Microcystis Oligotype Analysis

Michelle Berry

July 2016

Import Data

```
# Import oligotypes
mc_oligos <- read.csv(
  file = "data/mc-oligo-counts.txt",
  sep = "\t"
)

# Remove isolate data (keeping just erie samples)
erie_oligos <-
  mc_oligos %>%
    filter(grepl(pattern = "E0", x = samples))

sampdat <- read.csv("data/erie-sampledata.csv")
```

Format data

```
# Join sample data to oligotype data
oligos_join <-
  sampdat %>%
    left_join(erie_oligos, by = c("SampleID" = "samples"))

# Scale oligotype data by total reads from samples
oligos_scale <-
  oligos_join %>%
    mutate(CTT = CTT/ReadSums) %>%
    mutate(CTG = CTG/ReadSums) %>%
    mutate(CCG = CCG/ReadSums) %>%
    mutate(CTG_CCG_ratio = CTG/CCG)

# Tidy oligotypes into one column
erie_oligo_df <-
  oligos_scale %>%
  select(-CCT, -TCG) %>%
  gather(key = Oligotype, value = count, CTG, CTT, CCG)
```

Plot data

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

erie_oligo_df <- order_dates(erie_oligo_df)
erie_oligo_df$Oligotype <- factor(erie_oligo_df$Oligotype, levels = c("CCG", "CTT", "CTG"))


# Plot oligotypes over time
oligoplot <-
  ggplot(data = erie_oligo_df, aes(x = Date, y = count, fill = Oligotype)) +
    facet_grid(~Station) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values = c("#b3002d", "#43a2ca", "#a6d854")) +
    scale_x_discrete(
      breaks = c("7/8", "8/4", "9/2", "10/6"),
      labels = c("Jul", "Aug", "Sep", "Oct"),
      drop = FALSE
    ) +
    theme(
      axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
      axis.title.x = element_blank()
    ) +
    ylab("Relative Abundance \n (% of total community)")


# Plot toxin over time
toxinplot <-
  ggplot(data = erie_oligo_df, aes(x = Date, y = ParMC, group = Station)) +
    facet_grid(~Station) +
    geom_line(color = "black") +
    geom_point() +
  scale_x_discrete(
      breaks = c("7/8", "8/4", "9/2", "10/6"),
      labels = c("Jul", "Aug", "Sep", "Oct"),
      drop = FALSE
    ) +
  ylab("Particulate Microcystin-LR \n (ug/L)") +
  theme(
      axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
      axis.title.x = element_blank(),
      plot.margin = unit(c(1, 0, 0, 0), "cm")
  )
```

```r
# grab grobs for PC plots
toxinGrob <- ggplotGrob(toxinplot)
oligoGrob <- ggplotGrob(oligoplot)
```

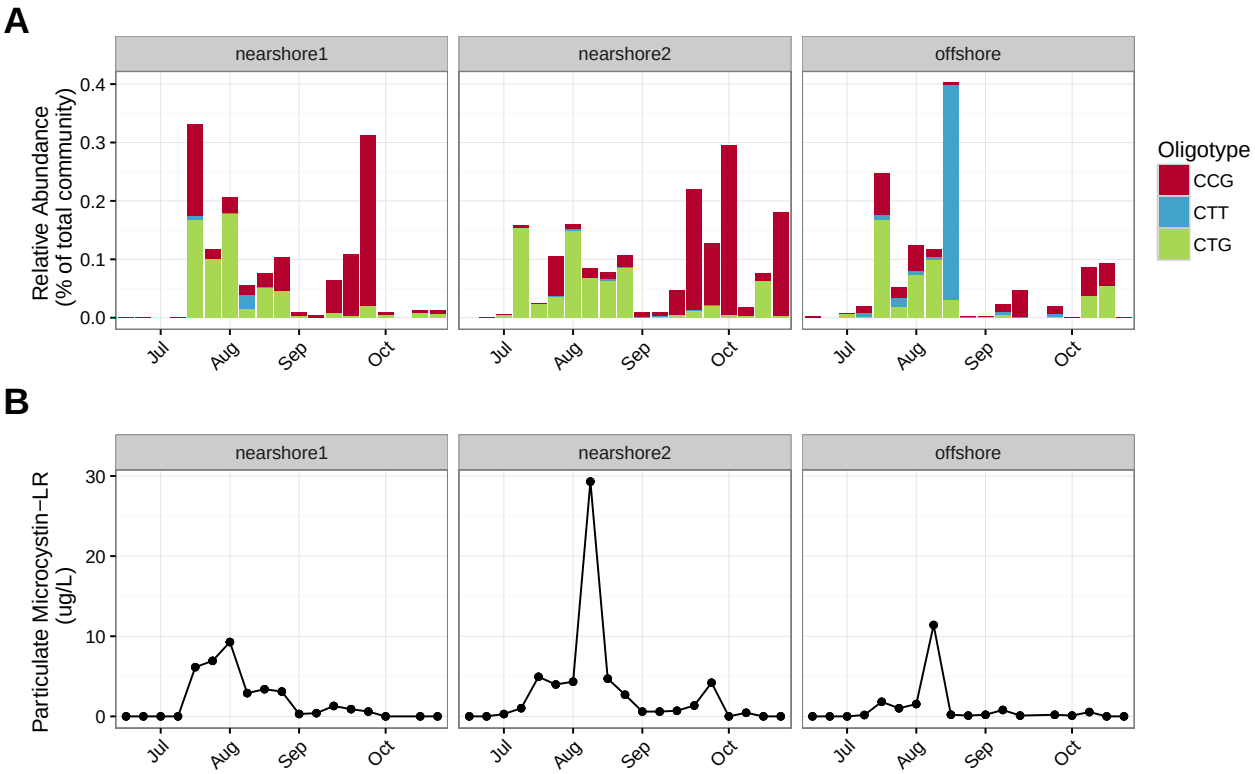
```
toxinGrobWider <- gtable_add_cols(toxinGrob, widths = oligoGrob$widths[10])

# rbind the two plots together
plot <- rbind(oligoGrob, toxinGrobWider, size = "first")


erie_plot <-
  ggdraw() +
    draw_plot(plot, x = 0.02, y = 0.02, width = 0.96, height = 0.94) +
    draw_plot_label(c("A", "B"), c(0, 0), c(1, .52), size = 20)

ggsave("../plots/raw-plots/erie-oligo-toxin-plot.pdf", plot = erie_plot, width = 10, height = 6)

erie_plot
```



Statistics

Median Chla

```
erie_oligo_df %>%
  group_by(Station) %>%
  summarise(median(Chla)) %>%
  pander()
```

Station	median(Chla)
nearshore1	18.555
nearshore2	13.720
offshore	5.905

Median TP levels

```
erie_oligo_df %>%
  group_by(Station) %>%
  summarise(median(TP)) %>%
  pander()
```

Station	median(TP)
nearshore1	46.25
nearshore2	44.30
offshore	18.70

CTG, CCG, and toxicity

Filter out dates with no microcystis

```
# Ratio of CTG to CCG
ratio <-
  oligos_scale %>%
  filter((CTG + CCG + CTT) > 0)
```

Look at the average ratio of CTG to CCG in July and August

```
ratio %>%
  filter(Month %in% c( "July", "August")) %>%
  summarise(median(CTG_CCG_ratio)) %>%
  pander()
```

median(CTG_CCG_ratio)
4.339

Look at the average ratio of CTG to CCG in September and October

```
ratio %>%
  filter(Month %in% c("September", "October")) %>%
  summarise(median(CTG_CCG_ratio)) %>%
  pander()
```

median(CTG_CCG_ratio)
0.2347

Is there a significant correlation between CTG relative abundance and ParMC?

```
CTG <-
  erie_oligo_df %>%
  filter(Oligotype == "CTG")

cor.test(CTG$count, CTG$ParMC, method = "spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  CTG$count and CTG$ParMC
## S = 6918.7, p-value = 5.537e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7046571
```

CTT and nutrient gradients

```
# Mean relative abundance
CTT <-
  erie_oligo_df %>%
    group_by(Shore) %>%
    filter(!is.na(count)) %>%
    filter(Oligotype == "CTT")

CTT %>%
  summarise(median(count)) %>%
  pander()
```

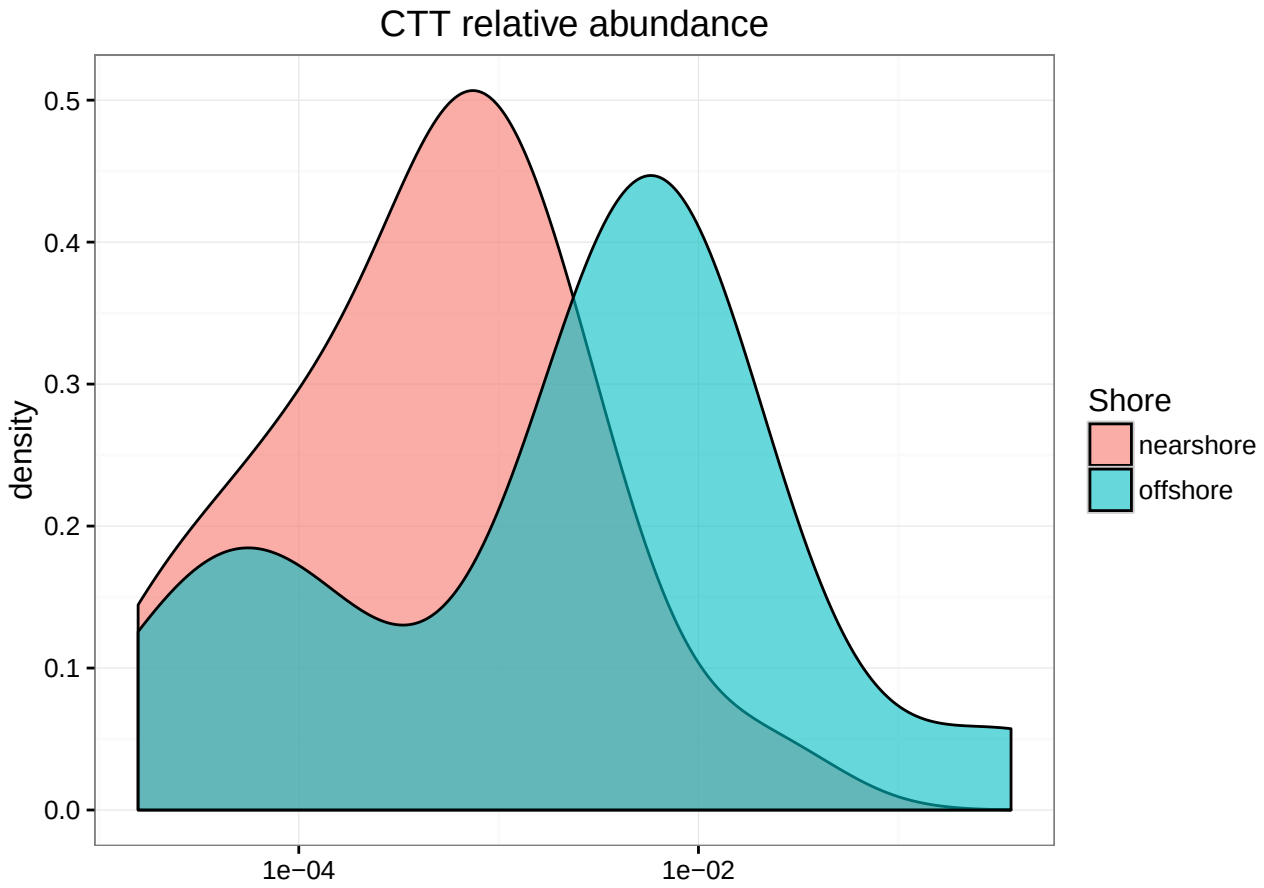
Shore	median(count)
nearshore	0.0002292592
offshore	0.0028197353

Distribution of nearshore and offshore abundance of CTT

```
CTT_plot <-
  ggplot(CTT, aes(x = count, group = Shore, fill = Shore)) +
    geom_density(alpha = .6, position = "identity") +
    scale_x_log10() +
    xlab("") +
    ggtitle("CTT relative abundance")

ggsave("../plots/raw-plots/CTT-distribution.pdf", plot = CTT_plot, width = 7, height = 5)

CTT_plot
```



Wilcoxon test to determine if there is a significant difference in CTT abundance at nearshore and offshore stations

```
wilcox.test(count ~ Shore, data = CTT)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  count by Shore
## W = 192.5, p-value = 0.04026
## alternative hypothesis: true location shift is not equal to 0
```

The wilcoxon test does not give very strong power because of small sample sizes and uneven classes. A permutational test for a significant difference in the medians would be more appropriate here.

```
# Permutation test to see if the medians of two distributions
# are significantly different.
#
# Args:
#   x: numeric vector for first distribution
#   y: numeric vector for second distribution
#   k: numer of permutations
# Returns:
#   a p-value
```

```
perm.test <- function(x, y, k = 10000) {
  n1 <- length(x)
  n2 <- length(y)
  # Generate k different permutations of x and y
  perms <- replicate(k, sample(c(x, y)))
  # Calculate the median for length x and length y
  mediandifs <- apply(perms[1:n1, ], 2, median) -
    apply(perms[(n1 + 1):(n2 + n1), ], 2, median)
  # What fraction of permuted difference in medians are larger than the observed difference?
  p <- sum(abs(mediandifs) > abs(median(x) - median(y)))/k
  return(p)
}

nearshoreCTT <- CTT %>% filter(Shore == "nearshore")
offshoreCTT <- CTT %>% filter(Shore == "offshore")

set.seed(1)
perm.test(x = nearshoreCTT$count, y = offshoreCTT$count, k = 10000)
```

```
## [1] 6e-04
```

The permutation test gives a much more significant result

Supplementary figure 2

Here we read in the RaxML tree for the cultures and calculate their patristic distances

```
# Read in best raxml tree
rax_tree <- read.tree("data//RAxML_bestTree.bs100_mlst")

# Calculate patristic distances
patristic <- cophenetic(rax_tree)

# Remove NIES
pat_dist <- as.dist(patristic[rownames(patristic) != "NIES483", colnames(patristic) != "NIES483"])
```

Here we calculate the hamming distance for the culture 16S-based oligotype sequence variants

```
# Filter to just culture samples
culture_oligos <-
  mc_oligos %>%
    mutate(sampID = substr(samples, 2, 10)) %>%
    filter(sampID %in% labels(pat_dist))


# Find the consensus oligotype for each culture
ConsOligo <- apply(culture_oligos, MARGIN = 1, function(x) {names(which.max(x[2:6]))} )
culture_oligos$ConsOligo <- ConsOligo

# Order the samples the same as they are ordered in the tree
target <- labels(pat_dist)
```

```
culture_oligos_sorted <- culture_oligos[match(target, culture_oligos$sampID), ]


# Calculate hamming distance
ham_dist <- adist(culture_oligos_sorted$ConsOligo)
rownames(ham_dist) <- culture_oligos_sorted$sampID
colnames(ham_dist) <- culture_oligos_sorted$sampID
ham_dist <- as.dist(ham_dist)

# Sanity check
labels(ham_dist) == labels(pat_dist)
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [43] TRUE TRUE TRUE TRUE
```

Plot hamming distance vs patristic distance

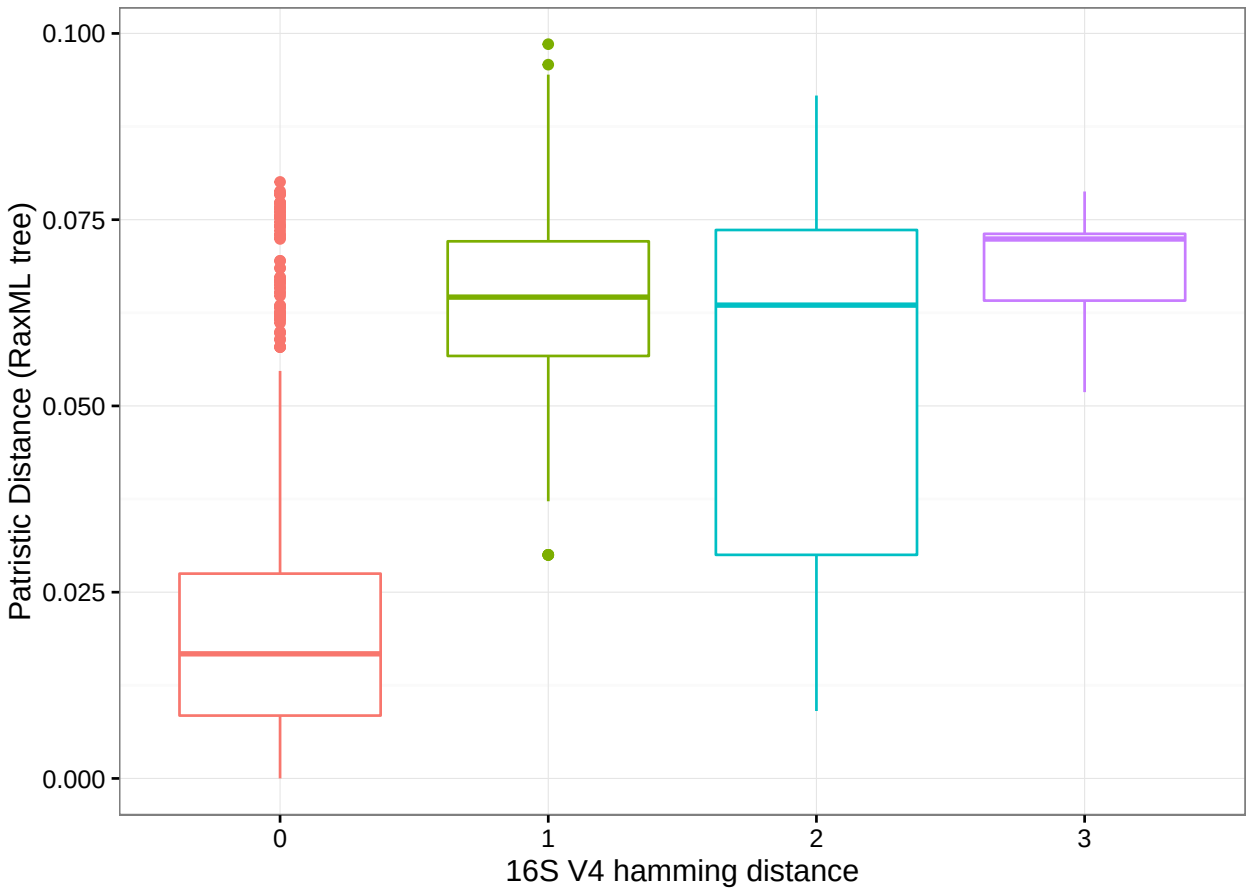
```
df <- data.frame(patDist = as.vector(pat_dist), hamDist = as.vector(ham_dist))

df$hamDist <- as.factor(df$hamDist)

hamdist_plot <-
  ggplot(df, aes(x = hamDist, y = patDist, color = hamDist)) +
    geom_boxplot() +
    ylab("Patristic Distance (RaxML tree)") +
    xlab("16S V4 hamming distance") +
    theme(legend.position = "none")

ggsave("../plots/raw-plots/hamdist-vs-patristic-plot.pdf", plot = hamdist_plot, width = 7, height = 5)

hamdist_plot
```


0.100
0.075
0.050
0.025
0.000
Patristic Distance (RaxML tree)
0
1
2
3
16S V4 hamming distance