Analysis Metabolism Pond B3P3 2022

Emanuel Mauch

2023-07-26

Introduction

The following analysis attempts to model ecosystem metabolism of pond B3P3 in 2022 based on dissolved oxygen measurements (do). The analysis is made in the context of the Stickelback experiments of Blakes group in Greenland and is based on the paper "LakeMetabolizer: an R package for estimating lake metabolism from free-water oxygen using diverse statistical models" by Winslow et al. (2016). The analysis of B3P3 is meant to serve as a possible blueprint for other ponds in the future. Additionally metadata was downloaded from two different webpages, which are described in the section "Data sources". This report is deliberatly made as transparent and reproducible as possible to maintain verifyable for possible readers.

Step 0: set up R-script

```
rm(list= ls())

setwd("~/ZIVI_EAWAG/project_22")
source("~/ZIVI_EAWAG/project_22/Moritz_Luehrig_paper_stuff/methods_packages.R")

# Estimating lake metabolism with free-water oxygen
library(LakeMetabolizer)

#demo('fig_metab', package='LakeMetabolizer')

# For plotting
library(GGally)

# Nonparametric Missing Value Imputation using Random Forest
# library(missForest)

# This is our complete dataset of the sondes measurements 2022
# Measurement Interval: 2 minutes
all <- fread("~/ZIVI_EAWAG/project_22/data/ponds_sonde_data_all_uncut.txt")

# This is the wind dataset
# Measurement Interval: 1 hour
wind <- fread("~/ZIVI_EAWAG/project_22/data_metabolism_wind/wind_narsarsuaq_radiosonde_2022_complete.tx
# The wind variable is the average hourly wind speed in m/s, measured with
# a radiosonde in Narsarsuaq, Greenland
```

Step 1: data formatting

```
# Select columns to keep from our time series
all <- all[, c("Pond", "Date_time", "ODO_mgL", "Temp_C")]

# Select Pond B3P3 (Treatment: Sticklebacks from single population; source: pond L26)
all <- subset(all, Pond == "B3P3")

# Rename
all <- setnames(all,c("Pond", "datetime", "do.obs", "wtr"))

# Isolate the desired period (described in the introduction)
all <- all[c(255:7455),]

# Ommit last row (not included in our timeseries)
all <- all[-nrow(all), ]

# format the wind dataset
wind$datetime <- as.POSIXct(wind$datetime, format = "%d.%m.%Y %H:%M")

wind <- wind %>%
  mutate(datetime = as.POSIXct(format(datetime, format = "%Y-%m-%d %H:01:00")))

# Create a new inflated dataset with wind measurements every 2 minutes
# The hourly mean will be assigned to every observation (2min-interval) within that hour
wind_ext <- wind %>%
  complete(datetime = seq(from = min(datetime), to = max(datetime), by = "2 min"))

# Fill the missing Wind_speed_ms values with the corresponding hourly values using 'fill'
wind_ext <- wind_ext %>%
  fill(wsp, .direction = "up")

# Shift Wind_speed_ms one row up using 'lag'
wind_ext <- wind_ext %>%
  mutate(wsp = lead(wsp, default = last(wsp)))

# Ommit last row (not included in our timeseries)
wind_ext <- wind_ext[-nrow(wind_ext), ]

# Reset row names for the final result
rownames(wind_ext) <- NULL

head(wind_ext,3)
```

```
## # A tibble: 3 x 2
##   datetime              wsp
##   <dttm>              <dbl>
## 1 2022-06-23 00:01:00   6.2
## 2 2022-06-23 00:03:00   6.2
## 3 2022-06-23 00:05:00   6.2
```

Step 2: merge sonde and wind dataset

```
all <- merge(wind_ext, all, by = "datetime")

# Save
fwrite(all, "~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_B3P3_2022.txt", sep="\t")

head(all,3)
```

```
##              datetime wsp Pond do.obs    wtr
## 1 2022-06-23 00:01:00 6.2 B3P3  10.83 13.188
## 2 2022-06-23 00:03:00 6.2 B3P3  10.82 13.171
## 3 2022-06-23 00:05:00 6.2 B3P3  10.83 13.180
```

Step 3: Missing data imputation

We need a complete ODO_mgL timeseries. Therefore we will impute the missing values here, with the package "missForest" (Nonparametric Missing Value Imputation using Random Forest).

Update: since ODO_mgL of B3P3 is complete, we don't need to do that here But for other ponds, we might need this code.

```
#####
# all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_B3P3_2022.txt", sep="\t")
#
# # Correct shifted Date_time
# Date_time_col <- all$Date_time
# Shift_date_time <- Date_time_col + hours(2)
# all$Date_time <- Shift_date_time

# # Prepare for imputation
# all$Pond <- factor(all$Pond)
# all_temp <- all[, c("Wind_speed_ms", "Pond", "ODO_mgL", "Temp_C")]
#
# # Impute
# imp_all_temp <- missForest(all_temp, verbose = T, variablewise = T)
# # Retrieve imputed dataset
# imp_all_temp <- imp_all_temp$ximp
#
# # Append imputed variable to all
# all$ODO_mgL_imp <- imp_all_temp$ODO_mgL
#
# colSums(is.na(all))
#
# fwrite(all,"~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_imp_wind_B3P3_2022.txt", sep="\t")
#####
```

Step 4: Model choice

Gas transfer coefficient model: kcole() (see paper linked in the introduction, https://doi.org/10.1080/IW-6.4.883)

- only needs wind speed and sensor height (Assumption: wind speed is measured at 10m, since the data are measured by a radiosonde)

Lake metabolism model: I'll try all the models mentioned in the paper:

- Bookkeeping
- Estimation via Ordinary Least Squares (OLS)
- via Maximum Likelihood Estimation (MLE)
- via Maximum Likelihood Estimation + Kalman filter
- Bayesian approach

Step 5: Calculate k600 using the Cole method

```
all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_B3P3_2022.txt", sep="\t")

# Correct shifted Date_time
Date_time_col <- all$datetime
Shift_date_time <- Date_time_col + hours(2)
all$datetime <- Shift_date_time

# This function needs specific column names
all$wnd <- all$wsp
all$wsp <- NULL

# Also only the 2 columns of interest, otherwise it gets confused
all.600 <- all[,c("datetime","wnd")]
k600.cole <- k.cole(all.600)

# Merge
all <- merge(all, k600.cole, by = "datetime")
head(all,3)
```

```
##                datetime Pond do.obs    wtr wnd     k600
## 1: 2022-06-23 00:01:00 B3P3  10.83 13.188 6.2 1.644202
## 2: 2022-06-23 00:03:00 B3P3  10.82 13.171 6.2 1.644202
## 3: 2022-06-23 00:05:00 B3P3  10.83 13.180 6.2 1.644202
```

Step 6: Calculate gas-specific k600

```
kgas <- k600.2.kGAS(all)

# Merge
all <- merge(all, kgas, by = "datetime")

# only keep k.gas
all$k600 <- NULL

head(all,3)
```

```
##               datetime Pond do.obs    wtr wnd    k.gas
## 1: 2022-06-23 00:01:00 B3P3  10.83 13.188 6.2 1.464477
## 2: 2022-06-23 00:03:00 B3P3  10.82 13.171 6.2 1.463805
## 3: 2022-06-23 00:05:00 B3P3  10.83 13.180 6.2 1.464161
```

Step 7: Calculate dissolved oxygen saturated (do.sat)

```
# Instead of the sonde-included calculations, I'll use a function
# provided by the authors.

o2.sat <- o2.at.sat(all[,c('datetime','wtr')])

# Merge
all <- merge(all, o2.sat, by = "datetime")

head(all,3)
```

```
##               datetime Pond do.obs    wtr wnd    k.gas   do.sat
## 1: 2022-06-23 00:01:00 B3P3  10.83 13.188 6.2 1.464477 10.49129
## 2: 2022-06-23 00:03:00 B3P3  10.82 13.171 6.2 1.463805 10.49526
## 3: 2022-06-23 00:05:00 B3P3  10.83 13.180 6.2 1.464161 10.49316
```

```
fwrite(all,"~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param_B3P3_2022.txt", sep="\t")
```

Step 8: Calculate Actively mixed layer depth (z.mix)

Pond B3P3 is about 5 meters deep According to the profiling plots, B3P3 had an actively mixed layer depth of only 0.5 m on 2022-06-21. However, wind speed was low at that time. Since Blake told me that I can assume an average mixed layer depth of about 2-3 meters, if a pond has a max depth of 5 meters, I am going to take 2.5m for the actively mixed layer depth. Of course, this is a simplification, and to do it perfectly, one would need to do a temperature time-series of different layers of the same pond. This could of course be also incorporated in the following models.

```
all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param_B3P3_2022.txt", sep="\t")

# Set Actively mixed layer depth to 2.5m
all$z.mix <- 2.5
head(all,3)
```

```
##               datetime Pond do.obs    wtr wnd    k.gas   do.sat z.mix
## 1: 2022-06-23 00:01:00 B3P3  10.83 13.188 6.2 1.464477 10.49129   2.5
## 2: 2022-06-23 00:03:00 B3P3  10.82 13.171 6.2 1.463805 10.49526   2.5
## 3: 2022-06-23 00:05:00 B3P3  10.83 13.180 6.2 1.464161 10.49316   2.5
```

```
# Save
fwrite(all,"~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param2_B3P3_2022.txt", sep="\t")
```

Step 9: Derive photosynthetically active radiation (par) from shortwave irradiation (sw)

Here, I take the downwelling shortwave irradiation (tilt corrected) from the QAS_L automated weather station a couple of kilometers away from Narsarsuaq It's the closest weather station measuring this parameter, that I could find.

```
# Import irradiation dataset
ir <- fread("~/ZIVI_EAWAG/project_22/data_metabolism_wind/data_down_shortwave_irr_tilt_corr_2022_comple

# Format datetime
ir$datetime <- as.POSIXct(ir$datetime, format = "%d.%m.%Y %H:%M")

# Align it with the observations of the sonde (always 01, 03, 05, etc.)
ir <- ir %>%
  mutate(datetime = as.POSIXct(format(datetime, format = "%Y-%m-%d %H:01:00")))

# Create a new inflated dataset with wind measurements every 2 minutes
# The hourly mean will be assigned to every observation (2min-interval) within that hour
ir_ext <- ir %>%
  complete(datetime = seq(from = min(datetime), to = max(datetime), by = "2 min"))

# Fill the missing dsr_cor values with the corresponding hourly values using 'fill'
ir_ext <- ir_ext %>%
  fill(dsr_cor, .direction = "up")

# Shift dsr_cor one row up using 'lag'
ir_ext <- ir_ext %>%
  mutate(dsr_cor = lead(dsr_cor, default = last(dsr_cor)))

# Ommit last row (not included in our timeseries)
ir_ext <- ir_ext[-nrow(ir_ext), ]

# Reset row names for the final result
rownames(ir_ext) <- NULL
head(ir_ext,3)
```

```
## # A tibble: 3 x 2
##   datetime            dsr_cor
##   <dttm>                <dbl>
## 1 2022-06-23 00:01:00   0.127
## 2 2022-06-23 00:03:00   0.127
## 3 2022-06-23 00:05:00   0.127
```

```
# Derive par from sw
par <- sw.to.par(ir_ext, sw.col = "dsr_cor")
rownames(par) <- NULL
head(par,3)
```

```
## # A tibble: 3 x 2
##   datetime              par
##   <dttm>              <dbl>
```

```
## 1 2022-06-23 00:01:00 0.268
## 2 2022-06-23 00:03:00 0.268
## 3 2022-06-23 00:05:00 0.268
```

```
# Specify correct timezone
# par$datetime <- as.POSIXct(par$datetime, tz = "UTC")

# Correct shifted datetime col
Date_time_col <- par$datetime
Shift_date_time <- Date_time_col + hours(2)
par$datetime <- Shift_date_time

# Merge with our metabolism dataset
# Import
all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param2_B3P3_2022.txt", sep="\t")

# Merge
all <- merge(all, par, by = "datetime")
head(all,3)
```

```
##                datetime Pond do.obs    wtr wnd    k.gas   do.sat z.mix       par
## 1: 2022-06-23 00:01:00 B3P3  10.83 13.188 6.2 1.464477 10.49129   2.5 0.2680552
## 2: 2022-06-23 00:03:00 B3P3  10.82 13.171 6.2 1.463805 10.49526   2.5 0.2680552
## 3: 2022-06-23 00:05:00 B3P3  10.83 13.180 6.2 1.464161 10.49316   2.5 0.2680552
```

```
# Save
fwrite(all,"~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param3_B3P3_2022.txt", sep="\t")
```

Step 10: Fit models

```
# Now lets read in our prepared metabolism dataset
all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param3_B3P3_2022.txt", sep="\t")

# Convert z.mix_m
all$z.mix <- as.numeric(all$z.mix)

# Choose columns to keep
all <- all[,c("datetime", "do.obs", "do.sat", "k.gas", "z.mix", "par", "wtr")]
head(all,3)
```

```
##                datetime do.obs   do.sat    k.gas z.mix       par    wtr
## 1: 2022-06-23 00:01:00  10.83 10.49129 1.464477   2.5 0.2680552 13.188
## 2: 2022-06-23 00:03:00  10.82 10.49526 1.463805   2.5 0.2680552 13.171
## 3: 2022-06-23 00:05:00  10.83 10.49316 1.464161   2.5 0.2680552 13.180
```

```
# Save again
fwrite(all,"~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param4_B3P3_2022.txt", sep="\t")
```

```
# bookkeep model:
# for some reason, it dosen't run yet
bookkeep.res <- metab(all, method = "bookkeep", wtr.name='wtr', do.obs.name='do.obs',
                      irr.name="par", z.mix = "z.mix")
bookkeep.res <- metab(all,irr.name="par")

# OLS model:
ols.res <- metab(all, method = "ols", wtr.name='wtr', do.obs.name='do.obs',
                      irr.name="par", z.mix = "z.mix")
fwrite(ols.res, "~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_ols_2022.txt", sep="\t")

# MLE model:
mle.res <- metab(all, method = "mle", wtr.name='wtr', do.obs.name='do.obs',
                 irr.name="par", z.mix = "z.mix")
fwrite(mle.res, "~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_mle_2022.txt", sep="\t")

# MLE + Kalman filter:
kalman.res <- metab(all, method='kalman', wtr.name='wtr', do.obs.name='do.obs',
                    irr.name='par', z.mix = "z.mix")
fwrite(kalman.res, "~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_kalman_2022.txt", sep="\t")

# Bayesian
bayesian.res <- metab(all, method='bayesian', wtr.name='wtr', do.obs.name='do.obs',
                      irr.name='par', z.mix = "z.mix")
fwrite(bayesian.res, "~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_bayesian_2022.txt", sep="\t")
```

Step 11: Visualize the results

Plots of ecosystem parameters

Plots of metabolism

```
ols.res <- fread("~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_ols_2022.txt", sep="\t")
mle.res <- fread("~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_mle_2022.txt", sep="\t")
kalman.res <- fread("~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_kalman_2022.txt", sep="\t")
bayesian.res <- fread("~/ZIVI_EAWAG/project_22/metabolism_results/B3P3_bayesian_2022.txt", sep="\t")

# Modify result datasets to have a column specifiying the model used
ols.res$Model <- "OLS"
mle.res$Model <- "MLE"
kalman.res$Model <- "MLE + Kalman filter"
bayesian.res$Model <- "Bayesian"

# Merge the 4 result datasets from the 4 different models
merged.res <- rbind(ols.res, mle.res, kalman.res, bayesian.res)

# Create a date variable
merged.res$Date <- as.Date(paste(merged.res$year, "-", merged.res$doy, sep = ""), format = "%Y-%j")

# Select columns to keep
```

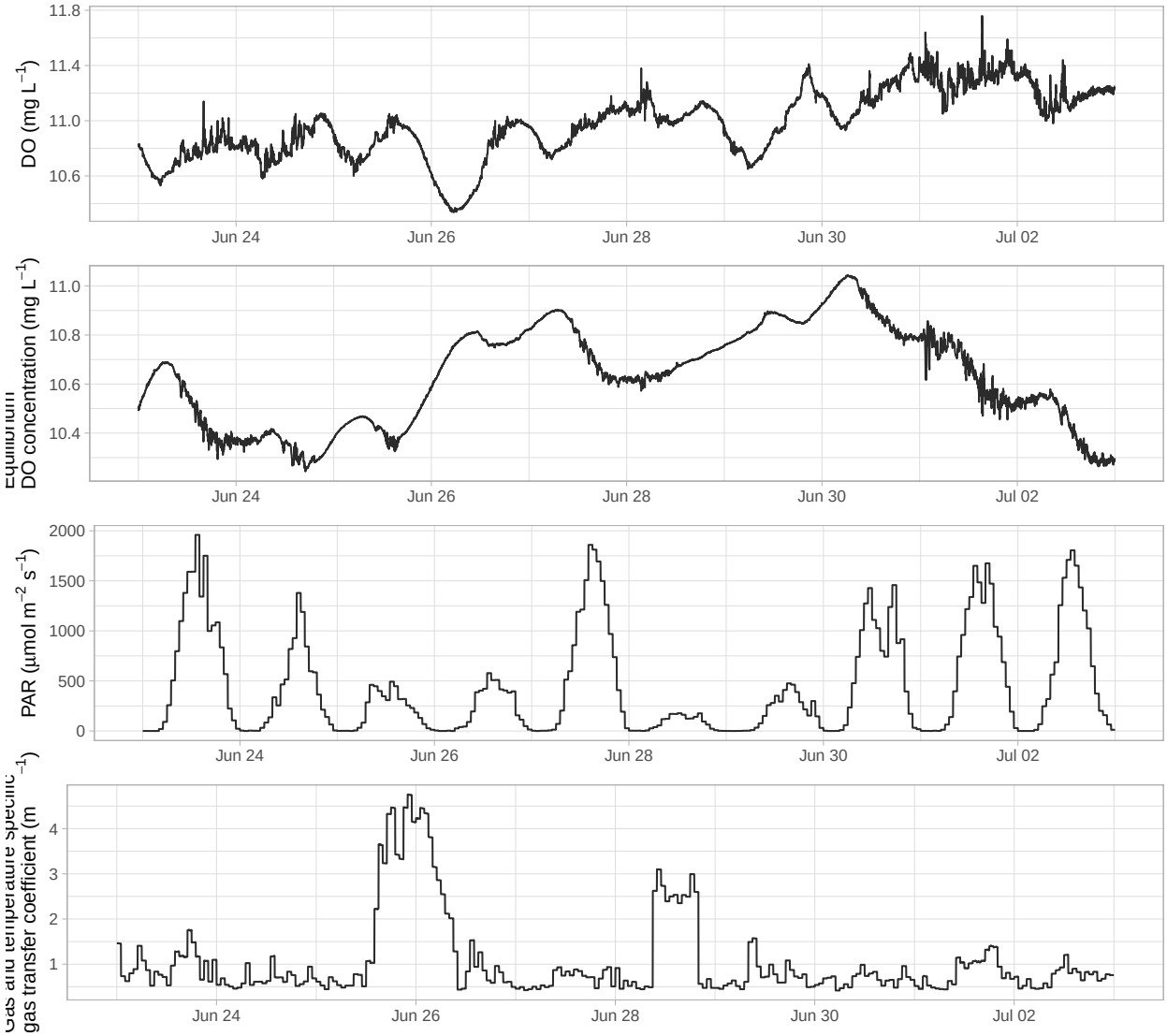

Figure 1: Plots of ecosystem parameters used to predict metabolism of B3P3.

```r
merged.res <- merged.res[,c("Date", "GPP", "R", "NEP", "Model")]

# Save
fwrite(merged.res, "~/ZIVI_EAWAG/project_22/metabolism_results/merged_results_B3P3_2022.txt", sep="\t")
```

Figure 2: estimated metabolism of pond B3P3 during a period of 9 days, estimated with 4 different models.

After talking to Blake, we suggest to take the Bayesian approach to model metabolism, since its the most flexible approach, allowing us to yield the maximum amount of information out of our data.

According to the Bayesian model, B3P3 is productive during the whole time-span (NEP > 0).

12: Data Sources

- Wind data: https://www.dmi.dk/lokationarkiv/show/GL/3421711/Narsarsuaq/#arkiv
- Irradiation data: https://dataverse.geus.dk/dataset.xhtml?persistentId=doi:10.22008/FK2/IW73UU

13: Appendix

Version and packages used to generate this report:

```
## 2023-08-03 16:03:06.385525 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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggpubr_0.6.0           GGally_2.1.2           LakeMetabolizer_1.5.5
##  [4] rLakeAnalyzer_1.11.4.1 zoo_1.8-12             viridis_0.6.3
##  [7] viridisLite_0.4.2      forcats_1.0.0          stringr_1.5.0
## [10] dplyr_1.1.2            purrr_1.0.1            readr_2.1.4
## [13] tidyr_1.3.0            tibble_3.2.1           ggplot2_3.4.2
## [16] tidyverse_2.0.0        lubridate_1.9.2        data.table_1.14.8
## [19] cowplot_1.1.1          bit64_4.0.5            bit_4.0.5
## [22] pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3         generics_0.1.3     rstatix_0.7.2      stringi_1.7.12
##  [5] lattice_0.21-8     hms_1.1.3          digest_0.6.32      magrittr_2.0.3
##  [9] RColorBrewer_1.1-3 evaluate_0.21      timechange_0.2.0   fastmap_1.1.1
## [13] plyr_1.8.8         backports_1.4.1    reshape_0.8.9      gridExtra_2.3
## [17] fansi_1.0.4        scales_1.2.1       abind_1.4-5        cli_3.6.1
## [21] rlang_1.1.1        munsell_0.5.0      withr_2.5.0        yaml_2.3.7
## [25] tools_4.3.1        tzdb_0.4.0         ggsignif_0.6.4     colorspace_2.1-0
## [29] broom_1.0.5        vctrs_0.6.3        R6_2.5.1           lifecycle_1.0.3
## [33] car_3.1-2          pkgconfig_2.0.3    pillar_1.9.0       gtable_0.3.3
## [37] Rcpp_1.0.10        glue_1.6.2         xfun_0.39          tidyselect_1.2.0
## [41] rstudioapi_0.14    knitr_1.43         farver_2.1.1       htmltools_0.5.5
```

```
## [45] labeling_0.4.2     carData_3.0-5      rmarkdown_2.23     compiler_4.3.1
```

Code used to generate plots:

```
# Plots of ecosystem parameters
all <- fread("~/ZIVI_EAWAG/project_22/data_metabolism/merged_ODO_wind_param3_B3P3_2022.txt", sep="\t")

p4 = ggplot((all)) + theme_light() +
  geom_line(aes(y=do.obs, x=datetime), color = "gray17") +
  rremove("xlab") +
  ylab(expression(paste("DO (mg L",NULL^-1,")")))

p5 = ggplot((all)) + theme_light() +
  geom_line(aes(y=do.sat, x=datetime), color = "gray17") +
  rremove("xlab") +
  ylab(expression(paste("Equilibrium\nDO concentration (mg L",NULL^-1,")")))

p6 = ggplot((all)) + theme_light() +
  geom_line(aes(y=par, x=datetime), color = "gray17") +
  rremove("xlab") +
  ylab(expression(paste("PAR (",mu,"mol m",NULL^-2," s",NULL^-1,")")))

p7 = ggplot((all)) + theme_light() +
  geom_line(aes(y=k.gas, x=datetime), color = "gray17") +
  rremove("xlab") +
  ylab(expression(paste("Gas and temperature specific\ngas transfer coefficient (m",NULL^-1,")")))

fig2 <- ggarrange(p4, p5, p6, p7,
                  nrow=5)

# Plots of metabolism
# GPP
p1 = ggplot(subset(merged.res, Model=="OLS")) + theme_light() +
  geom_line(aes(y=GPP, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE"), aes(y=GPP, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE + Kalman filter"), aes(y=GPP, x=Date, color = Model),l
  geom_line(data = subset(merged.res, Model=="Bayesian"), aes(y=GPP, x=Date, color = Model),lwd = 1.25)
  ylab(expression(paste("GPP (", O[2], " in mg L", NULL^-1, " day", NULL^-1, ")")))

# R
p2 = ggplot(subset(merged.res, Model=="OLS")) + theme_light() +
  geom_line(aes(y=R, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE"), aes(y=R, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE + Kalman filter"), aes(y=R, x=Date, color = Model),lwd
  geom_line(data = subset(merged.res, Model=="Bayesian"), aes(y=R, x=Date, color = Model),lwd = 1.25) +
  ylab(expression(paste("R (", O[2], " in mg L", NULL^-1, " day", NULL^-1, ")")))

# NEP
p3 = ggplot(subset(merged.res, Model=="OLS")) + theme_light() +
  geom_line(aes(y=NEP, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE"), aes(y=NEP, x=Date, color = Model),lwd = 1.25) +
  geom_line(data = subset(merged.res, Model=="MLE + Kalman filter"), aes(y=NEP, x=Date, color = Model),l
```

```
  geom_line(data = subset(merged.res, Model=="Bayesian"), aes(y=NEP, x=Date, color = Model),lwd = 1.25)
  ylab(expression(paste("NEP (", O[2], " in mg L", NULL^-1, " day", NULL^-1, ")")))

fig1 <- ggarrange(p1 + rremove("xlab"), p2 + rremove("xlab"), p3 + rremove("xlab"),
                  nrow=3, common.legend = TRUE, legend="right")

annotate_figure(fig1, top = text_grob("Estimated metabolism of pond B3P3"))
```

14: References

Winslow, Luke A., Jacob A. Zwart, Ryan D. Batt, Hilary A. Dugan, R. Iestyn Woolway, Jessica R. Corman, Paul C. Hanson, and Jordan S. Read. 2016. "LakeMetabolizer: An r Package for Estimating Lake Metabolism from Free-Water Oxygen Using Diverse Statistical Models." *Inland Waters* 6 (4): 622–36. https://doi.org/10.1080/iw-6.4.883.