatment, kind=
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf-
ordiellipse(cbind(Brassica.total.no[ ,1], Brassica.total.no[ ,2]), Brassica.total.no$treatment, kind="s
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)

# Add Legend Outside
par(mar = c(4, 0, 5, 1) + 0.5)
plot.new()
legend(0, 1, c("Active, -Rpf", "Active, +Rpf", "Total, -Rpf", "Total, +Rpf"),
       pch = c(21, 21, 24, 24),
       col = c("Black", "Black", "Black", "Black"),
       pt.bg = c("lightgrey", "grey15", "lightgrey", "grey15"),
       bty = "n", y.intersp = 1, pt.cex = 3.2, cex = 2, lwd= 2, lty =  NA)

# Sample number label
mtext(text = expression(italic("n")~" = 5"), line = 1, adj = -0.65, col="black", cex=1.5)

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```r
graphics.off()

# Show Plot
img <- readPNG("../figures/Suppl.Fig4.Bacteria.png")
grid.raster(img)

# Statistical test
# Add factor for pot number to account of paired/match design of study
Brassica.div$pot <- c(1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10)
Brassica.div$pot <- as.factor(Brassica.div$pot) #unranked order

# PERMANOVA to test Rpf and metabolic status effects on soil bacterial community structure
all.bray.permanova <- adonis(OTU.REL.log ~ type + treatment + pot +
                          type*treatment , data = Brassica.div,
                         method = "bray", binary = FALSE, permutations = 999)
all.bray.permanova
```

```
##
## Call:
## adonis(formula = OTU.REL.log ~ type + treatment + pot + type *      treatment, data = Brassica.div, p
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                 Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## type             1   0.54671 0.54671  7.2402 0.22571  0.001 ***
## treatment        1   0.14742 0.14742  1.9523 0.06086  0.032 *
## pot              8   1.02737 0.12842  1.7007 0.42415  0.001 ***
## type:treatment   1   0.09661 0.09661  1.2794 0.03988  0.168
## Residuals        8   0.60409 0.07551         0.24940
## Total           19   2.42219                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Indentify indicator species
#spe.corr <- add.spec.scores(Brassica.PCoA, OTU.REL.log,
#                            method = "cor.scores")$cproj
#corrcut <- 0.7 # User defined cutoff
#imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]
#imp.spp
#imp.otu <- as.vector(rownames(imp.spp))
#imp.otu

#imp.spp.lst <- OTU.tax[ which(OTU.tax$OTU == imp.otu | OTU.tax$OTU == "Otu00014" | OTU.tax$OTU == "Otu
#imp.spp.lst
```

Gram positive bacteria alpha and beta- diversity

```r
# Subset Gram positive OTU from taxonomy file
GramPositive.OTU.tax <- OTU.tax[ which(OTU.tax$Phylum == 'Actinobacteria' | OTU.tax$Phylum == 'Firmicute
# Subset OTU table to contain only Gram positive bacteria
GramPositive.OTU <- match(GramPositive.OTU.tax$OTU, colnames(OTU))
GramPositive.OTU <- sort(c(GramPositive.OTU-1, GramPositive.OTU))
Gpos.OTU <- OTU[, GramPositive.OTU]

# Remove OTUs with less than two occurences across all sites #
Gpos.OTU <- Gpos.OTU[, which(colSums(Gpos.OTU) >= 2)]

# Remove mock community #
Gpos.OTU <- Gpos.OTU[1:20, ]

### Alpha diversity
# Resampling code to estimate alpha diversity (used if run.all = T)
if (run.all == TRUE){
  rich <- round(richness.iter(input = Gpos.OTU, size = 1000,
                              iters = 100, shared = "FALSE"), 3)
  even <- round(evenness.iter(input = Gpos.OTU, size = 1000,
                              iters = 100, shared = "FALSE",
                              method = "simp_even"), 3)
  rare <- rarefy(Gpos.OTU, 1000, se = FALSE, MARGIN = 1)
  # Write output to files
  write.table(rich, "../data/rich.txt", sep = "\t",
              col.names = T, row.names = T)
  write.table(even, "../data/even.txt", sep = "\t",
              col.names = T, row.names = T)
}

# Read in alpha diversity files from above
rich2 <- read.table("../data/rich.txt", sep = "\t")
even2 <- read.table("../data/even.txt", sep = "\t")

# Merge data to design and calculate mean and sem per sample
rich.data <- merge(design, rich2, by = "row.names")
row.names(rich.data) <- rich.data$Row.names
rich.data <- rich.data[sort(row.names(rich.data)), ]
rich.mean <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, mean, na.rm = TRUE),3)
rich.sem <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, sem, na.rm = TRUE), 3)

even.data <- merge(design, even2, by = "row.names")
row.names(even.data) <- even.data$Row.names
even.data <- even.data[sort(row.names(even.data)), ]
even.mean <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, mean, na.rm = TRUE),3)
even.sem <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, sem, na.rm = TRUE),4)

# Make new dataframe merging design file and mean diversity
Brassica.div <- data.frame(design[sort(row.names(design)), ], rich.mean, even.mean)

# Take averages of technial reps
rich.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, rich = mean(rich.mean))
even.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, even = mean(even.mean))

# Reshape data
rich.2 <- reshape(rich.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")

even.2 <- reshape(even.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")


## Statistical test: two-way ANOVA of Rpf and metabolic status
# Soil bacterial richness
rich.anova.c <- aov(rich.mean ~ type + treatment + treatment*type, Brassica.div)
summary(rich.anova.c)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## type             1    850   849.7   0.744  0.401
## treatment        1     33    33.5   0.029  0.866
## type:treatment   1    296   295.7   0.259  0.618
## Residuals       16  18276  1142.2
```

```
# Soil bacterial evenness
even.anova.c <- aov(even.mean ~ type + treatment + treatment*type, Brassica.div)
summary(even.anova.c)
```

```
##                 Df   Sum Sq  Mean Sq F value  Pr(>F)
## type             1 0.027602 0.027602  14.436 0.00157 **
## treatment        1 0.001674 0.001674   0.876 0.36329
## type:treatment   1 0.001566 0.001566   0.819 0.37883
## Residuals       16 0.030594 0.001912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Beta Diversity
# Make presence-absence matrix
Gpos.OTU.PA <- (Gpos.OTU > 0) * 1

# Make relative abundence matrix
Gpos.OTU.REL <- Gpos.OTU
for (i in 1:dim(Gpos.OTU)[1]){
  Gpos.OTU.REL[i,] <- Gpos.OTU[i,]/sum(Gpos.OTU[i,])
  }

# Log-transform relative abundances
Gpos.OTU.REL.log <- decostand(Gpos.OTU, method="log")

Brassica.bc.dis <- vegdist(Gpos.OTU.REL.log, method = "bray", binary = "FALSE")
Brassica.dis.mean <- mean(Brassica.bc.dis)

# Principal Coordinates Analysis (PCoA)
Brassica.PCoA <- cmdscale(Brassica.bc.dis, eig = TRUE, k = 3)
explainvar1 <- round(Brassica.PCoA$eig[1] / sum(Brassica.PCoA$eig), 3) * 100
explainvar2 <- round(Brassica.PCoA$eig[2] / sum(Brassica.PCoA$eig), 3) * 100
explainvar3 <- round(Brassica.PCoA$eig[3] / sum(Brassica.PCoA$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

# Gpos.OTU Scores
Gpos.OTU.scores <- t(cor(Brassica.PCoA$points, Gpos.OTU.REL))
Gpos.OTU.scores <- as.matrix(Gpos.OTU.scores)[,1:2]
Gpos.OTU.scores <- Gpos.OTU.scores[abs(Gpos.OTU.scores[,1]) > 0.7|abs(Gpos.OTU.scores[,2]) > 0.7,]

# Average BC Distance Between Treatments
Brassica.bc.dis.m <- as.matrix(Brassica.bc.dis)
all.equal(row.names(Brassica.div), rownames(Brassica.bc.dis.m))
```

```
## [1] TRUE
```

```
treatment.div <- unique(Brassica.div$treatment)
treatment.dis <- rep(NA, length(treatment.div))
for(i in 1:length(treatment.div)){
  temp <- row.names(Brassica.div[Brassica.div$treatment == treatment.div[i], ])
  treatment.dis[i] <- Brassica.bc.dis.m[temp[1], temp[2]]
}

mean(treatment.dis)
```

```
## [1] 0.464032
```

```
# Plot figure # Gram positive ordination
png(filename="../figures/Suppl.Fig3.GramPositive.png",
    width = 1800, height = 800, res = 96*2)

layout(matrix(1:3, 1, 3), widths = c(20, 9, 2.5))

par(mar = c(7, 10, 1, 0) + 0.5)

plot(Brassica.PCoA$points[ ,1], Brassica.PCoA$points[ ,2],
     ylim = c(-0.4, 0.4), xlim = c(-0.5, 0.4),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""), line = 5,
     #xlab = "", ylab = "", xaxt = "n", yaxt = "n",
     pch = 22, cex = 2.0, type = "n", cex.lab = 2.5, cex.axis = 3,
     axes = FALSE)
```

```
## Warning in plot.window(...): "line" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "line" is not a graphical parameter
```

```
# Add Axes
axis(side = 1, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 2, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 3, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
axis(side = 4, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
abline(h = 0, v = 0, lty = 3)

box(lwd = 2)

# Subset data
all.equal(row.names(Brassica.PCoA$points), rownames(Brassica.div))
```

```
## [1] TRUE
```

```
Brassica.points <- data.frame(Brassica.PCoA$points, Brassica.div)

# Active community
Brassica.active.rpf <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.active.no <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf-"), ]
# Total community
Brassica.total.rpf <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.total.no <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf-"), ]

# Add points
# Active community Rpf+
points(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Active community Rpf-
points(Brassica.active.no[ ,1], Brassica.active.no[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)
# Total community Rpf+
points(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Total community Rpf-
points(Brassica.total.no[ ,1], Brassica.total.no[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)

# Add elipcses
# Active Rpf+
ordiellipse(cbind(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2]), Brassica.active.rpf$treatment, ki
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Active Rpf-
ordiellipse(cbind(Brassica.active.no[ ,1], Brassica.active.no[ ,2]), Brassica.active.no$treatment, kind
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf+
ordiellipse(cbind(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2]), Brassica.total.rpf$treatment, kind
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf-
ordiellipse(cbind(Brassica.total.no[ ,1], Brassica.total.no[ ,2]), Brassica.total.no$treatment, kind="s
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)


# Add Legend Outside
par(mar = c(4, 0, 5, 1) + 0.5)
plot.new()
legend(0, 1, c("Active, -Rpf", "Active, +Rpf", "Total, -Rpf", "Total, +Rpf"),
       pch = c(21, 21, 24, 24),
       col = c("Black", "Black", "Black", "Black"),
       pt.bg = c("lightgrey", "grey15", "lightgrey", "grey15"),
       bty = "n", y.intersp = 1, pt.cex = 3.2, cex = 2, lwd= 2, lty =  NA)

# Sample number label
mtext(text = expression(italic("n")~" = 5"), line = 1, adj = -0.65, col="black", cex=1.5)

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Suppl.Fig3.GramPositive.png")
grid.raster(img)

# Factoring pot number for match/pair design
Brassica.div$pot <- c(1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10)
Brassica.div$pot <- as.factor(Brassica.div$pot)

## Statistical test: PERMANOVA to test Rpf and metabolic status effects on gram positive bacteria compo
gpos.bray.permanova <- adonis(Gpos.OTU.REL ~ type + treatment + pot +
                           type*treatment, data = Brassica.div,
                         method = "bray", binary = FALSE, permutations = 999)
gpos.bray.permanova
```

```
##
## Call:
## adonis(formula = Gpos.OTU.REL ~ type + treatment + pot + type *      treatment, data = Brassica.div,
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                 Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## type             1   0.70147 0.70147 13.4025 0.27823  0.001 ***
## treatment        1   0.18241 0.18241  3.4851 0.07235  0.002 **
## pot              8   1.12593 0.14074  2.6890 0.44659  0.001 ***
## type:treatment   1   0.09265 0.09265  1.7702 0.03675  0.075 .
## Residuals        8   0.41871 0.05234         0.16608
## Total           19   2.52117                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Indentify indicator species
#spe.corr <- add.spec.scores(Brassica.PCoA, Gpos.OTU.REL,
#                              method = "cor.scores")$cproj
#corrcut <- 0.94 # User defined cutoff
#imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]
#imp.spp
#imp.otu <- as.vector(rownames(imp.spp))
#imp.otu

#imp.spp.lst <- OTU.tax[ which(OTU.tax$OTU == imp.otu), ]
#imp.spp.lst
```

Actinobacteria alpha and beta- diversity

```
# Subset Gram positive OTU from taxonomy file
Actino.OTU.tax <- OTU.tax[ which(OTU.tax$Phylum == 'Actinobacteria'), ]
```

```r
# Subset OTU table to contain only Actinobacteria
Actino.OTU <- match(Actino.OTU.tax$OTU, colnames(OTU))
Actino.OTU <- sort(c(Actino.OTU-1, Actino.OTU))
Actino.OTU <- OTU[, Actino.OTU]

# Remove OTUs with less than two occurences across all sites #
Actino.OTU <- Actino.OTU[, which(colSums(Actino.OTU) >= 2)]

# Remove mock community #
Actino.OTU <- Actino.OTU[1:20, ]


### Alpha diversity
# Resampling code to estimate alpha diversity (used if run.all = T)
if (run.all == TRUE){
  rich <- round(richness.iter(input = Actino.OTU, size = 1000,
                              iters = 100, shared = "FALSE"), 3)
  even <- round(evenness.iter(input = Actino.OTU, size = 1000,
                              iters = 100, shared = "FALSE",
                              method = "simp_even"), 3)
  rare <- rarefy(Actino.OTU, 1000, se = FALSE, MARGIN = 1)
  # Write output to files
  write.table(rich, "../data/rich.txt", sep = "\t",
              col.names = T, row.names = T)
  write.table(even, "../data/even.txt", sep = "\t",
              col.names = T, row.names = T)
}

# Read in alpha diversity files from above
rich2 <- read.table("../data/rich.txt", sep = "\t")
even2 <- read.table("../data/even.txt", sep = "\t")

# Merge data to design and calculate mean and sem per sample
rich.data <- merge(design, rich2, by = "row.names")
row.names(rich.data) <- rich.data$Row.names
rich.data <- rich.data[sort(row.names(rich.data)), ]
rich.mean <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, mean, na.rm = TRUE),3)
rich.sem <- round(apply(rich.data[5:(4 + dim(rich2)[2])], 1, sem, na.rm = TRUE), 3)

even.data <- merge(design, even2, by = "row.names")
row.names(even.data) <- even.data$Row.names
even.data <- even.data[sort(row.names(even.data)), ]
even.mean <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, mean, na.rm = TRUE),3)
even.sem <- round(apply(even.data[5:(4 + dim(even2)[2])], 1, sem, na.rm = TRUE),4)

# Make new dataframe merging design file and mean diversity
Brassica.div <- data.frame(design[sort(row.names(design)), ], rich.mean, even.mean)

# Take averages of technial reps
rich.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, rich = mean(rich.mean))
even.rep.ave <- ddply(Brassica.div, .(treatment, type, rep), summarize, even = mean(even.mean))

# Reshape data
```

```r
rich.2 <- reshape(rich.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")

even.2 <- reshape(even.rep.ave[,1:4], timevar = "type",
                   idvar = c("treatment", "rep"), direction = "wide")


## Statistical test: two-way ANOVA of Rpf and metabolic status
# Soil bacterial richness
rich.anova.c <- aov(rich.mean ~ type + treatment + treatment*type, Brassica.div)
summary(rich.anova.c)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## type             1      3     2.5   0.003  0.957
## treatment        1     17    17.2   0.021  0.887
## type:treatment   1    509   509.4   0.619  0.443
## Residuals       16  13173   823.3
```

```r
# One-way ANOVA based on Metabolic status
trans <- Brassica.div[ which(Brassica.div$type == 'cDNA'), ]
gDNA <- Brassica.div[ which(Brassica.div$type == 'DNA'), ]

trans.aov <- aov(rich.mean ~ treatment, trans)
summary(trans.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1    170   169.7   0.142  0.716
## Residuals    8   9551  1193.9
```

```r
gDNA.aov <- aov(rich.mean ~ treatment, gDNA)
summary(gDNA.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1    357   356.9   0.788  0.401
## Residuals    8   3622   452.8
```

```r
# Soil bacterial evenness
even.anova.c <- aov(even.mean ~ type + treatment + treatment*type, Brassica.div)
summary(even.anova.c)
```

```
##                 Df   Sum Sq  Mean Sq F value Pr(>F)  
## type             1 0.010215 0.010215   5.799 0.0285 *
## treatment        1 0.001345 0.001345   0.763 0.3952  
## type:treatment   1 0.000218 0.000218   0.124 0.7297  
## Residuals       16 0.028184 0.001762                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
trans.e.aov <- aov(even.mean ~ treatment, trans)
summary(trans.e.aov)
```

```
##             Df   Sum Sq   Mean Sq F value Pr(>F)
## treatment    1 0.000240 0.0002401   0.438  0.527
## Residuals    8 0.004382 0.0005478
```

```r
gDNA.e.aov <- aov(even.mean ~ treatment, gDNA)
summary(gDNA.e.aov)
```

```
##              Df   Sum Sq  Mean Sq F value Pr(>F)
## treatment     1 0.001322 0.001323   0.444  0.524
## Residuals     8 0.023802 0.002975
```

```
### Beta Diversity
# Make presence-absence matrix
Actino.OTU.PA <- (Actino.OTU > 0) * 1

# Make relative abundence matrix
Actino.OTU.REL <- Actino.OTU
for (i in 1:dim(Actino.OTU)[1]){
  Actino.OTU.REL[i,] <- Actino.OTU[i,]/sum(Actino.OTU[i,])
  }

# Log-transform relative abundances
Actino.OTU.REL.log <- decostand(Actino.OTU, method="log")

Brassica.bc.dis <- vegdist(Actino.OTU.REL.log, method = "bray", binary = "FALSE")
Brassica.dis.mean <- mean(Brassica.bc.dis)

# Principal Coordinates Analysis (PCoA)
Brassica.PCoA <- cmdscale(Brassica.bc.dis, eig = TRUE, k = 3)
explainvar1 <- round(Brassica.PCoA$eig[1] / sum(Brassica.PCoA$eig), 3) * 100
explainvar2 <- round(Brassica.PCoA$eig[2] / sum(Brassica.PCoA$eig), 3) * 100
explainvar3 <- round(Brassica.PCoA$eig[3] / sum(Brassica.PCoA$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

# Actino.OTU Scores
Actino.OTU.scores <- t(cor(Brassica.PCoA$points, Actino.OTU.REL))
Actino.OTU.scores <- as.matrix(Actino.OTU.scores)[,1:2]
Actino.OTU.scores <- Actino.OTU.scores[abs(Actino.OTU.scores[,1]) > 0.7|abs(Actino.OTU.scores[,2]) > 0.7

# Average BC Distance Between Treatments
Brassica.bc.dis.m <- as.matrix(Brassica.bc.dis)
all.equal(row.names(Brassica.div), rownames(Brassica.bc.dis.m))
```

```
## [1] TRUE
```

```
treatment.div <- unique(Brassica.div$treatment)
treatment.dis <- rep(NA, length(treatment.div))
for(i in 1:length(treatment.div)){
  temp <- row.names(Brassica.div[Brassica.div$treatment == treatment.div[i], ])
  treatment.dis[i] <- Brassica.bc.dis.m[temp[1], temp[2]]
}

mean(treatment.dis)
```

```
## [1] 0.4451541
```

```
# Plot figure #
png(filename="../figures/Figure5-ActinoOrdination.png",
    width = 1800, height = 800, res = 96*2)

layout(matrix(1:3, 1, 3), widths = c(20, 9, 2.5))

par(mar = c(7, 10, 1, 0) + 0.5)
```

```
plot(Brassica.PCoA$points[ ,1], Brassica.PCoA$points[ ,2],
     ylim = c(-0.4, 0.4), xlim = c(-0.5, 0.4),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""), line = 5,
     #xlab = "", ylab = "", xaxt = "n", yaxt = "n",
     pch = 22, cex = 2.0, type = "n", cex.lab = 2.5, cex.axis = 3,
     axes = FALSE)
```

```
## Warning in plot.window(...): "line" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "line" is not a graphical parameter
```

```
# Add Axes
axis(side = 1, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 2, labels = T, lwd.ticks = 3, cex.axis = 2, las = 1, tck=-0.025)
axis(side = 3, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
axis(side = 4, labels = F, lwd.ticks = 3, cex.axis = 1, las = 1, tck=-0.025)
abline(h = 0, v = 0, lty = 3)

box(lwd = 2)

# Subset data
all.equal(row.names(Brassica.PCoA$points), rownames(Brassica.div))
```

```
## [1] TRUE
```

```
Brassica.points <- data.frame(Brassica.PCoA$points, Brassica.div)

# Active community
Brassica.active.rpf <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.active.no <- Brassica.points[ which(Brassica.points$type == "cDNA" &
                                   Brassica.points$treatment == "Rpf-"), ]
# Total community
Brassica.total.rpf <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf+"), ]
Brassica.total.no <- Brassica.points[ which(Brassica.points$type == "DNA" &
                                   Brassica.points$treatment == "Rpf-"), ]

# Add points
# Active community Rpf+
points(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Active community Rpf-
points(Brassica.active.no[ ,1], Brassica.active.no[ ,2], pch = 21,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)
# Total community Rpf+
points(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "grey15", lwd= 2.5)
# Total community Rpf-
points(Brassica.total.no[ ,1], Brassica.total.no[ ,2], pch = 24,
       cex = 3.5, col = "Black", bg = "lightgrey", lwd= 2.5)

# Add elipcses
# Active Rpf+
ordiellipse(cbind(Brassica.active.rpf[ ,1], Brassica.active.rpf[ ,2]), Brassica.active.rpf$treatment, ki
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Active Rpf-
ordiellipse(cbind(Brassica.active.no[ ,1], Brassica.active.no[ ,2]), Brassica.active.no$treatment, kind=
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf+
ordiellipse(cbind(Brassica.total.rpf[ ,1], Brassica.total.rpf[ ,2]), Brassica.total.rpf$treatment, kind=
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)
# Total Rpf-
ordiellipse(cbind(Brassica.total.no[ ,1], Brassica.total.no[ ,2]), Brassica.total.no$treatment, kind="s
            lwd=2, lty=3, draw="lines", col="black", label=FALSE)

# Add Legend Outside
par(mar = c(4, 0, 5, 1) + 0.5)
plot.new()
legend(0, 1, c("Active, -Rpf", "Active, +Rpf", "Total, -Rpf", "Total, +Rpf"),
       pch = c(21, 21, 24, 24),
       col = c("Black", "Black", "Black", "Black"),
       pt.bg = c("lightgrey", "grey15", "lightgrey", "grey15"),
       bty = "n", y.intersp = 1, pt.cex = 3.2, cex = 2, lwd= 2, lty =  NA)

# Sample number label
# mtext(text = expression(italic("n")~" = 5"), line = 1, adj = -0.65, col="black", cex=1.5)

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Figure5-ActinoOrdination.png")
grid.raster(img)


# Adding pot number factor
Brassica.div$pot <- c(1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10)
Brassica.div$pot <- as.factor(Brassica.div$pot)

## Statistical test: PERMANOVA to test Rpf and metabolic status effects on Actinobacteria composition
act.bray.permanova <- adonis(Actino.OTU.REL ~ Brassica.div$type + Brassica.div$treatment +
                               Brassica.div$pot + Brassica.div$type*Brassica.div$treatment, method
                             = "bray", binary = FALSE, permutations = 999)
act.bray.permanova
```

```
##
## Call:
## adonis(formula = Actino.OTU.REL ~ Brassica.div$type + Brassica.div$treatment +      Brassica.div$pot
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Terms added sequentially (first to last)
##
##                                            Df SumsOfSqs MeanSqs F.Model
## Brassica.div$type                           1   0.80455 0.80455 16.0765
## Brassica.div$treatment                      1   0.16252 0.16252  3.2475
## Brassica.div$pot                            8   0.96850 0.12106  2.4191
## Brassica.div$type:Brassica.div$treatment    1   0.09792 0.09792  1.9566
## Residuals                                   8   0.40036 0.05004
## Total                                      19   2.43384
##                                                 R2 Pr(>F)
## Brassica.div$type                          0.33057  0.001 ***
## Brassica.div$treatment                     0.06677  0.004 **
## Brassica.div$pot                           0.39793  0.001 ***
## Brassica.div$type:Brassica.div$treatment 0.04023  0.056 .
## Residuals                                  0.16450
## Total                                      1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Indentify indicator species
spe.corr <- add.spec.scores(Brassica.PCoA, Actino.OTU.REL.log,
                            method = "cor.scores")$cproj
corrcut <- 0.7 # User defined cutoff
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]
imp.spp <- as.matrix(imp.spp)
imp.spp
```

```
##                 Dim1        Dim2         Dim3
## Otu00041  0.94381916  0.04814077  0.030967097
## Otu00043  0.76747042 -0.40614482  0.155918685
## Otu00088  0.72771554  0.12731720  0.223078014
## Otu00089  0.71206071  0.09387614  0.004679730
## Otu00107  0.96034286 -0.03379011  0.060128607
## Otu00123  0.35679161 -0.74185740 -0.099933755
## Otu00326  0.85921722 -0.02808484  0.061608151
## Otu00333  0.18382962 -0.76512659  0.320588965
## Otu00345  0.74148325 -0.39538094  0.129141400
## Otu00361  0.76285456  0.10517675  0.178617067
## Otu00362  0.72606371 -0.46317397  0.114809959
## Otu00396  0.92868880  0.10510438 -0.058518292
## Otu00444 -0.80940522 -0.12705206 -0.031491239
## Otu00445 -0.02434157 -0.76578469  0.394396102
## Otu00450  0.73826260  0.21670886 -0.315001896
## Otu00584  0.03294415 -0.72248314 -0.636192663
## Otu00588  0.94158759  0.05400797  0.013540349
## Otu00593  0.85116440  0.02216679 -0.090761376
## Otu00651  0.76075683 -0.26257639 -0.435906141
## Otu00689  0.74826755 -0.01220389  0.076093423
## Otu00712  0.89351956  0.02111334  0.105495379
## Otu00715 -0.80969329 -0.38173470  0.141080840
## Otu00723  0.78431568 -0.05579502  0.011597332
## Otu00724 -0.10052924 -0.79285640 -0.437547967
## Otu00739  0.15920435 -0.74623341  0.274755412
## Otu00745  0.79036612 -0.18697157 -0.105621360
## Otu00756  0.85990913  0.11222778  0.126606599
## Otu00759  0.87141381  0.04031918  0.110509999
## Otu00760  0.92321740  0.13203911  0.027538908
## Otu00763  0.92258282  0.06969988 -0.002099196
## Otu00774  0.19395639 -0.72674543 -0.321544522
## Otu00813  0.76841557 -0.20481054  0.026234766
## Otu00825  0.76991638  0.17225716 -0.160550974
## Otu00842 -0.01057316 -0.73557720 -0.350104641
## Otu00856 -0.08636043 -0.71878749 -0.062097879
## Otu00878 -0.19195504 -0.77447642 -0.471994619
## Otu00888  0.84831542  0.17994736 -0.217073986
## Otu00901  0.84731317 -0.10311912  0.004000382
## Otu00922 -0.09193357 -0.81327040  0.261037274
## Otu00932 -0.17259958 -0.74802641 -0.536560909
## Otu00933 -0.24154212 -0.83420833 -0.293960006
## Otu00957  0.76300096  0.03372929 -0.294099078
## Otu00958 -0.08090690 -0.82218252 -0.347714221
## Otu00986  0.03031647 -0.70112091  0.326914240
## Otu01000  0.24591308 -0.74562844  0.321822650
## Otu01013 -0.28227690 -0.72404705 -0.321193807
## Otu01098  0.71368384  0.14422806  0.249789863
## Otu01124  0.86916933  0.09327884 -0.255325138
## Otu01131  0.70155128 -0.30441649 -0.025081632
## Otu01178 -0.71004010 -0.55493149  0.187954145
## Otu01182 -0.75998207 -0.40892933  0.162553824
## Otu01201  0.25618214 -0.76734621  0.156302420
## Otu01201  0.25618214 -0.76734621  0.156302420
## Otu01213  0.12523198 -0.84924422  0.238270116
## Otu01373  0.86926126  0.08107373  0.081135035
## Otu01377  0.02030233 -0.77579802  0.205400064
## Otu01416 -0.15176134 -0.83628086 -0.460432539
## Otu01428  0.79177671 -0.01106019  0.360036165
## Otu01429 -0.21174689 -0.75413767 -0.415809264
## Otu01529 -0.87226912 -0.37098201  0.135003701
## Otu01531 -0.92069802 -0.23560267  0.116830492
## Otu01532  0.76488494 -0.38144633 -0.033365950
## Otu01544 -0.11646615 -0.84234001  0.233543212
## Otu01659  0.82445195 -0.05925605  0.049568086
## Otu01660 -0.49204701 -0.73938048  0.047598442
## Otu01680 -0.85606534 -0.16763737  0.106878296
## Otu01681 -0.45872532 -0.72942543  0.302201702
## Otu01693 -0.32673701 -0.70062192  0.046527733
## Otu01694  0.33203447 -0.76923285  0.116829942
## Otu01704  0.08768080 -0.86309152 -0.059373532
## Otu01726  0.79631757  0.06829790 -0.027216588
## Otu01878 -0.83575449 -0.15159472  0.176659455
## Otu01880 -0.16126686 -0.75017163 -0.329670170
## Otu01890 -0.08675671 -0.78227928  0.238140892
## Otu01896 -0.13140521 -0.79893093 -0.377451958
## Otu01897  0.14028872 -0.71839035  0.373101141
## Otu01903  0.72274109 -0.29922032 -0.267397434
## Otu01903  0.72274109 -0.29922032 -0.267397434
## Otu01904 -0.09219415 -0.79880465 -0.443575302
## Otu01928 -0.03938488 -0.77518163  0.082032167
## Otu01992  0.20854352 -0.77715788  0.173461530
## Otu02036  0.92687369  0.12922367  0.061688230
## Otu02075 -0.86572205 -0.32444381  0.036945765
## Otu02076 -0.37701728 -0.77714025 -0.128797482
## Otu02114 -0.77495267 -0.11988964  0.246189516
## Otu02169  0.79005749 -0.04603300  0.064850610
## Otu02388 -0.70333729 -0.39063688  0.137643387
## Otu02417 -0.20853884 -0.71579693 -0.436059832
## Otu02499 -0.70453786 -0.18444295 -0.326875901
## Otu02520  0.17377413 -0.79189333 -0.272104115
## Otu02566 -0.76649331 -0.17396218  0.076034771
## Otu02567 -0.23776256 -0.80184394 -0.377049771
## Otu02571  0.89485161 -0.02323996  0.116926125
## Otu02653  0.79640750  0.07521834  0.052122250
## Otu02725  0.05411807 -0.76433527 -0.185866812
## Otu02751 -0.04955620 -0.73246241  0.361264799
## Otu02760 -0.22536168 -0.80536498 -0.293029578
## Otu02802  0.01704045 -0.81423645 -0.174957838
## Otu03021 -0.31013801 -0.76229666 -0.170649031
## Otu03157  0.70358076  0.22148167  0.181039797
## Otu03198 -0.75279377 -0.13957898  0.018556493
## Otu03231  0.76918439  0.01075444  0.116768179
## Otu03233  0.72934743  0.08339050  0.009881609
## Otu03277  0.77494079 -0.09218918  0.229586402
## Otu03325 -0.26292237 -0.72625785 -0.408091887
## Otu03328  0.71595568  0.27210352 -0.145862863
## Otu03350  0.07789583 -0.71737784 -0.276124139
## Otu03532  0.73999685 -0.29067449  0.296543422
## Otu03545  0.71008310  0.29124450  0.213096324
## Otu03548  0.76509529  0.16740101 -0.069834111
## Otu03572  0.78869771  0.14218137 -0.168501407
## Otu04357  0.71330667  0.20025483 -0.083480816
## Otu04394  0.80472799  0.02142752  0.093810715
## Otu04397  0.73347732  0.02158065  0.254017812
## Otu04398  0.76764141  0.13158320 -0.047492433
## Otu04735  0.83743813  0.16102724  0.092104639
## Otu05581  0.75723284  0.17083136 -0.172265928
## Otu05925  0.71068597  0.06824637  0.278048352
## Otu05962  0.71406301 -0.08310418  0.083123954
## Otu06174  0.80491581 -0.11868848  0.220953336
## Otu06738  0.77733257 -0.12573453 -0.029234751
## Otu06866  0.27692177 -0.72148358  0.115500871
## Otu07469  0.17085264 -0.71805056  0.008971341
## Otu09869  0.70301475  0.03917627 -0.115245892
## Otu11200  0.72522449 -0.02015755  0.356809823
## Otu11200  0.72522449 -0.02015755  0.356809823
```

```
imp.otu <- as.vector(rownames(imp.spp))
imp.otu
```

```
##   [1] "Otu00041" "Otu00043" "Otu00088" "Otu00089" "Otu00107" "Otu00123"
##   [7] "Otu00326" "Otu00333" "Otu00345" "Otu00361" "Otu00362" "Otu00396"
##  [13] "Otu00444" "Otu00445" "Otu00450" "Otu00584" "Otu00588" "Otu00593"
##  [19] "Otu00651" "Otu00689" "Otu00712" "Otu00715" "Otu00723" "Otu00724"
##  [25] "Otu00739" "Otu00745" "Otu00756" "Otu00759" "Otu00760" "Otu00763"
##  [31] "Otu00774" "Otu00813" "Otu00825" "Otu00842" "Otu00856" "Otu00878"
##  [37] "Otu00888" "Otu00901" "Otu00922" "Otu00932" "Otu00933" "Otu00957"
##  [43] "Otu00958" "Otu00986" "Otu01000" "Otu01013" "Otu01098" "Otu01124"
##  [49] "Otu01131" "Otu01178" "Otu01182" "Otu01201" "Otu01201" "Otu01213"
##  [55] "Otu01373" "Otu01377" "Otu01416" "Otu01428" "Otu01429" "Otu01529"
##  [61] "Otu01531" "Otu01532" "Otu01544" "Otu01659" "Otu01660" "Otu01680"
##  [67] "Otu01681" "Otu01693" "Otu01694" "Otu01704" "Otu01726" "Otu01878"
##  [73] "Otu01880" "Otu01890" "Otu01896" "Otu01897" "Otu01903" "Otu01903"
##  [79] "Otu01904" "Otu01928" "Otu01992" "Otu02036" "Otu02075" "Otu02076"
##  [85] "Otu02114" "Otu02169" "Otu02388" "Otu02417" "Otu02499" "Otu02520"
##  [91] "Otu02566" "Otu02567" "Otu02571" "Otu02653" "Otu02725" "Otu02751"
##  [97] "Otu02760" "Otu02802" "Otu03021" "Otu03157" "Otu03198" "Otu03231"
## [103] "Otu03233" "Otu03277" "Otu03325" "Otu03328" "Otu03350" "Otu03532"
## [109] "Otu03545" "Otu03548" "Otu03572" "Otu04357" "Otu04394" "Otu04397"
## [115] "Otu04398" "Otu04735" "Otu05581" "Otu05925" "Otu05962" "Otu06174"
## [121] "Otu06738" "Otu06866" "Otu07469" "Otu09869" "Otu11200" "Otu11200"
```

```
imp.spp.lst.1 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[1:48]), ]
imp.spp.lst.2 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[48:57]), ]
imp.spp.lst.3 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[57:68]), ]
imp.spp.lst.4 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[69:77]), ]
imp.spp.lst.5 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[78:89]), ]
imp.spp.lst.6 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[90:99]), ]
imp.spp.lst.7 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[100:109]), ]
imp.spp.lst.8 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[110:119]), ]
imp.spp.lst.8 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[120:125]), ]
imp.spp.lst.9 <- OTU.tax[ which(OTU.tax$OTU == imp.otu[126]), ]

imp.spp.lst <- rbind(imp.spp.lst.1, imp.spp.lst.2, imp.spp.lst.3, imp.spp.lst.4, imp.spp.lst.5,
                     imp.spp.lst.6, imp.spp.lst.7, imp.spp.lst.8, imp.spp.lst.9)

fit <- envfit(Brassica.PCoA, Actino.OTU.REL.log, perm = 999)
```

6) Supplementary materials

Gram positive: Gram negative ratios

```
# Subsetting
OTU.Rpf <- OTU[c(6:10, 16:20), ]
Gpos.OTU.Rpf <- Gpos.OTU[c(6:10, 16:20), ]

OTU.Con <- OTU[c(1:5, 11:15), ]
Gpos.OTU.Con <- Gpos.OTU[c(1:5, 11:15), ]

# Calculate sum reads for Gram positive
Gram.pos.Rpf <- as.vector(rowSums(Gpos.OTU.Rpf))
All.Rpf <- as.vector(rowSums(OTU.Rpf))
Gram.pos.Con <- as.vector(rowSums(Gpos.OTU.Con))
All.Con <- as.vector(rowSums(OTU.Con))

# Calculate Gram negative read sum
Gram.neg.Rpf <- All.Rpf - Gram.pos.Rpf
Gram.neg.Con <- All.Con - Gram.pos.Con
```

```
# Calculate ratio of Gram postive : Gram negative values
Rpf.Gram.Ratio <- Gram.pos.Rpf / Gram.neg.Rpf  ## points for Rpf+ Ratio
Con.Gram.Ratio <- Gram.pos.Con / Gram.neg.Con  ## points for Rpf- Ratio

# Generate data table for figure
dat <- cbind(Rpf.Gram.Ratio, Con.Gram.Ratio)
dat.m <- melt(dat)
dat.m <- dat.m[, 2:3]
colnames(dat.m) <- c("Treatment", "Ratio")
dat.m$Treatment <- gsub('Rpf.Gram.Ratio', 'Rpf+', dat.m$Treatment)
dat.m$Treatment <- gsub('Con.Gram.Ratio', 'Rpf-', dat.m$Treatment)

# Gram Ratio table #
GramRatio.mean <- aggregate(dat.m$Ratio ~ Treatment, dat.m, mean)
GramRatio.sem <- aggregate(dat.m$Ratio ~ Treatment, dat.m, sem)
GramRatio.sem.LL <- GramRatio.mean[2] + GramRatio.sem[2]
GramRatio.sem.UL <- GramRatio.mean[2] - GramRatio.sem[2]
GramRatio.table <- data.frame(GramRatio.mean[1], GramRatio.mean[2], GramRatio.sem[2],
          GramRatio.sem.LL[1], GramRatio.sem.UL[1])
colnames(GramRatio.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
GramRatio.table <- GramRatio.table[order(GramRatio.table$mean),]


# Plotting Gram Ratio #
png(filename="../figures/Suppl.Fig2a.GramRatio.png",
    width = 800, height = 800, res = 96*2)
par(mar = c(5, 5, 1, 1))

arabid.fig <- plot(jitter(rep(1, length(Con.Gram.Ratio)), amount = 0.1), Con.Gram.Ratio,
      ylim = c(0, 0.2), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(Rpf.Gram.Ratio)), amount = 0.1), Rpf.Gram.Ratio, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data point for each treatment #
points(1, mean(Con.Gram.Ratio), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(Rpf.Gram.Ratio), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis labels
mtext(expression('Gram positive : Gram negative'), side = 2,
      outer = FALSE, cex = 1.5, line = 3.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0", "0.1", "0.2"),
     at = c(0.0, 0.1, 0.2))
```

```
axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0, 0.1, 0.2), labels = F, tck = -0.02)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("Rpf-", "Rpf+"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.02)

# Adding SEM #
arrows(x0 = c(2,1), y0 = GramRatio.table$mean, y1 = GramRatio.table$LCI, angle = 90,
       length = 0.25, lwd = 2)
arrows(x0 = c(2,1), y0 = GramRatio.table$mean, y1 = GramRatio.table$UCI, angle = 90,
       length=0.25, lwd = 2)

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Suppl.Fig2a.GramRatio.png")
grid.raster(img)


# t-test for effect of RPf
Gram.Ratio.ttest <- t.test(Con.Gram.Ratio, Rpf.Gram.Ratio, alternative="greater")
Gram.Ratio.ttest  # Non-significant: t = 1.2977, df = 15.203, p = 0.1069
```

```
##
##  Welch Two Sample t-test
##
## data:  Con.Gram.Ratio and Rpf.Gram.Ratio
## t = 1.2977, df = 15.203, p-value = 0.1069
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.004341151          Inf
## sample estimates:
##  mean of x  mean of y
## 0.09961739 0.08720269
```

Actinobacteria: proprotions

```
# Subsetting based on Rpf treatment
# +Rpf treatment
OTU.Rpf <- OTU[c(6:10, 16:20), ]
Actino.OTU.Rpf <- Actino.OTU[c(6:10, 16:20), ]
# -Rpf treatment
OTU.Con <- OTU[c(1:5, 11:15), ]
Actino.OTU.Con <- Actino.OTU[c(1:5, 11:15), ]
```

```
# Calculate within sample sum reads for Actino and all 16S based on treatment
Actino.Rpf <- as.vector(rowSums(Actino.OTU.Rpf))
All.Rpf <- as.vector(rowSums(OTU.Rpf))
Actino.Con <- as.vector(rowSums(Actino.OTU.Con))
All.Con <- as.vector(rowSums(OTU.Con))

# Calculate Gram negative read sum
#Gram.neg.Rpf <- All.Rpf - Actino.Rpf
#Gram.neg.Con <- All.Con - Actino.Con

# Calculate reads ratio of Actinobacteria : all 16S rRNA
Rpf.Act.Ratio <- Actino.Rpf / All.Rpf
Con.Act.Ratio <- Actino.Con / All.Con

# Generate data table for figure
dat <- cbind(Rpf.Act.Ratio, Con.Act.Ratio)
dat.m <- melt(dat)
dat.m <- dat.m[, 2:3]
colnames(dat.m) <- c("Treatment", "Ratio")
dat.m$Treatment <- gsub('Rpf.Act.Ratio', 'Rpf+', dat.m$Treatment)
dat.m$Treatment <- gsub('Con.Act.Ratio', 'Rpf-', dat.m$Treatment)

# Gram Ratio table #
ActRatio.mean <- aggregate(dat.m$Ratio ~ Treatment, dat.m, mean)
ActRatio.sem <- aggregate(dat.m$Ratio ~ Treatment, dat.m, sem)
ActRatio.sem.LL <- ActRatio.mean[2] + ActRatio.sem[2]
ActRatio.sem.UL <- ActRatio.mean[2] - ActRatio.sem[2]
ActRatio.table <- data.frame(ActRatio.mean[1], ActRatio.mean[2], ActRatio.sem[2],
          ActRatio.sem.LL[1], ActRatio.sem.UL[1])
colnames(ActRatio.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
ActRatio.table <- ActRatio.table[order(ActRatio.table$mean),]


# Plotting Actino Ratio #
png(filename="../figures/Suppl.Fig2.ActinoProportion.png",
    width = 800, height = 800, res = 96*2)
par(mar = c(5, 5, 1, 1))

arabid.fig <- plot(jitter(rep(1, length(Con.Act.Ratio)), amount = 0.1), Con.Act.Ratio,
      ylim = c(0, 0.2), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(Rpf.Act.Ratio)), amount = 0.1), Rpf.Act.Ratio, pch = 21,
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data point for each treatment #
points(1, mean(Con.Act.Ratio), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(Rpf.Act.Ratio), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)
```

```
# Y axis labels
mtext(expression('Proportion Actinobacteria'), side = 2,
      outer = FALSE, cex = 1.5, line = 3.5, adj = 0.5)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0", "0.1", "0.2"),
     at = c(0.0, 0.1, 0.2))

axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0, 0.1, 0.2), labels = F, tck = -0.02)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.02)

# Adding SEM #
arrows(x0 = c(2,1), y0 = ActRatio.table$mean, y1 = ActRatio.table$LCI, angle = 90,
       length = 0.25, lwd = 2)
arrows(x0 = c(2,1), y0 = ActRatio.table$mean, y1 = ActRatio.table$UCI, angle = 90,
       length=0.25, lwd = 2)

# p-value
mtext(text = expression(italic("P")~" = 0.097") , side =3, line = -1.2, adj = 0.925, col="black", cex=1
# Sample number label
#mtext(text = expression(italic("n")~" = 10"), side = 3, line = -2.2, adj = 0.925, col="black", cex=1.2

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Suppl.Fig2.ActinoProportion.png")
grid.raster(img)


# Statistics
Act.Ratio.ttest <- t.test(Con.Act.Ratio, Rpf.Act.Ratio, alternative="greater")
Act.Ratio.ttest  # Non-significant: t = 1.3691, df = 13.303, p = 0.09682
```

```
##
##  Welch Two Sample t-test
##
## data:  Con.Act.Ratio and Rpf.Act.Ratio
## t = 1.3653, df = 13.669, p-value = 0.09711
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.002782881          Inf
## sample estimates:
##  mean of x  mean of y
## 0.06326911 0.05374770
```

Control Arabidopsis experiment

```
# Load dataset
seedling <- read.delim("~/Github/BrassicaRpf/data/seedlingbiomass.txt", sep = ",", head = TRUE)

# Calculate relative biomass #
seedling$Biomass <- (seedling$BiomassPixel)/(seedling$PlatePixel)
seedling$RelativeBiomass <- (seedling$Biomass)/(seedling$Seedlings)*100

# Biomass data points #
seedling.rpf <- seedling[ which(seedling$Treatment == "Rpf+"),]
seedling.con <- seedling[ which(seedling$Treatment == "Rpf-"),]

# Biomass data table #
seedling.mean <- aggregate(seedling$RelativeBiomass ~ Treatment, seedling, mean)
seedling.sem <- aggregate(seedling$RelativeBiomass ~ Treatment, seedling, sem)
seedling.sem.LL <- seedling.mean[2] + seedling.sem[2]
seedling.sem.UL <- seedling.mean[2] - seedling.sem[2]
seedling.table <- data.frame(seedling.mean[1], seedling.mean[2], seedling.sem[2],
          seedling.sem.LL[1], seedling.sem.UL[1])
colnames(seedling.table) <- c("Treatment", "mean", "sem", "LCI", "UCI")
seedling.table <- seedling.table[order(seedling.table$mean),]

# Plotting Arabidopsis biomass #
png(filename="../figures/Suppl.Fig5.Arabidopsis.png",
    width = 800, height = 800, res = 96*2)
par(mar = c(5, 5, 1, 1))

arabid.fig <- plot(jitter(rep(1, length(seedling.con$RelativeBiomass)), amount = 0.1), seedling.con$Rela
      ylim = c(0, 1), xlim = c(0.5, 2.5), pch = 21, col = "lightgrey", bg = "lightgrey", lwd = 3.5,
      cex = 1.7, yaxt = "n", xaxt = "n", cex.lab = 2, cex.axis = 2,
      las = 1, ylab = "", xlab = "")
      box(lwd = 2)
points(jitter(rep(2, length(seedling.rpf$RelativeBiomass)), amount = 0.1), seedling.rpf$RelativeBiomass
       bg = "lightgrey", col = "lightgrey", lwd = 2, cex = 1.7)

# Adding mean data point for each treatment #
points(1, mean(seedling.con$RelativeBiomass), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)
points(2, mean(seedling.rpf$RelativeBiomass), pch = 21, col = "black",
       bg = "NA", lwd = 2, cex = 2.5)

box(lwd = 2)

# Y axis labels
mtext(expression('Relative Biomass'), side = 2,
      outer = FALSE, cex = 1.5, line = 3.5, adj = 0.5)

# Major Axes
```

```
axis(side = 2, lwd.ticks = 2, cex.axis = 1, las = 1,
     labels = c("0.0", "0.5", "1.0"),
     at = c(0.0, 0.5, 1.0))

axis(side = 4, lwd.ticks = 2, cex.axis = 1, las = 1,
     at=c(0.0, 0.5, 1.0), labels = F, tck = -0.02)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1,
     labels = c("-Rpf", "+Rpf"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
     at = c(1, 2), labels = F, tck = -0.02)

# Adding SEM #
arrows(x0 = c(2,1), y0 = seedling.table$mean, y1 = seedling.table$LCI, angle = 90,
       length = 0.25, lwd = 2)
arrows(x0 = c(2,1), y0 = seedling.table$mean, y1 = seedling.table$UCI, angle = 90,
       length=0.25, lwd = 2)

# p-value
mtext(text = expression(italic("P")~" = 0.320") , side =3, line = -1.2, adj = 0.925, col="black", cex=1
# Sample number label
#mtext(text = expression(italic("n")~" = 4"), side = 3, line = -2.2, adj = 0.925, col="black", cex=1.25

# Close Plot Device
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG("../figures/Suppl.Fig5.Arabidopsis.png")
grid.raster(img)


# Statistical test: t-test of Rpf effects on Arabdiopsis plant biomass
Arabid.rpf <- seedling[ which(seedling$Treatment == "Rpf+"), ]
Arabid.con <- seedling[ which(seedling$Treatment == "Rpf-"), ]
Arabid.ttest <- t.test(Arabid.rpf$RelativeBiomass, Arabid.con$RelativeBiomass)
Arabid.ttest
```

```
##
##  Welch Two Sample t-test
##
## data:  Arabid.rpf$RelativeBiomass and Arabid.con$RelativeBiomass
## t = -1.1134, df = 4.6148, p-value = 0.3201
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3878870  0.1575483
## sample estimates:
## mean of x mean of y
## 0.5348800 0.6500494
```

```r
anova <- aov(seedling$RelativeBiomass ~ seedling$Treatment, data = seedling)
summary(anova)
```

```
##                    Df  Sum Sq Mean Sq F value Pr(>F)
## seedling$Treatment  1 0.02274 0.02274   1.214  0.321
## Residuals           5 0.09363 0.01873
```

```r
TukeyHSD(anova)  # Non-significant: p = 0.321
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = seedling$RelativeBiomass ~ seedling$Treatment, data = seedling)
##
## $`seedling$Treatment`
##                 diff        lwr       upr     p adj
## Rpf+-Rpf- -0.1151693 -0.3838394 0.1535007 0.3206918
```