Analysis Illumina data

Heleen

8/6/2020

1. Pre blast

Load table and check read numbers

Functions:

```
getData <- function(){
  otu <- fread("~/Documents/derep_illum/changedheader/otu.about")
  before <- nrow(otu)
  # Remove X. or X from colnames
  names(otu) <- sub("#", "", names(otu))
  otu <- column_to_rownames(otu, var = "OTU ID")}

# optional
remove_chimeras <- function(){
  chimeras <- read.csv("~/Documents/IlluminaAdaptertrimmedAllreps/thingremoved", header=FALSE, sep=";")
  otu <- column_to_rownames(otu, var = "OTU.ID")
  otu <- otu[ ! sub("^.*?:", "", otu$OTU.ID) %in% chimeras$V1,] ##remove all chimeras
  after <- nrow(otu)
  cat(paste("removed", before-after, "chimeric sequences\n\n"))
  rownames(otu) <- NULL
  return(otu)}

# optional chimera removal: otu <- remove_chimeras()

# print results nicely:
myOTUcat <- function(){
  #total read sum in all clusters
  total_reads <- sum(rowSums(otu))
  cat(paste('total reads (grand total with which clustering was done):\n',
            total_reads))
  cat("\n\n", 'Summary statistics of number of reads per OTU:\n')
  print(summary(rowSums(otu)))
  cat(paste("\n\n",
            'Total number of OTUs (including singletons):\n', nrow(otu)))
}
```

Execution:

```
otu <- getData()
myOTUcat()
```

```
## total reads (grand total with which clustering was done):
##  2512237
##
##  Summary statistics of number of reads per OTU:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     1.00     1.00     1.00     8.46     2.00 86074.00
##
##
##  Total number of OTUs (including singletons):
##  296884
```

Prepare Saba location data

Functions:

```
#function to change decimal to comma in one
decimal_to_comma <- function(data, column){
  data[,column] <- sub(",", ".",
                       data[,column],
                       fixed = TRUE)}

prepLocSaba <- function(){
  ## load the Saba sample location data
  locdata_saba <- read.delim("~/Downloads/NICO5-eDNA-64PE432-Metadata-MinIon - DataFilterSaba.txt")

  ## Change samplenames, colnames in metadatafile so they match the OTU file making merging is possible
  ## Change decimal to comma for computation.
  locdata_saba[,1] <- gsub("(?<![0-9])0+", "", locdata_saba[,1], perl = TRUE)
  locdata_saba[,1] <- gsub("\\.", "_", locdata_saba[,1], perl = TRUE)
  locdata_saba[,1] <- tolower(locdata_saba[,1])

  ## change long colnams to lat,long, altitude
  names(locdata_saba)[names(locdata_saba)=="geo_lat..in.decimalen..WGS84."] <- "lat"
  names(locdata_saba)[names(locdata_saba)=="geo_lon..in.decimalen..WGS84."] <- "long"
  names(locdata_saba)[names(locdata_saba)=="altitude..in.meters.aasl."] <- "altitude"
  names(locdata_saba)[1] <- "sample"

  ## change decimals to commas
  for (col in c("lat", "long")){
    locdata_saba[, col] <- as.numeric(decimal_to_comma(locdata_saba, col))}
  return(locdata_saba)}

write.table(prepLocSaba(), file = "locationdata_saba.tsv", sep = "\t", col.names = TRUE, quote = FALSE)
```

Execution:

```
#  establish samples and controls
controls <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71", "0", "unicon1",
              "unicon1A", "neg_controle")
samples <- names(otu)[-which(names(otu) %in% controls)]

locdata_saba <- prepLocSaba()

# take only the columns needed, and the samplenames needed
locdata_saba <- locdata_saba %>%
  select(sample, lat, long, altitude, habitat) %>%
  filter(sample %in% samples)
```

load and prepare Statia data

select only relevant columns and rows, and setnames, and change the numbers to depth

```
# take relevant columns, take out the samples that are not in the OTU table, and set the colnames to sa
locdata_statia <- read.delim("~/statia_location.txt") %>%
  select(Field.nr., lat, long, Average.depth) %>%   # select relevant columns
  filter(!Field.nr. %in% c(528, 529)) %>% # discard irrelevant rows
  setNames(c("sample", "lat", "long", "altitude")) %>% # change column names
  mutate(altitude = as.numeric(gsub('[+]', '', altitude)) * -1) # mutate altidue column to negative
```

Bind Saba and statia data by row (to get merged data frame (mdf))

```
mdf <- plyr::rbind.fill(locdata_saba, locdata_statia)
```

estimate boxcore altitude data

boxcore altitude data is missing, so it's estimated by taking the nearest point geographically of which altitude data is available

```
#fill in missing boxcore altitude data wth nearestby latitude , the lowest value of that
mdf <- mdf %>%
  group_by(lat) %>%
  # arrange the groups by descending altitude within the groups
  arrange(desc(altitude), .by_group = TRUE) %>%
  # make new column with lowest altitude of group if the value is missing
  mutate(altitude = ifelse(is.na(altitude), min(altitude, na.rm = TRUE), altitude)) %>%
  # because for some boxcore samples it was taken at a slgihty different latitude, it does not belong t
  # Thus, R introduces infinite values which this command changes to NA values
  mutate(altitude = ifelse(is.infinite(altitude), NA, altitude)) %>%
  # needs to be ungrouped to fill it with the nearest & lowest altitude
  ungroup() %>%
  fill(altitude, .direction = 'down')
```

```
## Warning in min(altitude, na.rm = TRUE): no non-missing arguments to min;
## returning Inf

## Warning in min(altitude, na.rm = TRUE): no non-missing arguments to min;
## returning Inf
```

Put every sample in north, south or statia catogery based on latitude, to enable exchange testing. Add a tag indication what region the sampling location is in: Saba north, Saba south, or Statia.

```
mdf$tag <- ifelse(grepl("^[0-9]+$", mdf$sample), 'Statia',
                  ifelse(mdf$lat > 17.55, "Saba North",
                         "Saba South")) %>% as.factor()
```

Control reads

investigate control reads

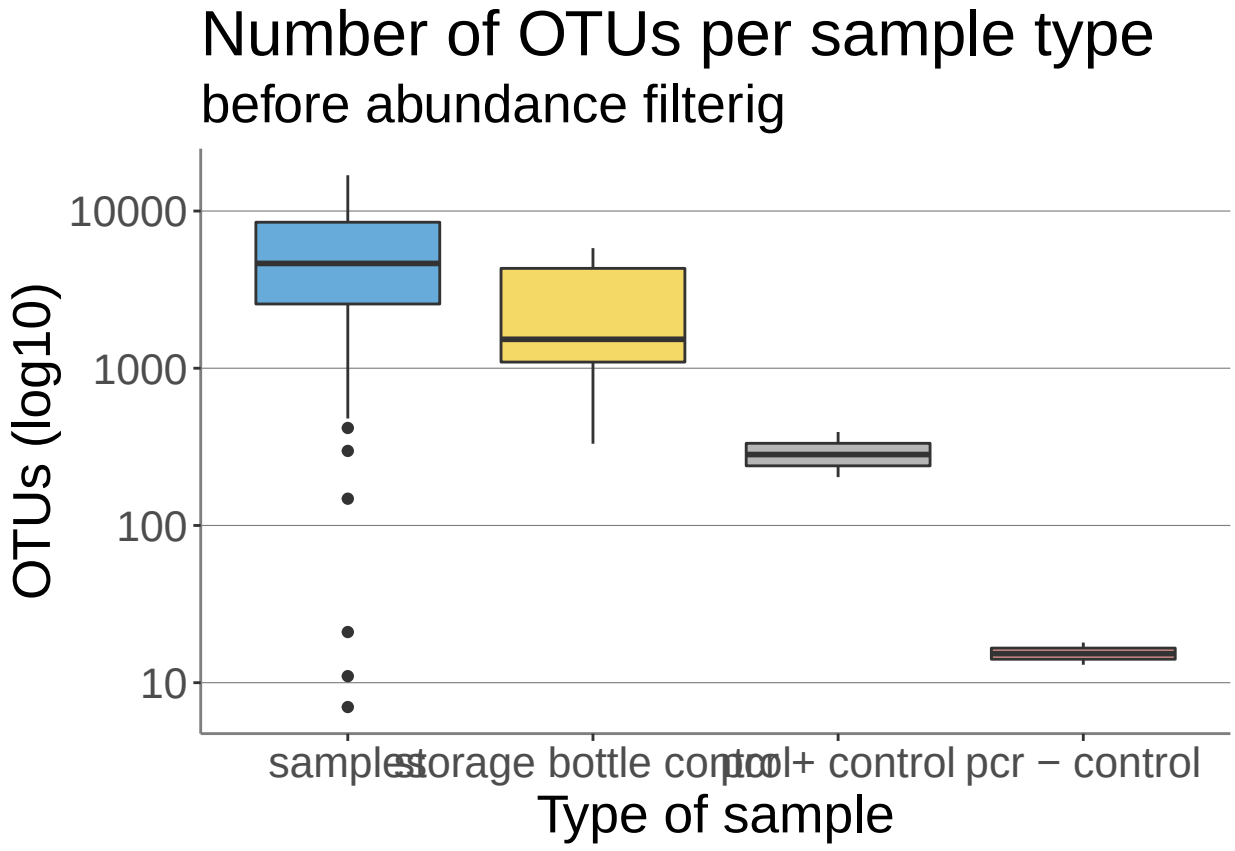
####Functions:

```
#prep data
controlDf <- function(){
  copy <- otu
  copy[copy>0] <- 1
  per_type <- copy %>% colSums() %>%
    as.data.frame() %>% rownames_to_column(var = "sample")%>%
    mutate(type = ifelse(sample %in% bottlecontrol, "storage bottle control",
                               ifelse(grepl("unicon", sample), "pcr + control",
                                      ifelse(sample %in% c("0", "neg_controle"),
                                             "pcr - control", "samples")))) %>%
    mutate(type = fct_reorder(type, desc(.)))
  return(per_type)}


plot_pertype <- function(df){
  control_plotOTU <-
    ggplot(df, aes(x = type,
                   y = .,
                   fill = type)) +
    geom_boxplot() +
    labs(title = "Number of OTUs per sample type",
         subtitle = "before abundance filterig",
       x = "Type of sample",
       y = "OTUs (log10)") +
    scale_fill_jco(alpha = 0.6) +
    scale_y_log10() +
  # edit lines and background
    theme(text = element_text(size = 20),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line("gray50", size = 0.2),
        panel.background = element_blank(),
        axis.line = element_line("gray50"),
```

```
        legend.position = "none")
  control_plotOTU}
```

Execution:

```
controls <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71", "0", "unicon1",
              "unicon1A", "neg_controle")
bottlecontrol <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71")


controlDf <- controlDf()
plot_pertype(controlDf)
```



anova of storage bottle control

```
res.aov <- aov(d = controlDf, . ~ type)
summary(res.aov)
```

```
##             Df    Sum Sq  Mean Sq F value Pr(>F)
```

```
## type         3 1.763e+08 58764044   3.444 0.0198 *
## Residuals   96 1.638e+09 17060989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(res.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = . ~ type, data = controlDf)
##
## $type
##                                           diff        lwr       upr     p adj
## storage bottle control-samples       -3137.205  -7376.563  1102.152 0.2204976
## pcr + control-samples                -5496.348 -13218.159  2225.463 0.2517563
## pcr - control-samples                -5778.848 -13500.659  1942.963 0.2117674
## pcr + control-storage bottle control -2359.143 -11018.102  6299.817 0.8919758
## pcr - control-storage bottle control -2641.643 -11300.602  6017.317 0.8553299
## pcr - control-pcr + control           -282.500 -11082.120 10517.120 0.9998844
```

```
# check for assumptions
check_assumption <- function(){
  plot(res.aov, 1) # homogeneity of variances
  plot(res.aov, 2) # normality of residuals
  shapiro.test(residuals(res.aov))} # shapiro wilk of anova residuals
```

Post-blast

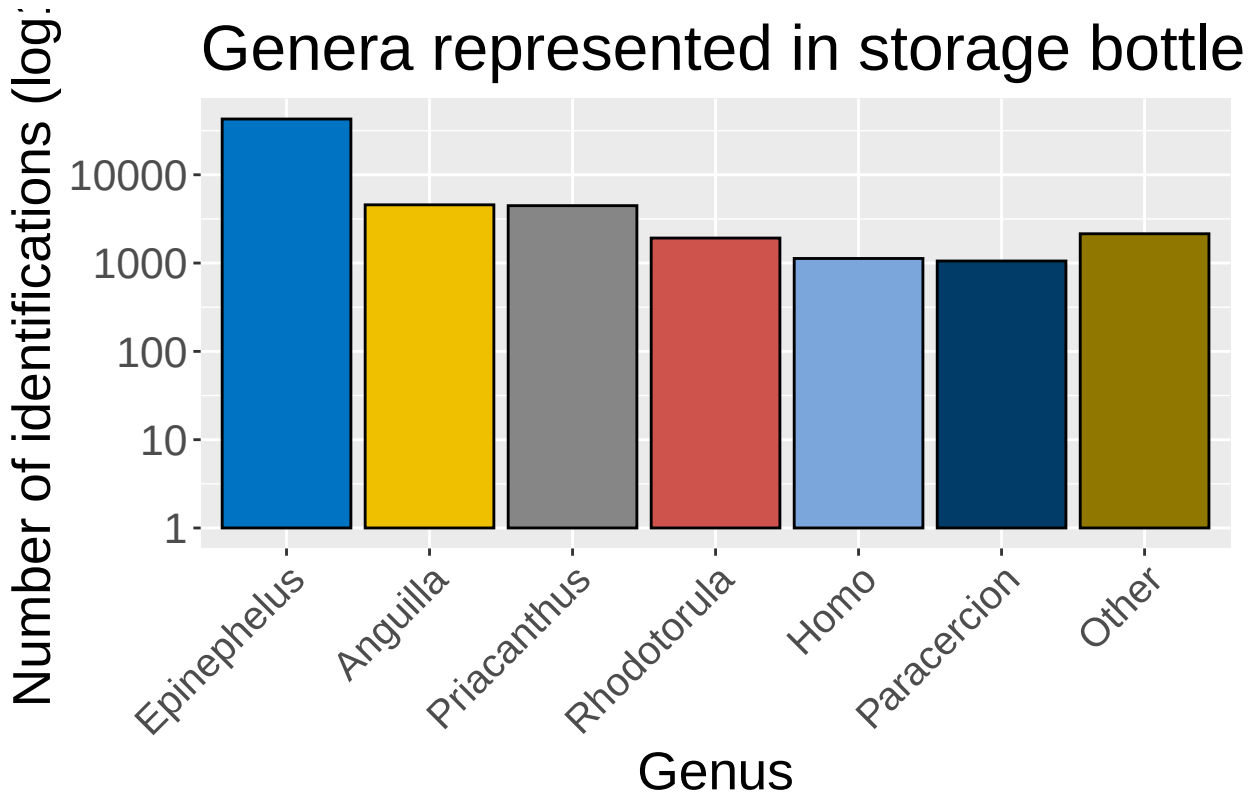
investigate storage bottle control identifications

```
`lca storage` <- read.delim("~/Documents/derep_illum/controls/underep/taxadded/lca") ## load lca file o

species <- table(`lca storage`$X.genus) %>%
  data.frame() %>%
  mutate(Var1 = ifelse(Freq < 1000, "Other", as.character(Var1))) %>%
  filter(!Var1=="no identification") %>%
  group_by(Var1) %>%
  dplyr::summarise(Freq = sum(Freq)) %>%
  mutate(Prop = (Freq/sum(Freq))*100) %>%
  ungroup() %>%
  mutate(Var1 = fct_reorder(Var1, desc(Freq))) %>%
  mutate(Var1 = fct_relevel(Var1, "Other", after = Inf))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

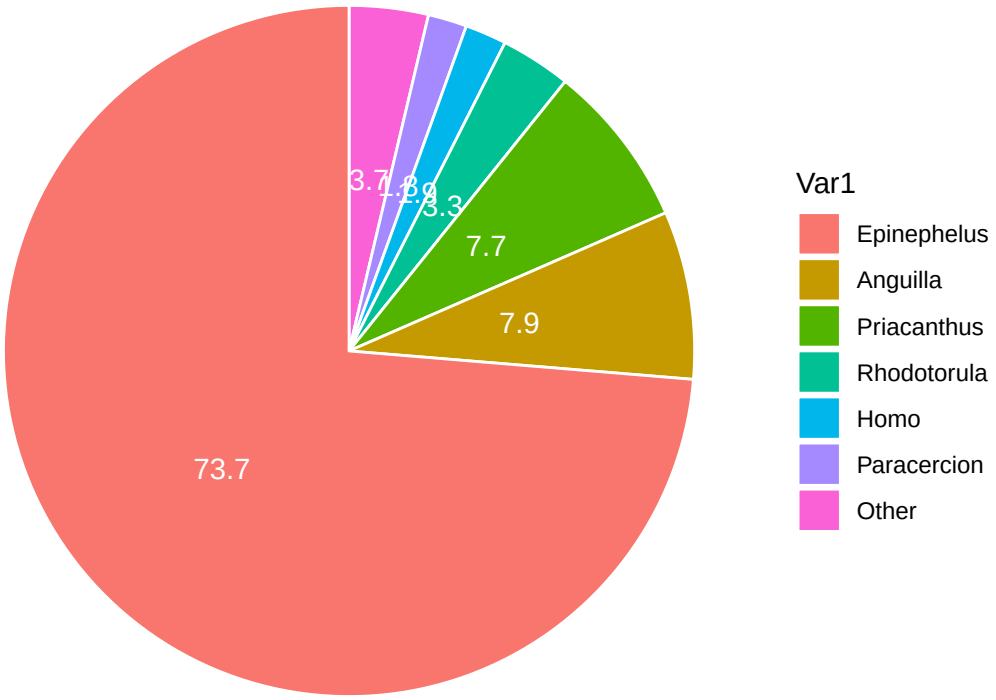
```
get_col <- function(){
  colorcount <- length(genuscount$Var1)
  qual_col <- brewer.pal.info[brewer.pal.info$category == "qual",]
```

```
  col_vector <- unlist(mapply(brewer.pal,
                              qual_col$maxcolors, rownames(qual_col)))
  mycol <- sample(col_vector, colorcount)}


ggplot(species, aes(x = Var1, y = Freq, fill = Var1)) +
  geom_bar(stat = "identity", color = "black") +
  theme(text = element_text(size = 20),
        axis.text.x = element_text(angle = 45, hjust =1, size = 15),
        legend.position = "none") +
  scale_fill_jco() +
  labs(title = "Genera represented in storage bottles",
       x = "Genus\n" ,
       y = "Number of identifications (log10)") +
  scale_y_log10()
```




```
# Pie Chart
# add position of label
count.data <- species %>%
  arrange(desc(Var1)) %>%
  mutate(lab.ypos = cumsum(Prop) - 0.5*Prop)

ggplot(count.data, aes(x = "", y = Prop, fill = Var1)) +
     geom_bar(width = 1, stat = "identity", color = "white") +
     coord_polar("y", start = 0)+
```

```
    geom_text(aes(y = lab.ypos, label = round(Prop,1)), color = "white")+
    theme_void()
```



Filter out out controls

If a OTU also contains control reads, these need to be filtered out of the samples contain them in frequencies that are close to the control frequencies. This could be contamination from the bottles the sample was stored in, or PCR contamination.

```
posContamination <- function(){
  contam <- otu %>% filter(unicon1 > 5000) %>% select(!c("unicon1",
              "unicon1A")) %>% sum()
  total <- otu %>% filter(unicon1 > 5000) %>% sum()
  freq <- contam/total
  cat(paste("Contamination percentage of positive control in other samples: \t", round(freq*100, 5), "\
  return(freq)}

negContamination <- function(){
  contam <- otu %>% select(neg_controle) %>% sum()
  total <- otu %>% filter(neg_controle>0) %>% sum()
  freq <- contam/total
  cat(paste("Contamination percentage in negative samples: \t", round(freq*100, 5)))}
```

```
rate <- posContamination()
```

```
## Contamination percentage of positive control in other samples:    0.00671
```

```
negContamination()
```

```
## Contamination percentage in negative samples:     0.0612
```

Low abundance filter

the rate of contamination in the positive control was used as low abundance filter rate.

```
lowAbuncanceFilter <- function(rate){
  before <- nrow(otu)
  colsum <- colSums(otu)
  min_read <- colsum * rate  # if OTU contains less than this many reads, filter out
  otu <-
    mapply(col = otu, min = min_read, function(col, min){
    col[col < min] <- 0
    col}) %>%
    as.data.frame () %>%
    `rownames<-`(rownames(otu)) %>% filter(!rowSums(.[samples]) == 0) # take out "empty" otus
  after <- nrow(otu)
  percenage_ret <- ((before-after)/before)*100
  cat(paste("filtered out ", before-after, " OTUs, which is ",
            round(percenage_ret, 2), "% of original OTUs
            \n\n",
            after, " OTUs were retained", sep = ""))
  return(otu)
}

controls <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71", "0", "unicon1",
              "unicon1A", "neg_controle")
bottlecontrol <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71")
samples <- names(otu)[-which(names(otu) %in% controls)]

otu <- lowAbuncanceFilter(rate = rate)
```

```
## filtered out 244221 OTUs, which is 82.26% of original OTUs
##
##
## 52663 OTUs were retained
```

remove singleton OTUs

```
remove_singletons <- function(){
  OTUbefore <- nrow(otu)
  otu <- otu %>% filter(!rowSums(.[samples]) < 2) # remove all rows where rowSum == 1 (singleton OTU)
  OTUafter <- nrow(otu)
  removed <- OTUbefore-OTUafter
  retained_percent <- round(OTUafter/OTUbefore*100, 2)
  cat(paste("removed", removed, "singletons\n",
            "retained", OTUafter, "OTUs, which means", retained_percent, "percent of OTUs was retained"
            sep = " "))
  return(otu)}

otu <- remove_singletons()
```

```
## removed 12254 singletons
##  retained 40409 OTUs, which means 76.73 percent of OTUs was retained
```

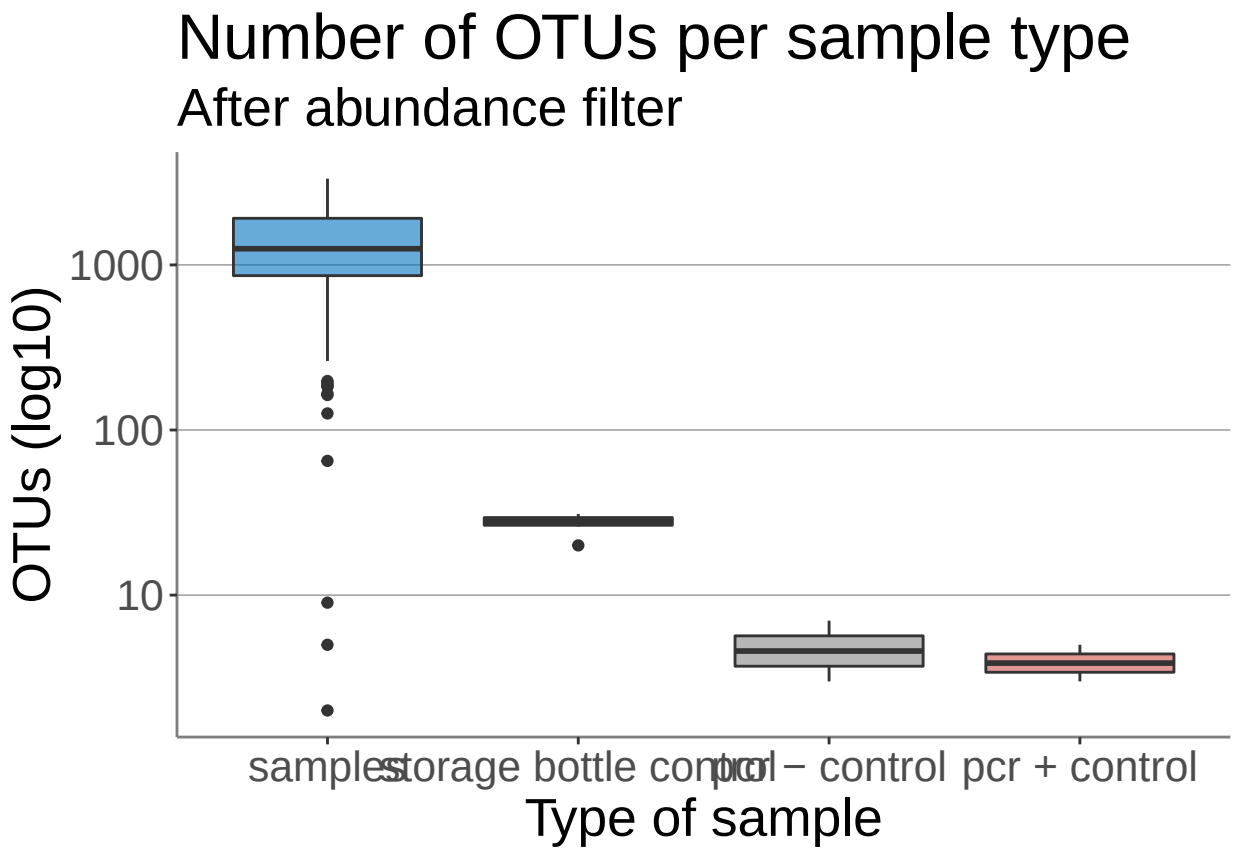
plot number of otus per sample

```
saba <- samples[grepl("sxm", samples)]


copy <- otu
copy[copy>0] <- 1
copy <- copy %>% colSums() %>%
  as.data.frame() %>% rownames_to_column(var = "sample")%>%
  mutate(type = ifelse(sample %in% bottlecontrol, "storage bottle control",
                               ifelse(grepl("unicon", sample), "pcr + control",
                                      ifelse(sample %in% c("0", "neg_controle"),
                                             "pcr - control", "samples")))) %>%
  mutate(type = fct_reorder(type, desc(.)))


control_plotOTU <-
  ggplot(copy, aes(x = type,
                           y = .,
                           fill = type)) +
  geom_boxplot() +
  labs(title = "Number of OTUs per sample type",
       subtitle = "After abundance filter",
       x = "Type of sample",
       y = "OTUs (log10)") +
  scale_fill_jco(alpha = 0.6) +
  scale_y_log10() +
  # edit lines and background
  theme(text = element_text(size = 20),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line("gray50", size = 0.2),
        panel.background = element_blank(),
        axis.line = element_line("gray50"),
        legend.position = "none")
```

```
control_plotOTU
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
ggsave("controlplot AFTER abundance filter", plot = control_plotOTU, device = "png", height = 7, width =
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

ANOVA

```
copy <- otu
copy[copy>0] <- 1
copy <- copy %>% colSums() %>%
  as.data.frame() %>% rownames_to_column(var = "sample")%>%
  mutate(type = ifelse(sample %in% bottlecontrol, "storage bottle control",
                               ifelse(grepl("unicon", sample), "pcr + control",
                                      ifelse(sample %in% c("0", "neg_controle"),
```

```
                                          "pcr - control", "samples")))) %>%
  mutate(type = fct_reorder(type, desc(.)))

res.aov <- aov(d = copy, . ~ type)
summary(res.aov)
```

```
##             Df   Sum Sq Mean Sq F value   Pr(>F)
## type         3 17706977 5902326   9.744 1.13e-05 ***
## Residuals   96 58149328  605722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(res.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = . ~ type, data = copy)
##
## $type
##                                              diff       lwr        upr     p adj
## storage bottle control-samples       -1338.00642 -2136.800 -539.21259 0.0001758
## pcr - control-samples                -1356.29213 -2811.261   98.67709 0.0769222
## pcr + control-samples                -1357.29213 -2812.261   97.67709 0.0765985
## pcr - control-storage bottle control   -18.28571 -1649.836 1613.26414 0.9999909
## pcr + control-storage bottle control   -19.28571 -1650.836 1612.26414 0.9999893
## pcr + control-pcr - control             -1.00000 -2035.900 2033.90018 1.0000000
```

```
# check for assumptions
check_assumption <- function(){
  plot(res.aov, 1) # homogeneity of variances
  plot(res.aov, 2) # normality of residuals
  shapiro.test(residuals(res.aov))} # shapiro wilk of anova residuals
```

Additional bottle control contamination check for saba samples

And remove controls from otu table

```
# establish controls and samples
controls <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71", "0", "unicon1",
              "unicon1A", "neg_controle")
bottlecontrol <- c("sxm_2018_62", "sxm_2018_63", "sxm_2018_64",
              "sxm_2018_65", "sxm_2018_66", "sxm_2018_70",
              "sxm_2018_71")
samples <- names(otu)[-which(names(otu) %in% controls)]
saba <- samples[grepl("sxm", samples)]

# go over the rows(OTUs) where there are control reads and change any reads to 0 if they contain less t
filter_controls <- function(){
```

```
  OTUbefore <- nrow(otu)
  mcr <- do.call(pmax, otu[bottlecontrol]) # max control value for each otu
  mcp <- mcr > 0                      # control values  > 0
  otu[mcp, saba][otu[mcp, saba] <  2*mcr[mcp]] <- 0
  # discard controls, and OTUs that have no reads associated bc of control filter
  otu <- otu %>%
    select(all_of(samples)) %>% #only keep samples
    filter(!rowSums(.) == 0)# discard OTUs that have no reads because of filtering
  ncolbefore <- ncol(otu)
  OTUafter <- nrow(otu)
  cat(paste("Control filtering removed", OTUbefore-OTUafter, "OTUs, which is ",
            round(((OTUbefore-OTUafter)/OTUbefore)*100, 2)), "%")
  otu <- otu[,colSums(otu) > 2000]
  # select only samples that have read counts of higher than two thousand
  ncolafter <- ncol(otu)
  after2000 <- nrow(otu)
  cat(paste("\n\nanother", OTUafter-after2000, "OTUs, were removed by removing ", ncolbefore-ncolafter,
  return(otu)}

otu <- filter_controls()
```

```
## Control filtering removed 33 OTUs, which is  0.08 %
##
## another 0 OTUs, were removed by removing  3 samples that head read numbers below 2000 reads
```

write sequences to blast to file

now that all control reads and singletons have been filtered out, the remaining OTUs can be blasted. For this, the OTU centroid sequences from filtering step at 98% are extracted and then blasted -> taxadded -> dummyadded(for LCA script to work) -> lca script. Then its back to R

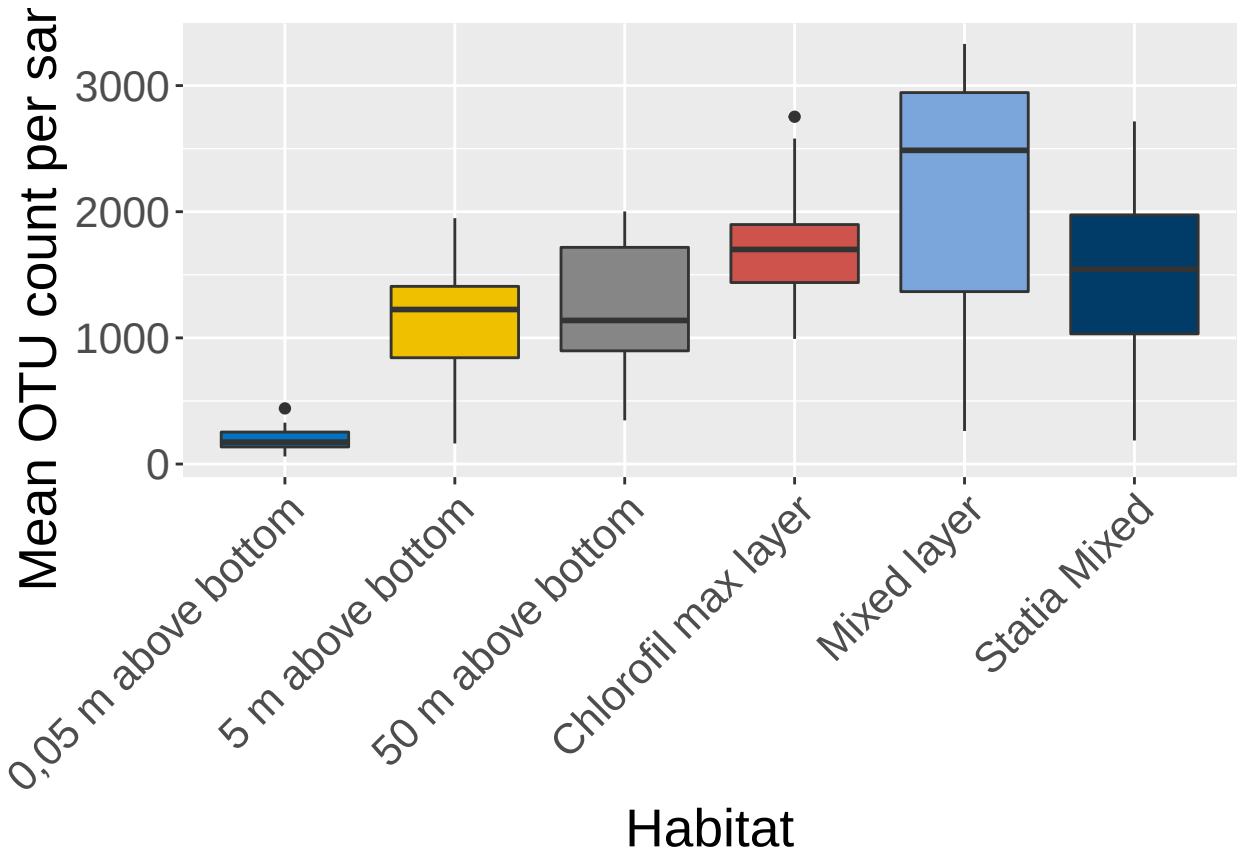
```
# make file with which sequences to be blasted can be selected (singletons filtered out)
write.table(rownames(otu),
            file = '~/OTUcentroids.txt',
            row.names = FALSE,
            quote = FALSE,
            col.names = FALSE)
nrow(otu)
```

```
## [1] 40376
```

number of reads by habitat

```
# prepare data for OTU per smample analysis by habitat
newotu <- otu
newotu[newotu > 0] <- 1

otu_habitat <- cbind(sample = names(colSums(newotu)),
                     OTUs = colSums(newotu)) %>%
  `colnames<-`(c("sample", "OTUs")) %>%
```

```
  merge(mdf, by="sample") %>%
  mutate(habitat = as.character(habitat)) %>%
  mutate(habitat = replace_na(habitat, "Statia Mixed")) %>%
  filter(!habitat == "100 m above bottom") %>%
  mutate(habitat = factor(habitat, levels = c("0,05 m above bottom", "5 m above bottom",
                                              "50 m above bottom", "Chlorofil max layer",
                                              "Mixed layer", "Statia Mixed")))

otu_habitat$OTUs <- as.numeric(as.character(otu_habitat$OTUs))

ggplot(otu_habitat, aes(x=habitat, y=OTUs, fill = habitat, group = habitat)) +
  geom_boxplot() +
  theme(text = element_text(size = 20),
        axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = "Habitat",
       y= "Mean OTU count per sample") +
  scale_fill_jco()
```




```
res.aov <- aov(OTUs ~ habitat, data=otu_habitat)
data.frame(table(otu_habitat$habitat))
```

```
##                  Var1 Freq
## 1 0,05 m above bottom    7
## 2    5 m above bottom   15
```

```
## 3   50 m above bottom   14
## 4 Chlorofil max layer   12
## 5         Mixed layer   12
## 6        Statia Mixed   25
```

```
tapply(otu_habitat$OTUs, otu_habitat$habitat, mean)
```

```
## 0,05 m above bottom    5 m above bottom   50 m above bottom Chlorofil max layer
##            207.5714           1071.7333           1262.8571           1750.6667
##         Mixed layer        Statia Mixed
##           2180.1667           1495.5200
```

```
tapply(otu_habitat$OTUs, otu_habitat$habitat, sd)
```

```
## 0,05 m above bottom    5 m above bottom   50 m above bottom Chlorofil max layer
##            131.3530            542.0267            510.3753            523.3188
##         Mixed layer        Statia Mixed
##           1030.1210            627.7306
```

```
summary(res.aov)
```

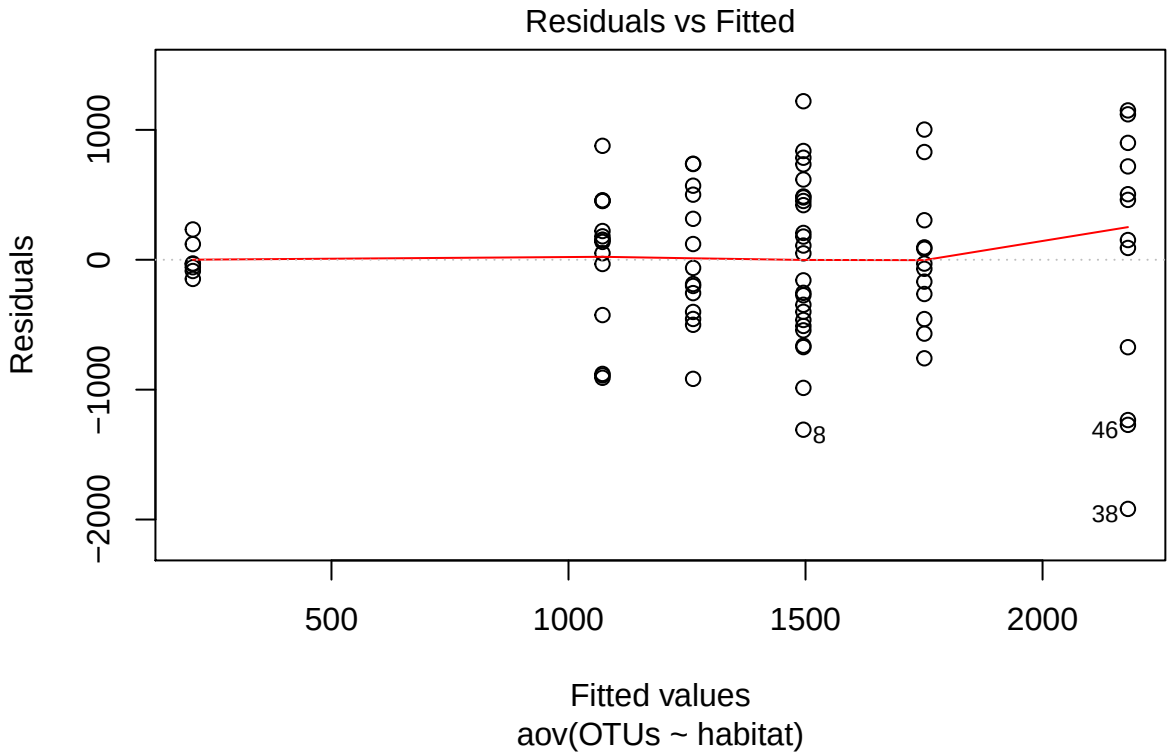
```
##             Df   Sum Sq Mean Sq F value   Pr(>F)
## habitat      5 20833607 4166721   10.37 1.14e-07 ***
## Residuals   79 31745127  401837
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

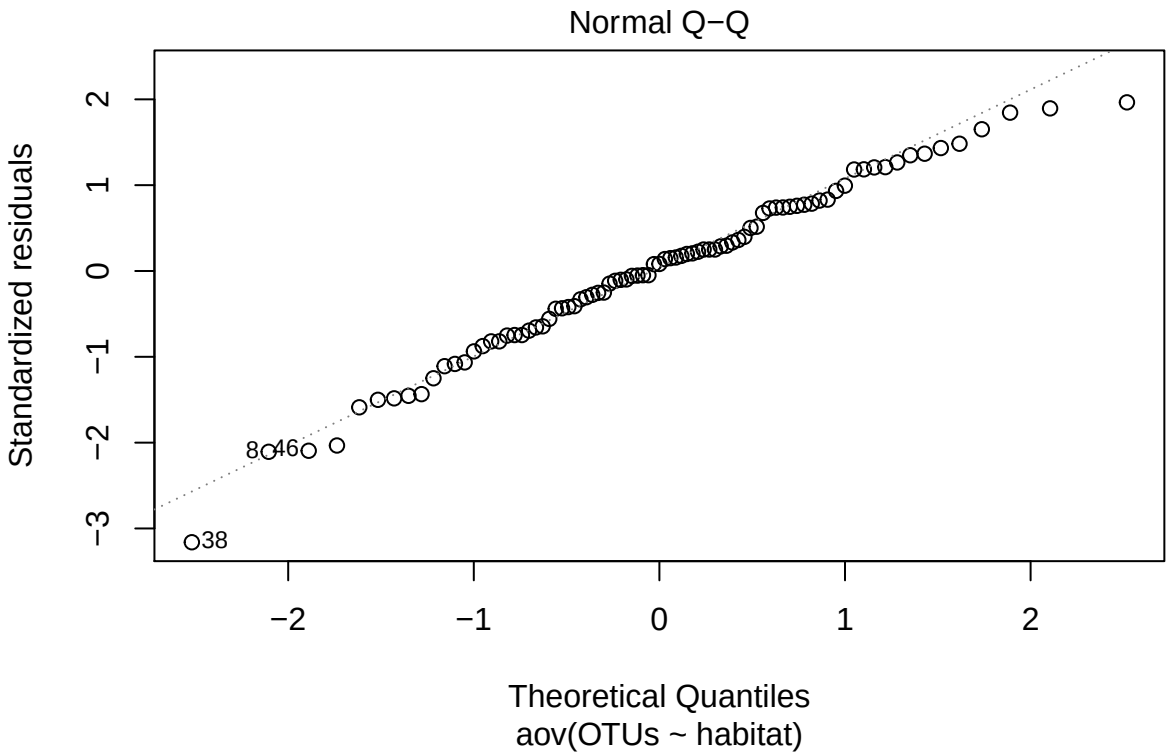
```
TukeyHSD(res.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = OTUs ~ habitat, data = otu_habitat)
##
## $habitat
##                                             diff         lwr        upr
## 5 m above bottom-0,05 m above bottom     864.1619    16.64377 1711.68004
## 50 m above bottom-0,05 m above bottom   1055.2857   198.19080 1912.38063
## Chlorofil max layer-0,05 m above bottom 1543.0952   662.51392 2423.67656
## Mixed layer-0,05 m above bottom         1972.5952  1092.01392 2853.17656
## Statia Mixed-0,05 m above bottom        1287.9486   496.19820 2079.69894
## 50 m above bottom-5 m above bottom       191.1238  -496.92882  879.17644
## Chlorofil max layer-5 m above bottom     678.9333   -38.16372 1396.03039
## Mixed layer-5 m above bottom            1108.4333   391.33628 1825.53039
## Statia Mixed-5 m above bottom            423.7867  -180.92267 1028.49600
## Chlorofil max layer-50 m above bottom    487.8095  -240.58109 1216.20014
## Mixed layer-50 m above bottom            917.3095   188.91891 1645.70014
## Statia Mixed-50 m above bottom           232.6629  -385.39708  850.72279
## Mixed layer-Chlorofil max layer          429.5000  -326.38667 1185.38667
## Statia Mixed-Chlorofil max layer        -255.1467  -905.38496  395.09163
## Statia Mixed-Mixed layer                -684.6467 -1334.88496  -34.40837
```

```
##                                             p adj
## 5 m above bottom-0,05 m above bottom    0.0430280
## 50 m above bottom-0,05 m above bottom   0.0071611
## Chlorofil max layer-0,05 m above bottom 0.0000306
## Mixed layer-0,05 m above bottom         0.0000001
## Statia Mixed-0,05 m above bottom        0.0001268
## 50 m above bottom-5 m above bottom      0.9646407
## Chlorofil max layer-5 m above bottom    0.0739654
## Mixed layer-5 m above bottom            0.0003072
## Statia Mixed-5 m above bottom           0.3258382
## Chlorofil max layer-50 m above bottom   0.3766839
## Mixed layer-50 m above bottom           0.0055200
## Statia Mixed-50 m above bottom          0.8801491
## Mixed layer-Chlorofil max layer         0.5621783
## Statia Mixed-Chlorofil max layer        0.8604275
## Statia Mixed-Mixed layer                0.0331370
```

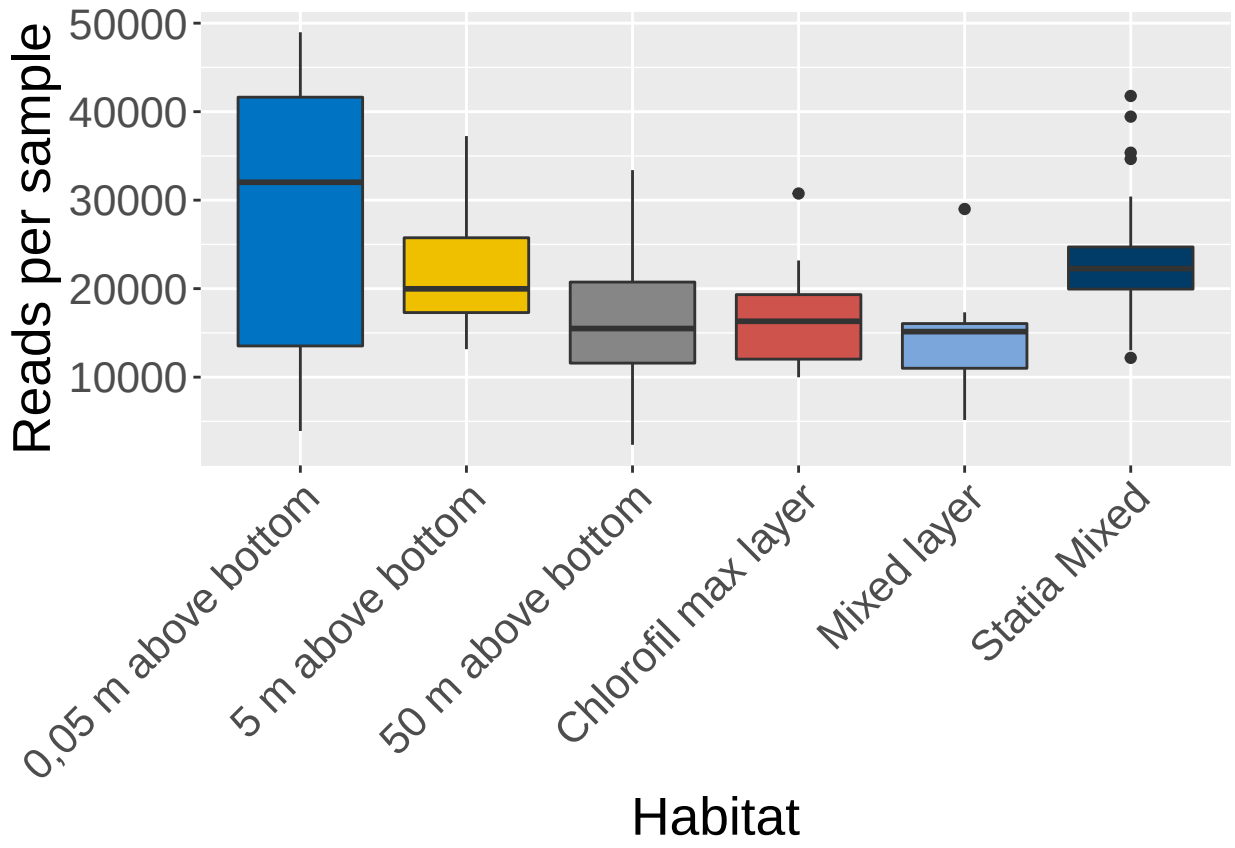
```
check_assumption()
```






```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(res.aov)
## W = 0.98533, p-value = 0.4502
```

```
reads_habitat <- cbind(sample = names(colSums(otu[,-1])),
                       OTUs = colSums(otu[,-1])) %>%
  `colnames<-`(c("sample", "OTUs")) %>%
  merge(mdf, by="sample") %>%
  mutate(habitat = as.character(habitat)) %>%
  mutate(habitat = replace_na(habitat, "Statia Mixed")) %>%
  filter(!habitat == "100 m above bottom") %>%
  mutate(habitat = factor(habitat, levels = c("0,05 m above bottom", "5 m above bottom",
                                              "50 m above bottom", "Chlorofil max layer",
                                              "Mixed layer", "Statia Mixed")))
reads_habitat$OTUs <- as.numeric(as.character(reads_habitat$OTUs))

ggplot(reads_habitat, aes(x=habitat, y=OTUs, fill = habitat, group = habitat)) +
  geom_boxplot() +
  theme(text = element_text(size = 20),
        axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = "Habitat",
       y= "Reads per sample") +
  scale_fill_jco()
```




```
res.aov <- aov(OTUs ~ habitat, data=reads_habitat)
data.frame(table(reads_habitat$habitat))
```

```
##                  Var1 Freq
## 1 0,05 m above bottom    7
## 2    5 m above bottom   15
## 3   50 m above bottom   14
## 4 Chlorofil max layer   12
## 5         Mixed layer   12
## 6        Statia Mixed   24
```

```
tapply(reads_habitat$OTUs, reads_habitat$habitat, mean)
```

```
## 0,05 m above bottom    5 m above bottom   50 m above bottom Chlorofil max layer
##            27890.57            21917.40            16210.71            16842.08
##         Mixed layer        Statia Mixed
##            14146.00            23696.42
```

```
tapply(reads_habitat$OTUs, reads_habitat$habitat, sd)
```

```
## 0,05 m above bottom    5 m above bottom   50 m above bottom Chlorofil max layer
##           17614.839            7034.437            7689.281            6002.955
```

```
##         Mixed layer        Statia Mixed
##            6143.475            7677.992
```

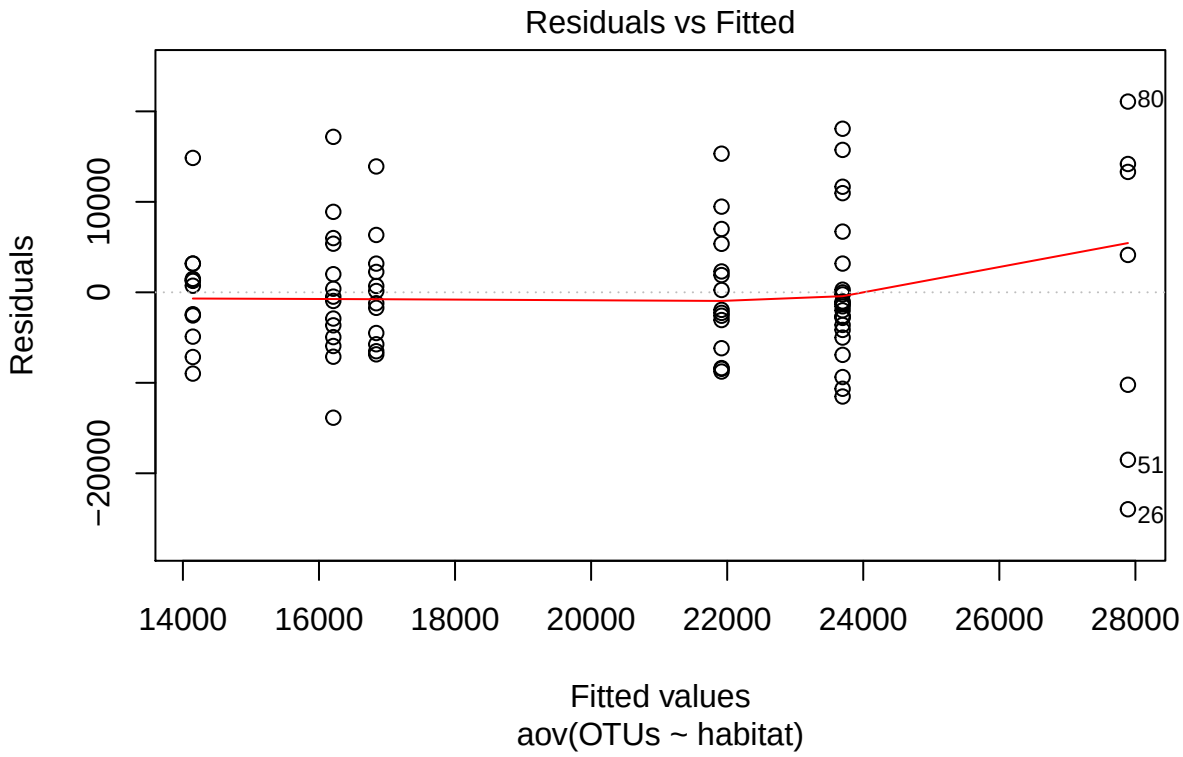
```
summary(res.aov)
```

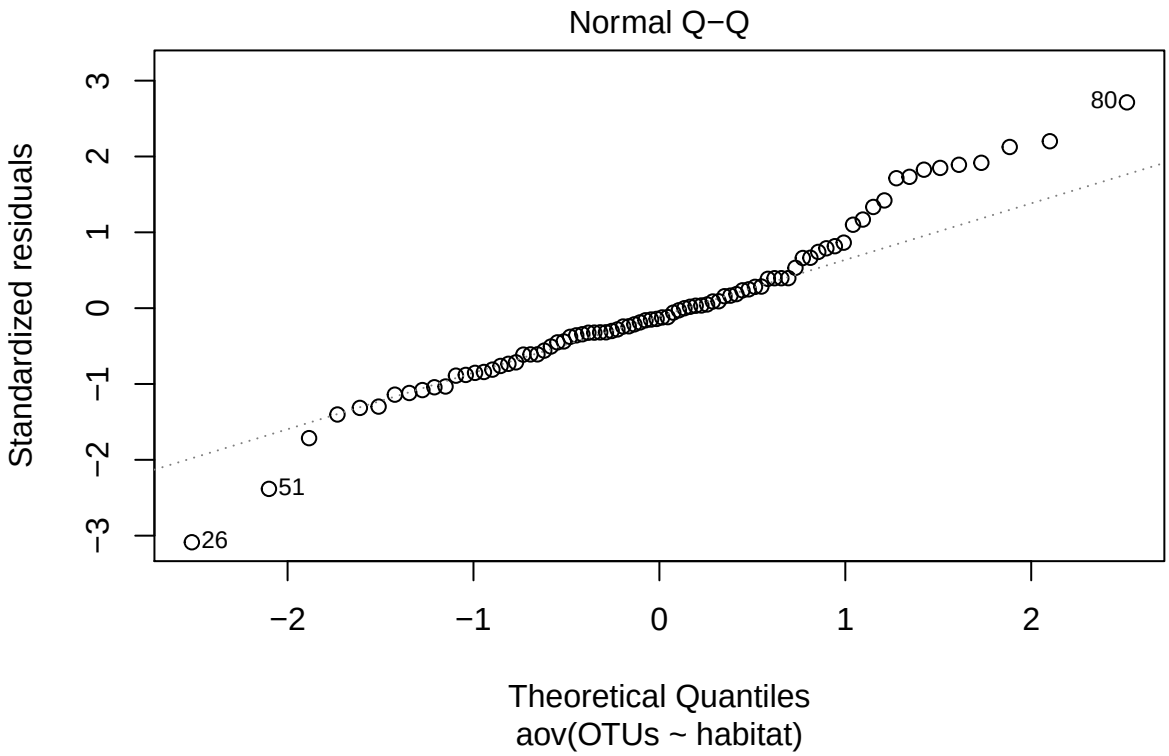
```
##             Df    Sum Sq   Mean Sq F value  Pr(>F)
## habitat      5 1.549e+09 309848387   4.402 0.00139 **
## Residuals   78 5.491e+09  70391392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(res.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = OTUs ~ habitat, data = reads_habitat)
##
## $habitat
##                                             diff         lwr        upr
## 5 m above bottom-0,05 m above bottom     -5973.171 -17193.9084  5247.5655
## 50 m above bottom-0,05 m above bottom   -11679.857 -23027.3861  -332.3282
## Chlorofil max layer-0,05 m above bottom -11048.488 -22706.9658   609.9896
## Mixed layer-0,05 m above bottom         -13744.571 -25403.0492 -2086.0937
## Statia Mixed-0,05 m above bottom         -4194.155 -14724.2160  6335.9065
## 50 m above bottom-5 m above bottom       -5706.686 -14816.1753  3402.8038
## Chlorofil max layer-5 m above bottom     -5075.317 -14569.3405  4418.7072
## Mixed layer-5 m above bottom             -7771.400 -17265.4239  1722.6239
## Statia Mixed-5 m above bottom             1779.017  -6289.3522  9847.3855
## Chlorofil max layer-50 m above bottom      631.369  -9012.1762 10274.9143
## Mixed layer-50 m above bottom            -2064.714 -11708.2596  7578.8310
## Statia Mixed-50 m above bottom            7485.702   -758.0863 15729.4910
## Mixed layer-Chlorofil max layer          -2696.083 -12703.6632  7311.4965
## Statia Mixed-Chlorofil max layer          6854.333  -1812.4851 15521.1517
## Statia Mixed-Mixed layer                  9550.417    883.5983 18217.2351
##                                             p adj
## 5 m above bottom-0,05 m above bottom    0.6299328
## 50 m above bottom-0,05 m above bottom   0.0399313
## Chlorofil max layer-0,05 m above bottom 0.0734731
## Mixed layer-0,05 m above bottom         0.0114872
## Statia Mixed-0,05 m above bottom        0.8525107
## 50 m above bottom-5 m above bottom      0.4526956
## Chlorofil max layer-5 m above bottom    0.6257014
## Mixed layer-5 m above bottom            0.1719390
## Statia Mixed-5 m above bottom           0.9871962
## Chlorofil max layer-50 m above bottom   0.9999632
## Mixed layer-50 m above bottom           0.9887965
## Statia Mixed-50 m above bottom          0.0968816
## Mixed layer-Chlorofil max layer         0.9689354
## Statia Mixed-Chlorofil max layer        0.2022706
## Statia Mixed-Mixed layer                0.0222346
```

```
check_assumption()
```






```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(res.aov)
## W = 0.9678, p-value = 0.03359
```

3. LCA

remove bacteria hits

The entire dataset was blasted to genbank in galaxy. The resulting lca file is imported here to remova any reads belonging to bacteria.

```
genbank <- read.delim("~/Downloads/Galaxy6-[filtOTUseqs40376.fasta_BLAST_original_taxonomy_lca].tabular

bact <- genbank %>% filter(X.kingdom == "Bacteria")

toremove <- bact$X.Query
length(toremove)
```

```
## [1] 22428
```

```
otu <- otu[!row.names(otu) %in% toremove,]
nrow(otu)
```

```
## [1] 17948
```

import and prepare lca data

Then, the lca files at bitscore 8 and 12 percent are imported and compared

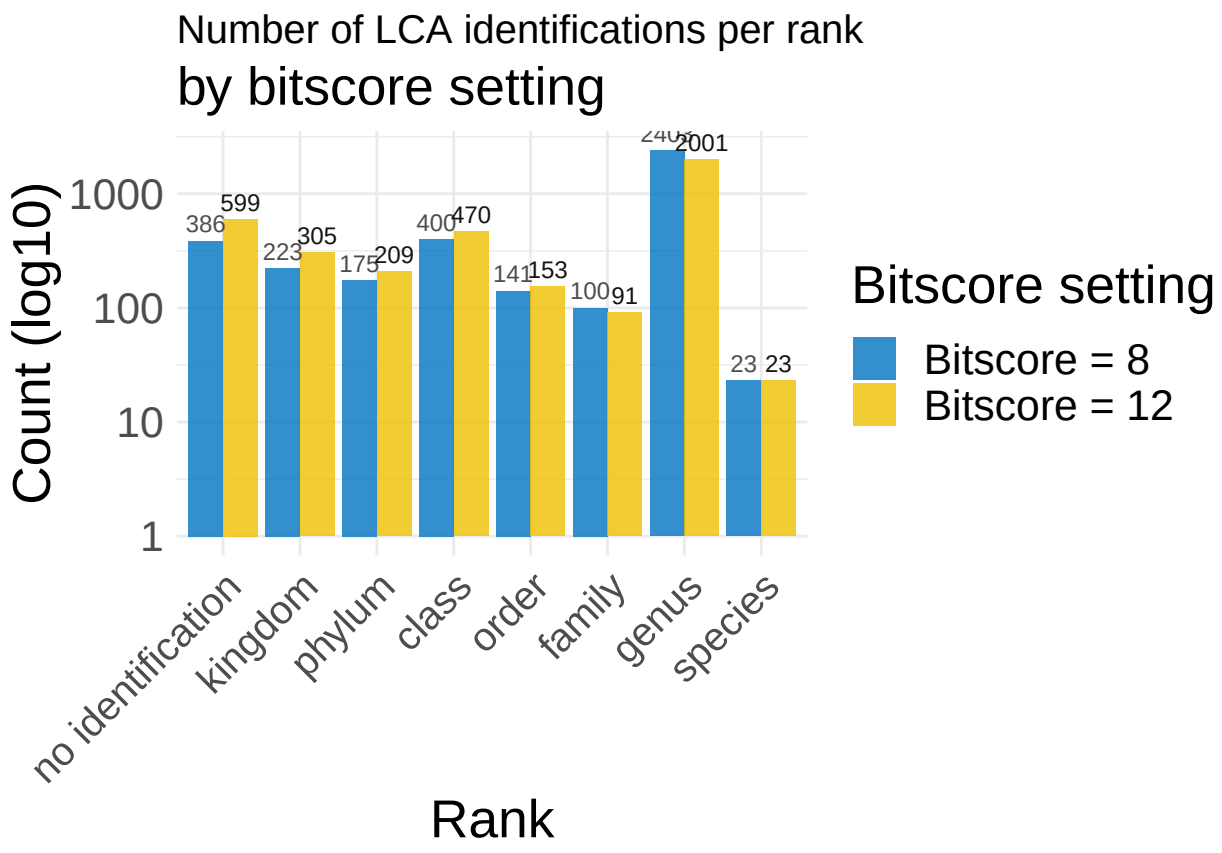
```
getLca <- function(){
  df <- read.delim("~/Documents/derep_illum/changedheader/taxadded/bit8range")
  df2 <- read.delim("~/Documents/derep_illum/changedheader/taxadded/bit12range")
  dfs <- list(df, df2)
  # remove X. from cols and name the dfs
  dfs <-
    lapply(dfs, function(x){setNames(x, sub("^X.", "", names(x)))}) %>%
    `names<-`(c("Bitscore = 8", "Bitscore = 12"))}

#execute the functions
lcas <- getLca()
lcas <- lapply(lcas, function(x) {x <- x[!x$Query %in% toremove,]})

# add information on how many reads were captured by the bitscore threshold
merged_dfs <-
  lapply(1:length(lcas), function(x) lcas[[x]] %>%
    data.frame) %>%
  # create extra column with what bitscore was used and bind the dataframes
  Map(cbind, ., Bitscore_setting = names(lcas)) %>%  # info of bitscore for bth dfs
  do.call(rbind, .) %>%  #combined them by row
  data.frame() %>%
  filter(!grepl("sp\\.", species)) # remove hits that contain sp. because theyre not informative.

# get factor levels in right order for nice looking plots
merged_dfs$lca.rank <- factor(merged_dfs$lca.rank, levels = c("no identification", colnames(merged_dfs)
```

plot number of taxa found per rank by bitscore setting

Bitscore 8 has a stronger bias towards genus level identifications because the it takes fewer reads into account for the lca deterimination so higher chance that theres only 1 read to do taxa determination with,

```
myPlot <- function(){
  merged_dfs %>%
    ggplot(aes(x= lca.rank, y = ..count.., group = Bitscore_setting)) +
    geom_bar(aes(fill=`Bitscore_setting`),
             position = "dodge", alpha = 0.8) +
    geom_text(stat = "count",
              aes(label = ..count.., colour = Bitscore_setting),
              position = position_dodge(0.9),
              vjust = -0.5, size = 3) +
    scale_fill_jco() +
    #scale_fill_brewer(type = "qual", palette = "Pastel2") +
```

```
    #scale_fill_manual(values = alpha(c("#00AFBB", "#FC4E07"), 0.8)) +
    theme_minimal() +
    scale_color_manual(values = c("gray30", "gray8"),
                       guide = F) +
    scale_y_log10() +
    labs(title = "Number of LCA identifications per rank",
         subtitle = "by bitscore setting",
         x = "Rank",
         y = "Count (log10)",
         fill = "Bitscore setting") +
    theme(plot.title = element_text(size = 15, vjust = 1),
          text = element_text(size = 20),
          axis.text.x = element_text(angle = 45, hjust =1, size = 15))}
p <- myPlot()
p
```




```
ggsave(filename = "Bitscore plot", p, device = "png", width = 10, height = 7.5)
```

The plot shows that bitscore 12 has less identifications but also less of a bias toward genus level, at it takes more reads into the LCA step.

Combine OTU and sampling location data

Compare the number of centroids supplied to the blast file to the number of blast hits found that had at least one blast hit of 70% identity and 70% coverage. The LCA file wiht bitscore 12 is chosen. get LCA:

numbers

```
merge_otu_lca  <- function(){
  otu <- rownames_to_column(otu, var = "Query")
  df_otu_lca <- merge(otu, lcas[[2]], by='Query', all.x = TRUE)
  total_otu <- nrow(otu)
  OTUs_hit <- length(unique(lcas[[2]]$Query))
  lca_hit <- length(lcas[[2]]$lca.rank[lcas[[2]]$lca.rank != "no identification"])
  cat(paste("\tnumber of rows in OTU file (number of OTUs found):\t", total_otu,
            "\n\n",
            "\tnumber of rows in LCA file (OTUs that had at least one blast hit):\t", OTUs_hit,
            "\n\nPercentage of dark taxa is: ", round((1-lca_hit/total_otu)*100,2), "%"))

  return(df_otu_lca)}

otu_lca <- merge_otu_lca()
```

```
##  number of rows in OTU file (number of OTUs found):   17948
##
##      number of rows in LCA file (OTUs that had at least one blast hit):   3848
##
## Percentage of dark taxa is:  81.87 %
```

Combine sampling location data with the OTU table

Functions

```
get_bin_tags <- function(df){
    mdf %>%
    filter(sample %in% colnames(df)) %>%
    mutate(habitat = as.character(habitat)) %>%
    mutate(habitat = replace_na(habitat, "Mixed")) %>%
    mutate(bin = paste(tag, habitat)) %>%
    t() %>%
    as.data.frame() %>%
    row_to_names(1) %>%
    rownames_to_column(var = "Query") %>%
    filter(Query == "bin")
}


get_tags <- function(df){
  mdf %>%
    filter(sample %in% colnames(df)) %>%
    t() %>%
    as.data.frame() %>%
    row_to_names(1) %>%
    rownames_to_column(var = "Query") %>%
    filter(Query == "tag")}

replace_colnames <- function(df){
```

```
  df <- df %>% rownames_to_column(var = "Query")
  with_bins <- rbind.fill(tags, df) %>%
  row_to_names(row_number = 1)
  colnames(with_bins)[1] <- "Query"
  return(with_bins)
}

# make a new df with one column calles Query (for merging), bind by col and remove rownames
#otu <- rownames_to_column(otu, var = "Query")
```

Execution:

```
tags <- get_bin_tags(otu)
for_network <- replace_colnames(otu)
```

```
## Warning in row_to_names(., row_number = 1): Row 1 does not provide unique names.
## Consider running clean_names() after row_to_names().
```

```
tags <- get_tags(otu)
for_shared <- replace_colnames(otu)
```

```
## Warning in row_to_names(., row_number = 1): Row 1 does not provide unique names.
## Consider running clean_names() after row_to_names().
```

Prep data for sumarising per region

```
mixedmdf <- mdf %>%
  mutate(habitat = as.character(habitat)) %>%
  mutate(habitat = replace_na(habitat, "Mixed")) %>%
  mutate(habitat = gsub(" layer", "", habitat)) %>%
  filter(!habitat == "100 m above bottom") %>%
  filter(!is.na(lat))

try <- otu %>%
  data.matrix() %>%
  t() %>%
  as.data.frame %>%
  rownames_to_column(var = "sample") %>%
  merge(mixedmdf, ., by = "sample", all.y = TRUE)

with_row <- column_to_rownames(try, var = "sample")
with_row[,6:ncol(with_row)][with_row[,6:ncol(with_row)] > 0 ] <- 1
with_row <- with_row %>% filter(!is.na(lat))

with_row <- with_row[!rowSums(with_row[,6:ncol(with_row)])==0,]
```

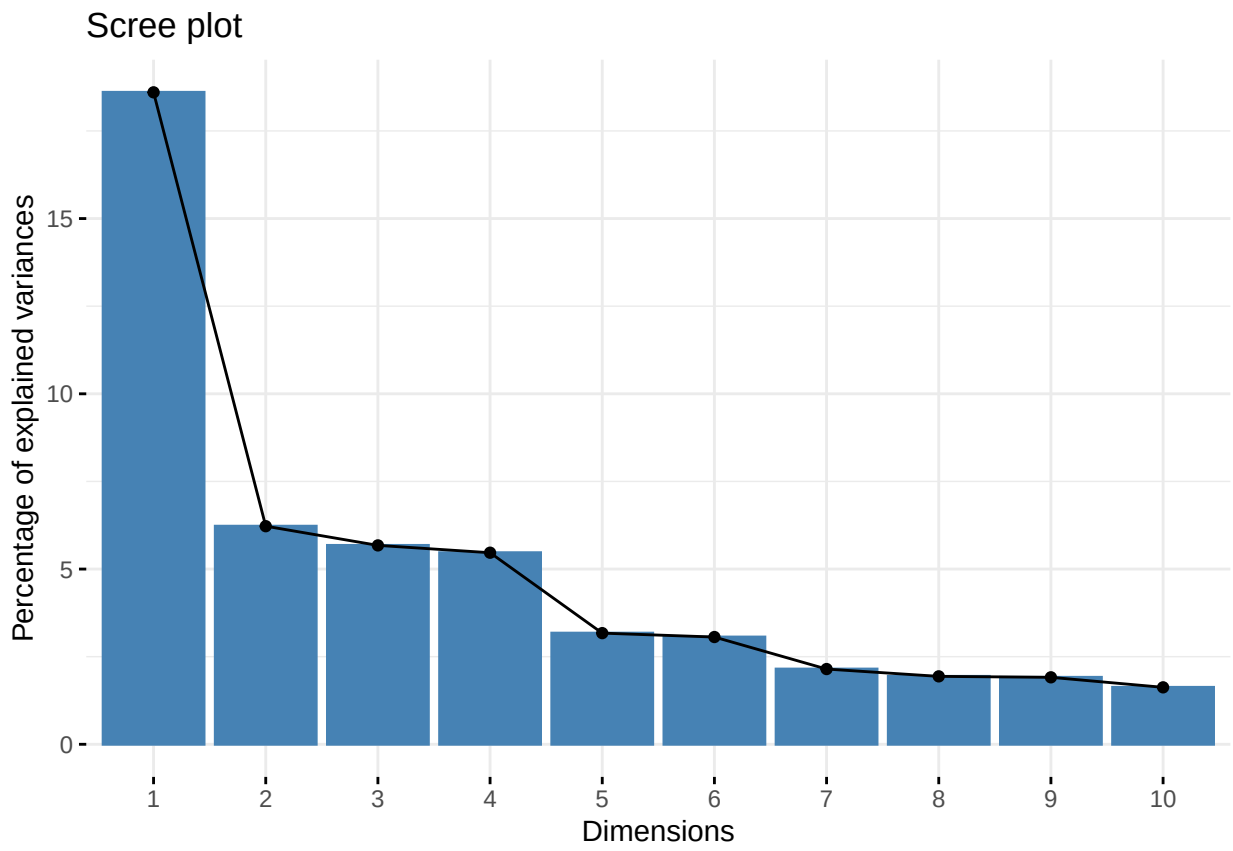
Region PCA

```
res.pca <- prcomp(with_row[,6:ncol(with_row)],
                  center = TRUE, scale. = FALSE)
get_eig(res.pca)
```

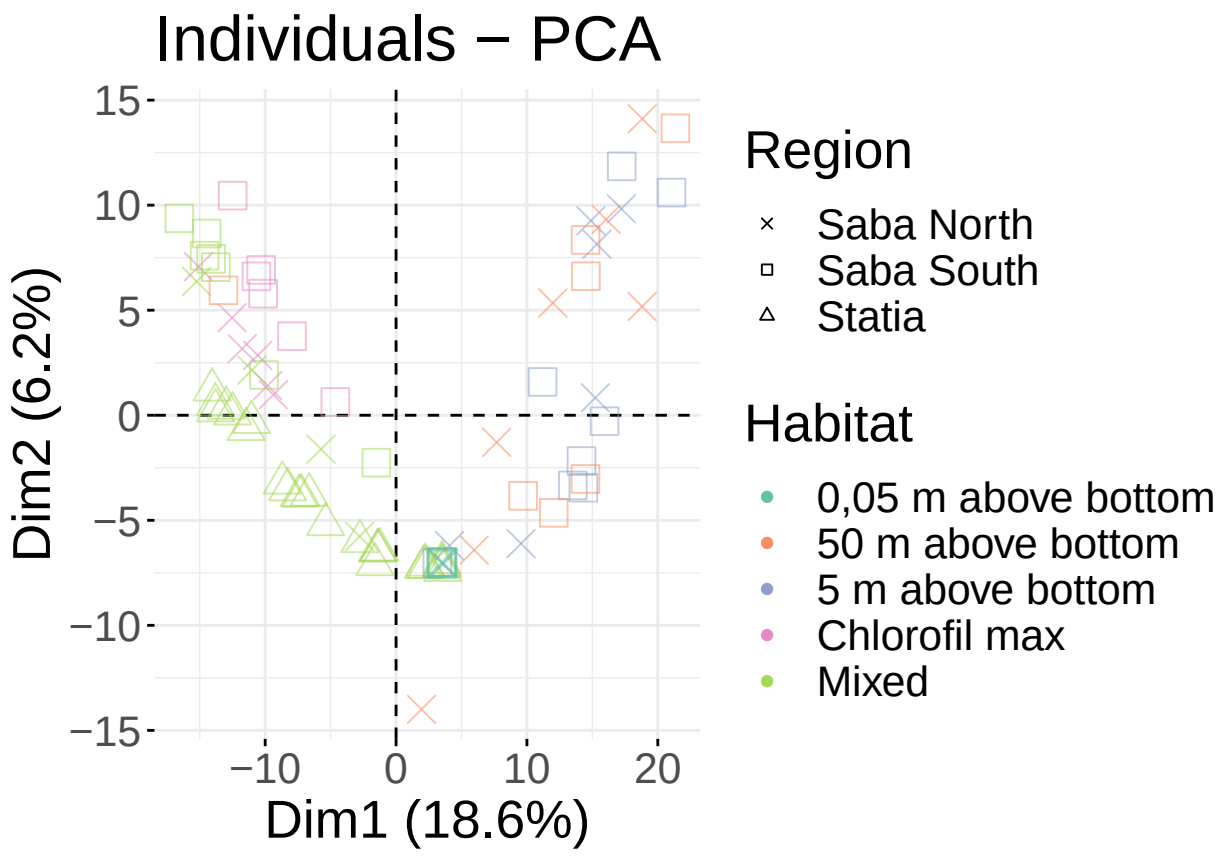
```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.235636e+02     1.860083e+01                       18.60083
## Dim.2  4.133834e+01     6.222927e+00                       24.82376
## Dim.3  3.769930e+01     5.675119e+00                       30.49888
## Dim.4  3.631158e+01     5.466217e+00                       35.96510
## Dim.5  2.106870e+01     3.171608e+00                       39.13670
## Dim.6  2.033141e+01     3.060618e+00                       42.19732
## Dim.7  1.426110e+01     2.146815e+00                       44.34414
## Dim.8  1.287627e+01     1.938348e+00                       46.28249
## Dim.9  1.269507e+01     1.911070e+00                       48.19356
## Dim.10 1.079537e+01     1.625097e+00                       49.81865
## Dim.11 1.029378e+01     1.549589e+00                       51.36824
## Dim.12 9.961104e+00     1.499509e+00                       52.86775
## Dim.13 9.367197e+00     1.410105e+00                       54.27786
## Dim.14 9.019676e+00     1.357790e+00                       55.63565
## Dim.15 8.318713e+00     1.252270e+00                       56.88792
## Dim.16 8.105968e+00     1.220244e+00                       58.10816
## Dim.17 8.056506e+00     1.212798e+00                       59.32096
## Dim.18 7.999300e+00     1.204186e+00                       60.52514
## Dim.19 7.833405e+00     1.179213e+00                       61.70436
## Dim.20 7.688611e+00     1.157416e+00                       62.86177
## Dim.21 7.593018e+00     1.143026e+00                       64.00480
## Dim.22 7.459436e+00     1.122917e+00                       65.12772
## Dim.23 7.397463e+00     1.113588e+00                       66.24130
## Dim.24 7.249856e+00     1.091368e+00                       67.33267
## Dim.25 7.128307e+00     1.073070e+00                       68.40574
## Dim.26 7.104494e+00     1.069485e+00                       69.47523
## Dim.27 6.996115e+00     1.053170e+00                       70.52840
## Dim.28 6.936785e+00     1.044239e+00                       71.57264
## Dim.29 6.902948e+00     1.039145e+00                       72.61178
## Dim.30 6.660621e+00     1.002666e+00                       73.61445
## Dim.31 6.501130e+00     9.786571e-01                       74.59310
## Dim.32 6.407848e+00     9.646147e-01                       75.55772
## Dim.33 6.306231e+00     9.493177e-01                       76.50704
## Dim.34 6.256552e+00     9.418392e-01                       77.44888
## Dim.35 6.210241e+00     9.348676e-01                       78.38374
## Dim.36 6.112079e+00     9.200906e-01                       79.30383
## Dim.37 5.929396e+00     8.925902e-01                       80.19642
## Dim.38 5.763768e+00     8.676572e-01                       81.06408
## Dim.39 5.631673e+00     8.477721e-01                       81.91185
## Dim.40 5.450972e+00     8.205700e-01                       82.73242
## Dim.41 5.389881e+00     8.113736e-01                       83.54380
## Dim.42 5.323411e+00     8.013674e-01                       84.34516
## Dim.43 5.267317e+00     7.929232e-01                       85.13809
## Dim.44 5.056815e+00     7.612351e-01                       85.89932
## Dim.45 4.881622e+00     7.348621e-01                       86.63418
```

```
## Dim.46 4.844067e+00     7.292086e-01                 87.36339
## Dim.47 4.763941e+00     7.171468e-01                 88.08054
## Dim.48 4.637051e+00     6.980453e-01                 88.77859
## Dim.49 4.560209e+00     6.864777e-01                 89.46506
## Dim.50 4.440668e+00     6.684825e-01                 90.13355
## Dim.51 4.281621e+00     6.445401e-01                 90.77809
## Dim.52 4.097152e+00     6.167707e-01                 91.39486
## Dim.53 3.960984e+00     5.962724e-01                 91.99113
## Dim.54 3.905791e+00     5.879640e-01                 92.57909
## Dim.55 3.783217e+00     5.695120e-01                 93.14860
## Dim.56 3.588145e+00     5.401466e-01                 93.68875
## Dim.57 3.485210e+00     5.246511e-01                 94.21340
## Dim.58 3.460860e+00     5.209855e-01                 94.73439
## Dim.59 3.422107e+00     5.151519e-01                 95.24954
## Dim.60 3.350430e+00     5.043619e-01                 95.75390
## Dim.61 3.237250e+00     4.873241e-01                 96.24123
## Dim.62 3.000120e+00     4.516274e-01                 96.69285
## Dim.63 2.251840e+00     3.389840e-01                 97.03184
## Dim.64 2.221142e+00     3.343628e-01                 97.36620
## Dim.65 2.051549e+00     3.088329e-01                 97.67503
## Dim.66 1.950424e+00     2.936099e-01                 97.96864
## Dim.67 1.563194e+00     2.353176e-01                 98.20396
## Dim.68 1.406802e+00     2.117750e-01                 98.41574
## Dim.69 1.358941e+00     2.045701e-01                 98.62031
## Dim.70 1.274400e+00     1.918436e-01                 98.81215
## Dim.71 1.101545e+00     1.658226e-01                 98.97797
## Dim.72 1.001609e+00     1.507787e-01                 99.12875
## Dim.73 8.549059e-01     1.286945e-01                 99.25745
## Dim.74 8.429984e-01     1.269020e-01                 99.38435
## Dim.75 7.747584e-01     1.166294e-01                 99.50098
## Dim.76 7.136975e-01     1.074375e-01                 99.60841
## Dim.77 6.076474e-01     9.147308e-02                 99.69989
## Dim.78 5.186211e-01     7.807138e-02                 99.77796
## Dim.79 4.197801e-01     6.319220e-02                 99.84115
## Dim.80 3.859904e-01     5.810562e-02                 99.89926
## Dim.81 3.474255e-01     5.230020e-02                 99.95156
## Dim.82 3.218061e-01     4.844355e-02                100.00000
## Dim.83 2.292289e-25     3.450731e-26                100.00000
```

```
fviz_eig(res.pca)
```




```
p <- fviz_pca_ind(res.pca,
             label = "none",
             alpha.ind = 0) + scale_color_brewer(palette = "Set2") + geom_point(aes(color = with_row$hal
p
```

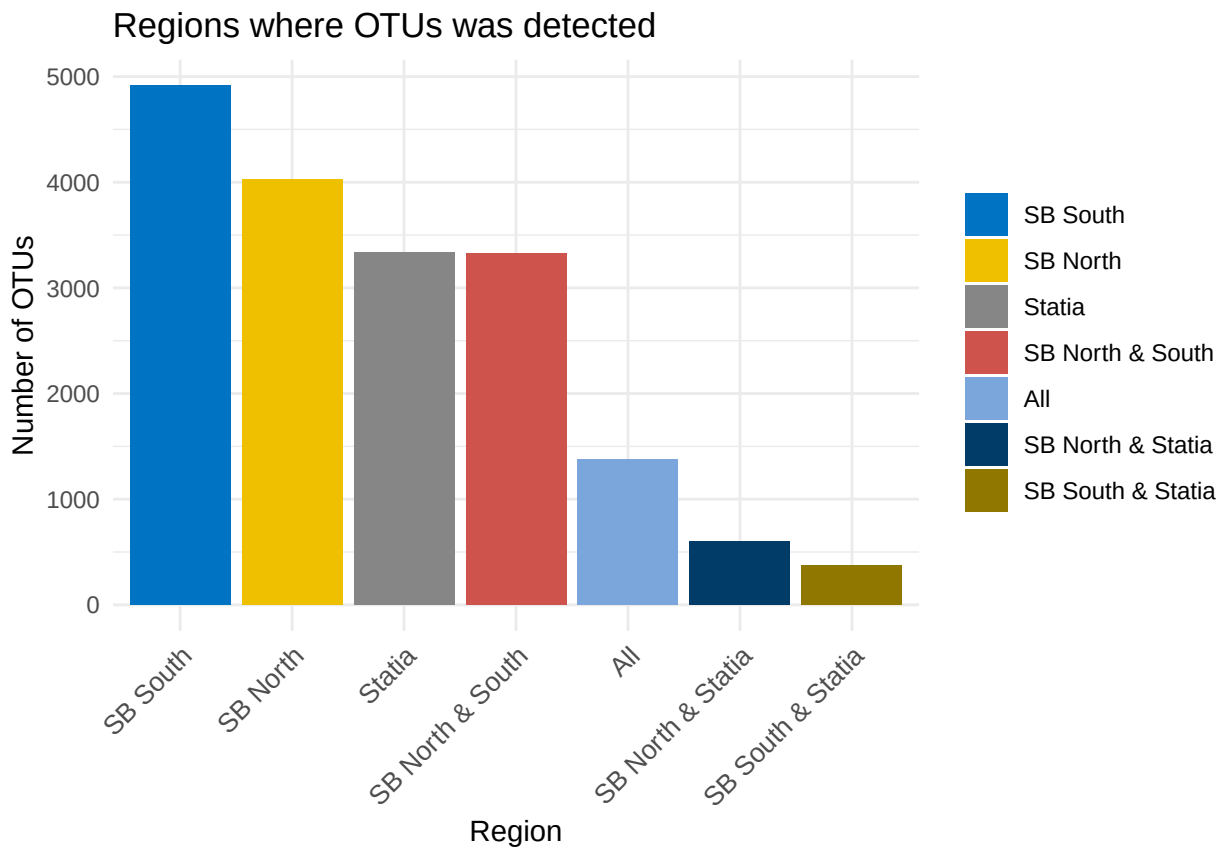



```
ggsave(p, filename = "pcatest", device = "png", height = 7, width = 12)
```

region bar plot

```
# sum the values per row
summed <-
  data.matrix(for_shared[,-1]) %>% t(.) %>%
  rowsum(., group = sub("\\.\\d+$", "", rownames(.))) %>%
  t() %>%
  data.frame %>% `rownames<-`(for_shared[,1]) %>%
  rownames_to_column(var = "Query")

plotdf <- melt(summed, id.vars = "Query", value.name = "total_present")

# make dummy: 0=not found & 1 = found
plotdf$dummy <- ifelse(plotdf$total_present > 0, yes=1, no=0)

plotdfcomb <- plotdf %>%
  group_by(variable) %>%
  mutate(presence = if_else(dummy == 1,
                            ifelse(variable ==  'Saba.North', 'SB North',
                                   ifelse(variable == 'Saba.South', 'SB South', 'Statia')),
                            NULL)) %>%
  na.omit() %>%
```

```
  group_by(Query) %>%
  summarise(presence = paste(presence, collapse = ' & '), .groups = 'drop') %>%
  mutate(shortpresence = ifelse(presence == 'SB North & SB South & Statia', 'All',
                                ifelse(presence == 'SB North & SB South', 'SB North & South',
                                       presence)))

pres <-
  data.frame(table(plotdfcomb$shortpresence)) %>%
  setNames(c('Region', 'Freq')) %>%
  arrange(desc(Freq))

pres$Region<- factor(pres$Region, levels = reorder(pres$Region, -pres$Freq))

pres %>%
  ggplot() +
  geom_bar(aes(x = Region,
               y = Freq,
               fill = Region),
           stat = 'identity') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = '10'),
        legend.title = element_blank()) +
  labs(title = 'Regions where OTUs was detected',
       x = 'Region',
       y = 'Number of OTUs') +
  scale_fill_jco()
```



Region plot

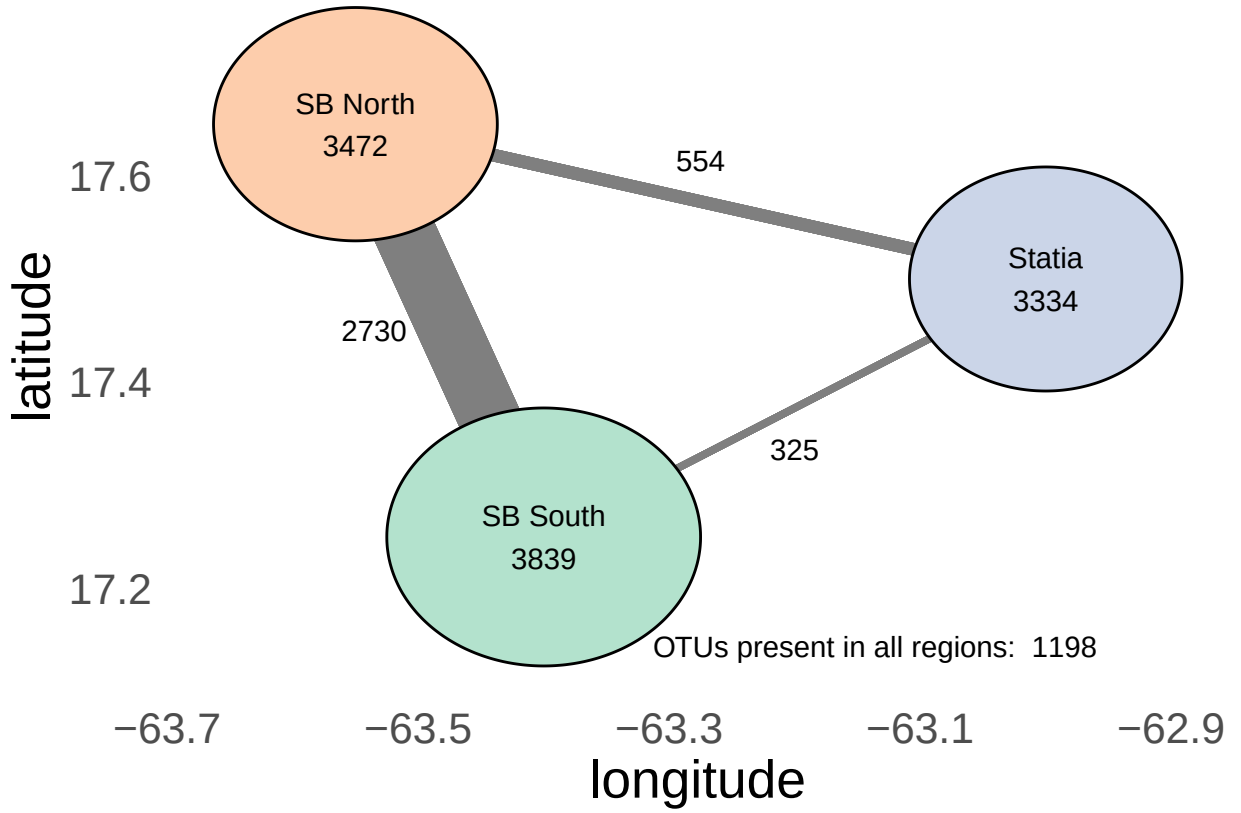
```
lat = c(17.25, 17.65, 17.5, NA, NA, NA, NA)
long = c(-63.4, -63.55, -63.0, NA, NA, NA, NA)
lat_text <- c(NA, NA, NA, mean(c(17.25, 17.65)), mean(c(17.25, 17.65, 17.5)), mean(c(17.65, 17.5))+0.04
long_text <- c(NA, NA, NA, mean(c(-63.4, -63.55))-0.06, mean(c(-63.4, -63.55, -63.0)), mean(c(-63.55, -
coordinates <- cbind(lat, long, lat_text, long_text)
test <- cbind(pres, coordinates)

northlength <- length(grep("Saba North", colnames(for_shared)))
southlength <- length(grep("Saba South", colnames(for_shared)))
statialength <- length(grep("Statia", colnames(for_shared)))

test$Freq <- round((test$Freq/c(southlength, northlength, statialength,
                          mean(c(southlength, northlength)),
                          mean(c(southlength, northlength, statialength)),
                          mean(c(northlength, statialength)),
                          mean(c(southlength, statialength)))) * 25, 0)

allregions <- paste("OTUs present in all regions: ", test[5,2])
test <- test %>% filter(!Region == "All")

p <- ggplot(data=test, aes(x0=long, y0=lat, r=Freq/(8*max(Freq)), fill = Region)) +
```

```
  geom_segment(aes(x=long[1],
                   y=lat[1],
                   xend=long[2],
                   yend=lat[2]), size = test$Freq[4]/250, color = "gray50") +
  geom_segment(aes(x=long[2],
                   y=lat[2],
                   xend=long[3],
                   yend=lat[3]), size = test$Freq[5]/250, color = "gray50") +
  geom_segment(aes(x=long[3],
                   y=lat[3],
                   xend=long[1],
                   yend=lat[1]), size = test$Freq[6]/250, color = "gray50") +
  geom_circle() +
  geom_text(data = test, aes(x=long, y=lat, label = paste(Region, "\n", as.character(Freq), sep = ""))) +
  geom_text(data = test, aes(x =long_text, y=lat_text, label = Freq)) +
  theme_minimal() +
  labs(
    #title = "OTUs that are shared between regions",
    #subtitle = "Circle correspond to number of OTUs found per region\nthickness of line corresponds to
       x = "longitude",
       y = "latitude") +
  theme(text = element_text(size = 20), panel.grid = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "none") +
  scale_fill_brewer(type = "qual", palette = "Pastel2") +
  annotate(geom = "text", label = allregions, x = Inf, y = -Inf, hjust = 1.3, vjust = -2)

p
```

```
## Warning: Removed 3 rows containing non-finite values (stat_circle).
```

```
## Warning: Removed 3 rows containing missing values (geom_text).
```

```
## Warning: Removed 3 rows containing missing values (geom_text).
```




```
ggsave("regions", p, device = "png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 3 rows containing non-finite values (stat_circle).
```

```
## Warning: Removed 3 rows containing missing values (geom_text).
```

```
## Warning: Removed 3 rows containing missing values (geom_text).
```

```
chisq.test(c(2730, 554, 325), p=c(1/3,1/3,1/3))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(2730, 554, 325)
## X-squared = 2929.2, df = 2, p-value < 2.2e-16
```

prepare data for summarising per habitat

```
prep <- function(df){
  df %>%
```

```
  `rownames<-`(NULL) %>%
  column_to_rownames(var = "Query") %>%
  data.matrix() %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column(var = "habitat") %>%
  filter(!habitat == "Saba.North.100.m.above.bottom") %>%
  mutate(habitat = sub("\\.\\d+$", "", habitat)) %>%
  filter(!habitat == "Saba.North.100.m.above.bottom")
}

aggr <- function(df){
  aggregate(df[,-1], list(habitat = df$habitat), mean) %>%
    mutate(habitat = habitat %>%
             sub("Saba.South", "SS", .) %>%
             sub("Saba.North", "SN", .) %>%
             sub(".above.bottom", "ab", .) %>%
             sub(".layer", "", .) %>%
             gsub("\\.", " ", .) %>%
             sub("0 05", "0.05", ., fixed = TRUE)) %>%
    column_to_rownames(var = "habitat") %>%
    t() %>%
    as.data.frame %>%
    rownames_to_column(var = "habitat")
}
```

Execution:

```
try <- prep(for_network)

try[,-1][try[,-1]>0] <- 1

agr <- try %>% aggr()
```

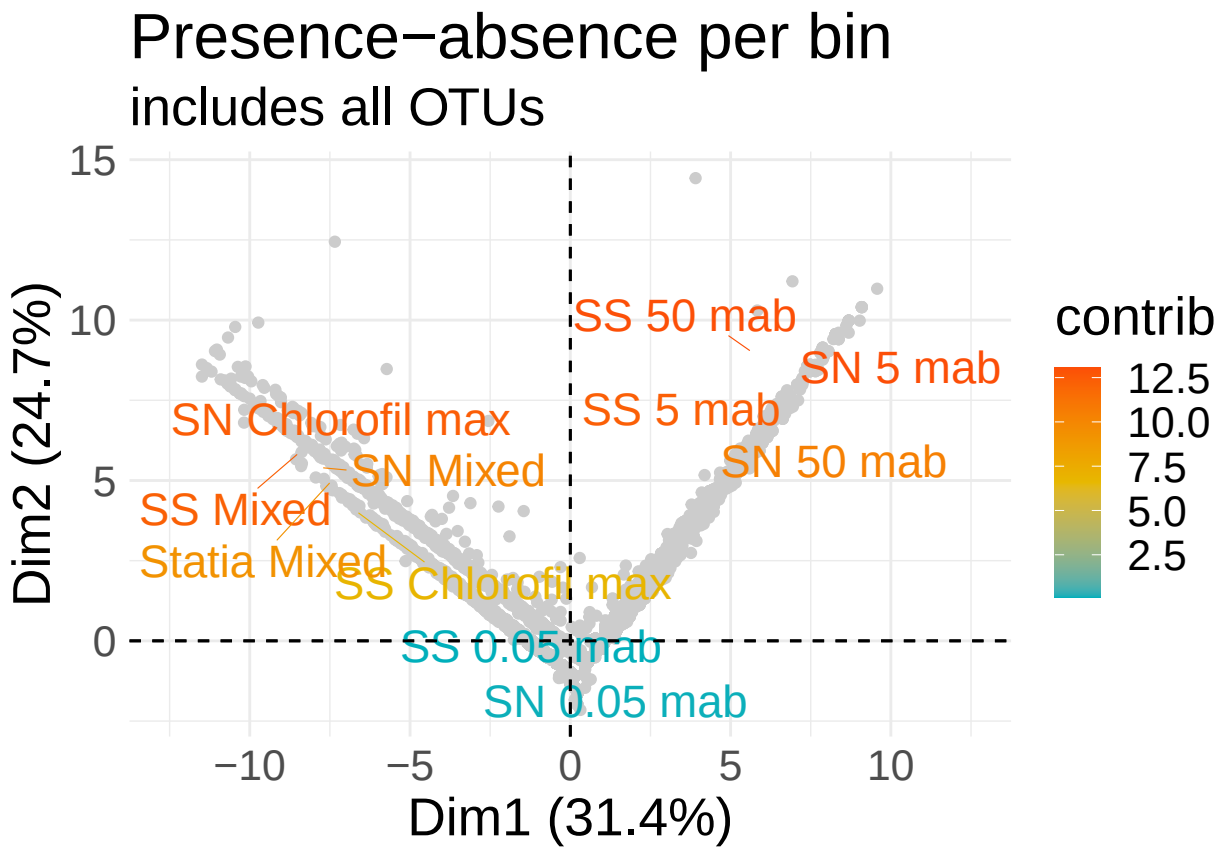
PCA analysis

```
myPca <- function(df){
  res.pca <- prcomp(column_to_rownames(df, var = "habitat"),
                    center = TRUE, scale. = TRUE)
  print(get_eig(res.pca))
  fviz_eig(res.pca)
  fviz_pca_biplot(res.pca,
                  label = "var",
                  col.var = "contrib",
                  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                  col.ind = "gray80",
                  repel = TRUE,
                  geom.ind = "point",
                  geom.var = "text",
                  labelsize = 6) + theme_minimal() + theme(text = element_text(size = 20))
```

```
}

p_agr <- myPca(agr) +
  labs(title = "Presence-absence per bin",
       subtitle = "includes all OTUs") + xlim(-12.5, 12.5)
```

```
##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1   3.4579667        31.436061                     31.43606
## Dim.2   2.7160396        24.691269                     56.12733
## Dim.3   0.9972303         9.065730                     65.19306
## Dim.4   0.9955104         9.050094                     74.24315
## Dim.5   0.7590921         6.900837                     81.14399
## Dim.6   0.5116063         4.650966                     85.79496
## Dim.7   0.4595809         4.178008                     89.97297
## Dim.8   0.3532729         3.211572                     93.18454
## Dim.9   0.2794858         2.540780                     95.72532
## Dim.10  0.2540913         2.309921                     98.03524
## Dim.11  0.2161238         1.964762                    100.00000
```

```
p_agr
```




```
ggsave("finalpcapresence", p_agr, device = "png")
```

```
## Saving 6.5 x 4.5 in image
```