Data analysis Manuscript Gutekunst et al. 2022

Cordula Gutekunst and Gerald Jurasinski

2022-07-29

Contents

1 Introduction

This document can be used to create all plots for the journal article Gutekunst et al. (unpublished): "Effects of brackish water inflow on methane cycling microbial communities in a freshwater rewetted coastal fen". The document consists of three parts and is separated according to: 1. Preparation and plotting of bubble plots, 2. Running an plotting NMDS ordinations and, 3. Creation of depth profiles from environmental and microbial data.

All required data and packages are loaded at the beginning of each subsection, so that they can be visited individually. The function `prep.dna()` however, is only defined once in the beginning and also the working directory, which is then used throughout. All to be loaded data sets and a folder named *"figure"* have to reside in the working directory together with the Rmd-file.

Sampling campaigns providing data used in this study took place in 2014 (freshwater rewetted baseline), 2018 (during a drought) and in spring as well as in autumn 2019 after a brackish water inflow in January 2019. Hereafter, these campaigns are in the above order addressed as the following: Baseline2014, Drought2018, Post-inflow Spring2019 and Post-inflow Autumn2019.

2 Bubble plots

2.1 Preparation

First, required packages have to be loaded after being installing via `install.packages("package.name")`.

```
library(dplyr)
library(vegan)
library(scales)
library(simba)
library(vegan)
```

2.1.1 Load pore water (environmental) data

Next, we need to load the data that contain pore water (environmental from now on) variables. These will be needed later on to be merged with the microbial data. For the bubble plots, we only need them to provide information about the sampling details such as the sampling campaign ("Date"), the location ("site") and the sampled depth zone ("depth", "depth_id" and "anchor"). Further, some sampling campaigns also distinguish between the hydrological and vegetational state ("WU" for wet un-vegetated and "DU" for dry un-vegetated).

```
setwd(file.path("Y:/C Gutekunst/Baltic Transcoast/G3 Projekt",
"Daten/Data analysis Hütelmoor Gutekunst et al"))
nms_new <- read.table("qpcr_nms_mudenv.txt", header = TRUE)
# Chose target data and discard rows, which contain NA's
nms_new <- nms_new[!is.na(nms_new$ID),]
# read in DNA names
nms_new_DNA<- nms_new[(nms_new$DNA_type=="DNA"),]
```

2.1.2 DNA-based abundances - preparation

Before we load the microbial ASV files (one for bacteria and one for archaea), we define a function that will create complex objects with three list entries that each contain a `data.frame` (data.frames: *otu*, *nms* and *tax*). This is useful because ASV files are not in a format, which is practical to work with, so we split the information to be able to address e.g. the genus level within the taxonomy individually. We also want to split the information that is contained in the ID into different rows, so we are able to address and modify them more easily. Most of the ID information will, however, be used to merge with the data set of the environmental variables and will be discarded later on.

```
######## DNA/cDNA data handling function ########
## Define preparation function for DNA/cDNA data
prep.dna <- function(file, group, what){
  otu <- file[,-c(ncol(file)-1, ncol(file))]
  nms.tmp <- strsplit(names(otu), "_")
  nms <- data.frame(
    ID = names(otu),
    when = sapply(nms.tmp, function(x) x[2]),
    month = sapply(nms.tmp, function(x) x[3]),
    what = what,
    site = sapply(nms.tmp, function(x) x[4]),
    depth = sapply(nms.tmp, function(x) x[5])
```

```
  )
  nms$depth <- as.character(nms$depth)
  nms$depth[nms$depth == "surface"] <- "00-00"
  nms$depth <- factor(nms$depth)
  nms$IDnew <- paste(nms$site, nms$depth, sep="_")
  otu <- otu[,order(nms$ID)]
  nms <- nms[order(nms$ID),]
  tax.tmp <- strsplit(as.character(file$taxonomy), split=";")
  tax <- data.frame(
    Do = sapply(tax.tmp, function(x) substr(x[1], 4, 40)),
    Ph = sapply(tax.tmp, function(x) substr(x[2], 4, 40)),
    Cl = sapply(tax.tmp, function(x) substr(x[3], 4, 40)),
    Or = sapply(tax.tmp, function(x) substr(x[4], 4, 40)),
    Fa = sapply(tax.tmp, function(x) substr(x[5], 4, 40)),
    Ge = sapply(tax.tmp, function(x) substr(x[6], 4, 40)),
    Sp = sapply(tax.tmp, function(x) substr(x[7], 4, 40)),
    group = group,
    stringsAsFactors = F
  )
  row.names(tax) <- names(tax.tmp)
  res <- list(otu = otu, nms = nms, tax = tax)
}
```

Now we read in the microbial ASV files, first bacteria. We also check bacteria and archaea data for referring to exactly the same samples. The created vector `sel.arc` will later be used to constrain the archaea data set the same samples, which will be done after archaea ASV is loaded. But for now, we stick with the bacteria data. Sometimes, it is necessary to remove archaea data from the bacteria data set, when universal primers are used, like in our case. However, here, archaea data were remove manually before.

```
## Bacteria
asv.bac <- read.table("DNA_bac.tsv", header=T, sep="\t",
                  stringsAsFactors=F, row.names=1)

####### Take care, that the same samples are addressed
asv.arc <- read.table("DNA_arc.tsv",
                      header=T, sep="\t", stringsAsFactors=F, row.names=1)
sel.arc <- match(names(asv.bac), names(asv.arc))

######## Optional: remove archaea from data set when universal primers are being used
#to remove archaea from bacteria data set
#pat0 = 'Archaea'
#get subset
#otu = otu[!grepl(pat0,otu$taxonomy,ignore.case=T),]
```

Next step is to take strings for targeting bacterial groups, such as methanotrophs or sulfate reducing bacteria (SRB). The used strings have to occur somewhere in the taxonomy, but it does not matter, whether they occur in the domain or the genus or any other level. Further, we used `grepl` to remove unwanted taxa that we defined manually before. Here, it is possible to either give strings or full name of species, genus, family, order or class groups. This way, taxa will be excluded, which e.g. have the defined string *"methylo"* in their name, but are known to not be able to perform methane oxidation. In addition, a column named "group" is added, which marks all taxa of one group, like, e.g., *"methanotroph"*. Within that group, we normalized the data using Wisconsin double standardization to account for the different sizes of the bacterial and the archaeal data sets and strongly varying count numbers across taxonomic units.

```
#FOR METHANOTROPHS
# for methanotrophs in reprocessed data set
pat1 = 'methylo'
# get subset
asv.mtrophs = asv.bac[grepl(pat1,asv.bac$taxonomy,ignore.case=T),]
# further clean patterns, these will be removed
toremove = 'Methylolige|Methylophil|Methylobacterium|Methylovirg|Methylopila|
  Sh765B-TzT-35|Rokubacteriales|Burkholderiales|MIZ17|Methylorosula'
# remove unwanted and get wanted subset
asv.mtrophs = asv.mtrophs[!grepl(toremove,asv.mtrophs$taxonomy,ignore.case=T),]
# add column that specifies the data set
asv.mtrophs$group <- "methanotroph"
asv.mtrophs$group <- as.character(asv.mtrophs$group)

# normalize counts with Wisconsin double standardization
tmp <- asv.mtrophs[,!names(asv.mtrophs) %in% c("taxonomy", "group")]
asv.mtrophs[,!names(asv.mtrophs) %in% c("taxonomy", "group")] <- t(wisconsin(t(tmp)))

#FOR sulfate reducing bacteria (SRB)
pat2 = 'desulfo'
# get subset
asv.srb = asv.bac[grepl(pat2,asv.bac$taxonomy,ignore.case=T),]
# further clean patterns, these will be removed
toremove = 'Syntrophus|Smithella|Syntrophorhabdus|Syntrophales|Desulfuromonadia|
  bacteriap25|Carboxydothermales|Ammonificaceae|Desulfallas'
# remove unwanted and get wanted subset
asv.srb = asv.srb[!grepl(toremove,asv.srb$taxonomy,ignore.case=T),]
# add column that specifies the data set
asv.srb$group <- "SRB"
asv.srb$group <- as.character(asv.srb$group)

# normalize counts with Wisconsin double standardization
tmp <- asv.srb[,!names(asv.srb) %in% c("taxonomy", "group")]
asv.srb[,!names(asv.srb) %in% c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))
```

Next, archaea are chosen that can either perform methanogenesis or are known ANME (anaerobic methanotrophs). The process is similar to the one above for bacteria, but we used now the ASV file for the archaea, which was loaded above.

```
#FOR METHANOGENS DNA
asv.arc <- asv.arc[,sel.arc] # This is the step, where the bacteria and archaea data sets
                     # are constrained to the same samples

##
pat3='methano'
# get subset
asv.mgens = asv.arc[grepl(pat3,asv.arc$taxonomy,ignore.case=T),]
# further clean patterns, these will be removed
toremove = 'Methanopere|ANME'
# remove unwanted and get wanted subset
asv.mgens.oA = asv.mgens[!grepl(toremove,asv.mgens$taxonomy,ignore.case=T),]
# add column that specifies the data set
asv.mgens.oA$group <- "methanogen"
```

```
asv.mgens.oA$group <- as.character(asv.mgens.oA$group)

# normalize counts with Wisconsin double standardization
tmp <- asv.mgens.oA[,!names(asv.mgens.oA) %in% c("taxonomy", "group")]
asv.mgens.oA[,!names(asv.mgens.oA) %in%
               c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))


#FOR ANME DNA
# get subset
asv.mgens.ANME = asv.arc[grepl("ANME|Methanopere", asv.arc$taxonomy, ignore.case=T),]
# add column that specifies the data set
asv.mgens.ANME$group <- "ANME"
asv.mgens.ANME$group <- as.character(asv.mgens.ANME$group)

# normalize counts with Wisconsin double standardization
tmp <- asv.mgens.ANME[,!names(asv.mgens.ANME) %in% c("taxonomy", "group")]
asv.mgens.ANME[,!names(asv.mgens.ANME) %in%
                 c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))
```

The function `prep.dna` is now applied to the four microbial taxonomical groups (*methanogens*, *methanotrophs*, *SRB* and *ANME*). In parallel, the names (nms) data.frame, which contains the information about sampling details (date, location, depth, replicate, etc.) is merged with the environmental variables. Here, only the ID and the depth from the former nms data set are kept, while sampling details will be taken from the environmental data. However, environmental variables, including absolute microbial abundances, derived from qPCR, are also excluded and will only be used for depth profile plots.

```
### Transform for better handling of taxonomies and such
## Methanotrophs
dna.mtrophs <- prep.dna(asv.mtrophs, "methanotroph", "DNA")
# join with pore water variables
dna.mtrophs$nms <- merge(dna.mtrophs$nms[,c(1,6)], nms_new_DNA[,c(2:8,17,22)],
                         by="ID")
# don't use data that is not comparable
dna.mtrophs$otu <- dna.mtrophs$otu[,!is.na(dna.mtrophs$nms$depth_id)]
dna.mtrophs$nms <- dna.mtrophs$nms[!is.na(dna.mtrophs$nms$depth_id),]

##SRB
dna.desulfo <- prep.dna(asv.srb, "SRB", "DNA")
# join with pore water variables
dna.desulfo$nms <- merge(dna.desulfo$nms[,c(1,6)], nms_new_DNA[,c(2:8,17,22)],
                         by="ID")
# don't use data that is not comparable
dna.desulfo$otu <- dna.desulfo$otu[,!is.na(dna.desulfo$nms$depth_id)]
dna.desulfo$nms <- dna.desulfo$nms[!is.na(dna.desulfo$nms$depth_id),]

## Methanogens without ANME
dna.mgens.oA <- prep.dna(asv.mgens.oA, "methanogen", "DNA")
# join with pore water variables
dna.mgens.oA$nms <- merge(dna.mgens.oA$nms[,c(1,6)], nms_new_DNA[,c(2:8,17,22)],
                          by="ID")
# don't use data that is not comparable
dna.mgens.oA$otu <- dna.mgens.oA$otu[,!is.na(dna.mgens.oA$nms$depth_id)]
```

```
dna.mgens.oA$nms <- dna.mgens.oA$nms[!is.na(dna.mgens.oA$nms$depth_id),]

## ANME
dna.mgens.ANME <- prep.dna(asv.mgens.ANME, "ANME", "DNA")
# join with pore water variables
dna.mgens.ANME$nms <- merge(dna.mgens.ANME$nms[,c(1,6)], nms_new_DNA[,c(2:8,17,22)],
                            by="ID")
# don't use data that is not comparable
dna.mgens.ANME$otu <- dna.mgens.ANME$otu[,!is.na(dna.mgens.ANME$nms$depth_id)]
dna.mgens.ANME$nms <- dna.mgens.ANME$nms[!is.na(dna.mgens.ANME$nms$depth_id),]
```

After creating four single data sets that only contain the results of the members of the specific groups of organisms (*methanogens*, *methanotrophs*, *ANME*, *SRB*), we merge them all into one object for further handling. Since the object is a complex object like the ones produced with `prep.dna()` we have to combine things separately for the different list entries. This is, however not necessary for the `nms` object since it is the same across all groups.

```
### Put them all together
dna <- list(otu = rbind(dna.mtrophs$otu, dna.mgens.oA$otu, dna.mgens.ANME$otu,
                        dna.desulfo$otu),
            nms = data.frame(dna.mtrophs$nms[,-2]),
            tax = rbind(dna.mtrophs$tax, dna.mgens.oA$tax, dna.mgens.ANME$tax,
                        dna.desulfo$tax))
names(dna$nms)[c(4,5,9)] <- c("depth", "dzone","what")
dna$nms <- data.frame(dna$nms[,1:2], dna$nms[,3:9])
```

Now we choose the taxonomic level, which will be displayed in the plot later on. The preparation by `prep.dna()` now allows for easy addressing of the taxonomic level. Because we did not chose the same level for all four groups, we had to do this individually for each group. We chose the order level for *methanogens* and *methanotrophs*. For *SRB* we selected the class level, because of too many entries at the order level. *ANME* does not have many entries, so we decided to show them on the genus level.

Sometimes it is necessary to jump up one level, in case there are NA entries in the chosen taxonomic level. In the code below and, hence, for our analysis, this was only necessary for the SRB.

```
### Summarize over the chosen taxonomical level
tax.lev1 <- dna.mtrophs$tax$Or

tax.lev2 <- dna.mgens.oA$tax$Or

tax.lev3 <- dna.mgens.ANME$tax$Ge

tax.lev4 <- dna.desulfo$tax$Cl
#in case of NA's, display next taxonomical level (phylum level)
tax.lev4[(is.na(tax.lev4)) | (grepl("uncult", tax.lev4, ignore.case=T))] <-
      dna.desulfo$tax$Ph[(is.na(tax.lev4)) |
      (grepl("uncult", tax.lev4, ignore.case=T))]
```

After we defined the level of each group, we create a data frame and merge them together at the end.

```
# sum up values per taxonomic level and sample
dna.pl1 <- data.frame(tax = tax.lev1, dna.mtrophs$otu) %>%
  group_by(tax) %>%
```

```
  summarise_all(sum)

dna.pl2 <- data.frame(tax = tax.lev2, dna.mgens.oA$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

dna.pl3 <- data.frame(tax = tax.lev3, dna.mgens.ANME$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

dna.pl4 <- data.frame(tax = tax.lev4, dna.desulfo$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

## Put them all together
dna.pl<- rbind(dna.pl1, dna.pl2, dna.pl3, dna.pl4)
```

The row "group" was ignored until here. Now, we create a data frame, containing the information about which taxa belong to which group. Later on, we bind them together again.

```
# get the groupings right
dna.groups1 <- data.frame(tax = tax.lev1, dna.mtrophs$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

dna.groups2 <- data.frame(tax = tax.lev2, dna.mgens.oA$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

dna.groups3 <- data.frame(tax = tax.lev3, dna.mgens.ANME$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

dna.groups4 <- data.frame(tax = tax.lev4, dna.desulfo$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

## Put them all together
dna.groups<- rbind(dna.groups1, dna.groups2, dna.groups3, dna.groups4)
```

2.1.3 cDNA-based abundandances - preparation

The above code was dealing with the preparation of the DNA-based abundance analysis. Further, we will repeat exactly the same process with the cDNA-based data. Note, that we need to re-define the environmental data, because we now want to address sampling campaigns only, when cDNA was extracted.

```
## only works when the working directory was set correctly above
# read in cDNA names
nms_new_cdna<- nms_new[(nms_new$DNA_type=="cDNA"),]
```

Now, like before for DNA read in the microbial ASV files, first bacteria. We also check bacteria and archaea data for referring to exactly the same samples. The created vector `sel.arc` will later be used to constrain

the archaea data set the same samples, which will be done after archaea ASV is loaded. But for now, we stick with the bacteria data.

```
### FOR METHANOTROPHS cDNA
asv.bac <- read.table("cDNA_bac.tsv",
                  header=T, sep="\t",stringsAsFactors=F, row.names=1)

####### Take care, that the same samples are addressed
asv.arc <- read.table("cDNA_arc.tsv",
                      header=T, sep="\t", stringsAsFactors=F, row.names=1)
sel.bac <- match(names(asv.arc), names(asv.bac))

#remove data point that does not exist for archaea
asv.bac <- asv.bac[,sel.bac]
```

Next step is to choose strings for target bacterial groups, such as methanotrophs and sulfate reducing bacteria. The chosen string has to occur somewhere in the taxonomy, but it does not matter, whether it occurs in the domain or the genus level. Further, we use `grepl` to remove unwanted taxa that we define manually before. Here, it is possible to either give strings or full name of species, genus, family, order or class groups. This way, taxa will be excluded, which e.g., have the defined string "methylo" in their name, but are known to not be able to perform methane oxidation. In addition a column named "group" is added, which contains all taxa of one string type. Within that group, we normalize the data using Wisconsin double standardization to account for the different sizes of the bacterial and the archaea data sets and strongly varying count numbers across taxonomic units.

```
## define search patterns
pat1 = 'methylo'
# get subset
cdna.mtrophs = asv.bac[grepl(pat1,asv.bac$taxonomy,ignore.case=T),]
# further clean pattern, these will be removed
toremove = 'Methylolige|Methylophil|Methylobacterium|Methylovirg|Methylopila|
  Sh765B-TzT-35|Rokubacteriales|Burkholderiales|MIZ17|Methylorosula'
# remove unwanted and get wanted subset
cdna.mtrophs = cdna.mtrophs[!grepl(toremove,cdna.mtrophs$taxonomy,ignore.case=T),]
# add column that specifies the data set
cdna.mtrophs$group <- "methanotroph"
cdna.mtrophs$group <- as.character(cdna.mtrophs$group)

# normalize counts with Wisconsin double standardization
tmp <- cdna.mtrophs[,!names(cdna.mtrophs) %in% c("taxonomy", "group")]
cdna.mtrophs[,!names(cdna.mtrophs) %in%
               c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))


pat2 = 'desulfo'
# get subset
cdna.srb = asv.bac[grepl(pat2,asv.bac$taxonomy,ignore.case=T),]
# further clean pattern, these will be removed
toremove = 'Syntrophus|Smithella|Syntrophorhabdus|Syntrophales|Desulfuromonadia|
  bacteriap25|Carboxydothermales|Ammonificaceae|Desulfallas'
# remove unwanted and get wanted subset
cdna.srb = cdna.srb[!grepl(toremove,cdna.srb$taxonomy,ignore.case=T),]
# add column that specifies the data set
cdna.srb$group <- "SRB"
```

```
cdna.srb$group <- as.character(cdna.srb$group)

# normalize counts with Wisconsin double standardization
tmp <- cdna.srb[,!names(cdna.srb) %in% c("taxonomy", "group")]
cdna.srb[,!names(cdna.srb) %in% c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))
```

Next, archaea are chosen that can either perform methanogenesis or are known ANME. The process is similar to the one above for bacteria.

```
###FOR METHANOGENS cDNA

## define search patterns
pat2 = 'methano'
# get subset
cdna.mgens = asv.arc[grepl(pat2,asv.arc$taxonomy,ignore.case=T),]
# further clean pattern, these will be removed
toremove = 'Methanopere|ANME'
# remove unwanted and get wanted subset
cdna.mgens.oA = cdna.mgens[!grepl(toremove,cdna.mgens$taxonomy,ignore.case=T),]
# add column that specifies the data set
cdna.mgens.oA$group <- "methanogen"
cdna.mgens.oA$group <- as.character(cdna.mgens.oA$group)

# normalize counts with Wisconsin double standardization
tmp <- cdna.mgens.oA[,!names(cdna.mgens.oA) %in% c("taxonomy", "group")]
cdna.mgens.oA[,!names(cdna.mgens.oA) %in%
                c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))


#FOR ANME cDNA
cdna.mgens.ANME = asv.arc[grepl("ANME|Methanopere", asv.arc$taxonomy, ignore.case=T),]
# add column that specifies the data set
cdna.mgens.ANME$group <- "ANME"
cdna.mgens.ANME$group <- as.character(cdna.mgens.ANME$group)

# normalize counts with Wisconsin double standardization
tmp <- cdna.mgens.ANME[,!names(cdna.mgens.ANME) %in% c("taxonomy", "group")]
cdna.mgens.ANME[,!names(cdna.mgens.ANME) %in%
                  c("taxonomy", "group")] <-  t(wisconsin(t(tmp)))
```

The earlier defined function `prep.dna` is now applied again to the four microbial taxonomical groups (*methanogens*, *methanotrophs*, *SRB* and *ANME*). In parallel, the names (`nms`) object, which contains the information about sampling details (date, location, depth, replicate, etc.) is merged with the environmental variables. Here, only the ID and the depth from former *nms* data set are kept, while sampling details will be taken from the environmental data. However, environmental variables, including absolute microbial abundances, derived from qPCR, are also excluded and will only be used for depth profile plots later.

```
### Transform for better handling of taxonomies
## Methanotrophs
cdna.mtrophs <- prep.dna(cdna.mtrophs, "methanotroph", "cDNA")
# join with names
cdna.mtrophs$nms <- merge(cdna.mtrophs$nms[,c(1,6)],
                          nms_new_cdna[,c(2:7, 17, 22)], by="ID")
```

```
# don't use data that is not comparable
cdna.mtrophs$otu <- cdna.mtrophs$otu[,!is.na(cdna.mtrophs$nms$depth_id)]
cdna.mtrophs$nms <- cdna.mtrophs$nms[!is.na(cdna.mtrophs$nms$depth_id),]

#SRB
cdna.srb<- prep.dna(cdna.srb, "SRB", "cDNA")
# join with names
cdna.srb$nms <- merge(cdna.srb$nms[,c(1,6)],
                      nms_new_cdna[,c(2:7, 17, 22)], by="ID")
# don't use data that is not comparable
cdna.srb$otu <- cdna.srb$otu[,!is.na(cdna.srb$nms$depth_id)]
cdna.srb$nms <- cdna.srb$nms[!is.na(cdna.srb$nms$depth_id),]

## Methanogens without ANME
cdna.mgens.oA <- prep.dna(cdna.mgens.oA, "methanogen", "cDNA")
# join with names
cdna.mgens.oA$nms <- merge(cdna.mgens.oA$nms[,c(1,6)],
                           nms_new_cdna[,c(2:7, 17, 22)], by="ID")
# don't use data that is not comparable
cdna.mgens.oA$otu <- cdna.mgens.oA$otu[,!is.na(cdna.mgens.oA$nms$depth_id)]
cdna.mgens.oA$nms <- cdna.mgens.oA$nms[!is.na(cdna.mgens.oA$nms$depth_id),]

## ANME
cdna.mgens.ANME <- prep.dna(cdna.mgens.ANME, "ANME", "cDNA")
# join with names
cdna.mgens.ANME$nms <- merge(cdna.mgens.ANME$nms[,c(1,6)],
                             nms_new_cdna[,c(2:7, 17, 22)], by="ID")
# don't use data that is not comparable
cdna.mgens.ANME$otu <- cdna.mgens.ANME$otu[,!is.na(cdna.mgens.ANME$nms$depth_id)]
cdna.mgens.ANME$nms <- cdna.mgens.ANME$nms[!is.na(cdna.mgens.ANME$nms$depth_id),]
```

After creating four single data sets (one per metabolic group), we merge them all into one.

```
### Put them all together
cdna <- list(otu = rbind(cdna.mtrophs$otu, cdna.mgens.oA$otu,
                         cdna.mgens.ANME$otu, cdna.srb$otu),
             nms = data.frame(cdna.mtrophs$nms[,-2]),
             tax = rbind(cdna.mtrophs$tax, cdna.mgens.oA$tax,
                         cdna.mgens.ANME$tax, cdna.srb$tax))
names(cdna$nms)[c(4,5,8)] <- c("depth", "dzone","what")
```

Now we choose the taxonomic level, which will be displayed in the plot later on. Because we did not choose the same level for all four groups, we had to do this individually for each group. We chose the order level for *methanogens* and *methanotrophs*. For *SRB* we selected the class level, because of too many entries in the case of order level. *ANME* does not have many entries, so we decided to show them as detailed as possible on the genus.

```
### Summarize over the chosen taxonomical level
# determine the taxonomical level
tax.lev1 <- cdna.mtrophs$tax$Or

tax.lev2 <- cdna.mgens.oA$tax$Or
```

```
tax.lev3 <- cdna.mgens.ANME$tax$Ge

tax.lev4 <- cdna.srb$tax$Cl
tax.lev4[(is.na(tax.lev4)) | (grepl("uncult", tax.lev4, ignore.case=T))] <-
    cdna.srb$tax$Ph[(is.na(tax.lev4)) |
    (grepl("uncult", tax.lev4, ignore.case=T))]
```

After we define the level of each group, we create a `data.frame` and merge them together at the end.

```
# sum up per taxonomical level and sample
cdna.pl1 <- data.frame(tax = tax.lev1, cdna.mtrophs$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

cdna.pl2 <- data.frame(tax = tax.lev2, cdna.mgens.oA$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

cdna.pl3 <- data.frame(tax = tax.lev3, cdna.mgens.ANME$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

cdna.pl4 <- data.frame(tax = tax.lev4, cdna.srb$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

cdna.pl<- rbind(cdna.pl1, cdna.pl2, cdna.pl3, cdna.pl4)
```

The column "group" was ignored until here. Now, we create a data frame, containing the information about which taxa belong to which group. Later on, we bind them together again.

```
# get the groupings right
cdna.groups1 <- data.frame(tax = tax.lev1, cdna.mtrophs$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

cdna.groups2 <- data.frame(tax = tax.lev2, cdna.mgens.oA$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

cdna.groups3 <- data.frame(tax = tax.lev3, cdna.mgens.ANME$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

cdna.groups4 <- data.frame(tax = tax.lev4, cdna.srb$tax) %>%
  group_by(tax) %>%
  summarise(group = first(group))

cdna.groups<- rbind(cdna.groups1, cdna.groups2, cdna.groups3, cdna.groups4)
```

2.2 Plotting with ggplot

2.2.1 Plot preparation

```
#load required packages
library(reshape2)
library(forcats)
library(ggplot2)
library(ggpubr)
library(scales)
library(ggthemes)
```

To plot the data with ggplot2, we need to transform them from wide format to molten format using the function `melt`. Here, we re-name the given labels and then merge the former ASV data with the environmental data in order to have all sampling campaign information. In addition, we also give the final labels that we want to have in the bubble plot by creating a new column according to information extracted from the column "Date". Then, we add the taxonomic groups to the data and merge according to the taxa.

```
## melt data
dna.mltn <- melt(dna.pl)
names(dna.mltn) <- c("tax", "ID", "abundance")
## Merge with names
dna.mltn <- merge(dna.mltn, dna$nms, by = "ID")
dna.mltn$prepost <- NA
dna.mltn$prepost[dna.mltn$Date == "base14"] <- "Baseline2014"
dna.mltn$prepost[dna.mltn$Date == "post-infl19_aut"] <- "Post-inflow Aut.19"

## Merge with groups
dna.mltn <- merge(dna.mltn, dna.groups[,1:2], by = "tax")
```

Because we do not want to display the biological replicates individually, we summarize the data by the location, depth zone, date and taxonomy, so that the mean of the replicates will be shown.

```
## Summarize over replicates
dna.mltn <- dna.mltn %>%
  group_by(site, dzone, Date, tax) %>%
  summarise(
    abun = mean(abundance),
    prepost = first(prepost),
    group = first(group)
  )
```

We want to compare the microbial community along the depth profile with those from previous years. Therefore, we decided to only include depth sections, that were measured in all sampling campaigns and exclude measurements from below 50 cm.

```
dna.mltn<- dna.mltn[(!dna.mltn$dzone=="5"),]
```

2.2.2 DNA locations HC1-4

Now, we want to create different kinds of bubble plots. First, data from all locations from Baseline2014 and Post-inflow Autumn2019 will be plotted. Herefore, subsets with the sampling campaign we wish for

have to be created, as well as identifiers, consisting of location name ("site"), depth zone ("dzone") and the taxonomic group ("tax").

```
## Filter to only have pre and post
dna.mltn.pp <- filter(dna.mltn, !is.na(prepost))

dna.mltn.pp <- dna.mltn.pp %>%
  mutate(
    sitedepth = paste(site, dzone),
    sitedepthtax = paste(sitedepth, tax)
  )

dna.mltn.pp$group<-as.factor(dna.mltn.pp$group)
dna.mltn.pp$dzone<-as.factor(dna.mltn.pp$dzone)
dna.mltn.pp$sitedepth<-as.factor(dna.mltn.pp$sitedepth)
```

Then the plotting can start. First, we define a color palette that is friendly towards people with color vision deficiencies. The same palette will be used for all bubble plots.

```
y_cols = colorblind_pal()(4)
names(y_cols) = levels(factor(dna.mltn.pp$group))
```

Then, we plot using ggplot2. We plot into an object because we later will combine plots to a complex plot with several subplots using `ggarrange` of package `ggpubr`.

```
dna1 <- dna.mltn.pp %>%
  mutate(tax = fct_reorder2(tax, tax, group, .desc=TRUE)) %>%
  ggplot( aes(sitedepth, tax, size = sqrt(abun), color = group)) +
  geom_point(alpha = 0.5) +
  scale_size(range = c(0.1, 10), name = "Relative abundance") +
  scale_color_manual(breaks = names(y_cols), values=y_cols)+
  facet_wrap(vars(prepost)) +
  labs(x = "Location and depths", col = "Tax. groups", title = "a) DNA")+
  guides(colour = guide_legend(override.aes = list(size=5)))+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        legend.position = "none",
        legend.text = element_text(size=16),
        legend.title = element_text(size=16),
        legend.key = element_rect(fill = NA),
        strip.background = element_blank(),
        strip.text = element_text(hjust = 0, size = 18),
        plot.title = element_text(size = 20),
        axis.title.x = element_text(size = 16),
        axis.text.x = element_text(size = 16, angle = 90, vjust = 1, hjust=0.5),
        axis.title.y = element_blank(),
        axis.text.y = element_text(size = 16,
            color= y_cols[factor(sort(dna.groups$group, decreasing = TRUE))]))+
  geom_vline(xintercept = c(4.5, 8.5, 12.5), color = "grey", linetype = 2)
```

2.2.3 DNA location HC2

Second, DNA-based data from location HC2 ought to be plotted at four different sampling campaigns. The process is similar to the DNA-based data from location HC1-4, but here, there is more space on the x-axes legend, because of the common location HC2. So, depth zones will be written as numbers, not coded.

```
dna.mltn.hc2<- dna.mltn[(dna.mltn$site=="HC2"),]
dna.mltn.hc2<- dna.mltn.hc2[(!dna.mltn.hc2$dzone=="5"),]

dna.mltn.hc2$prepost[dna.mltn.hc2$Date == "drought18"] <- "Drought2018"
dna.mltn.hc2$prepost[dna.mltn.hc2$Date == "post-infl19_spr"] <- "Post-inflow Spr.19"
dna.mltn.hc2$prepost[dna.mltn.hc2$Date == "post-infl19_aut"] <- "Post-inflow Aut.19"

dna.mltn.hc2$dzone[dna.mltn.hc2$dzone=="1"] <-"0-5"
dna.mltn.hc2$dzone[dna.mltn.hc2$dzone=="2"] <-"5-20"
dna.mltn.hc2$dzone[dna.mltn.hc2$dzone=="3"] <-"20-40"
dna.mltn.hc2$dzone[dna.mltn.hc2$dzone=="4"] <-"40-50"


dna.mltn.hc2$group<-as.factor(dna.mltn.hc2$group)

dna.mltn.hc2$dzone <- factor(dna.mltn.hc2$dzone,
                             levels=c("0-5", "5-20", "20-40", "40-50"))
dna.mltn.hc2$prepost <- factor(dna.mltn.hc2$prepost,
                             levels=c("Baseline2014", "Drought2018",
                                      "Post-inflow Spr.19", "Post-inflow Aut.19"))

names(y_cols) = levels(factor(dna.mltn.hc2$group))

#plot
dna2<-dna.mltn.hc2 %>%
  mutate(tax = fct_reorder2(tax, tax, group, .desc=TRUE)) %>%
  ggplot( aes(dzone, tax, size = sqrt(abun), color = group)) +
  geom_point(alpha = 0.5) +
  scale_size(range = c(0.1, 10), name = "Relative abundance") +
  facet_grid(~prepost) +
  labs(col = "Tax. group", shape = "Location", x="Depth (cm)",
       title = "b) DNA location HC2")+
  scale_color_manual(breaks = names(y_cols), values = y_cols)+
  guides(colour = guide_legend(override.aes = list(size=5)))+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        legend.position = "none",
        legend.text = element_text(size=16),
        legend.title = element_text(size=16),
        legend.key = element_rect(fill = NA),
        strip.background = element_blank(),
        strip.text = element_text(hjust = 0, size = 18),
        plot.title = element_text(size = 20),
        axis.title.x = element_text(size = 16),
        axis.text.x = element_text(size = 16, angle = 90),
        axis.title.y = element_blank(),
        axis.text.y = element_text(size = 16,
```

```
              color= y_cols[factor(sort(dna.groups$group, decreasing = TRUE))])
  )
```

2.2.4 cDNA location HC2

Third, cDNA-based abundances from location HC2 on four sampling campaigns will be plotted. Therefore, we want to ensure, that y-axes label are equivalent to the DNA-based data, so we merge them with the labels from the DNA-based data set. After this, the process is similar to DNA-based relative abundances at location HC2 (see above).

```
### four lines take care that taxa names are the same in DNA and cDNA
oempf <- merge(dna.pl[,1], cdna.pl, by="tax", all.x = TRUE)
oempf[is.na(oempf)] <- 0
cdna.groups <- merge(oempf[,1:2], data.frame(cdna.groups), by = "tax",
                     all.x = TRUE)[,-2]
cdna.groups$group[is.na(cdna.groups$group)] <- c("SRB", "methanotroph")
### now go on as before
cdna.mltn <- melt(oempf)
names(cdna.mltn) <- c("tax", "ID", "abundance")
## Merge with names
cdna.mltn <- merge(cdna.mltn, cdna$nms, by = "ID")
cdna.mltn$prepost <- NA
cdna.mltn$prepost[cdna.mltn$Date == "drought18"] <- "Drought2018"
cdna.mltn$prepost[cdna.mltn$Date == "post-infl19_spr"] <- "Post-inflow Spr.19"
cdna.mltn$prepost[cdna.mltn$Date == "post-infl19_aut"] <- "Post-inflow Aut.19"
## Merge with groups
cdna.mltn <- merge(cdna.mltn, cdna.groups[,1:2], by = "tax")

## Summarize over replicates
cdna.mltn <- cdna.mltn %>%
  group_by(dzone, Date, tax) %>%
  summarise(
    abun = mean(abundance),
    prepost = first(prepost),
    group = first(group)
  )

#make subset with plot HC2 only
cdna.mltn$dzone[cdna.mltn$dzone=="1"] <-"0-5"
cdna.mltn$dzone[cdna.mltn$dzone=="2"] <-"5-20"
cdna.mltn$dzone[cdna.mltn$dzone=="3"] <-"20-40"
cdna.mltn$dzone[cdna.mltn$dzone=="4"] <-"40-50"

cdna.mltn$group <- as.factor(cdna.mltn$group)
cdna.mltn$dzone <- factor(cdna.mltn$dzone,
                          levels=c("0-5", "5-20", "20-40", "40-50"))
cdna.mltn$prepost <- factor(cdna.mltn$prepost,
                          levels=c("Baseline2014", "Drought2018",
                                   "Post-inflow Spr.19", "Post-inflow Aut.19"))

names(y_cols) = levels(factor(cdna.mltn$group))

#plot
```

```
cdna<-cdna.mltn %>%
  mutate(tax = fct_reorder2(tax, tax, group, .desc=TRUE)) %>%
  ggplot( aes(dzone, tax, size = sqrt(abun), color = group)) +
  geom_point(alpha = 0.5) +
  scale_size(range = c(0.1, 10), name = "Relative abundance") +
  facet_grid(~prepost) +
  labs(col = "Tax. group", shape = "Location", x="Depth (cm)",
       title = "c) cDNA location HC2")+
  scale_color_manual(breaks = names(y_cols), values = y_cols)+
  guides(colour = guide_legend(override.aes = list(size=5)))+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        legend.position = "none",
        legend.text = element_text(size=16),
        legend.title = element_text(size=16),
        legend.key = element_rect(fill = NA),
        strip.background = element_blank(),
        strip.text = element_text(hjust = 0, size = 18),
        plot.title = element_text(size = 20),
        axis.title.x = element_text(size = 16),
        axis.text.x = element_text(size = 16, angle = 90),
        axis.title.y = element_blank(),
        axis.text.y = element_text(size = 16,
            color= y_cols[factor(sort(cdna.groups$group, decreasing = TRUE))])
  )
```

2.2.5 Final edits

The three independent plots are pasted together with `ggarrange`by package ggpubr. However, some finishing touches before submission where done in a graphic program.

```
# paste graphs together
figure1 <- ggarrange(dna2, cdna,
                     widths = c(1.3,1),
                     ncol = 2, nrow = 1,  legend = "bottom",
                     common.legend = TRUE)

figure2<- ggarrange(dna1, figure1,
                    widths = c(1.1,1),
                    ncol = 1, nrow = 2,  legend = "none",
                    common.legend = TRUE)

# save to file
ggsave(figure2, filename = "figures/Fig4_bubbleplot.pdf",
       width = 23, height = 16)
```

3 NMDS plotting

For a new plot, we load the required data again. To plot an NMDS ordination including a PostHoc-fit of the environmental data, we need the environmental data and the microbial ASV files. First we load

the environmental data and make our usual adaptations. In addition, we change the unit of the CO_2 concentrations from μM to mM. Then we load the packages.

```
## Works when working directory is set correctly above at the beginning
## read in cDNA names
nms_new <- read.table("qpcr_nms_mudenv.txt", header = TRUE)
nms_new <- nms_new[!is.na(nms_new$ID),]
nms_new_DNA<- nms_new[(nms_new$DNA_type=="DNA"),]


## Transform units of the CO2 flux from micromole to millimole
nms_new_DNA$co2_mM <- nms_new_DNA$co2_miM/1000

# load necessary libraries
library(dplyr)
library(simba)
library(vegan)
```

3.1 Preparations

3.1.1 Bacteria

Now read in the microbial ASV files, first bacteria. Check bacteria and also archaea data for similarity in a separate step. As before we use some lines of code to take care, that there are the same samples in the two data sets.

```
asv.bac <- read.table("DNA_bac.tsv", header=T, sep="\t",
                  stringsAsFactors=F, row.names=1)

####### Take care, that the same samples are addressed
asv.arc <- read.table("DNA_arc.tsv", header=T, sep="\t",
                      stringsAsFactors=F, row.names=1)
sel.arc <- match(names(asv.bac), names(asv.arc))
```

This time, our target groups of interest are bacteria and archaea as a domain, so we normalize the data using Wisconsin double standardization to account for the different sizes of the whole bacterial and the archaeal data set and strongly varying count numbers across taxonomic units. We remove depths, which do not have a counterpart in other sampling campaigns and apply the earlier defined function `prep.dna` to the group bacteria as a whole.

```
## Wisconsin double standard transformation (see reasoning)
asv.bac[,-ncol(asv.bac)] <- t(wisconsin(t(asv.bac[,-ncol(asv.bac)])))

#remove depths that are not comparable
asv.bac$c_2014_oct_HC2_50.55a <- NULL
asv.bac$c_2014_oct_HC2_50.55b <- NULL

asv_bac <- prep.dna(asv.bac, "bac", "DNA")
# join with names
asv_bac$nms <- merge(asv_bac$nms[,c(1,6)], nms_new_DNA[,-c(1,18:21)] , by= "ID")
# don't use data that is not comparable
asv_bac$otu <- asv_bac$otu[,!is.na(asv_bac$nms$depth_id)]
asv_bac$nms <- asv_bac$nms[!is.na(asv_bac$nms$depth_id),]
```

Now we choose the taxonomic level, which will be displayed in the plot later on. Here, we take the genus level. Then, we define the minimum threshold of measured abundances to be displayed. Taxa with abundances lower than 0.01 % will not be considered for the ordination.

```
# determine the taxonomic level
tax.lev <- apply(asv_bac$tax[,2:6], 1, paste, collapse="_")
# sum up per taxonomic level and sample
asv_bac.fl <- data.frame(tax = tax.lev, asv_bac$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

## extract the taxa which have enough abundance (> 0.01%)
sel <- rowSums(asv_bac.fl[,-1]/sum(asv_bac.fl[,-1])*nrow(asv_bac.fl)) > 0.01
sel2 <- c(TRUE, colSums(asv_bac.fl[,-1]) > 0)
asv_bac.fl <- asv_bac.fl[sel,sel2]
asv_bac.fl.tax <- asv_bac$tax[sel,]
```

3.1.2 Archaea

Now we move on to the archaeal ASV files. In a separate step, similarity of bacteria and archaea data set is ensured. We also normalize the archaeal data using Wisconsin double standardization. We remove depths, which do not have a counterpart in other sampling campaigns and apply the earlier defined function `prep.dna`to the group archaea as a whole. Then we choose the taxonomic level, which will be displayed in the plot later on. Here, we chose the genus level and also define the minimum threshold of 0.01 %.

```
asv.arc <- asv.arc[,sel.arc] # Here, we constrain the archea data set to the
                      # samples that also were in the bacterial data set

## Wisconsin double standard transformation (see reasoning)
asv.arc[,-ncol(asv.arc)] <- t(wisconsin(t(asv.arc[,-ncol(asv.arc)])))

#remove depths that are not comparable
asv.arc$c_2014_oct_HC2_50.55a <- NULL
asv.arc$c_2014_oct_HC2_50.55b <- NULL

asv_arc <- prep.dna(asv.arc, "arc", "DNA")
# join with names
asv_arc$nms <- merge(asv_arc$nms[,c(1,6)], nms_new_DNA [,-c(1,18:21)], by="ID")
# don't use data that is not comparable
asv_arc$otu <- asv_arc$otu[,!is.na(asv_arc$nms$depth_id)]
asv_arc$nms <- asv_arc$nms[!is.na(asv_arc$nms$depth_id),]


# determine the taxonomic level
tax.lev <- apply(asv_arc$tax[,2:6], 1, paste, collapse="_")
# sum up per taxonomic level and sample
asv_arc.fl <- data.frame(tax = tax.lev, asv_arc$otu) %>%
  group_by(tax) %>%
  summarise_all(sum)

## extract the taxa which have enough abundance (> 0.1%)
sel <- rowSums(asv_arc.fl[,-1]/sum(asv_arc.fl[,-1])*nrow(asv_arc.fl)) > 0.01
sel2<- c(TRUE, colSums(asv_arc.fl [,-1])>0)
```

```
asv_arc.fl <- asv_arc.fl[sel,sel2]
asv_arc.fl.tax <- asv_arc$tax[sel,]
```

3.2 NMDS Ordination

NMDS is run using function `metaMDS` found in the vegan package. For more stability across platforms we are setting a seed for the random number generator. And for sparing the runs being printed to the output during knitting, we set `trace = 0`.

```
set.seed(1)
asv_bac.nmds <- metaMDS(t(asv_bac.fl[,-1]), trymax=100, trace=0)
asv_arc.nmds <- metaMDS(t(asv_arc.fl[,-1]), trymax=100, trace=0)
```

3.3 Further prepare environmental data

Before we plot the NMDS ordination, we need to merge the environmental data with bacteria and archaea data set separately. Then we use `envfit` to do a Post-Hoc fit of environmental variables on the NMDS ordination configurations.

```
# Bacteria
scrs_bac <- data.frame(scores(asv_bac.nmds))
scrs_bac$ID <- rownames(scrs_bac)
env_bac <- merge(scrs_bac, nms_new_DNA, by="ID")[,c("ID","depth_id", "pH", "EC_mScm",
                    "sulfate_mM","co2_mM", "ch4_miM", "d13C_CO2", "d13C_CH4")]
bac.efit <- envfit(asv_bac.nmds, env_bac[,-1], na.rm=TRUE)
asv_bac$nms <- merge(scrs_bac, asv_bac$nms, by = "ID")

# Archaea
scrs_arc <- data.frame(scores(asv_arc.nmds))
scrs_arc$ID <- rownames(scrs_arc)
env_arc <- merge(scrs_arc, nms_new_DNA, by="ID")[,c("ID", "depth_id", "pH", "EC_mScm",
                    "sulfate_mM",  "co2_mM", "ch4_miM", "d13C_CO2", "d13C_CH4")]
arc.efit <- envfit(asv_arc.nmds, env_arc[,-1], na.rm=TRUE)
asv_arc$nms <- merge(scrs_arc, asv_arc$nms, by = "ID")
```

##Plot NMDS Ordination

To plot, we proceed step by step using base R. First, we create an external file, then we set the frame and then boundaries. Then, bacteria and archaea data are plotted, including the legends for the sampling locations, sampling campaigns, sampling depths and the sulfate concentration range (size of the symbols). In order to increase the size of the environmental fit arrows, we manipulated the plot manually.

```
## Plot to file
pdf("figures/Fig6_NMDS.pdf", 32/2.54, 16/2.54)
# Prepare color vector
clrs <- c("#fdae61", "#d7191c","#2c7bb6","#abd9e9")
# prepare empty plot
par(mfrow=c(1,2), las=1, mar=c(4,4.5,2.6,0.5))
## Bacteria
# plot empty
ps <- plot(asv_bac.nmds, type="n", display="sites",
```

```
          main=paste("a) Bacteria, stress = ", round(asv_bac.nmds$stress,2)),
          cex.main = 1.8, cex.axis = 1.3, cex.lab = 1.5, xlim=c(-1.5, 2), ylim=c(-2,1))
# plot points
points(asv_bac.nmds, display="sites",
       pch=c(21:24)[factor(asv_bac$nms$site)],
       bg=grey(symbol.size(as.numeric(asv_bac$nms$depth_id),0.8,0.2)),
       col=clrs[factor(asv_bac$nms$Date)],
       cex=symbol.size(env_bac$sulfate_mM, 1, 5), lwd =2)
# plot legends
legend("bottomright", bty="n", pch=c(21:24), cex=1.5,
       legend=levels(factor(paste(asv_bac$nms$site))))
legend("bottomleft", bty="n", pch=22, pt.lwd=2, cex=1.5,
       col=clrs, legend=levels(factor(asv_arc$nms$Date))[c(1:2,4,3)], lty=1)
# add polygons
mmh <- ordihull(asv_bac.nmds, groups=factor(asv_bac$nms$Date), "sites", col=clrs)
# add environmental data (those that are well correlated to ordination result)
bac.pfeile<- data.frame(bac.efit$vectors$arrows)
sel<- bac.efit$vectors$pvals < 0.05
bac.pfs<- bac.pfeile[sel,]
bac.pfs$NMDS1a<- bac.pfs$NMDS1 + c(0.2, -0.05, 0.25, 0.5, 0.45)
bac.pfs$NMDS2a<- bac.pfs$NMDS2 + c(-0.1, 0.05, 0.1, 0, 0)
lbls=c("Depth", "pH", "EC (mS/cm)", "Sulfate (mM)",
       expression(paste(CO[2], " (mM)")), expression(paste(CH[4], " (µM)")),
       expression(paste(delta,"13C-", DIC)), expression(paste(delta,"13C-",
                                                              CH[4])))
arrows(rep(0,nrow(bac.pfs)),rep(0,nrow(bac.pfs)),bac.pfs$NMDS1, bac.pfs$NMDS2,
       lwd= 2, col= "green4")
text(bac.pfs$NMDS1a, bac.pfs$NMDS2a, labels= lbls[sel], cex=1.5, col="green4")

## Archaea
# plot empty
ps <- plot(asv_arc.nmds, type="n", display="sites",
          main=paste("b) Archaea, stress = ", round(asv_bac.nmds$stress,2)),
          cex.main = 1.8, cex.axis = 1.3, cex.lab = 1.5, xlim=c(-1.7, 1.3))
# plot points
points(asv_arc.nmds, display="sites",
       pch=c(21:24)[factor(asv_bac$nms$site)],
       bg=grey(symbol.size(as.numeric(asv_bac$nms$depth_id),0.8,0.2)),
       col=clrs[factor(asv_arc$nms$Date)],
       cex=symbol.size(env_arc$sulfate_mM, 1, 5), lwd =2)
# plot legends
legend("topright", bty="n", pch=15, cex=1.5,
       col=grey(symbol.size(1:5,0.8,0.2)),
       legend=c("0-5cm", "5-20cm", "20-40cm", "40-50cm"))
legend("bottomright", bty="n", pch=22, cex= 1.5,
       pt.cex= symbol.size(range(env_arc$sulfate_mM, na.rm=TRUE), 1, 5),
       legend=round(range(env_arc$sulfate_mM, na.rm=TRUE), 1), y.intersp = 1.2,
       adj= c(-0.1, 0.5), lty= 1, pt.bg= "black", col= "white")
# add polygons
mmh <- ordihull(asv_arc.nmds, groups=factor(asv_arc$nms$Date), "sites", col=clrs)
# add environmental data (those that are well correlated to ordination result)
arc.pfeile<- data.frame(arc.efit$vectors$arrows)
sel<- arc.efit$vectors$pvals < 0.05
```

```
arc.pfs<- arc.pfeile[sel,]
lbls=c("Depth", "pH", "EC (mS/cm)", "Sulfate (mM)",
         expression(paste(CO[2], " (mM)")), expression(paste(CH[4], " (µM)")),
         expression(paste(delta,"13C-", DIC)), expression(paste(delta,"13C-",
                                                               CH[4])))
arrows(rep(0,nrow(arc.pfs)),rep(0,nrow(arc.pfs)),arc.pfs$NMDS1, arc.pfs$NMDS2,
       lwd= 2, col= "green4")
text(arc.pfs$NMDS1, arc.pfs$NMDS2, labels= lbls[sel], pos= 2, offset= 0.1,
       cex=1.6, col = "green4")
dev.off()
```

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

4 Environmental and microbial abundance depthprofiles

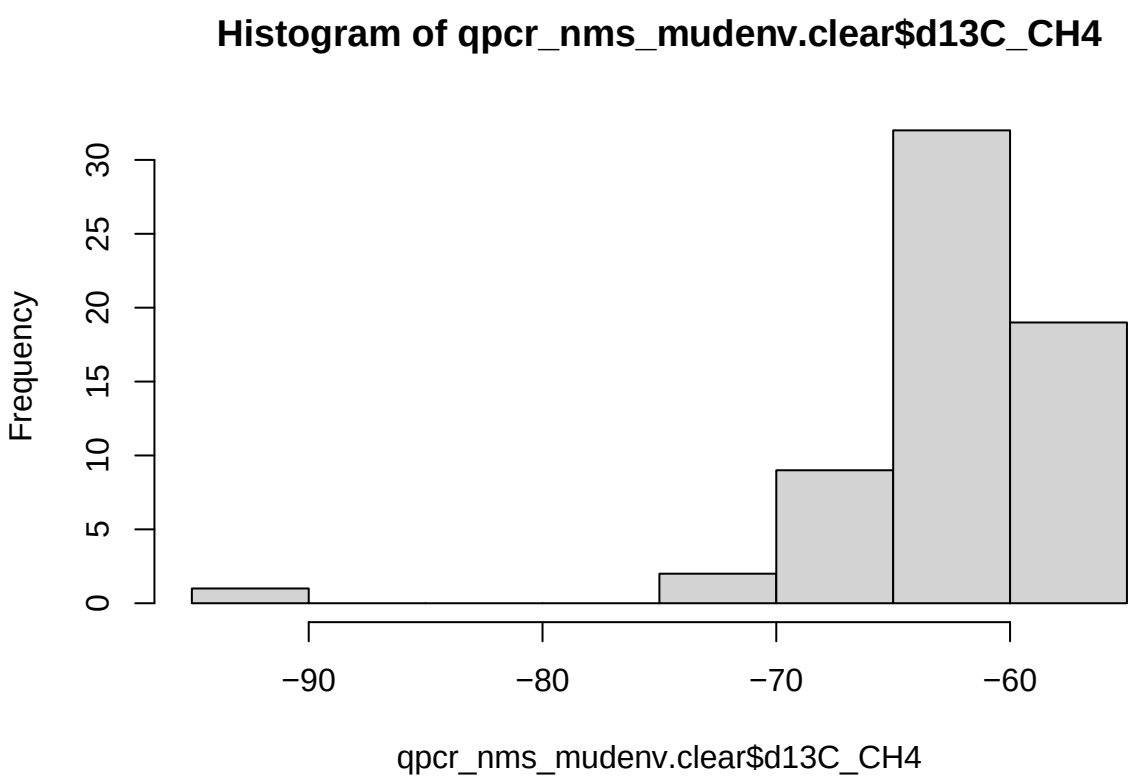
4.1 Preparation

Load the data and packages, choose data of interest and make minor changes to the data set. This changes imply creating a subset that only contains DNA-based abundances, changing the unit of CO_2 concentrations from μM to mM and make subsets for 1) the sampling campaign Baseline2014 and in Post-inflow Autumn2019, excluding extra sampling campaigns at HC2 during the Drought2018 and in Post-inflow Spring2019 and 2) all four sampling campaigns at HC2. Here, we found that two separate data sets have to be created to exclude outliers from $\delta^{13}C - CH_4$ below -90 ‰.

```
library(ggplot2)
library(ggpubr)
library(patchwork)
library(psych)
## Read in should work when working directory is correctly set at the very top
qpcr_nms_mudenv <- read.table("qpcr_nms_mudenv.txt", header = TRUE)

# make DNA subset
qpcr_nms_mudenv<- qpcr_nms_mudenv[(qpcr_nms_mudenv$DNA_type=="DNA"),]
#adapt units
qpcr_nms_mudenv$co2_mM<- qpcr_nms_mudenv$co2_miM/1000

# make subset and exclude extra HC2 sampling
qpcr_nms_mudenv.clear <- qpcr_nms_mudenv[(!qpcr_nms_mudenv$Date=="drought18"),]
qpcr_nms_mudenv.clear <-
  qpcr_nms_mudenv.clear[(!qpcr_nms_mudenv.clear$Date=="post-infl19_spr"),]
# exclude outliers
hist(qpcr_nms_mudenv.clear$d13C_CH4)
```




```
qpcr_nms_mudenv.clear2<- subset(qpcr_nms_mudenv.clear, d13C_CH4 > -80)

# make subset with all sampling campaigns at location HC2
qpcr_nms_mudenv.hc2<- qpcr_nms_mudenv[(qpcr_nms_mudenv$site=="HC2"),]
```

4.2 Absolute microbial abundances

Plots are done using ggplot2. First, absolute abundances of target genes *16S rRNA*, *mcrA*, *pmoA* and *dsrB* are plotted at locations HC1-4 from sampling campaign Baseline2014 and Post-inflow Autumn2019. Later, abundances of the same genes are plotted at location HC2 from all sampling campaigns, including Drought2018 and Post-inflow Spring2019. All plots are plotted into objects that are later used with `ggarrange` to produce complex plots with several sub-plots.

4.2.1 16S plot

```
X16S <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=log10(X16S),
                                               col = Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(8, 11))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#2c7bb6"), 0.15))+
```

```
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, method = "loess", level = 0.95, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance 16S
       rRNA (copies/g)", col = "Date", shape = "Location", x="Depth (cm)",
       title = "Total prokaryotes")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.y = element_text(size = 12),
        axis.text.y = element_text(size = 11))
```

4.2.2 mcrA plot

```
mcrA <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=log10(mcrA),
                                               color= Date, fill=Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14",  "post-infl19_aut"),
                     values=c("#fdae61", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  guides(fill=FALSE, col = FALSE)+
  geom_smooth(show.legend=FALSE, method = "loess", level = 0.95, span = 0.5) +
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance mcrA
       (copies/g)", shape = "Location", x="Depth (cm)", title = "methanogens")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_line(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.2.3 pmoA plot

```
pmoA <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=log10(pmoA),
                                               color= Date, fill=Date))+
```

```
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14",  "post-infl19_aut"),
                     values=c("#fdae61", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  guides(fill=FALSE, col = FALSE)+
  geom_smooth(show.legend=FALSE, method = "loess", level = 0.95, span = 0.5)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance pmoA
       (copies/g)", shape = "Location", x="Depth (cm)", title = "methanotrophs")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.2.4 dsrB plot

```
dsrB <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=log10(dsrB),
                                               color = Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  geom_smooth(show.legend = FALSE, method = "loess", level = 0.95, span = 0.5) +
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance dsrB
       (copies/g)", x="Depth (cm)", title = "SRB", shape = "Location")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

Now we turn to the sub-plots for only location HC2

4.2.5 16S plot for HC2 only

```
# Location HC2 16S plot
X16S.hc2 <-ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=log10(X16S),
                                                color= Date, fill=Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(8, 11))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, method = "loess", level = 0.5, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance 16S
       rRNA (copies/g)", col = "Date", shape = "Location", x="Depth (cm)")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 11))
```

4.2.6 mcrA plot for HC2 only

```
mcrA.hc2 <-ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=log10(mcrA),
                                                color= Date, fill=Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  guides(fill=FALSE, shape = FALSE)+
  geom_smooth(show.legend = FALSE, method = "loess", level = 0.5, span = 0.5) +
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance mcrA
       (copies/g)", col = "Date", shape = "Location", x="Depth (cm)")+
```

```
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.2.7 pmoA plot for HC2 only

```
pmoA.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=log10(pmoA),
                                                 color= Date, fill=Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, level = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance pmoA
       (copies/g)", col = "Date", shape = "Location", x="Depth (cm)")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.2.8 dsrB plot for HC2 only

```
dsrB.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=log10(dsrB),
                                                 color= Date, fill=Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 11))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
```

```
                      values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, method = "loess", level = 0.5, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "log10 Abundance dsrB
       rRNA (copies/g)", col = "Date", shape = "Location", x="Depth (cm)")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.2.9 Plot final microbes figure

Plot all above defined figures into one using `wrap_plots` by ggpubr and save it.

```
p0<-list(X16S, mcrA, pmoA, dsrB,
         X16S.hc2, mcrA.hc2, pmoA.hc2, dsrB.hc2)

figure1.2 <- wrap_plots(p0, nrow= 2)+
  plot_layout(guides="collect") & theme(legend.position = "bottom") &
  plot_annotation(tag_levels = 'a')

# save locally in two different formats
ggsave(figure1.2, filename = "figures/Fig5_qpcr_depth_profiles.png",
       width = 10, height = 7, dpi = 300)

ggsave(figure1.2, filename = "figures/Fig5_qpcr_depth_profiles.pdf",
       width = 10, height = 7)
```

4.3 Environmental variables

Proceed as with absolute microbial abundances. First, we plot all locations, then location HC2.

4.3.1 pH depth profile

```
ph <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=pH, color= Date,
                                             fill= Date))+
```

```
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(5, 8))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "pH \n",x="Depth (cm)", shape = "Location", title = "pH")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.y = element_text(size = 12),
        axis.text.y = element_text(size = 11),
        axis.title.x = element_blank(),
        axis.text.x = element_blank())
```

4.3.2 Sulfate depth profile

```
sulfate <-ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=sulfate_mM,
                                                 color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 50))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Sulfate concentration (mM)", shape = "Location", x="Depth (cm)",
       title = "Sulfate")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.3 Chloride depth profile

```
Cl <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=cl_mM, color= Date,
                                             fill= Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 100))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Chloride concentration (mM)",x="Depth (cm)", shape = "Location",
       title =  "Chloride")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.4 EC (electric conductivity) depth profile

```
EC<- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=EC_mScm, color= Date,
                                            fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 15))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Electrical conductivity (mS/cm)",shape = "Location", x="Depth (cm)",
       title = "EC")+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.5 CH_4 concentration depth profile

```
ch4 <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=ch4_miM, color= Date,
                                              fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 600))+
  scale_color_manual(breaks = c("base14",  "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(CH[4], " ", concentration," ", "(",mu, M,")")),
       shape = "Location", x="Depth (cm)",
       title = expression(paste(CH[4])))+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.y = element_text(size = 12),
        axis.text.y = element_text(size = 11),
        axis.title.x = element_blank(),
        axis.text.x = element_blank())
```

4.3.6 CO_2 concentration depth profile

```
co2 <- ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=co2_mM, color= Date,
                                              fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 30))+
  scale_color_manual(breaks = c("base14",  "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(CO[2], " ", concentration, " ", mM)),
       shape = "Location",x="Depth (cm)",
       title = expression(paste(CO[2])))+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
```

```
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.7 $\delta^{13}C - CH_4$ depth profile

```
d13ch4<- ggplot(data=qpcr_nms_mudenv.clear2, aes(x=anchor, y=d13C_CH4,
                                                 color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(-75,-55))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(delta^13, C," ",CH[4], "\n (\211)")),
       shape = "Location", x="Depth (cm)",
       title= expression(paste(delta^13, C," ",CH[4])))+
  scale_y_continuous(minor_breaks = waiver())+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.8 $\delta^{13}C - CO_2$ depth profile

```
d13co2<-ggplot(data=qpcr_nms_mudenv.clear, aes(x=anchor, y=d13C_CO2,
                                               color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(-30, 10))+
  scale_color_manual(breaks = c("base14", "post-infl19_aut"),
                     values=c("#fdae61",  "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14",  "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61",  "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, col = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(delta^13, C," ",DIC, " ","(\211)")),
       shape = "Location", x="Depth (cm)",
       title= expression(paste(delta^13, C," ",DIC)))+
```

```
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title = element_blank(),
        axis.text = element_blank())
```

4.3.9 pH depth profile at HC2 only

```
# Location HC2
ph.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=pH,
                                               color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(5, 8))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "pH",x="Depth (cm)", col = "Date", shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title= element_text(size = 12),
        axis.text= element_text(size = 11))
```

4.3.10 Sulfate depth profile at HC2 only

```
sulfate.hc2<-ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=sulfate_mM,
                                                  color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 50))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
```

```
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Sulfate concentration (mM)", col = "Date", x="Depth (cm)",
       shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.3.11 Chloride depth profile at HC2 only

```
Cl.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=cl_mM,
                                               color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 100))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Chloride concentration (mM)",x="Depth (cm)", col = "Date",
       shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.3.12 EC depth profile at HC2 only

```
EC.hc2<- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=EC_mScm,
                                              color= Date, fill = Date))+
```

```
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(4, 15))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "aug2018", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = "Electrical conductivity (mS/cm)",x="Depth (cm)", shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.3.13 CH_4 concentration depth profile at HC2 only

```
ch4.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=ch4_miM,
                                                color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 600))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(CH[4], " ", concentration," ", "(",mu, M,")")),
       col = "Date", x="Depth (cm)",
       shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
```

```
        axis.title= element_text(size = 12),
        axis.text= element_text(size = 11))
```

4.3.14 CO_2 concentration depth profile at HC2 only

```
co2.hc2 <- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=co2_mM,
                                                color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(0, 30))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(CO[2], " ", concentration, " ", (mM))),
       col = "Date",x="Depth (cm)",
       shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.3.15 $\delta^{13}C - CH_4$ depth profile at HC2 only

```
d13ch4.hc2<- ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=d13C_CH4,
                                                  color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(-75,-55))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
```

```
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE) +
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(delta^13, C," ",CH[4], " %")), col = "Date",
       x="Depth (cm)", shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

4.3.16 $\delta^{13}C - CO_2$ depth profile at HC2 only

```
d13co2.hc2<-ggplot(data=qpcr_nms_mudenv.hc2, aes(x=anchor, y=d13C_CO2,
                                                 color= Date, fill = Date))+
  geom_point(size = 1.5, aes(shape = site))+
  expand_limits(x=c(0,50), y=c(-30, 10))+
  scale_color_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                                "post-infl19_aut"),
                     values=c("#fdae61", "#d7191c","#abd9e9", "#2c7bb6"))+
  scale_fill_manual(breaks = c("base14", "drought18", "post-infl19_spr",
                               "post-infl19_aut"),
                    values = scales::alpha(c("#fdae61", "#d7191c","#abd9e9",
                                             "#2c7bb6"), 0.15))+
  scale_shape_manual(breaks = c("HC1", "HC2", "HC3", "HC4"),
                     values=c(21, 22, 24, 25))+
  geom_smooth(show.legend = FALSE, span = 0.5) +
  guides(fill=FALSE, shape = FALSE)+
  scale_x_reverse()+
  coord_flip()+
  labs(y = expression(paste(delta^13, C," ",DIC, " %")), col = "Date",
       x="Depth (cm)", shape = "Location")+
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(linetype = "solid", fill = NA),
        axis.ticks = element_line(),
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 11),
        axis.title.y = element_blank(),
        axis.text.y = element_blank())
```

The final figure is created by using function `wrap_plot` by patchwork

```
p1<-list(ph, EC, sulfate, Cl,  ph.hc2, EC.hc2, sulfate.hc2, Cl.hc2,
            ch4, co2, d13ch4, d13co2,
            ch4.hc2, co2.hc2, d13ch4.hc2, d13co2.hc2)
```

```
figure <- wrap_plots(p1,nrow= 4)+
  plot_layout(guides="collect") &
              theme (legend.position = "bottom",
                     legend.title = element_text(size = 12),
                     legend.text =element_text(size=12)) &
              plot_annotation(tag_levels = 'a')

# save to harddrive
ggsave(figure, filename = "figures/Fig3_porewater_depth_profiles.png",
       width = 11, height = 14, dpi = 300)

ggsave(figure, filename = "figures/Fig3_porewater_depth_profiles.pdf",
       width = 11, height = 14)
```

Last, but not least, we also calculated averages using `describeBy`(package psych) to avoid creating subsets for locations combinations of interest.

```
# calculate means
## absolute microbial abundances
describeBy(qpcr_nms_mudenv.clear$X16S,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$mcrA,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$pmoA,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$dsrB,qpcr_nms_mudenv.clear$Date)

describeBy(qpcr_nms_mudenv.hc2$X16S,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$mcrA,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$pmoA,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$dsrB,qpcr_nms_mudenv.hc2$Date)

## environmental variables
describeBy(qpcr_nms_mudenv.clear$sulfate_mM,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$EC_mScm,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$cl_mM,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$ch4_miM,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear$co2_mM,qpcr_nms_mudenv.clear$Date)
describeBy(qpcr_nms_mudenv.clear2$d13C_CH4,qpcr_nms_mudenv.clear2$Date)
describeBy(qpcr_nms_mudenv.clear$d13C_CO2,qpcr_nms_mudenv.clear$Date)

describeBy(qpcr_nms_mudenv.hc2$sulfate_mM,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$EC_mScm,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$cl_mM,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$ch4_miM,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$co2_mM,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$d13C_CH4,qpcr_nms_mudenv.hc2$Date)
describeBy(qpcr_nms_mudenv.hc2$d13C_CO2,qpcr_nms_mudenv.hc2$Date)
```

Further, we test the data for normal distribution and significant differences between sampling campaigns. In order to remove all NA's from the data sets, we created individual data sets for each variables to avoid unnecessary cutting from all variables. Then, we test for normal distribution of the data. In our case, all data (absolute microbial abundances of specific genes and environmental data) were not normally distributed, so we used the Kruskal-Wallis-Test to test for significant differences between the sampling campaigns ("Date") and applied Wilcox-test as a Post-Hoc test.

```
# remove NA's
## absolute microbial abundances
qpcr1 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$X16S),]
qpcr2 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$mcrA),]
qpcr3 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$pmoA),]
qpcr4 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$dsrB),]

## environmental variables
qpcr_nms_mudenv1 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$pH),]
qpcr_nms_mudenv2 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$EC_mScm),]
qpcr_nms_mudenv3 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$sulfate_mM),]
qpcr_nms_mudenv4 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$cl_mM),]
qpcr_nms_mudenv5 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$ch4_miM),]
qpcr_nms_mudenv6 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$co2_miM),]
qpcr_nms_mudenv7 <- subset(qpcr_nms_mudenv, d13C_CH4 > -80)
qpcr_nms_mudenv7 <- qpcr_nms_mudenv7[!is.na(qpcr_nms_mudenv7$d13C_CH4),]
qpcr_nms_mudenv8 <- qpcr_nms_mudenv[!is.na(qpcr_nms_mudenv$d13C_CO2),]


#test for normal distribution
## absolute microbial abundances
shapiro.test(qpcr1$X16S)
shapiro.test(qpcr2$mcrA)
shapiro.test(qpcr3$pmoA)
shapiro.test(qpcr4$dsrB)

## environmental variables
shapiro.test(qpcr_nms_mudenv1$pH)
shapiro.test(qpcr_nms_mudenv2$EC_mScm)
shapiro.test(qpcr_nms_mudenv3$sulfate_mM)
shapiro.test(qpcr_nms_mudenv4$cl_mM)
shapiro.test(qpcr_nms_mudenv5$ch4_miM)
shapiro.test(qpcr_nms_mudenv6$co2_mM)
shapiro.test(qpcr_nms_mudenv7$d13C_CH4)
shapiro.test(qpcr_nms_mudenv8$d13C_CO2)


# test plot for significance for not normally distributed variables
## absolute microbial abundances
kruskal.test(qpcr1$X16S~qpcr1$Date)
pairwise.wilcox.test(qpcr1$X16S, qpcr1$Date, Paired = FALSE,
                     p.adjust = "bonferroni")

kruskal.test(qpcr2$mcrA~qpcr2$Date)
pairwise.wilcox.test(qpcr2$mcrA, qpcr2$Date, Paired = FALSE,
                     p.adjust = "bonferroni")

kruskal.test(qpcr3$pmoA~qpcr3$Date)
pairwise.wilcox.test(qpcr3$pmoA, qpcr3$Date, Paired = FALSE,
                     p.adjust = "bonferroni")

kruskal.test(qpcr4$dsrB~qpcr4$Date)
pairwise.wilcox.test(qpcr4$dsrB, qpcr4$Date, Paired = FALSE,
```

```
                    p.adjust = "bonferroni")


## environmental parameters
kruskal.test(qpcr_nms_mudenv1$pH~qpcr_nms_mudenv1$Date)
pairwise.wilcox.test(qpcr_nms_mudenv1$pH, qpcr_nms_mudenv1$Date, Paired = FALSE,
                    p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv2$EC~qpcr_nms_mudenv2$Date)
pairwise.wilcox.test(qpcr_nms_mudenv2$EC, qpcr_nms_mudenv2$Date, Paired = FALSE,
                    p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv3$sulfate_mM~qpcr_nms_mudenv3$Date)
pairwise.wilcox.test(qpcr_nms_mudenv3$sulfate_mM, qpcr_nms_mudenv3$Date,
                    Paired = FALSE, p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv4$cl_mM~qpcr_nms_mudenv4$Date)
pairwise.wilcox.test(qpcr_nms_mudenv4$cl_mM, qpcr_nms_mudenv4$Date, Paired = FALSE,
                    p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv5$ch4_miM~qpcr_nms_mudenv5$Date)
pairwise.wilcox.test(qpcr_nms_mudenv5$ch4_miM, qpcr_nms_mudenv5$Date,
                    Paired = FALSE, p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv6$co2_mM~qpcr_nms_mudenv6$Date)
pairwise.wilcox.test(qpcr_nms_mudenv6$co2_mM, qpcr_nms_mudenv6$Date,
                    Paired = FALSE, p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv7$d13C_CH4~qpcr_nms_mudenv7$Date)
pairwise.wilcox.test(qpcr_nms_mudenv7$d13C_CH4, qpcr_nms_mudenv7$Date,
                    Paired = FALSE, p.adjust = "bonferroni")

kruskal.test(qpcr_nms_mudenv8$d13C_CO2~qpcr_nms_mudenv8$Date)
pairwise.wilcox.test(qpcr_nms_mudenv8$d13C_CO2, qpcr_nms_mudenv8$Date,
                    Paired = FALSE, p.adjust = "bonferroni")
```