Report data processing 2021

Emanuel Mauch

2023-07-14

Introduction

This is a report of what I have done so far regarding the sondes data of 2021. For the analysis, I adapted Moritz' code from his paper (https://doi.org/10.1002/ecy.3371). I have cleaned all the raw datasets, stored them as .txt files (folder "Raw_sonde_2021_2"), merged them with RStudio, processed the merged dataset and plotted some figures for the parameters of interest discussed with Blake and Danina. The code was adapted from Moritz' file "make_data.R", using his "method_packages.R". All the files I've used to make this report are in the "project_21" folder that I have uploaded.

Steps done so far

Step 0: Clean raw datafiles in Excel, store them as .txt files

I cleaned all the raw .csv files with Excel (e.g. reassembling split files, fixing date bugs which lead to wrong .txt file conversions etc.). They can be found in the folder "Raw_sonde_2021_2".

Step 1: set up R-script

```
rm(list= ls())

setwd("~/ZIVI_EAWAG/project_21")

source("~/ZIVI_EAWAG/project_21/Moritz_Luehrig_paper_stuff/methods_packages.R")
```

```
## Lade nötiges Paket: pacman
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
sonde_key = fread( "~/ZIVI_EAWAG/project_21/sondes_key_2021.txt", header=T)
```

Step 2: merge all raw data files into a single big one

```
path = "~/ZIVI_EAWAG/project_21/raw_data/Raw_sonde_2021_2"
filenames <- list.files(path=path, pattern=".txt")
files<-data.table(NULL)

system.time(for(i in filenames){
  filepath <- file.path(path,paste(i,sep=""))
  dummy1<-fread(filepath,
                colClasses = "character" ,
                sep = "\t",
                header = FALSE,
                fill=TRUE,
                encoding ="UTF-8")
  rownumb<-dummy1[V1 %like% "(MM/DD/YYYY)", which=TRUE]+1
  colname<-dummy1[V1 %like% "(MM/DD/YYYY)"]
  colname<-gsub(" ", "",(gsub("[[:punct:]]", "",c(lapply(colname,as.character)))))
  colname<-unlist(strsplit(colname,"\t"))
  dummy2<-dummy1[rownumb:.N]
  setnames(dummy2,colname)
  dummy2[,(colname[c(3,5:length(dummy2))]):=lapply(.SD, as.numeric),.SDcols=colname[c(3,5:length(dummy2
  dummy2$Pond<-sub("_Sept.*", "", i)
  dummy2$Source_file<-i
  files<-rbindlist(list(files, dummy2), fill=TRUE)
})
```

```
##        User      System verstrichen
##        0.80        0.17        4.95
```

```
all <- files

# remove last 3 columns (some variables only measured by few sondes)
all <- all[,!c("Depthm", "Pressurepsia", "VerticalPositionm")]

# replace "µ" with "" in colnames
setnames(all, gsub("µ", "",  names(all)))

# save intermediate stage 1 (FULL RAW DATA, NO REMOVAL)
fwrite(all, "~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_1.txt"
print(head(all,3))
```

```
##    DateMMDDYYYY TimeHHMMSS TimeFractSec   SiteName ChlorophyllRFU
## 1:    9/16/2021   15:09:50            0 greeenland           0.23
## 2:    9/16/2021   15:11:50            0 greeenland           0.12
## 3:    9/16/2021   15:13:50            0 greeenland           0.13
##    ChlorophyllugL ConduScm fDOMQSU fDOMRFU nLFConduScm ODOsat ODOlocal ODOmgL
## 1:           2.21     39.0    7.09    2.28        60.5   95.7     96.9  11.63
## 2:           1.86     38.9    7.16    2.30        60.4   95.6     96.9  11.62
## 3:           1.90     38.9    7.19    2.31        60.4   95.7     97.0  11.63
##    Salpsu SpConduScm BGAPCRFU BGAPCugL TDSmgL WiperPositionvolt   pH  pHmV
## 1:   0.03       59.5    -1.64    -1.64     39             1.218 7.64 -48.7
## 2:   0.03       59.4    -1.73    -1.73     39             1.225 7.63 -48.4
```

```
## 3:   0.03       59.4    -1.73    -1.73     39             1.224 7.61 -47.2
##    TempC BatteryV CablePwrV Pond          Source_file
## 1: 6.942     5.86         0 B1P1 B1P1_Sept24_2021.txt
## 2: 6.951     5.85         0 B1P1 B1P1_Sept24_2021.txt
## 3: 6.959     5.85         0 B1P1 B1P1_Sept24_2021.txt
```

Step 3: add exp. design, phases and format time

```
all <- fread("~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_1.txt

# format date & time
all$TimeHHMMSS<-substr(all$TimeHHMMSS,1,5) #gsub("\\D+", "", cleaning$TimeHHMMSS)
all$DateMMDDYYYY<-mdy(all$DateMMDDYYYY)
all$Date_time<-ymd_hm(paste(all$DateMMDDYYYY, all$TimeHHMMSS))
setnames(all,1:2,c("Date","Time"))

# Cut before and after to obtain the same period for all the ponds
# 2021-09-16 22:00:00 to 2021-09-24 03:00:00
all <- subset(all, Date_time>"2021-09-16 22:00:00") # For some reason: starts at 2021-09-16 20:00:00
all <- subset(all, Date_time<"2021-09-24 05:00:00") # For some reason: ends at 2021-09-24 03:00:00

# add fractions of day since start
all[,Time_seq:=round(((as.numeric(Date_time))-
                        (as.numeric(Date_time)[1]))/86400,7)]

# merge with key
all<-merge(all,sonde_key, by = "Pond")

# reorder and revalue
setorderv(all, c("Pond", "Date_time"), c(1, 1))

# select columns to keep
all<-all[,c("Pond", "Treatment", "Date_time", "Time_seq",
            "ChlorophyllRFU", "ChlorophyllugL", "ConduScm" , "BGAPCRFU",
            "BGAPCugL", "ODOsat", "ODOmgL", "pH", "fDOMRFU", "SpConduScm", "TempC")]

# rename
setnames(all, c("Pond", "Treatment", "Date_time", "Time_seq",
                "Chlorophyll_RFU", "Chlorophyll_ugL", "Cond_uScm", "BGAPC_RFU",
                "BGAPC_ugL", "ODO_sat", "ODO_mgL", "pH", "fDOM_RFU", "SpCond_uScm", "Temp_C"))

# debug: rm duplicate value
all = unique(all, by=c("Pond", "Time_seq"))

fwrite(all, "~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_2.txt"
print(head(all,3))
```

```
##    Pond Treatment           Date_time  Time_seq Chlorophyll_RFU Chlorophyll_ugL
## 1: B1P1         S 2021-09-16 20:01:00 0.0000000            0.26            2.29
## 2: B1P1         S 2021-09-16 20:03:00 0.0013889            0.47            2.96
## 3: B1P1         S 2021-09-16 20:05:00 0.0027778            0.99            4.58
##    Cond_uScm BGAPC_RFU BGAPC_ugL ODO_sat ODO_mgL   pH fDOM_RFU SpCond_uScm
```

```
## 1:      38.9     -1.63     -1.63    96.9   11.78 7.41     2.29        59.4
## 2:      38.9     -1.61     -1.61    96.9   11.78 7.41     2.29        59.4
## 3:      38.9     -1.51     -1.51    97.0   11.79 7.41     2.27        59.4
##    Temp_C
## 1:  6.956
## 2:  6.954
## 3:  6.944
```

Step 4.1: remove outliers

The outlier plots are in the folder "outliers".

```
all <- fread("~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_2.txt

save_path = "outliers/"

for(i in  c("Chlorophyll_RFU", "BGAPC_RFU", "ODO_mgL")){
  all[,  (i) := lapply(.SD[,..i],
                       outlier_removal,
                       window=48,
                       threshold=10,
                       plot=T,
                       return="removed",
                       save=T,
                       path=save_path,
                       name=paste0(i,"_",Pond)),
      by = c("Pond")]
}

for(i in c("fDOM_RFU")){
  all[,  (i) := lapply(.SD[,..i],
                       outlier_removal,
                       window=96,
                       threshold=10,
                       plot=F,
                       return="removed",
                       save=T,
                       path=save_path,
                       name=paste0(i,"_",Pond)),
      by = c("Pond")]
}

for(i in c("Temp_C", "SpCond_uScm", "Cond_uScm")){
  all[,  (i) := lapply(.SD[,..i],
                       outlier_removal,
                       window=96,
                       threshold=20,
                       plot=F,
                       return="removed",
                       save=T,
                       path=save_path,
                       name=paste0(i,"_",Pond)),
```

```
      by = c("Pond")]
}

# save
fwrite(all, "~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_3.1.tx
```

Step 4.2: remove anomalies

We inspected the data for anomalies and remove them here - for details see Russo, S., M. Lührig, W. Hao, B. Matthews, and K. Villez. 2020. Active learning for anomaly detection in environmental data. Environmental Modelling & Software 134:104869.

I didn't do this step, since I would need some time to refresh my python skills, but if you think it is worth, I will manage.

Step 5: add light and precipitation data

I do not have this data available.

Step 6: intercept correction

This uses the results form a cross correlation survey to adjust the sondes for "off factory differences, i.e. remove any intercept variation among the data.

I do not have the sonds correlation factors.

Step 7: save

```
# order and remove rownames
all = all[order(Pond, Time_seq), ]
rownames(all)<-NULL

fwrite(all, "~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")
```

```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")
print(head(all,3))
```

```
##    Pond Treatment           Date_time  Time_seq Chlorophyll_RFU Chlorophyll_ugL
## 1: B1P1         S 2021-09-16 20:01:00 0.0000000            0.26            2.29
## 2: B1P1         S 2021-09-16 20:03:00 0.0013889            0.47            2.96
## 3: B1P1         S 2021-09-16 20:05:00 0.0027778            0.99            4.58
##    Cond_uScm BGAPC_RFU BGAPC_ugL ODO_sat ODO_mgL   pH fDOM_RFU SpCond_uScm
## 1:      38.9     -1.63     -1.63    96.9   11.78 7.41     2.29        59.4
## 2:      38.9     -1.61     -1.61    96.9   11.78 7.41     2.29        59.4
## 3:      38.9     -1.51     -1.51    97.0   11.79 7.41     2.27        59.4
##    Temp_C
## 1:  6.956
## 2:  6.954
## 3:  6.944
```

Step 8: check completeness

```
all <- fread("~/ZIVI_EAWAG/project_21/intermediate_processing_steps/ponds_sonde_data_intermediate_3.1.t
all$Date_time <- as.POSIXct((all$Time_seq*86400), origin="2021-09-16 20:00:00", tz="UTC")
# is that the correct code for as.POSIXTct in our case?

all$Date <-date(all$Date_time)

# N per date
all[, N := uniqueN(Date), by = Pond]
complete <- setDT(data.frame(unclass(table(all$Date, all$Pond))), keep.rownames=T)
colnames(complete)[1]<-"Date"
fwrite(complete, "~/ZIVI_EAWAG/project_21/data/sonde_data_completeness.txt", sep="\t")
# Some major data-gaps for B1P1!

print(complete)
```

```
##          Date B1P1 B1P4 B2P2 B2P3 B2P4 B3P0 B3P1 B3P2 B3P3 ERL122 ERL152 ERL85
## 1: 2021-09-16  120  120  120  120  120  120  120  120  120    120    120   120
## 2: 2021-09-17  720  720  720  720  720  720  720  720  720    720    720   720
## 3: 2021-09-18  720  720  720  720  720  720  720  720  720    635    720   720
## 4: 2021-09-19  720  720  720  720  720  720  720  720  720    720    720   720
## 5: 2021-09-20  720  720  720  720  720  720  720  720  720    720    720   720
## 6: 2021-09-21  415  720  720  720  720  720  720  720  720    720    720   720
## 7: 2021-09-22    0  720  720  720  720  720  720  720  720    720    720   720
## 8: 2021-09-23  253  720  720  720  720  720  720  720  720    720    720   720
## 9: 2021-09-24    0   90   90   89   90   89   89   90   90     89     89    90
```

Step 9: calculate daily averages

Does it make sense to calculate daily averages in our case?

```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")

all[,Time_seq:=as.integer(Time_seq)]

parameters = c("Chlorophyll_RFU", "Chlorophyll_ugL", "Cond_uScm", "BGAPC_RFU", "BGAPC_ugL",
               "ODO_sat", "ODO_mgL", "pH", "fDOM_RFU", "SpCond_uScm",
               "Temp_C")
groups = c("Time_seq", "Pond", "Treatment")

all.summ = all[,lapply(.SD, m_r, 2), by=groups, .SDcols = parameters]

fwrite(all.summ, "~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_daily_avg.txt", sep=",")

table(all.summ$Pond, all.summ$Treatment)
```

```
##
##          D NF S
##   B1P1   0  0 6
##   B1P4   0  8 0
```

```
##   B2P2   0  0 8
##   B2P3   8  0 0
##   B2P4   0  8 0
##   B3P0   0  8 0
##   B3P1   8  0 0
##   B3P2   8  0 0
##   B3P3   0  0 8
##   ERL122 0  8 0
##   ERL152 0  8 0
##   ERL85  0  8 0
```

```
print(head(all.summ,3))
```

```
##    Time_seq Pond Treatment Chlorophyll_RFU Chlorophyll_ugL Cond_uScm BGAPC_RFU
## 1:        0 B1P1         S            0.39            2.80     38.88     -1.59
## 2:        1 B1P1         S            0.45            2.97     39.22     -1.60
## 3:        2 B1P1         S            0.55            3.25     39.11     -1.59
##    BGAPC_ugL ODO_sat ODO_mgL   pH fDOM_RFU SpCond_uScm Temp_C
## 1:     -1.58   96.54   11.73 7.33     2.33       59.40   6.96
## 2:     -1.60   98.07   11.82 7.32     2.26       59.32   7.28
## 3:     -1.59   98.50   11.86 7.31     2.26       59.05   7.32
```

Step 10: hourly averages

```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")

parameters = c("Chlorophyll_RFU", "Cond_uScm", "BGAPC_RFU", "ODO_mgL", "pH",
               "fDOM_RFU", "SpCond_uScm", "Temp_C")

all$Time_seq = as.integer(all$Time_seq)
all$Hour <- as.numeric(substr(as.character(all$Date_time),12,13))

groups = c("Time_seq", "Hour", "Pond", "Treatment")

all.summ = all[,lapply(.SD, m_r, 2), by=groups, .SDcols = parameters]

fwrite(all.summ, "~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_hourly_avg.txt", sep=",")
print(head(all.summ,3))
```

```
##    Time_seq Hour Pond Treatment Chlorophyll_RFU Cond_uScm BGAPC_RFU ODO_mgL
## 1:        0   20 B1P1         S            0.54      38.9     -1.57   11.79
## 2:        0   21 B1P1         S            0.60      38.9     -1.56   11.78
## 3:        0   22 B1P1         S            0.65      38.9     -1.54   11.76
##      pH fDOM_RFU SpCond_uScm Temp_C
## 1: 7.40     2.29        59.4   6.94
## 2: 7.40     2.32        59.4   6.91
## 3: 7.39     2.33        59.4   6.88
```

Step 11: 10-minute averages

I thought that maybe the hour interval is too large for our time series, so maybe a 10-minute interval could be more appropriate (or just plot¨ the absolute values).

```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")

parameters = c("Chlorophyll_RFU", "Chlorophyll_ugL", "Cond_uScm", "BGAPC_RFU", "BGAPC_ugL",
               "ODO_sat", "ODO_mgL", "pH", "fDOM_RFU", "SpCond_uScm",
               "Temp_C")

all$Date_time <- as.POSIXct((all$Time_seq*86400), origin="2021-09-16 20:00:00", tz="UTC")
# is that the correct code for as.POSIXTct in our case?

all$Time_seq = as.integer(all$Time_seq)
all$Hour <- as.numeric(substr(as.character(all$Date_time),12,13))
all$Ten_min <- as.numeric(substr(as.character(all$Date_time),15,15))


groups = c("Time_seq", "Hour", "Ten_min", "Pond", "Treatment")

all.summ = all[,lapply(.SD, m_r, 2), by=groups, .SDcols = parameters]

fwrite(all.summ, "~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_ten_min_avg.txt", sep=",")
print(head(all.summ,3))
```

```
##    Time_seq Hour Ten_min Pond Treatment Chlorophyll_RFU Chlorophyll_ugL
## 1:        0   20       0 B1P1         S            0.66            3.55
## 2:        0   20       1 B1P1         S            0.65            3.52
## 3:        0   20       2 B1P1         S            0.43            2.83
##    Cond_uScm BGAPC_RFU BGAPC_ugL ODO_sat ODO_mgL   pH fDOM_RFU SpCond_uScm
## 1:      38.9     -1.54     -1.54   96.95   11.79 7.40     2.29        59.4
## 2:      38.9     -1.55     -1.55   97.00   11.79 7.41     2.31        59.4
## 3:      38.9     -1.61     -1.61   97.00   11.79 7.40     2.29        59.4
##    Temp_C
## 1:   6.95
## 2:   6.94
## 3:   6.94
```

Step 12: Calculate overall averages of individual ponds

```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")

parameters = c("Chlorophyll_RFU", "Cond_uScm", "BGAPC_RFU", "ODO_mgL", "pH",
               "fDOM_RFU", "SpCond_uScm", "Temp_C")
groups = c("Pond","Treatment")

all.summ = all[,lapply(.SD, m_r, 2), by=groups, .SDcols = parameters]

fwrite(all.summ, "~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_overall_avg.txt", sep=",")

table(all.summ$Pond, all.summ$Treatment)
```

```
##
##          D NF S
##   B1P1   0  0 1
```

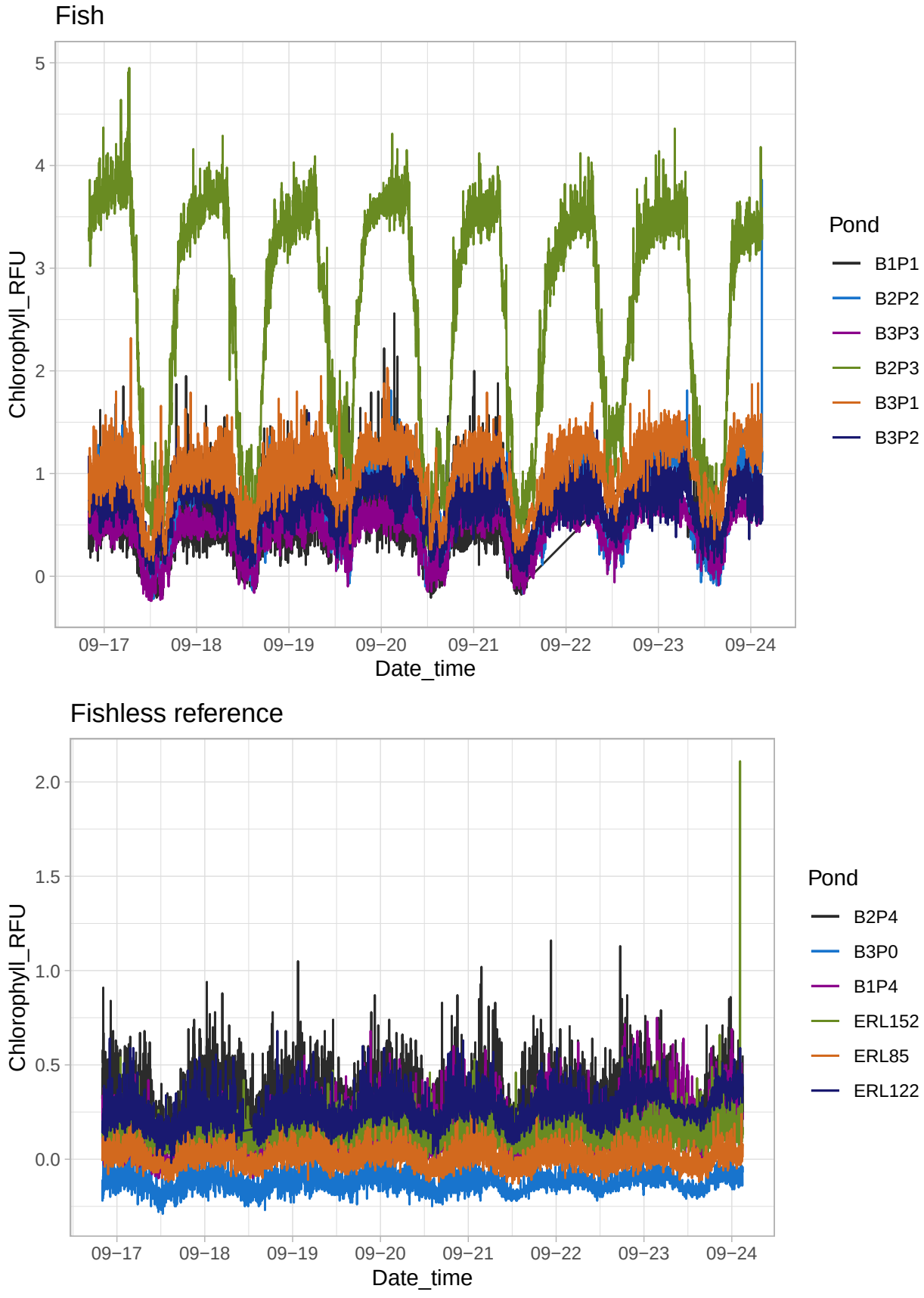
```
##   B1P4   0  1 0
##   B2P2   0  0 1
##   B2P3   1  0 0
##   B2P4   0  1 0
##   B3P0   0  1 0
##   B3P1   1  0 0
##   B3P2   1  0 0
##   B3P3   0  0 1
##   ERL122 0  1 0
##   ERL152 0  1 0
##   ERL85  0  1 0
```

Step 13: control plots

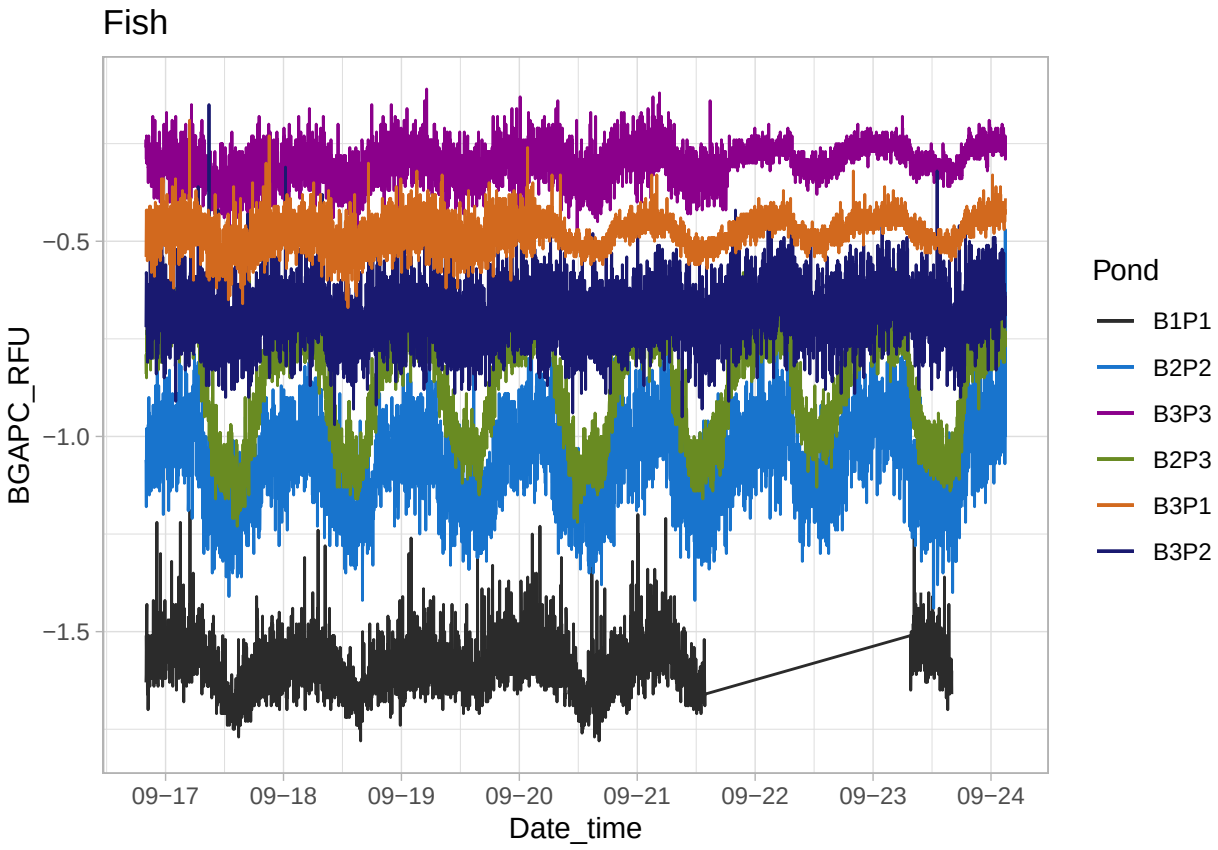
```
all<-fread("~/ZIVI_EAWAG/project_21/data/ponds_sonde_data_all.txt")

all$Date_time <- as.POSIXct((all$Time_seq*86400), origin="2021-09-16 20:00:00", tz="UTC")

table(all$Pond, all$Treatment)
```

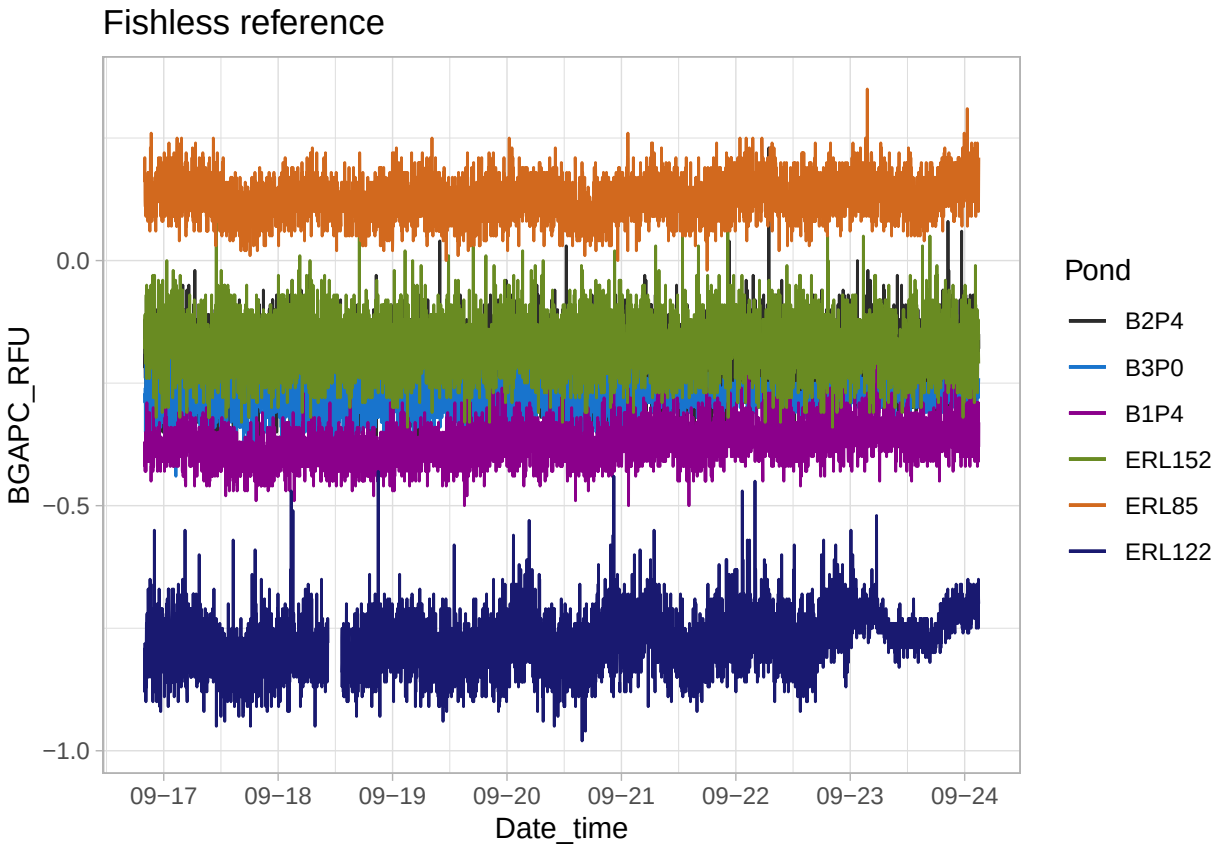
```
##
##             D   NF    S
##   B1P1      0    0 3668
##   B1P4      0 5250    0
##   B2P2      0    0 5250
##   B2P3   5249    0    0
##   B2P4      0 5250    0
##   B3P0      0 5249    0
##   B3P1   5249    0    0
##   B3P2   5250    0    0
##   B3P3      0    0 5250
##   ERL122    0 5164    0
##   ERL152    0 5249    0
##   ERL85     0 5250    0
```

Chlorophyll RFU

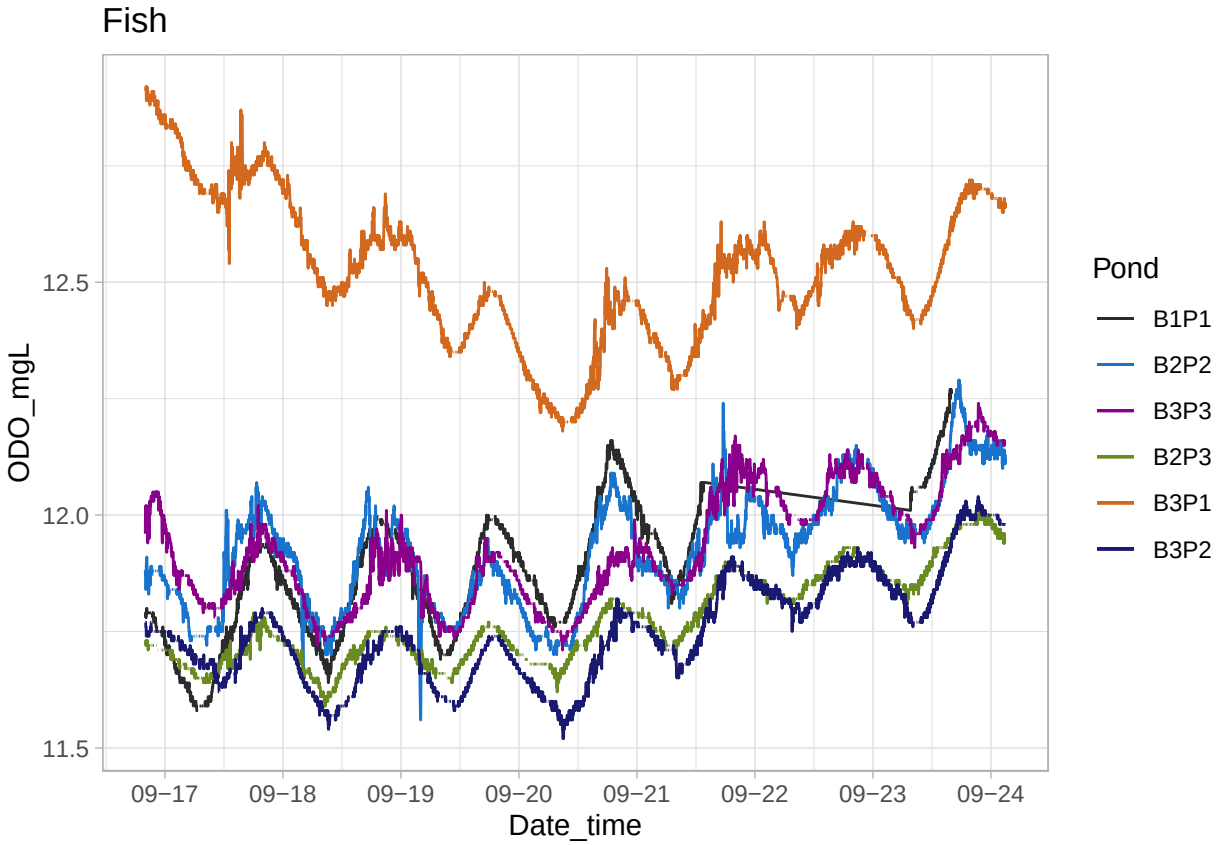


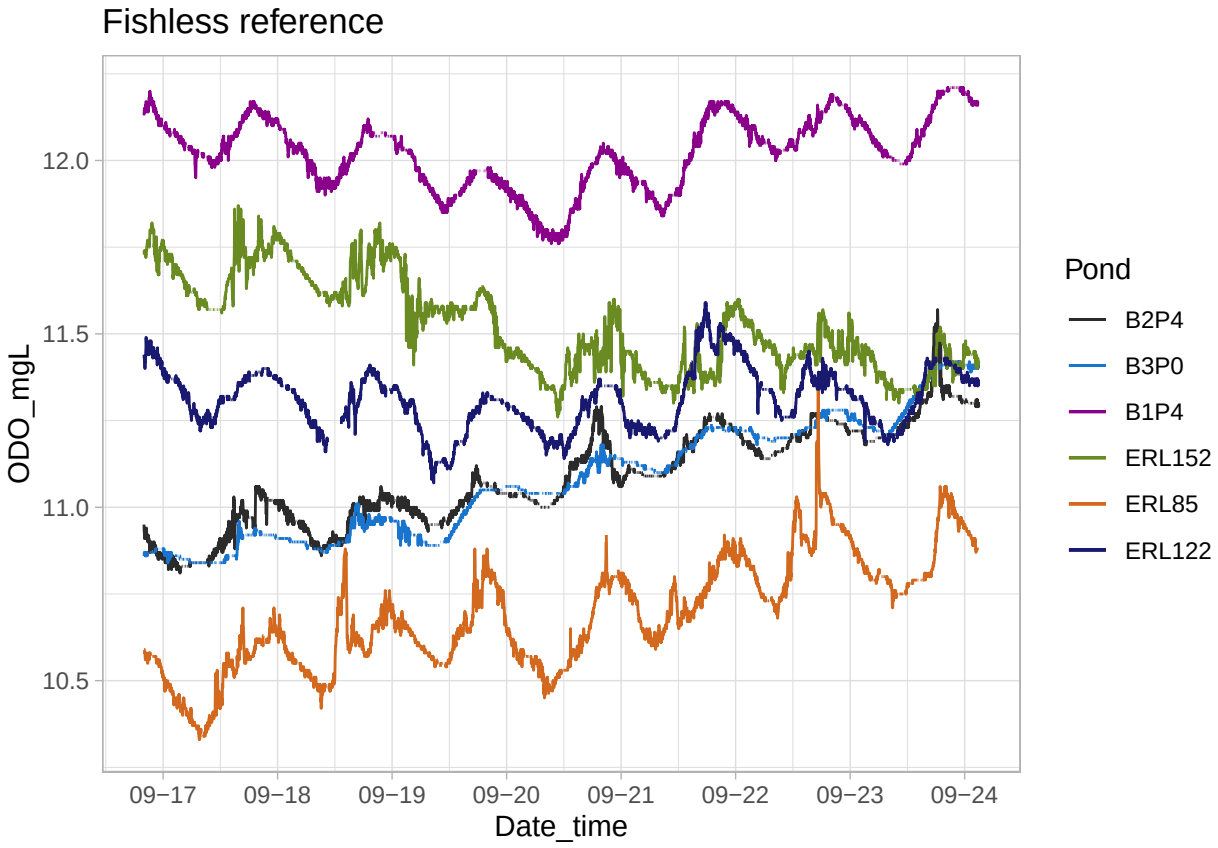
BGAPC RFU


Fish
BGAPC_RFU
−0.5
−1.0
−1.5
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B1P1
B2P2
B3P3
B2P3
B3P1
B3P2

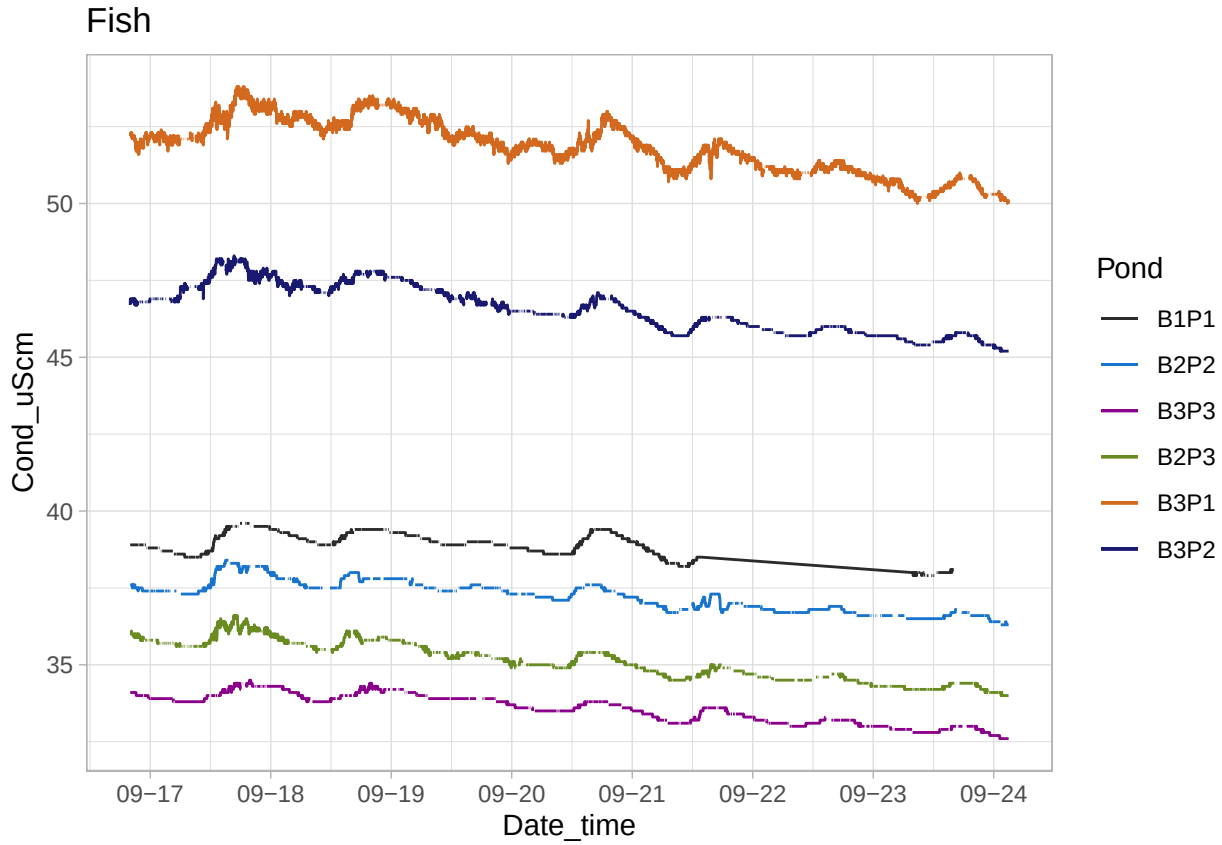

Fishless reference
BGAPC_RFU
0.0
−0.5
−1.0
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B2P4
B3P0
B1P4
ERL152
ERL85
ERL122

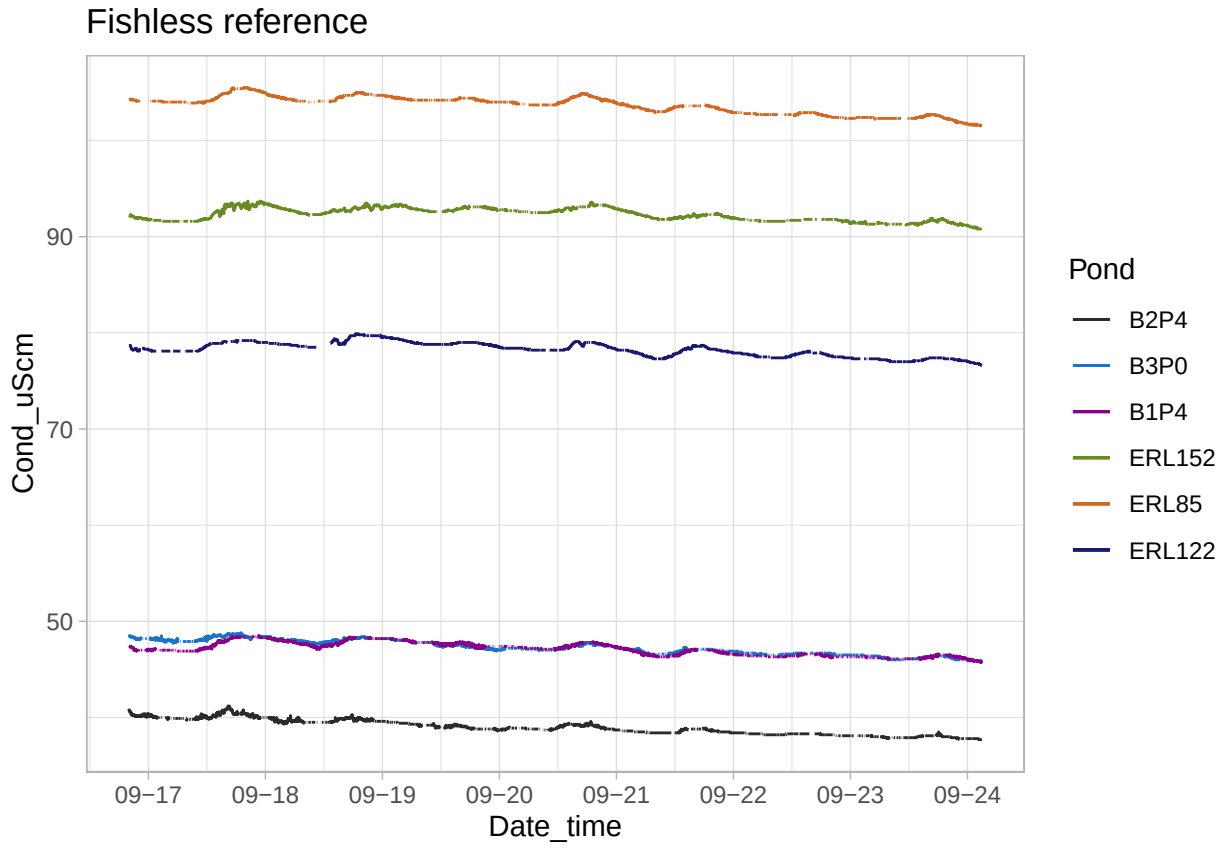
ODO mg/L


Fish
ODO_mgL
12.5
12.0
11.5
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B1P1
B2P2
B3P3
B2P3
B3P1
B3P2

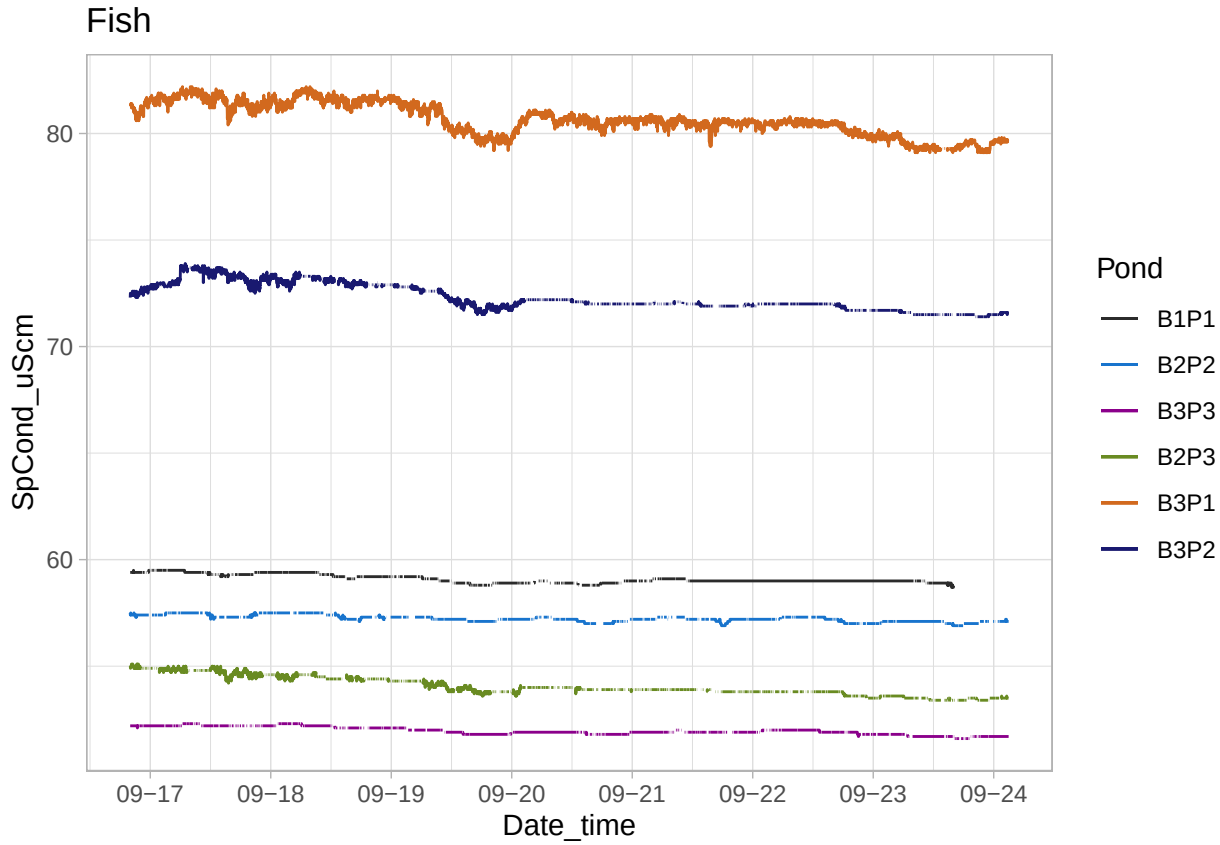

Fishless reference
ODO_mgL
12.0
11.5
11.0
10.5
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B2P4
B3P0
B1P4
ERL152
ERL85
ERL122

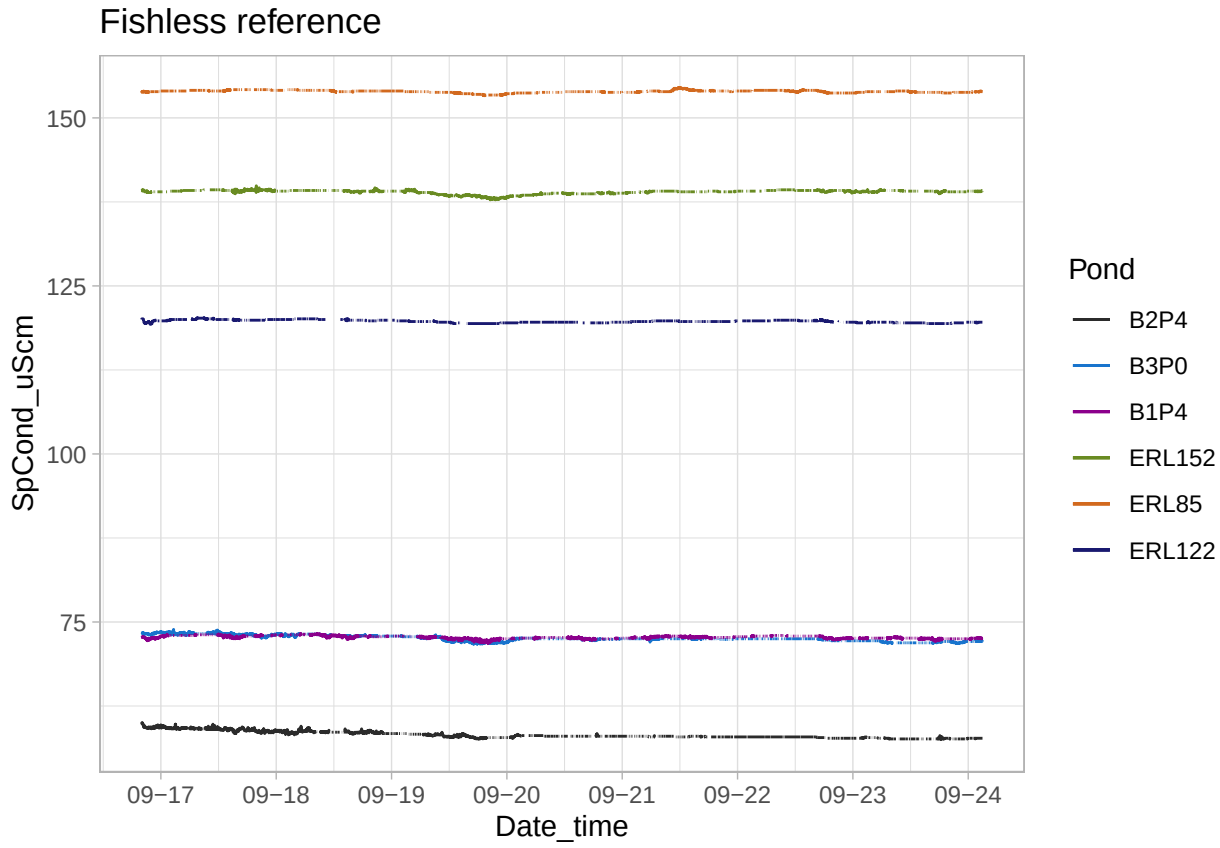
Cond uS/cm


Fish
Cond_uScm
50
45
40
35
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B1P1
B2P2
B3P3
B2P3
B3P1
B3P2

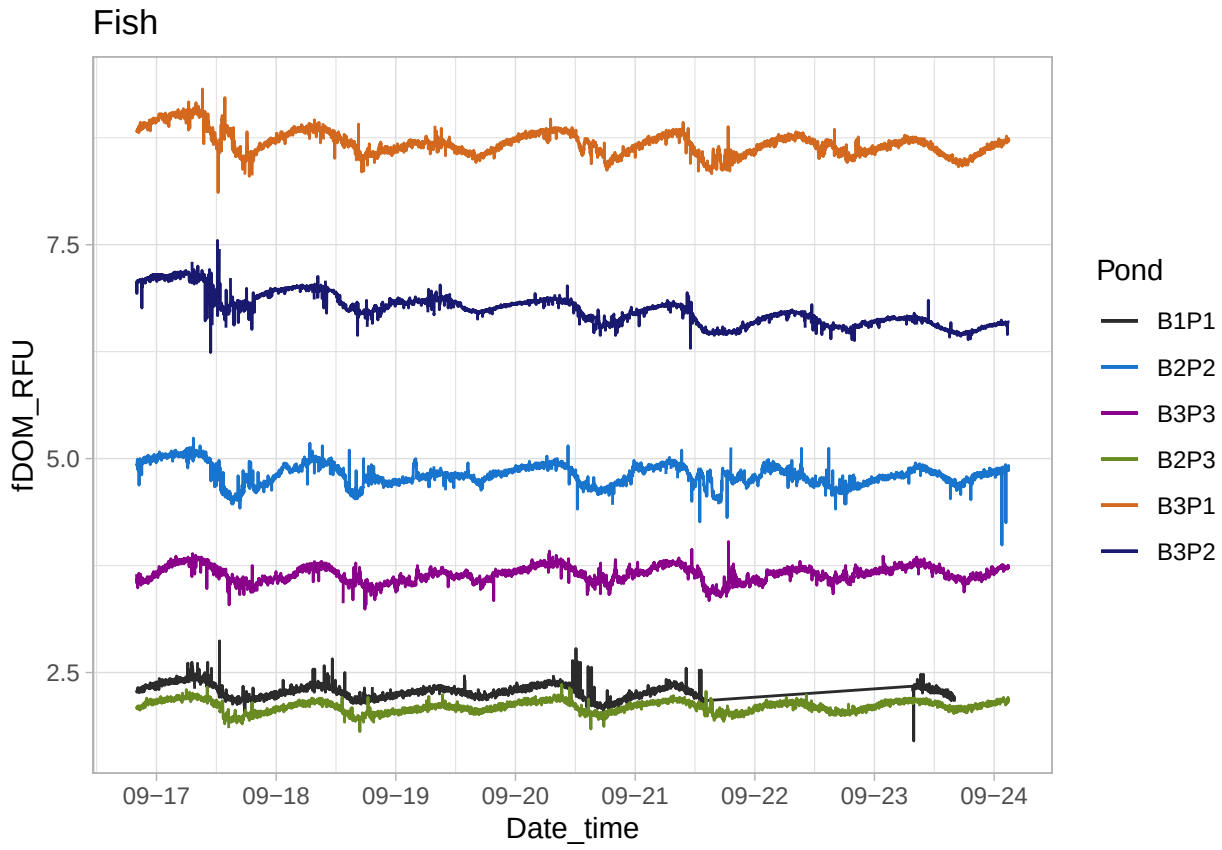

Fishless reference
Cond_uScm
90
70
50
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B2P4
B3P0
B1P4
ERL152
ERL85
ERL122

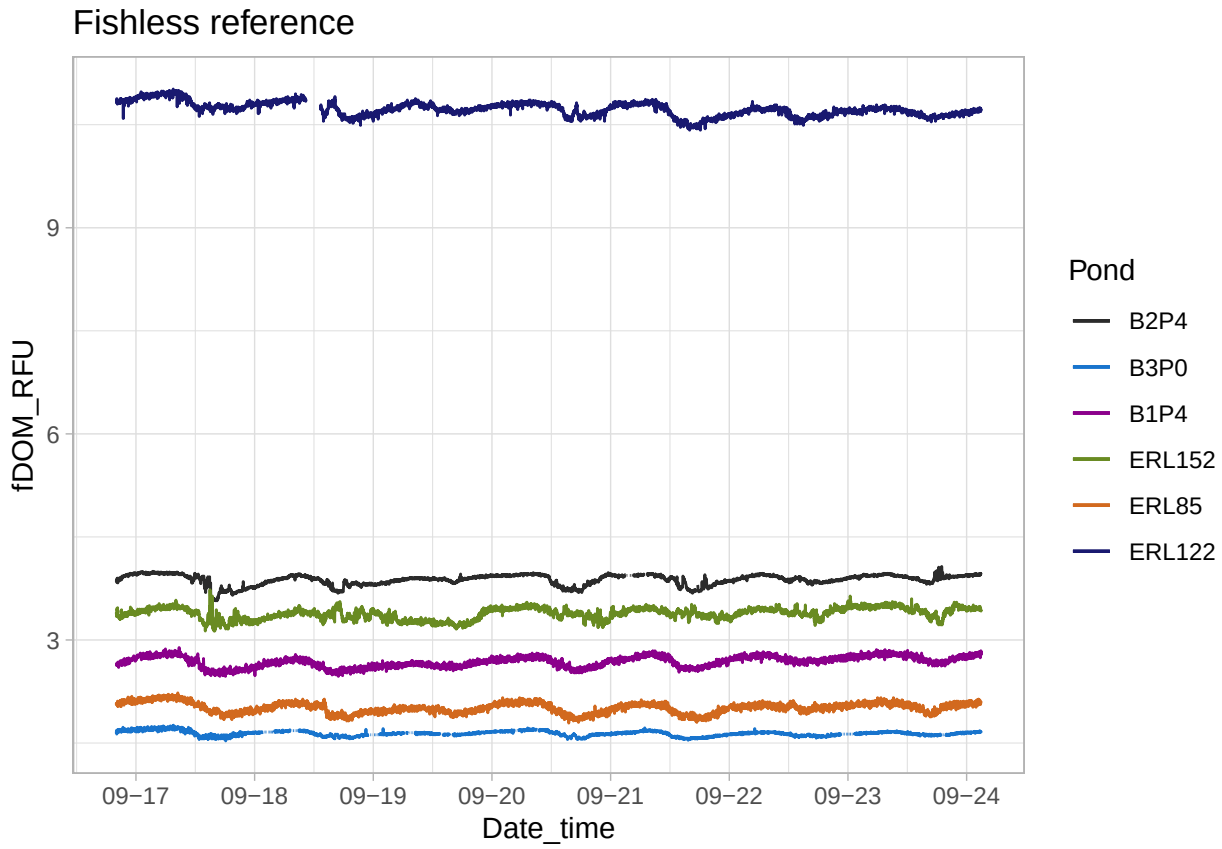
Specific Conductivity uS/cm





fDOM RFU


Fish
fDOM_RFU
7.5
5.0
2.5
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B1P1
B2P2
B3P3
B2P3
B3P1
B3P2


Fishless reference
fDOM_RFU
9
6
3
09−17
09−18
09−19
09−20
09−21
09−22
09−23
09−24
Date_time
Pond
B2P4
B3P0
B1P4
ERL152
ERL85
ERL122

14: Appendix

Version and packages used to generate this report:

```
## 2023-08-07 11:25:08.992352 Europe/Zurich

## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=German_Switzerland.utf8  LC_CTYPE=German_Switzerland.utf8
## [3] LC_MONETARY=German_Switzerland.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Switzerland.utf8
##
## time zone: Europe/Zurich
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] GGally_2.1.2      zoo_1.8-12        viridis_0.6.3     viridisLite_0.4.2
##  [5] forcats_1.0.0     stringr_1.5.0     dplyr_1.1.2       purrr_1.0.1
##  [9] readr_2.1.4       tidyr_1.3.0       tibble_3.2.1      ggplot2_3.4.2
## [13] tidyverse_2.0.0   lubridate_1.9.2   data.table_1.14.8 cowplot_1.1.1
## [17] bit64_4.0.5       bit_4.0.5         pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3         generics_0.1.3     stringi_1.7.12     lattice_0.21-8
##  [5] hms_1.1.3          digest_0.6.32      magrittr_2.0.3     RColorBrewer_1.1-3
##  [9] evaluate_0.21      grid_4.3.1         timechange_0.2.0   fastmap_1.1.1
## [13] plyr_1.8.8         reshape_0.8.9      gridExtra_2.3      fansi_1.0.4
## [17] scales_1.2.1       cli_3.6.1          rlang_1.1.1        munsell_0.5.0
## [21] withr_2.5.0        yaml_2.3.7         tools_4.3.1        tzdb_0.4.0
## [25] colorspace_2.1-0   vctrs_0.6.3        R6_2.5.1           lifecycle_1.0.3
## [29] pkgconfig_2.0.3    pillar_1.9.0       gtable_0.3.3       Rcpp_1.0.10
## [33] glue_1.6.2         highr_0.10         xfun_0.39          tidyselect_1.2.0
## [37] rstudioapi_0.14    knitr_1.43         farver_2.1.1       htmltools_0.5.5
## [41] labeling_0.4.2     rmarkdown_2.23     compiler_4.3.1
```

Code used to generate control plots:

```
# Chlorophyll RFU
p1 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
```

```
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p2 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=Chlorophyll_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p1, p2, nrow = 2, rel_heights =  c(0.229,0.2,0.2,0.32))

# BGAPC RFU
p3 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p4 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=BGAPC_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p3, p4, nrow = 2)
```

```
# ODO mg/L
p5 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p6 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=ODO_mgL, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p5, p6, nrow = 2)

# Cond uS/cm
p7 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p8 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=Cond_uScm, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
```

```
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p7, p8, nrow = 2)

# Specific Cond uS/cm
p9 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p10 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=SpCond_uScm, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p9, p10, nrow = 2)

# fDOM RFU
p11 = ggplot(subset(all, Pond=="B1P1")) + theme_light() +
  geom_line(aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P2"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P3"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B2P3"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P2"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P1"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B1P1', 'B2P2', 'B3P3', "B2P3", "B3P1", "B3P2"),
                     values=c('B1P1'="gray17", 'B2P2'='dodgerblue3', 'B3P3'='magenta4',
                              "B2P3"="olivedrab4", "B3P1"="chocolate", "B3P2"="midnightblue")) +
  ggtitle("Fish")

p12 = ggplot(subset(all, Pond=="B2P4")) + theme_light() +
  geom_line(aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B3P0"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="B1P4"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL152"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
```

```
  geom_line(data = subset(all, Pond=="ERL85"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  geom_line(data = subset(all, Pond=="ERL122"), aes(y=fDOM_RFU, x=Date_time, color = Pond)) +
  scale_x_datetime(date_breaks = "1 day", date_labels = "%m-%d") +
  scale_color_manual(breaks=c('B2P4', 'B3P0', 'B1P4', "ERL152", "ERL85", "ERL122"),
                     values=c('B2P4'="gray17", 'B3P0'='dodgerblue3', 'B1P4'='magenta4',
                              "ERL152"="olivedrab4", "ERL85"="chocolate", "ERL122"="midnightblue")) +
  ggtitle("Fishless reference")

plot_grid(p11, p12, nrow = 2)
```