Machine Learning Reveals Microbial Taxa Associated with a Swim Across the Pacific Ocean

Garry Lewis, Sebastian Reczek, Osayenmwen, Jarrad Hampton-Marcell Omozusi,

2024-05-30

# Set up

Load the necessary libraries.

# Load Packages  
library(phyloseq)  
library(ggplot2)  
library(ggsci)  
library(ggpubr)  
library(vegan)  
library(sf)  
library(rnaturalearth)  
library(rnaturalearthdata)  
library(ggspatial)  
library(ggrepel)  
library(googleway)  
library(cowplot)  
library(patchwork)  
library(microbiome)  
library(dplyr)  
library(broom)  
library(fpc)  
library(cluster)  
library(reshape)  
library(reshape2)  
library(randomForest)  
library(ggpmisc)  
library(knitr)

# Data Processing

# Load Data  
setwd("C:/Gut microbiome research/R studio raw files/Old codes")  
file <- import\_biom("swimmer\_w\_tax.biom")  
map <- import\_qiime\_sample\_data("swimmer\_mapping\_file\_modUPDATE.txt")  
  
# Merge Data Together  
swimmer <- merge\_phyloseq(file, map)  
  
# Remove Single OTUs  
swimmer.filt = prune\_taxa(taxa\_sums(swimmer) > 1, swimmer)  
  
# Normalize Data (10,000 seqs/sample)  
set.seed(8)  
swimmer10k = rarefy\_even\_depth(swimmer.filt, sample.size = 10000, rngseed = TRUE)  
  
# Rename Taxonomic Levels from Rank to Actual Name  
colnames(tax\_table(swimmer.filt)) <- c("Domain",  
 "Phylum",  
 "Class",  
 "Order",  
 "Family",  
 "Genus",  
 "Species")

# Quartile calculations for Table 1

#### Swim dist  
# Define the number of days in each quartile  
days\_per\_quartile <- c(27, 27, 26, 27)  
  
# make data frame  
swimdist <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, Daily\_Distance\_Nm=sample\_data(swimmer.filt)$Daily\_Distance\_Nm, Day =sample\_data(swimmer.filt)$Day)  
  
# Create a Quartile column based on the days  
swimdist <- swimdist %>%  
 mutate(Quartile = case\_when(  
 Day <= sum(days\_per\_quartile[1]) ~ "Q1",  
 Day <= sum(days\_per\_quartile[1:2]) ~ "Q2",  
 Day <= sum(days\_per\_quartile[1:3]) ~ "Q3",  
 TRUE ~ "Q4"  
 ))  
  
swimdist[swimdist == 0] = NA  
  
# Summarize data into quartiles  
quartile\_summary <- swimdist %>%  
 group\_by(Quartile) %>%  
 summarise(  
 Mean\_Daily\_Distance\_Nm = mean(Daily\_Distance\_Nm, na.rm = TRUE),  
 SD\_Daily\_Distance\_Nm = sd(Daily\_Distance\_Nm, na.rm = TRUE),  
 Count = n()  
 )  
  
# Print the summary table using kable  
kable(quartile\_summary, caption = "Summary of Daily Swim Distance by Quartile")

Summary of Daily Swim Distance by Quartile

| Quartile | Mean\_Daily\_Distance\_Nm | SD\_Daily\_Distance\_Nm | Count |
| --- | --- | --- | --- |
| Q1 | 22.38538 | 11.781160 | 18 |
| Q2 | 20.23714 | 6.915306 | 27 |
| Q3 | 14.66286 | 6.864890 | 26 |
| Q4 | 12.57867 | 5.037580 | 23 |

anova\_result <- aov(Daily\_Distance\_Nm ~ Quartile, data = swimdist)  
anova\_summary <- summary(anova\_result)  
print(anova\_summary)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Quartile 3 998 332.6 5.6 0.00175 \*\*  
## Residuals 66 3920 59.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 24 observations deleted due to missingness

######BP   
# Define the number of days in each quartile  
days\_per\_quartile <- c(27, 27, 26, 27)  
  
swimBP <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, BP = sample\_data(swimmer.filt)$Systolic\_BP, Day = sample\_data(swimmer.filt)$Day)  
  
# Create a Quartile column based on the days  
swimBP <- swimBP %>%  
 mutate(Quartile = case\_when(  
 Day <= sum(days\_per\_quartile[1]) ~ "Q1",  
 Day <= sum(days\_per\_quartile[1:2]) ~ "Q2",  
 Day <= sum(days\_per\_quartile[1:3]) ~ "Q3",  
 TRUE ~ "Q4"  
 ))  
  
swimBP[swimBP == 0] = NA  
  
# Summarize data into quartiles  
quartile\_summary\_BP <- swimBP %>%  
 group\_by(Quartile) %>%  
 summarise(  
 Mean\_BP = mean(BP, na.rm = TRUE),  
 SD\_BP = sd(BP, na.rm = TRUE),  
 Count = n()  
 )  
  
# Print the summary table using kable  
kable(quartile\_summary\_BP, caption = "Summary of Daily Blood pressure by Quartile")

Summary of Daily Blood pressure by Quartile

| Quartile | Mean\_BP | SD\_BP | Count |
| --- | --- | --- | --- |
| Q1 | 115.4000 | 5.272570 | 18 |
| Q2 | 127.2353 | 9.705214 | 27 |
| Q3 | 133.4545 | 10.727196 | 26 |
| Q4 | 132.5000 | 14.180973 | 23 |

anova\_result\_BP <- aov(BP ~ Quartile, data = swimBP)  
anova\_summary\_BP <- summary(anova\_result\_BP)  
print(anova\_summary\_BP)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Quartile 3 1249 416.5 3.862 0.0174 \*  
## Residuals 35 3774 107.8   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 55 observations deleted due to missingness

#### Heart Rate  
  
# Define the number of days in each quartile  
days\_per\_quartile <- c(27, 27, 26, 27)  
  
swimHR <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, Heart\_Rate = sample\_data(swimmer.filt)$Heart\_Rate, Day = sample\_data(swimmer.filt)$Day)  
  
# Create a Quartile column based on the days  
swimHR <- swimHR %>%  
 mutate(Quartile = case\_when(  
 Day <= sum(days\_per\_quartile[1]) ~ "Q1",  
 Day <= sum(days\_per\_quartile[1:2]) ~ "Q2",  
 Day <= sum(days\_per\_quartile[1:3]) ~ "Q3",  
 TRUE ~ "Q4"  
 ))  
  
swimHR[swimHR == 0] = NA  
  
# Summarize data into quartiles  
quartile\_summary\_HR <- swimHR %>%  
 group\_by(Quartile) %>%  
 summarise(  
 Mean\_Heart\_Rate = mean(Heart\_Rate, na.rm = TRUE),  
 SD\_Heart\_Rate = sd(Heart\_Rate, na.rm = TRUE),  
 Count = n()  
 )  
  
# Print the summary table using kable  
kable(quartile\_summary\_HR, caption = "Summary of Daily Heart rate by Quartile")

Summary of Daily Heart rate by Quartile

| Quartile | Mean\_Heart\_Rate | SD\_Heart\_Rate | Count |
| --- | --- | --- | --- |
| Q1 | 64.40000 | 9.071935 | 18 |
| Q2 | 67.35294 | 5.430713 | 27 |
| Q3 | 64.09091 | 6.625021 | 26 |
| Q4 | 64.83333 | 8.328665 | 23 |

anova\_result\_HR <- aov(Heart\_Rate ~ Quartile, data = swimHR)  
anova\_summary\_HR <- summary(anova\_result\_HR)  
print(anova\_summary\_HR)

## Df Sum Sq Mean Sq F value Pr(>F)  
## Quartile 3 87.8 29.28 0.646 0.591  
## Residuals 35 1586.8 45.34   
## 55 observations deleted due to missingness

##### Body Fat comp  
  
# Define the number of days in each quartile  
days\_per\_quartile <- c(27, 27, 26, 27)  
  
swimfatcomp <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, BFC = sample\_data(swimmer.filt)$Percent\_Fat, Day = sample\_data(swimmer.filt)$Day)  
  
# Create a Quartile column based on the days  
swimfatcomp <- swimfatcomp %>%  
 mutate(Quartile = case\_when(  
 Day <= sum(days\_per\_quartile[1]) ~ "Q1",  
 Day <= sum(days\_per\_quartile[1:2]) ~ "Q2",  
 Day <= sum(days\_per\_quartile[1:3]) ~ "Q3",  
 TRUE ~ "Q4"  
 ))  
  
swimfatcomp[swimfatcomp == 0] = NA  
  
# Summarize data into quartiles  
quartile\_summary\_fat <- swimfatcomp %>%  
 group\_by(Quartile) %>%  
 summarise(  
 Mean\_BFC = mean(BFC, na.rm = TRUE),  
 SD\_BFC = sd(BFC, na.rm = TRUE),  
 Count = n()  
 )  
  
# Print the summary table using kable  
kable(quartile\_summary\_fat, caption = "Summary of Daily Fat compostion by Quartile")

Summary of Daily Fat compostion by Quartile

| Quartile | Mean\_BFC | SD\_BFC | Count |
| --- | --- | --- | --- |
| Q1 | 21.00000 | 0.0000000 | 18 |
| Q2 | 20.42222 | 1.0084366 | 27 |
| Q3 | 18.18750 | 0.2587746 | 26 |
| Q4 | 18.00000 | 0.0000000 | 23 |

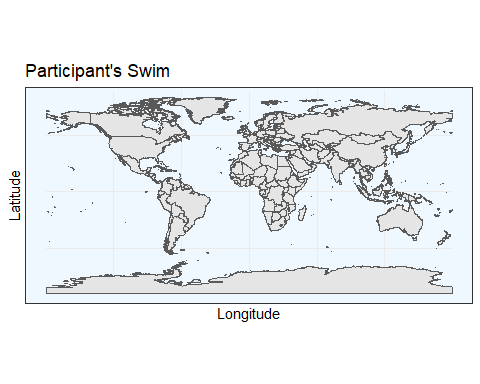
anova\_result\_fat <- aov(BFC ~ Quartile, data = swimfatcomp)  
anova\_summary\_fat <- summary(anova\_result\_fat)  
print(anova\_summary\_fat)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Quartile 3 42.93 14.311 38.26 4.17e-09 \*\*\*  
## Residuals 23 8.60 0.374   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 67 observations deleted due to missingness

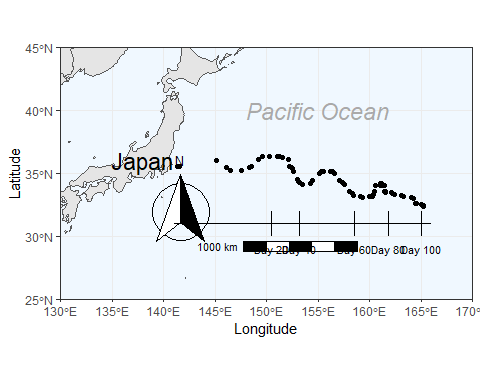
# Visualizing Daily Swim - Figure 1A

Map of participant’s swim.

# Map of participant's swim - Figure 1A  
map\_swimmer <- swimmer  
  
# Set theme for figure  
theme\_set(theme\_bw())  
  
# Load world map from internet  
world <- ne\_countries(scale = "medium", returnclass = "sf")  
  
ggplot(data = world) +  
 geom\_sf() +  
 xlab("Longitude") +  
 ylab("Latitude") +  
 ggtitle("Participant's Swim") +  
 theme(panel.background = element\_rect(fill = "aliceblue"))



# This is for the data to be put into dataframes  
endsites <- data.frame(longitude = sample\_data(map\_swimmer)$Stop\_Long, latitude = sample\_data(map\_swimmer)$Stop\_Lat)  
  
# This is the graph with the data on it  
bmap <- ggplot(data = world) +  
 geom\_sf() +  
 geom\_point(data = endsites, aes(x = longitude, y = latitude), size = 1.5, shape = 19, color = "black") +  
 coord\_sf(xlim = c(130, 170), ylim = c(25, 45), expand = FALSE) +  
 xlab("Longitude") +  
 ylab("Latitude") +  
 theme(panel.background = element\_rect(fill = "aliceblue")) +  
 annotation\_scale(location = "br", width\_hint = .4, pad\_x = unit(1.2, "in"), pad\_y = unit(.5, "in")) +  
 annotation\_north\_arrow(location = "bl", which\_north = "true", height = unit(1, "in"), width = unit(1, "in"),   
 pad\_x = unit(0.75, "in"), pad\_y = unit(0.5, "in"),  
 style = north\_arrow\_fancy\_orienteering) +  
 annotate(geom = "text", x = 138, y = 36, label = "Japan", color = "black", size = 6) +  
 annotate(geom = "text", x = 155, y = 40, label = "Pacific Ocean", fontface = "italic", color = "darkgrey", size = 6)  
  
# Add annotation markers for approximate day - horizontal line with markers  
fig1a <- bmap +  
 geom\_segment(aes(x = 141.1, y = 31, xend = 166, yend = 31)) +  
 geom\_segment(aes(x = 150.4776, y = 30, xend = 150.4776, yend = 32)) + # Day 20 (I used 21 bc 20 was absent)  
 annotate(geom = "text", x = 150.4776, y = 29, label = "Day 20", color = "black", size = 3) +  
 geom\_segment(aes(x = 153.1467, y = 30, xend = 153.1467, yend = 32)) + # Day 40  
 annotate(geom = "text", x = 153.1467, y = 29, label = "Day 40", color = "black", size = 3) +  
 geom\_segment(aes(x = 158.5011, y = 30, xend = 158.5011, yend = 32)) + # Day 60  
 annotate(geom = "text", x = 158.5011, y = 29, label = "Day 60", color = "black", size = 3) +  
 geom\_segment(aes(x = 161.8423, y = 30, xend = 161.8423, yend = 32)) + # Day 80  
 annotate(geom = "text", x = 161.8423, y = 29, label = "Day 80", color = "black", size = 3) +  
 geom\_segment(aes(x = 165.0494, y = 30, xend = 165.0494, yend = 32)) + # Day 100  
 annotate(geom = "text", x = 165.0494, y = 29, label = "Day 100", color = "black", size = 3)  
  
# Display the map with annotations  
fig1a

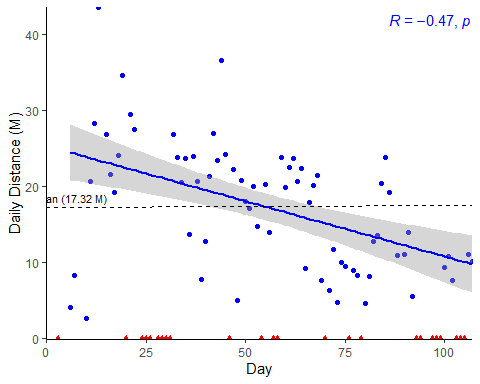


# Daily Swim Distance Over Time - Figure 1B

#Manipulating files  
dist\_swimmer <- swimmer  
  
#Makes a data frame out of sample data  
dailydist <- data.frame(sample\_data(dist\_swimmer)$Daily\_Distance\_Nm)  
  
#Turns 0 values into NA  
dailydist[dailydist == 0] = NA  
  
#Turns data frame into list  
dailylist <- as.list(dailydist)  
  
#Turn list into vector and sum total distance = 1212.59 nautical mi  
dailyvector <- unlist(dailylist, use.names = FALSE)  
dailyvectoromit <- dailyvector[!is.na(dailyvector)]  
totaldist <- sum(dailyvectoromit)  
  
#Participant swam for 70 days in total  
length(dailyvectoromit)

## [1] 70

#Average swim distance over actual days swam (1212.59/70) = 17.32271 nautical miles per day = 32.4355 km/day  
Avgswim <- totaldist/length(dailyvectoromit)  
  
#Turns 0 values into NA  
swimdist <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, Daily\_Distance\_Nm=sample\_data(swimmer.filt)$Daily\_Distance\_Nm, Day =sample\_data(swimmer.filt)$Day)  
swimdist2 <- swimdist %>% arrange(Day)  
swimdist2[swimdist2 == 0] = NA  
swimdist3 <- na.omit(swimdist2)  
  
#Standard deviation of swim distance = 10.51 nautical miles per day  
SwimSD <- sd(swimdist3$Daily\_Distance\_Nm)  
  
#Generate the figures  
fig1b <- ggplot(map, aes(x = Day, y = Daily\_Distance\_Nm, color = Swim)) +  
 geom\_point(data = subset(map, Swim == "Y"), color = "blue") +  
 geom\_point(data = subset(map, Swim != "Y"), shape = 17, color = "red") +  
 stat\_smooth(method = "glm", data = subset(map, Swim == "Y"), color = "blue") +  
 stat\_cor(method = "pearson", data = subset(map, Swim == "Y"), label.x = 86, color = "blue") +  
 theme\_classic() +  
 labs(x = "Day", y = "Daily Distance (M)") +  
 geom\_segment(aes(x = 0, y = 17.32, xend = 107, yend = 17.51), color = "black", linetype = "dashed") +  
 annotate(geom = "text", x = 6, y = 18.51, label = "Mean (17.32 M)", color = "black", size = 3) +  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, NA)) +  
 scale\_y\_continuous(expand = c(0, 0), limits = c(-.1, NA)) +  
 theme(legend.position = "none")  
  
#Display the figure   
fig1b

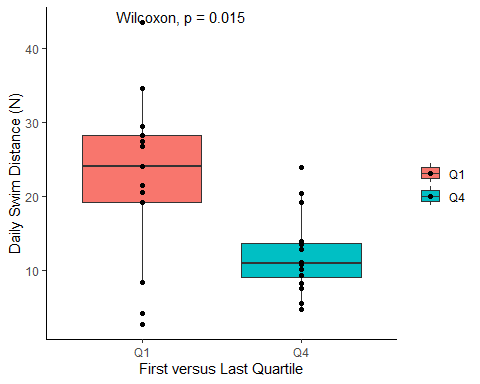


# Quartile Comparison of Daily Distance - Supplimental Figure 1

# First Quartile vs Fourth Quartile  
# Get first and last 27 DAYS, not observations  
swimdist4 <- head(swimdist3, 13)  
swimdist5 <- tail(swimdist3, 16)  
  
# Check for NA values  
swimdist4 <- na.omit(swimdist4)  
swimdist5 <- na.omit(swimdist5)  
  
# Ensure the data is numeric  
swimdist4$Daily\_Distance\_Nm <- as.numeric(swimdist4$Daily\_Distance\_Nm)  
swimdist5$Daily\_Distance\_Nm <- as.numeric(swimdist5$Daily\_Distance\_Nm)  
  
# ANOVA (Unpaired t-test equivalent here) For Distance Quartiles  
t.test(swimdist4$Daily\_Distance\_Nm, swimdist5$Daily\_Distance\_Nm)

##   
## Welch Two Sample t-test  
##   
## data: swimdist4$Daily\_Distance\_Nm and swimdist5$Daily\_Distance\_Nm  
## t = 2.9249, df = 15.856, p-value = 0.009987  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 2.829292 17.770227  
## sample estimates:  
## mean of x mean of y   
## 22.38538 12.08563

swimq1 <- swimdist4 #Quartile 1 data  
swimq4 <- swimdist5 #Quartile 4 data  
  
#Differentiate Quartile 1 and Quartile 2, make it easy by just changing the Swim column "Y" into Q1 or Q2.  
swimq1$Swim[swimq1$Swim == "Y"] <- "Q1"  
  
swimq4$Swim[swimq4$Swim == "Y"] <- "Q4"  
  
#Rename the "Swim" column into "Quartile", which is what they now represent  
colnames(swimq1)[which(names(swimq1) == "Swim")] <- "Quartile"  
colnames(swimq4)[which(names(swimq4) == "Swim")] <- "Quartile"  
  
#Merge Q1 and Q4 dfs together  
SwimQ1thru4 <- full\_join(swimq1, swimq4)  
  
# Generate quartile distance boxplot   
distbox <- ggplot(SwimQ1thru4, aes(x = Quartile, y = Daily\_Distance\_Nm, fill = Quartile)) +  
 geom\_boxplot() +  
 theme\_classic() +  
 geom\_point() +  
 stat\_compare\_means() +  
 labs(x = "First versus Last Quartile",  
 y = "Daily Swim Distance (N)",  
 fill = "")  
  
#Generate Figure  
distbox

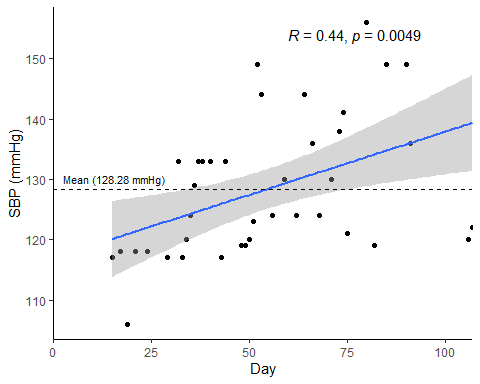


#Swim dist first and 2nd  
#Turns data frame into list  
swimdist4 <- as.list(swimdist4)$Daily\_Distance\_Nm  
swimfirsthalf <- unlist(swimdist4, use.names = FALSE)  
swimfirstdist <- sum(swimfirsthalf)  
swimfirstavg <- sum(swimfirsthalf)/length(swimfirsthalf)  
sda <- sd(swimdist4)  
  
swimdist5 <- as.list(swimdist5)$Daily\_Distance\_Nm  
swim2ndhalf <- unlist(swimdist5, use.names = FALSE)  
swim2nddist <- sum(swim2ndhalf)  
swim2ndavg <- sum(swim2ndhalf)/length(swim2ndhalf)  
sdb <- sd(swimdist5)  
  
#Cumulative distance BETWEEN the two 20 day periods  
totaldist-swimfirstdist-swim2nddist

## [1] 728.21

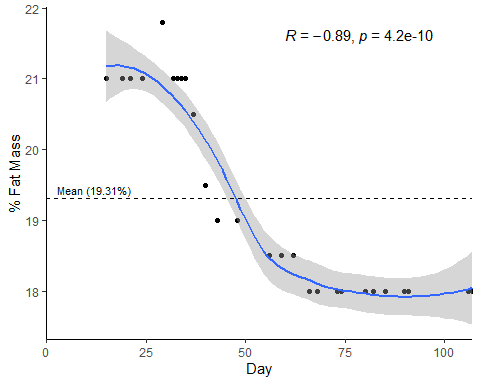
# Bloop Pressure (BP) Over Time - Figure 1C

#Avg Swim blood pressure  
swimBP <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, BP = sample\_data(swimmer.filt)$Systolic\_BP, Day = sample\_data(swimmer.filt)$Day)  
swimBP2 <- swimBP  
swimBP2[swimBP2 == 0] = NA  
#swimBP2[swimBP2 == "N"] = NA  
swimBP2 <- swimBP2 %>% arrange(Day)  
swimBP2 <- na.omit(swimBP2)  
swimBP2 <- as.list(swimBP2)$BP  
swimBPonly <- unlist(swimBP2, use.names = FALSE)  
swimBPsum <- sum(swimBPonly)  
swimBPavg <- sum(swimBPonly)/length(swimBPonly)  
sdBP <- sd(swimBPonly)  
  
#BP avg = 128.28 mmHg, SD = 11.50  
  
#Turns 0 values into NA  
BPdist <- swimBP  
BPdist2 <- BPdist %>% arrange(Day)  
BPdist3 <- na.omit(BPdist2)  
  
###### First Quartile vs Fouth Quartile  
#Get first 27 days of BP versus 27 days, NOT actual observations  
BPdist4 <- head(BPdist3, 5)  
BPdist5 <- tail(BPdist3, 7)  
  
#Swim dist first and 2nd  
#Turns data frame into list  
BPdist4 <- as.list(BPdist4)$BP  
BPfirsthalf <- unlist(BPdist4, use.names = FALSE)  
BPfirstdist <- sum(BPfirsthalf)  
BPfirstavg <- sum(BPfirsthalf)/length(BPfirsthalf) #115.40mmHg  
sBP1 <- sd(BPdist4)  
  
BPdist5 <- as.list(BPdist5)$BP  
BP2ndhalf <- unlist(BPdist5, use.names = FALSE)  
BP2nddist <- sum(BP2ndhalf)  
BP2ndavg <- sum(BP2ndhalf)/length(BP2ndhalf) #135.86 mmHg  
sBP2 <- sd(BPdist5)  
  
#Generate the figure  
fig1c <- ggplot(map, aes(x = Day, y = Systolic\_BP)) +  
 geom\_point() +  
 stat\_smooth(method = "glm") +  
 stat\_cor(label.x = 60) +  
 theme\_classic() +  
 labs(x = "Day", y = "SBP (mmHg)")+  
 geom\_segment(aes(x=0, y=128.28, xend = 107, yend=128.28), color = "black", linetype = "dashed")+  
 annotate(geom = "text", x=15.5, y=130 , label = "Mean (128.28 mmHg)", size = 3, color = "black") +  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, NA))  
  
#Display figure  
fig1c



# Body fat Percent Over Time - Figure 1D

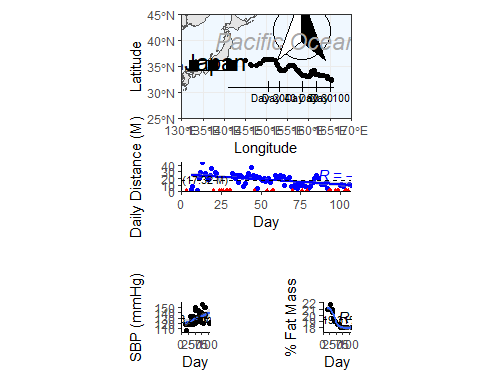
#Percent body fat avg  
swimfatcomp <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, BFC = sample\_data(swimmer.filt)$Percent\_Fat, Day = sample\_data(swimmer.filt)$Day)  
swimfatcomp2 <- swimfatcomp  
swimfatcomp2[swimfatcomp2 == 0] = NA  
swimfatcomp2 <- na.omit(swimfatcomp2)  
swimfatcomp2 <- as.list(swimfatcomp2)$BFC  
swimfconly <- unlist(swimfatcomp2, use.names = FALSE)  
swimfcsum <- sum(swimfconly)  
swimfcavg <- sum(swimfconly)/length(swimfconly)  
sdFC <- sd(swimfconly)  
  
#Generate figure  
fig1d <- ggplot(map, aes(x = Day, y = Percent\_Fat)) +  
 geom\_point() +  
 stat\_smooth(method = "loess") +  
 theme\_classic() +  
 stat\_cor(label.x = 60) +  
 labs(x = "Day", y = "% Fat Mass")+  
 geom\_segment(aes(x=0, y=19.31, xend = 107, yend=19.31), color = "black", linetype = "dashed") +  
 annotate(geom = "text", x=12, y=19.43, label = "Mean (19.31%)", color = "black", size=3) +  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, NA))  
  
#Display figure  
fig1d



# Cobmined Figure 1

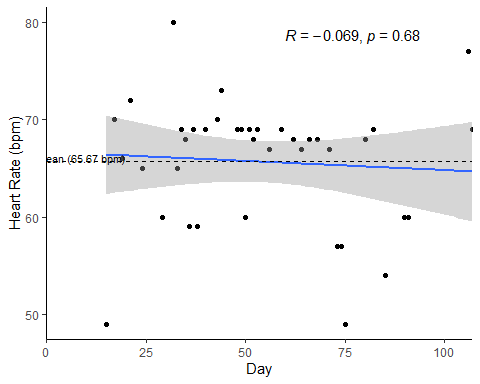
Combines figures B-D into one.

#Generate figure  
patch <- (fig1a) /  
 ((fig1b) /  
 theme(plot.margin = unit(c(0,0,50,0), "pt")) /  
 (fig1c +theme(plot.margin = unit(c(0,50,0,0), "pt")) + fig1d))  
  
#Display figure  
patch



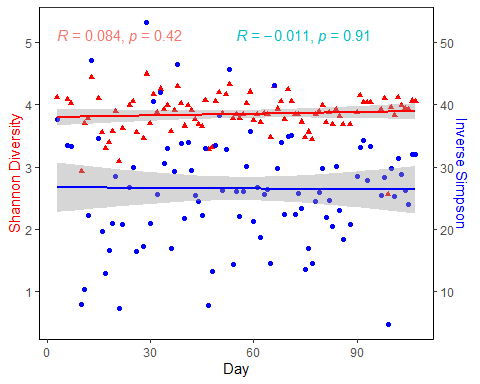
# Heart Rate Over Time - Supplimental Figure 2

#Avg heart rate  
swimHR <- data.frame(Swim = sample\_data(swimmer.filt)$Swim, Heart\_Rate = sample\_data(swimmer.filt)$Heart\_Rate, Day = sample\_data(swimmer.filt)$Day)  
swimHR2 <- swimHR  
swimHR2[swimHR2 == 0] = NA  
swimHR2 <- na.omit(swimHR2)  
swimHR2 <- as.list(swimHR2)$Heart\_Rate  
swimHRonly <- unlist(swimHR2, use.names = FALSE)  
swimHRsum <- sum(swimHRonly)  
swimHRavg <- sum(swimHRonly)/length(swimHRonly)  
sdHR <- sd(swimHRonly)  
  
supp2 <- ggplot(map, aes(x = Day, y = Heart\_Rate)) +  
 geom\_point() +  
 stat\_smooth(method = "glm") +  
 stat\_cor(label.x = 60) +  
 theme\_classic() +  
 labs(x = "Day", y = "Heart Rate (bpm)")+  
 geom\_segment(aes(x=0, y=65.67, xend = 107, yend=65.67), color = "black", linetype = "dashed")+  
 annotate(geom = "text", x=9.0, y=66.1 , label = "Mean (65.67 bpm)", size = 3, color = "black") +  
 scale\_x\_continuous(expand = c(0, 0), limits = c(0, NA))  
  
supp2



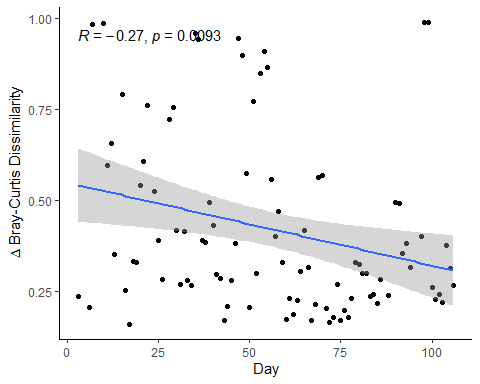
# Alpha Diverstiy - Supplimental Figure 3A

### Diversity Table ###  
alpha.table <- microbiome::alpha(swimmer.filt, index = "all")  
  
# Create SampleID Column  
alpha.table$X.SampleID <- rownames(alpha.table)  
  
df\_swim <- as(sample\_data(swimmer.filt), "data.frame")  
  
# Merge Tables  
alpha.table <- merge(alpha.table, df\_swim, by = "X.SampleID")  
  
# Simpson and Shannon combined   
#### The Stat values (R and P) are switched in the stat\_cor,   
#### so the diversity Simpson color red = the actually shannon diversiy and vice versa.  
coeff <- 10  
  
# Generate figure   
SupFig3a <- ggplot(alpha.table, aes(x = Day)) +  
 geom\_point(aes(y = diversity\_shannon), color = "red", shape=17) +  
 theme(axis.title.y.left = element\_text(color = "red"), axis.title.y.right = element\_text(color = "blue")) +  
 geom\_point(aes(y = diversity\_inverse\_simpson / coeff), color = "blue") +  
 scale\_y\_continuous(name = "Shannon Diversity", sec.axis = sec\_axis(~.\*coeff, name = "Inverse Simpson")) +  
 geom\_smooth(aes(y = diversity\_shannon), method = "lm", color = "red") +  
 geom\_smooth(aes(y = diversity\_inverse\_simpson / coeff), method = "lm", color = "blue") +  
 stat\_cor(aes(y = diversity\_inverse\_simpson / coeff, color = "red"), label.x.npc = "center", show.legend = FALSE) +  
 stat\_cor(aes(y = diversity\_shannon, color = "blue"), show.legend = FALSE)  
  
SupFig3a <- SupFig3a + theme(panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),  
 panel.background = element\_blank(), axis.line = element\_line(colour = "black"))  
  
#Display Figure  
SupFig3a



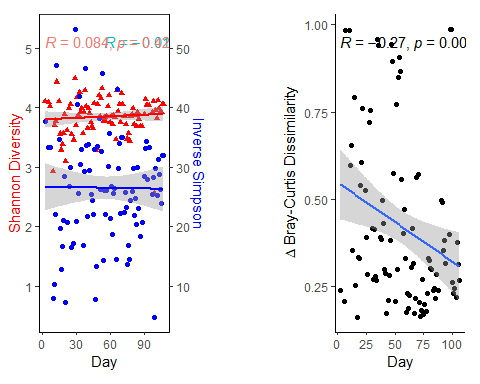
# Beta Diversity Supplimental Figure 3B

# Create Distance Matrix & Data Frame  
df\_swim <- as(sample\_data(swimmer.filt), "data.frame")  
d\_swim <- phyloseq::distance(swimmer.filt, "bray")  
  
# Add Clusters into Metadata  
df\_swim$bray\_cluster\_2 <- factor(pam(d\_swim, k=2, cluster.only = T))  
  
# Rename Cluster Levels  
df\_swim$bray\_cluster\_2 <- factor(df\_swim$bray\_cluster\_2,  
 levels = c("1",  
 "2"),  
 labels = c("Cluster 1",  
 "Cluster 2"))  
  
# Add Clusters into Metadata  
df\_swim$bray\_cluster\_2 <- factor(pam(d\_swim, k=2, cluster.only = T))  
  
# Rename Cluster Levels  
df\_swim$bray\_cluster\_2 <- factor(df\_swim$bray\_cluster\_2,  
 levels = c("1",  
 "2"),  
 labels = c("Cluster 1",  
 "Cluster 2"))  
  
# Add Cluster info back to sample data  
sample\_data(swimmer.filt) <- df\_swim  
  
#Reformat Distance Matrix  
bdiv.swim <- melt(as.matrix(d\_swim))  
  
# Change Column Names  
colnames(bdiv.swim) <- c("Sample1", "Sample2", "Distance")  
  
# Add Relevant Metadata  
bdiv.swim$Sample1\_Day <- df\_swim$Day[match(bdiv.swim$Sample1, row.names(df\_swim))]  
bdiv.swim$Sample2\_Day <- df\_swim$Day[match(bdiv.swim$Sample2, row.names(df\_swim))]  
  
bdiv.swim$Sample1\_Seq <- df\_swim$Sequence\_Sample[match(bdiv.swim$Sample1, row.names(df\_swim))]  
bdiv.swim$Sample2\_Seq <- df\_swim$Sequence\_Sample[match(bdiv.swim$Sample2, row.names(df\_swim))]  
  
# Remove Same Sample Comparisons  
bdiv.diff <- bdiv.swim[bdiv.swim$Sample1 != bdiv.swim$Sample2, ]  
  
# Keep Only Comparisons Between Consecutive Samples  
bdiv.diff <- bdiv.diff[as.numeric(bdiv.diff$Sample2\_Seq) ==  
 (as.numeric(bdiv.diff$Sample1\_Seq) + 1), ]  
  
# Generate Figure  
SupFig3b <- ggplot(bdiv.diff, aes(x = Sample1\_Day, y = Distance)) +  
 geom\_point() +  
 geom\_smooth(method = "glm") +  
 theme\_classic() +  
 labs(x = "Day", y = expression(Delta~"Bray-Curtis Dissimilarity")) +  
 stat\_cor()  
  
# Display figure   
SupFig3b



# Combine Supplimental Figure 3

SupFig3a + theme(plot.margin = unit(c(0,50,0,0), "pt")) + SupFig3b



# Firmicutes and Bacteroidetes info

# Taxa Data Frame for Phyla  
swimmer.genera <- swimmer.filt %>%  
 tax\_glom(taxrank = "Phylum") %>% # agglomerate at phylum level  
 transform\_sample\_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance  
 psmelt() %>% # Melt to long format  
 filter(Abundance > 0.01) %>% # Filter out low abundance taxa  
 arrange(Abundance) # Sort data frame alphabetically by abundance  
  
# Aggregrate Table  
swimmer.aggtab <- aggregate(Abundance ~ Phylum + Sample + Day + bray\_cluster\_2, swimmer.genera, FUN = sum)  
  
# Remove Underscores  
swimmer.aggtab$Phylum <- gsub("p\_\_", "", paste(swimmer.aggtab$Phylum))  
  
# Make Blank Cells Unspecified  
swimmer.aggtab$Phylum[swimmer.aggtab$Phylum == ""] <- "Unassigned"  
swimmer.aggtab$Phylum[swimmer.aggtab$Phylum == "NA"] <- "Unassigned"  
  
#Total phylum relative abundance  
swimmer.aggtot <- aggregate(Abundance ~ Phylum, swimmer.genera, FUN = sum)  
  
#Subset only Firmicutes or Bacteroidetes  
swimmer.aggFirm <- swimmer.aggtab %>% subset(Phylum == 'Firmicutes')  
swimmer.aggBact <- swimmer.aggtab %>% subset(Phylum == 'Bacteroidetes')  
  
#Get the average over the entire swim of Firmicutes and Bacteroidetes  
#Firmicutes  
swimmer.aggFirm

## Phylum Sample Day bray\_cluster\_2 Abundance  
## 2 Firmicutes BL3 3 Cluster 1 0.5383681  
## 6 Firmicutes BL6 6 Cluster 1 0.5576531  
## 11 Firmicutes BL7 7 Cluster 1 0.5324280  
## 16 Firmicutes BL10 10 Cluster 1 0.3985207  
## 21 Firmicutes BL11 11 Cluster 1 0.3937798  
## 26 Firmicutes BL12 12 Cluster 1 0.5294503  
## 30 Firmicutes BL13 13 Cluster 1 0.5300114  
## 34 Firmicutes BL15 15 Cluster 1 0.4509380  
## 37 Firmicutes BL16 16 Cluster 1 0.3448281  
## 40 Firmicutes BL17 17 Cluster 1 0.3553623  
## 42 Firmicutes BL18 18 Cluster 1 0.4299448  
## 44 Firmicutes BL19 19 Cluster 1 0.4755960  
## 47 Firmicutes BL20 20 Cluster 1 0.4898056  
## 49 Firmicutes BL21 21 Cluster 1 0.2766242  
## 53 Firmicutes BL24 24 Cluster 1 0.4772264  
## 57 Firmicutes BL25 25 Cluster 1 0.4083892  
## 61 Firmicutes BL26 26 Cluster 1 0.3719288  
## 66 Firmicutes BL29 29 Cluster 1 0.3812192  
## 72 Firmicutes BL30 30 Cluster 1 0.3462803  
## 78 Firmicutes BL31 31 Cluster 1 0.5779662  
## 84 Firmicutes BL33 33 Cluster 1 0.4980907  
## 90 Firmicutes BL34 34 Cluster 1 0.5195134  
## 95 Firmicutes BL35 35 Cluster 1 0.6751216  
## 99 Firmicutes BL36 36 Cluster 1 0.5293735  
## 104 Firmicutes BL37 37 Cluster 1 0.4586537  
## 109 Firmicutes BL38 38 Cluster 1 0.5270057  
## 115 Firmicutes BL39 39 Cluster 1 0.5477845  
## 121 Firmicutes BL41 41 Cluster 1 0.5619689  
## 125 Firmicutes BL42 42 Cluster 1 0.6391657  
## 129 Firmicutes BL43 43 Cluster 1 0.4369144  
## 133 Firmicutes BL44 44 Cluster 1 0.3935270  
## 137 Firmicutes BL45 45 Cluster 1 0.4375198  
## 141 Firmicutes BL46 46 Cluster 1 0.5643171  
## 145 Firmicutes BL47 47 Cluster 1 0.2687714  
## 149 Firmicutes BL48 48 Cluster 1 0.3723436  
## 153 Firmicutes BL49 49 Cluster 1 0.3796273  
## 158 Firmicutes BL50 50 Cluster 1 0.5848592  
## 162 Firmicutes BL51 51 Cluster 1 0.6269751  
## 167 Firmicutes BL52 52 Cluster 1 0.4777508  
## 173 Firmicutes BL53 53 Cluster 1 0.4368510  
## 177 Firmicutes BL54 54 Cluster 1 0.2218086  
## 182 Firmicutes BL55 55 Cluster 1 0.4885068  
## 186 Firmicutes BL56 56 Cluster 1 0.3338235  
## 191 Firmicutes BL58 58 Cluster 1 0.6012396  
## 195 Firmicutes BL70 70 Cluster 1 0.5362073  
## 199 Firmicutes BL91 91 Cluster 1 0.4031623  
## 204 Firmicutes BL94 94 Cluster 1 0.5573752  
## 209 Firmicutes BL98 98 Cluster 1 0.3586351  
## 213 Firmicutes BL99 99 Cluster 1 0.1814625  
## 217 Firmicutes BL22 22 Cluster 2 0.4616958  
## 221 Firmicutes BL28 28 Cluster 2 0.4533288  
## 226 Firmicutes BL32 32 Cluster 2 0.3630674  
## 230 Firmicutes BL40 40 Cluster 2 0.3733648  
## 234 Firmicutes BL57 57 Cluster 2 0.4030965  
## 238 Firmicutes BL59 59 Cluster 2 0.5712192  
## 244 Firmicutes BL60 60 Cluster 2 0.5105427  
## 248 Firmicutes BL61 61 Cluster 2 0.4361908  
## 253 Firmicutes BL62 62 Cluster 2 0.4902747  
## 258 Firmicutes BL63 63 Cluster 2 0.5193239  
## 263 Firmicutes BL64 64 Cluster 2 0.3813150  
## 269 Firmicutes BL65 65 Cluster 2 0.4130737  
## 274 Firmicutes BL66 66 Cluster 2 0.6264763  
## 279 Firmicutes BL67 67 Cluster 2 0.4591065  
## 284 Firmicutes BL68 68 Cluster 2 0.5479384  
## 289 Firmicutes BL69 69 Cluster 2 0.4898298  
## 294 Firmicutes BL71 71 Cluster 2 0.5678870  
## 299 Firmicutes BL72 72 Cluster 2 0.5061960  
## 304 Firmicutes BL73 73 Cluster 2 0.5226426  
## 309 Firmicutes BL74 74 Cluster 2 0.3881138  
## 314 Firmicutes BL75 75 Cluster 2 0.4181375  
## 320 Firmicutes BL76 76 Cluster 2 0.4343306  
## 325 Firmicutes BL77 77 Cluster 2 0.3170184  
## 330 Firmicutes BL78 78 Cluster 2 0.4047973  
## 335 Firmicutes BL79 79 Cluster 2 0.5100604  
## 340 Firmicutes BL80 80 Cluster 2 0.4804739  
## 347 Firmicutes BL81 81 Cluster 2 0.4374593  
## 352 Firmicutes BL82 82 Cluster 2 0.4506330  
## 356 Firmicutes BL83 83 Cluster 2 0.3029244  
## 362 Firmicutes BL84 84 Cluster 2 0.4109488  
## 367 Firmicutes BL85 85 Cluster 2 0.4333407  
## 371 Firmicutes BL86 86 Cluster 2 0.5245829  
## 375 Firmicutes BL88 88 Cluster 2 0.3630230  
## 380 Firmicutes BL90 90 Cluster 2 0.3728569  
## 384 Firmicutes BL92 92 Cluster 2 0.4972044  
## 388 Firmicutes BL93 93 Cluster 2 0.3769146  
## 393 Firmicutes BL97 97 Cluster 2 0.4663006  
## 397 Firmicutes BL100 100 Cluster 2 0.4496945  
## 401 Firmicutes BL101 101 Cluster 2 0.4692196  
## 406 Firmicutes BL102 102 Cluster 2 0.5542313  
## 411 Firmicutes BL103 103 Cluster 2 0.3920875  
## 414 Firmicutes BL104 104 Cluster 2 0.4223227  
## 418 Firmicutes BL105 105 Cluster 2 0.2645085  
## 422 Firmicutes BL106 106 Cluster 2 0.4046970  
## 426 Firmicutes BL107 107 Cluster 2 0.5863918

mean(swimmer.aggFirm$Abundance)

## [1] 0.4543991

#Bacteroides  
swimmer.aggBact

## Phylum Sample Day bray\_cluster\_2 Abundance  
## 1 Bacteroidetes BL3 3 Cluster 1 0.42862246  
## 5 Bacteroidetes BL6 6 Cluster 1 0.38775584  
## 10 Bacteroidetes BL7 7 Cluster 1 0.40102277  
## 15 Bacteroidetes BL10 10 Cluster 1 0.46421106  
## 20 Bacteroidetes BL11 11 Cluster 1 0.43634911  
## 25 Bacteroidetes BL12 12 Cluster 1 0.33600443  
## 29 Bacteroidetes BL13 13 Cluster 1 0.24413215  
## 33 Bacteroidetes BL15 15 Cluster 1 0.35708573  
## 36 Bacteroidetes BL16 16 Cluster 1 0.53743464  
## 39 Bacteroidetes BL17 17 Cluster 1 0.63790992  
## 41 Bacteroidetes BL18 18 Cluster 1 0.56331926  
## 43 Bacteroidetes BL19 19 Cluster 1 0.51634724  
## 46 Bacteroidetes BL20 20 Cluster 1 0.49330206  
## 48 Bacteroidetes BL21 21 Cluster 1 0.63403179  
## 52 Bacteroidetes BL24 24 Cluster 1 0.29876619  
## 56 Bacteroidetes BL25 25 Cluster 1 0.50941232  
## 60 Bacteroidetes BL26 26 Cluster 1 0.57952362  
## 64 Bacteroidetes BL29 29 Cluster 1 0.24906610  
## 71 Bacteroidetes BL30 30 Cluster 1 0.57641043  
## 76 Bacteroidetes BL31 31 Cluster 1 0.32837305  
## 82 Bacteroidetes BL33 33 Cluster 1 0.31573869  
## 88 Bacteroidetes BL34 34 Cluster 1 0.43482643  
## 93 Bacteroidetes BL35 35 Cluster 1 0.25948831  
## 98 Bacteroidetes BL36 36 Cluster 1 0.11037308  
## 102 Bacteroidetes BL37 37 Cluster 1 0.42033575  
## 107 Bacteroidetes BL38 38 Cluster 1 0.32009192  
## 113 Bacteroidetes BL39 39 Cluster 1 0.26068116  
## 119 Bacteroidetes BL41 41 Cluster 1 0.37158890  
## 123 Bacteroidetes BL42 42 Cluster 1 0.29054706  
## 127 Bacteroidetes BL43 43 Cluster 1 0.51503661  
## 131 Bacteroidetes BL44 44 Cluster 1 0.53652747  
## 135 Bacteroidetes BL45 45 Cluster 1 0.47209007  
## 139 Bacteroidetes BL46 46 Cluster 1 0.37497828  
## 143 Bacteroidetes BL47 47 Cluster 1 0.63127748  
## 148 Bacteroidetes BL48 48 Cluster 1 0.08648809  
## 151 Bacteroidetes BL49 49 Cluster 1 0.49538473  
## 156 Bacteroidetes BL50 50 Cluster 1 0.30296177  
## 161 Bacteroidetes BL51 51 Cluster 1 0.25231810  
## 165 Bacteroidetes BL52 52 Cluster 1 0.32431199  
## 171 Bacteroidetes BL53 53 Cluster 1 0.43119570  
## 176 Bacteroidetes BL54 54 Cluster 1 0.23277590  
## 181 Bacteroidetes BL55 55 Cluster 1 0.21597449  
## 184 Bacteroidetes BL56 56 Cluster 1 0.44503259  
## 189 Bacteroidetes BL58 58 Cluster 1 0.30247589  
## 194 Bacteroidetes BL70 70 Cluster 1 0.26831607  
## 198 Bacteroidetes BL91 91 Cluster 1 0.37114545  
## 203 Bacteroidetes BL94 94 Cluster 1 0.38825203  
## 207 Bacteroidetes BL98 98 Cluster 1 0.38298445  
## 212 Bacteroidetes BL99 99 Cluster 1 0.10280102  
## 216 Bacteroidetes BL22 22 Cluster 2 0.50874501  
## 220 Bacteroidetes BL28 28 Cluster 2 0.48102270  
## 224 Bacteroidetes BL32 32 Cluster 2 0.55450390  
## 228 Bacteroidetes BL40 40 Cluster 2 0.51719983  
## 232 Bacteroidetes BL57 57 Cluster 2 0.44406662  
## 237 Bacteroidetes BL59 59 Cluster 2 0.32728691  
## 242 Bacteroidetes BL60 60 Cluster 2 0.39821343  
## 246 Bacteroidetes BL61 61 Cluster 2 0.44646741  
## 251 Bacteroidetes BL62 62 Cluster 2 0.40972002  
## 256 Bacteroidetes BL63 63 Cluster 2 0.37531899  
## 261 Bacteroidetes BL64 64 Cluster 2 0.45209730  
## 267 Bacteroidetes BL65 65 Cluster 2 0.41583667  
## 272 Bacteroidetes BL66 66 Cluster 2 0.19466796  
## 277 Bacteroidetes BL67 67 Cluster 2 0.36681036  
## 282 Bacteroidetes BL68 68 Cluster 2 0.30636703  
## 287 Bacteroidetes BL69 69 Cluster 2 0.37139354  
## 292 Bacteroidetes BL71 71 Cluster 2 0.31579444  
## 297 Bacteroidetes BL72 72 Cluster 2 0.39048988  
## 302 Bacteroidetes BL73 73 Cluster 2 0.36281605  
## 307 Bacteroidetes BL74 74 Cluster 2 0.46155983  
## 312 Bacteroidetes BL75 75 Cluster 2 0.44811024  
## 318 Bacteroidetes BL76 76 Cluster 2 0.43642016  
## 323 Bacteroidetes BL77 77 Cluster 2 0.52786969  
## 328 Bacteroidetes BL78 78 Cluster 2 0.44525087  
## 333 Bacteroidetes BL79 79 Cluster 2 0.34712922  
## 339 Bacteroidetes BL80 80 Cluster 2 0.39294543  
## 345 Bacteroidetes BL81 81 Cluster 2 0.45303508  
## 350 Bacteroidetes BL82 82 Cluster 2 0.44143919  
## 354 Bacteroidetes BL83 83 Cluster 2 0.53590585  
## 360 Bacteroidetes BL84 84 Cluster 2 0.44831273  
## 365 Bacteroidetes BL85 85 Cluster 2 0.46368956  
## 369 Bacteroidetes BL86 86 Cluster 2 0.39207599  
## 373 Bacteroidetes BL88 88 Cluster 2 0.54349140  
## 378 Bacteroidetes BL90 90 Cluster 2 0.52939073  
## 383 Bacteroidetes BL92 92 Cluster 2 0.43319642  
## 387 Bacteroidetes BL93 93 Cluster 2 0.51667920  
## 391 Bacteroidetes BL97 97 Cluster 2 0.41780105  
## 395 Bacteroidetes BL100 100 Cluster 2 0.42777099  
## 399 Bacteroidetes BL101 101 Cluster 2 0.45621502  
## 404 Bacteroidetes BL102 102 Cluster 2 0.37702278  
## 409 Bacteroidetes BL103 103 Cluster 2 0.50738358  
## 413 Bacteroidetes BL104 104 Cluster 2 0.49457057  
## 417 Bacteroidetes BL105 105 Cluster 2 0.62535957  
## 421 Bacteroidetes BL106 106 Cluster 2 0.50574324  
## 425 Bacteroidetes BL107 107 Cluster 2 0.33126262

mean(swimmer.aggBact$Abundance)

## [1] 0.4095056

#Firm and Bact Cluster 1  
  
swimmer.aggFirmClust1 <- swimmer.aggFirm %>% subset(bray\_cluster\_2 == 'Cluster 1')  
mean(swimmer.aggFirmClust1$Abundance)

## [1] 0.4588709

swimmer.aggBactClust1 <- swimmer.aggBact %>% subset(bray\_cluster\_2 == 'Cluster 1')  
mean(swimmer.aggBactClust1$Abundance)

## [1] 0.3856139

#Firm and Bact Cluster 2  
  
swimmer.aggFirmClust2 <- swimmer.aggFirm %>% subset(bray\_cluster\_2 == 'Cluster 2')  
mean(swimmer.aggFirmClust2$Abundance)

## [1] 0.4495298

swimmer.aggBactClust2 <- swimmer.aggBact %>% subset(bray\_cluster\_2 == 'Cluster 2')  
mean(swimmer.aggBactClust2$Abundance)

## [1] 0.4355211

# Random forest model for classifying day swam - Figure 2

Using the Genera from the GM we can predict the day of swim the participant should be on. The mean decrease accuracy from this model identifies the genera most impact in making the prediction.

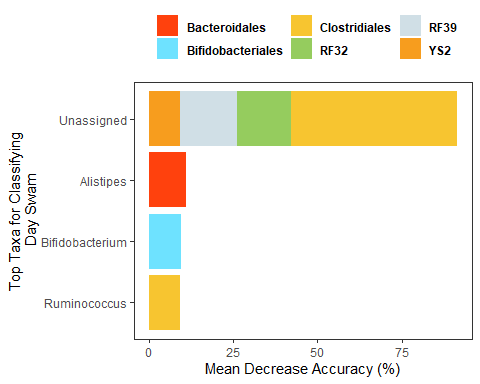
# Make Training Dataset  
predict <- t(otu\_table(swimmer.filt))  
dim(predict)

## [1] 94 856

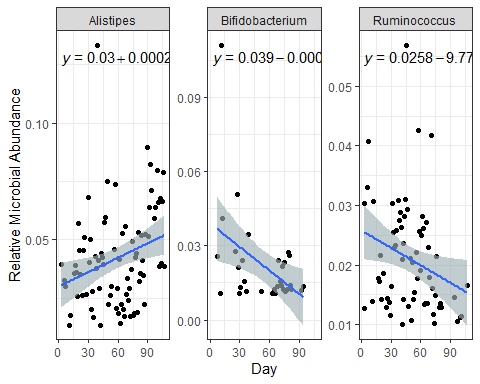
# We have 94 samples representing 856 OTUs  
  
# Create response variable (AGE\_CAT)  
res <- sample\_data(swimmer.filt)$Day  
  
# Combine them into 1 data frame  
rf <- data.frame(res, predict)  
  
# RandomForest Model (OOB)  
set.seed(28)  
class <- randomForest(res ~ ., data = rf, ntree = 1000, importance = TRUE)  
print(class)

##   
## Call:  
## randomForest(formula = res ~ ., data = rf, ntree = 1000, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 1000  
## No. of variables tried at each split: 285  
##   
## Mean of squared residuals: 88.60874  
## % Var explained: 89.3

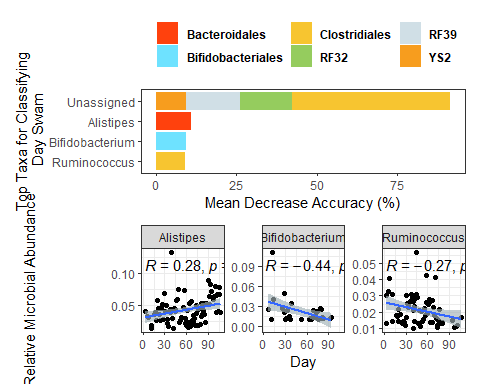
# Make a data frame with predictor names and their importance  
imp <- randomForest::importance(class)  
imp <- data.frame(predictors = rownames(imp), imp)  
  
# Order the predictor levels by importance  
imp.sort <- arrange(imp, desc(X.IncMSE))  
imp.sort$predictors <- factor(imp.sort$predictors, levels = imp.sort$predictors)  
  
# Select the top 10 predictors  
imp.10 <- imp.sort[1:10, ]  
  
# Change Column name to say "OTUID" rather than "predictors"  
colnames(imp.10)[which(names(imp.10) == "predictors")] <- "OTUID"  
  
# Remove "X" from OTUID column  
imp.10$OTUID <- gsub("X", "", paste(imp.10$OTUID))  
  
### Create Taxa Table ###  
  
# Make Taxa table a data frame  
otu\_df <- as.data.frame(tax\_table(swimmer.filt))  
  
# Make OTU IDs (row names) into column  
otu\_df$OTUID <- rownames(otu\_df)  
  
# Merge Two data frames using matched column  
imp10.merged <- merge(imp.10, otu\_df, by = "OTUID")  
  
# Remove level designations and underscores  
imp10.merged$Domain <- gsub("k\_\_", "", paste(imp10.merged$Domain))  
imp10.merged$Phylum <- gsub("p\_\_", "", paste(imp10.merged$Phylum))  
imp10.merged$Class <- gsub("c\_\_", "", paste(imp10.merged$Class))  
imp10.merged$Order <- gsub("o\_\_", "", paste(imp10.merged$Order))  
imp10.merged$Family <- gsub("f\_\_", "", paste(imp10.merged$Family))  
imp10.merged$Genus <- gsub("g\_\_", "", paste(imp10.merged$Genus))  
  
# Make Blank Cells Unspecified  
imp10.merged$Family[imp10.merged$Family==""] <- "Unassigned"  
imp10.merged$Genus[imp10.merged$Genus==""] <- "Unassigned"  
imp10.merged$Family[imp10.merged$Family=="NA"] <- "Unassigned"  
imp10.merged$Genus[imp10.merged$Genus=="NA"] <- "Unassigned"  
  
# Plot Feature Score  
# Feature score identifies important taxa in the RF model.   
# Those important taxa can be attributed to having the biggest change or impact on the GM.   
Rfday <- ggplot(imp10.merged, aes(x = reorder(Genus, X.IncMSE),  
 y = X.IncMSE,  
 fill = Order)) +  
 geom\_bar(stat = "identity") +  
 #geom\_segment(aes(y = 0, x = Genus, yend = MeanDecreaseAccuracy, xend = Genus)) +  
 #geom\_point(alpha = 0.5, size = 3) +  
 scale\_fill\_tron() +  
 coord\_flip() +  
 theme(legend.position = "top") +  
 labs(x = "Top Taxa for Classifying\n Day Swam",   
 y = "Mean Decrease Accuracy (%)",  
 fill = "") +  
 theme(  
 panel.background = element\_rect(fill = "transparent",  
 colour = NA\_character\_), # necessary to avoid drawing panel outline  
 panel.grid.major = element\_blank(), # get rid of major grid  
 panel.grid.minor = element\_blank(), # get rid of minor grid  
 plot.background = element\_rect(fill = "transparent",  
 colour = NA\_character\_), # necessary to avoid drawing plot outline  
 legend.background = element\_rect(fill = "transparent"),  
 legend.key = element\_rect(fill = "transparent"),  
 legend.text = element\_text(face = "bold")  
 )  
Rfday



#Analyze these important taxa using Spearman plots  
# Create Taxa Data Frame  
swimmer.genera <- swimmer.filt %>%  
 tax\_glom(taxrank = "Genus") %>% # agglomerate at phylum level  
 transform\_sample\_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance  
 psmelt() %>% # Melt to long format  
 filter(Abundance > 0.01) %>% # Filter out low abundance taxa  
 arrange(Order) # Sort data frame alphabetically by phylum  
  
  
target.otus <- subset(swimmer.genera, Genus == "g\_\_Alistipes" |  
 Genus == "g\_\_Bifidobacterium" |  
 Genus == "g\_\_Ruminococcus")  
  
target.otus$Genus <- gsub("g\_\_", "", target.otus$Genus)  
  
#Linear regression eq's with Spearman  
ggplot(target.otus, aes(x = Day, y = Abundance)) +  
 geom\_point(color = "black") +  
 geom\_smooth(method = "lm", se = TRUE, color = "blue", fill = "lightblue") +  
 facet\_wrap(~Genus, scale = "free\_y") +  
 scale\_color\_tron() +  
 labs(y = "Relative Microbial Abundance") +  
 stat\_poly\_line() +  
 stat\_poly\_eq(use\_label(c("eq", "R2")))



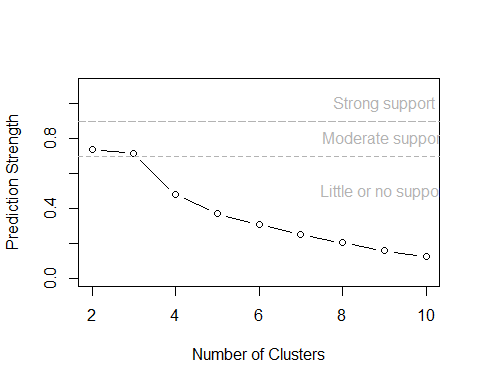
#P value with Spearman  
RfDaySpearman <- ggplot(target.otus, aes(x = Day, y = Abundance)) +  
 geom\_point(color = "black") +  
 geom\_smooth(method = "lm", se = TRUE, color = "blue", fill = "lightblue") +  
 facet\_wrap(~Genus, scale = "free\_y") +  
 scale\_color\_tron() +  
 labs(y = "Relative Microbial Abundance") +  
 stat\_poly\_line() +  
 stat\_cor()  
  
  
#Finished plot for Figure 2  
(Rfday) /   
 (RfDaySpearman)



# K Means Clustering Prediction Strength for Bray-Curtis Distances and PCoA - Supplemental Figure 4

K means clustering allows us to identify how many different communities existed within our data.

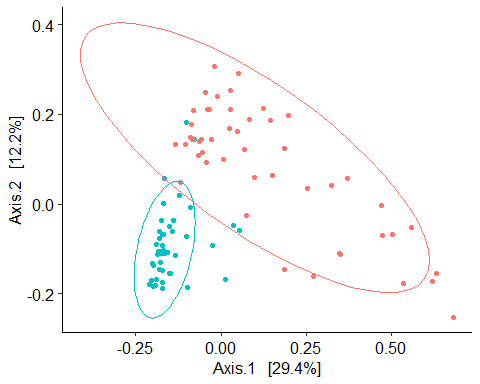
ps\_bray <- prediction.strength(d\_swim, Gmin = 2, Gmax = 10, clustermethod = pamkCBI)  
  
# Function to plot cluster validation measures  
plot\_cluster\_validation = function(bray, legend=T,...) {  
 plot(2:10, bray, type="b", pch=1, xlab="Number of Clusters", ...)  
 if(legend) legend("topright", legend = "Bray-Curtis", pch=c(1,2,22), lty=1:3)  
}  
  
# Plot Prediction Strength  
plot\_cluster\_validation(ps\_bray$mean.pred[2:10],   
 ylab = "Prediction Strength",  
 ylim = c(0, 1.1), legend = F)  
# Insert lines and labels for analysis  
abline(.9, 0, lty = 5, col = "grey70")  
abline(0.7, 0, lty = 8, col = "grey70")  
text("Strong support", x = 9, y = 1, col = "grey70")  
text("Moderate support", x = 9, y = .8, col = "grey70")  
text("Little or no support", x = 9, y = .5, col = "grey70")



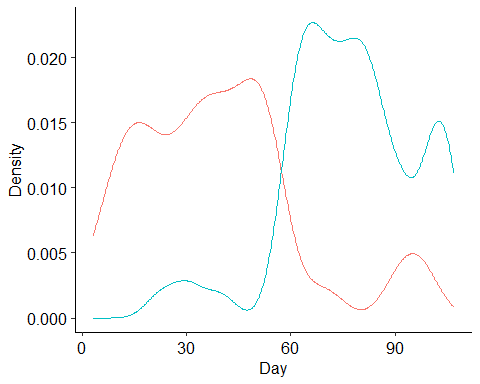
# Plot PCoA of Clusters and density plot- Figure 3A-B

PCOA allows us to determine if during course of the swim if there was a change in the GM community. Density plot allows us to easily visualize the changes.

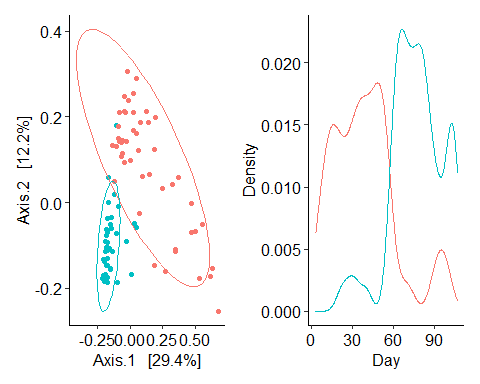
# Generate Figure 3A  
pcoa <- ordinate(swimmer.filt, method = "PCoA", distance = "bray")  
ClustPCoA <- plot\_ordination(swimmer.filt, pcoa, color = "bray\_cluster\_2") +  
 theme\_pubr() +  
 labs(color = "", shape = "") +  
 stat\_ellipse() +  
 theme(legend.position = "none")  
  
# Display figure 3A  
ClustPCoA



# Generate Figure 3B  
densityplot <- ggplot(swimmer.genera, aes(x = Day, color = bray\_cluster\_2, group = bray\_cluster\_2)) +  
 geom\_density(linewidth = .5, alpha = .5) +  
 labs(y = "Density", size = 3) +  
 theme\_pubr() +  
 theme(  
 legend.title = element\_blank(),  
 legend.position = "none"  
 )  
  
# Display Figure 3B  
densityplot



# Combined Figure 3A-B  
(ClustPCoA | densityplot)



# Cluster/Enterotypes Random forest - Figure 4A-B

Model used for genera of the Gm ability tot predict clusters identified from PCoA. The mean decrease accuarcy can be used to identify likely genera that are significantly different between the two clusters.

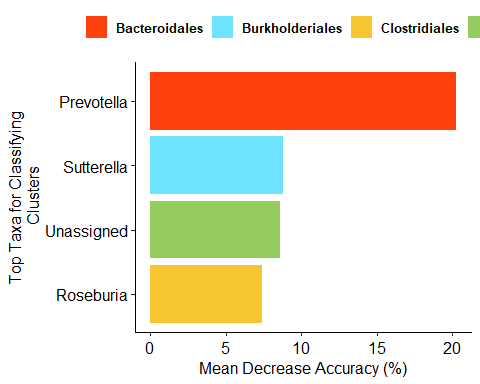
predict <- t(otu\_table(swimmer.filt))  
dim(predict)

## [1] 94 856

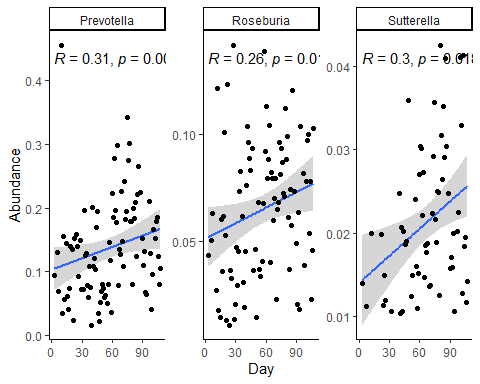
# We have 94 samples representing 856 OTUs  
# Create response variable (AGE\_CAT)  
clustres <- sample\_data(swimmer.filt)$bray\_cluster\_2  
# Combine them into 1 data frame  
rf2 <- data.frame(clustres, predict)  
# Generate Classification Model  
set.seed(28)  
clustclass <- randomForest(clustres ~ ., data = rf2, ntree = 1000, importance = TRUE)  
print(clustclass)

##   
## Call:  
## randomForest(formula = clustres ~ ., data = rf2, ntree = 1000, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 29  
##   
## OOB estimate of error rate: 10.64%  
## Confusion matrix:  
## Cluster 1 Cluster 2 class.error  
## Cluster 1 46 3 0.06122449  
## Cluster 2 7 38 0.15555556

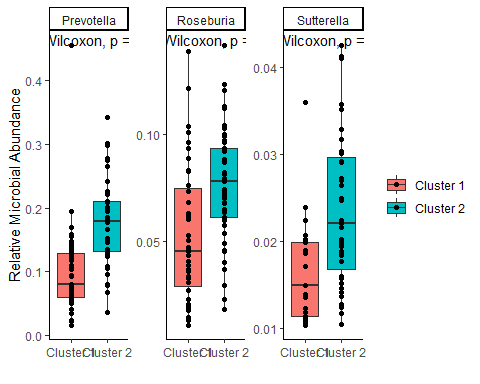
# Make a data frame with predictor names and their importance  
impcl <- randomForest::importance(clustclass)  
impcl <- data.frame(predictors = rownames(impcl), impcl)  
#General cluster prediction  
imp.sortcl3 <- arrange(impcl, desc(MeanDecreaseAccuracy))  
# Select the top 10 predictors  
imp.10cl3 <- imp.sortcl3[1:5, ]  
# Change Column name to say "OTUID" rather than "predictors"  
colnames(imp.10cl3)[which(names(imp.10cl3) == "predictors")] <- "OTUID"  
# Remove "X" from OTUID column  
imp.10cl3$OTUID <- gsub("X", "", paste(imp.10cl3$OTUID))  
### Create Taxa Table ###  
# Make Taxa table a data frame  
otu\_df <- as.data.frame(tax\_table(swimmer.filt))  
# Make OTU IDs (row names) into column  
otu\_df$OTUID <- rownames(otu\_df)  
# Merge Two data frames using matched column  
imp10.mergedcl3 <- merge(imp.10cl3, otu\_df, by = "OTUID")  
# Remove level designations and underscores  
imp10.mergedcl3$Domain <- gsub("k\_\_", "", paste(imp10.mergedcl3$Domain))  
imp10.mergedcl3$Phylum <- gsub("p\_\_", "", paste(imp10.mergedcl3$Phylum))  
imp10.mergedcl3$Class <- gsub("c\_\_", "", paste(imp10.mergedcl3$Class))  
imp10.mergedcl3$Order <- gsub("o\_\_", "", paste(imp10.mergedcl3$Order))  
imp10.mergedcl3$Family <- gsub("f\_\_", "", paste(imp10.mergedcl3$Family))  
imp10.mergedcl3$Genus <- gsub("g\_\_", "", paste(imp10.mergedcl3$Genus))  
# Make Blank Cells Unspecified  
imp10.mergedcl3$Family[imp10.mergedcl3$Family==""] <- "Unassigned"  
imp10.mergedcl3$Genus[imp10.mergedcl3$Genus==""] <- "Unassigned"  
imp10.mergedcl3$Family[imp10.mergedcl3$Family=="NA"] <- "Unassigned"  
imp10.mergedcl3$Genus[imp10.mergedcl3$Genus=="NA"] <- "Unassigned"  
# Plot Feature Scores  
#Combined for both clusters - Figure4A  
rfClust <- ggplot(imp10.mergedcl3, aes(x = reorder(Genus, MeanDecreaseAccuracy),  
 y = MeanDecreaseAccuracy,  
 fill = Order)) +  
 geom\_bar(stat = "identity") +  
 scale\_fill\_tron() +  
 coord\_flip() +  
 theme\_pubr() +  
 theme(legend.position = "top") +  
 labs(x = "Top Taxa for Classifying\n Clusters",  
 y = "Mean Decrease Accuracy (%)",  
 fill = "") +  
 theme(  
 legend.text = element\_text(face = "bold")  
 )  
# Display Figure 4A  
rfClust



#otus for both cluster 1 data and the overall cluster data?  
cl3.otus <- subset(swimmer.genera, Genus == "g\_\_Prevotella" |  
 Genus == "g\_\_Sutterella" |  
 Genus == "g\_\_Roseburia")  
cl3.otus$Genus <- gsub("g\_\_", "", cl3.otus$Genus)  
#Spearman plots  
ggplot(cl3.otus, aes(x = Day, y = Abundance)) +  
 geom\_smooth(method = "glm") +  
 facet\_wrap(~Genus, scales = "free\_y") +  
 theme\_classic() +  
 stat\_cor(method = "spearman") +  
 geom\_point()



#Box plots showing diffs b/w clusters - Figure 4B  
clustspear <- ggplot(cl3.otus, aes(x = bray\_cluster\_2, y = Abundance, fill = bray\_cluster\_2)) +  
 geom\_boxplot() +  
 facet\_wrap(~Genus, scales = "free\_y") +  
 theme\_classic() +  
 geom\_point() +  
 stat\_compare\_means(aes(group = bray\_cluster\_2)) +  
 labs(x = "",  
 y = "Relative Microbial Abundance",  
 fill = "")  
  
# Display box plot - Figure 4B  
clustspear



#Figure 4 A-B combined   
(rfClust) /  
 (clustspear)

