Data for: Partitioning inorganic carbon fluxes using paired O_2-CO_2 sensors in a headwater stream, Costa Rica, Submitted to Biogeochemistry

Nicholas S. Marzolf

Last updated: June 18, 2021

Contents

Publication information

Full authorship and affiliations:

Nicholas S. Marzolf[1 *], Gaston E. Small[2], Diana Oviedo-Vargas[3], Carissa N. Ganong[4], John H. Duff[5], Alonso Ramírez[6], Catherine M. Pringle[7], David P. Genereux[8], Marcelo Ardón[1,9]

1 Department of Forestry and Environmental Resources, North Carolina State University, NC, USA (ORCiD: 0000-0001-9146-1643)
2 College of Arts and Sciences, University of St. Thomas, MN, USA (ORDiD: 0000-0002-9018-7555)
3 Stroud Water Research Center, Avondale, PA, USA (ORCiD: 0000-0001-9333-0962)
4 Department of Biology, Missouri Western State University, MO, USA
5 US Geological Survey, Menlo Park, CA, USA
6 Department of Applied Ecology, North Carolina State University, NC, USA (ORCiD: 0000-0001-9985-5719)
7 Odum School of Ecology, University of Georgia, GA, USA (ORCiD: 0000-0002-3522-7315)
8 Marine, Earth, and Atmospheric Sciences, North Carolina State University, NC, USA (ORCiD: 0000-0001-5601-7088)
9 ORCiD: 0000-0001-7275-2672
* Corresponding author: nmarzol@ncsu.edu

Load packages and explore the data

```r
library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(cowplot)
library(ggrepel)
library(pander)
library(tidyquant)
library(scales)
library(lmodel2)
library(ellipse)
library(ggsci)
library(purrr)

# Read in hourly data file
Tac_all = readRDS("Tac_all.rds")
```

This file contains hourly data from April 1 - September 27, including: stream and well Vaisala pCO2 sensors (CO2_ppm, wellCO2), YSI sonde (temp.water, pH, DO.obs, DO.sat, cond), discharge (sumQ_m3_hr, meanQ_m3_s), and meterological data (relative humidity (RH), Rainfall, barometric pressure (bp_mmHg)) from the weather station located at La Selva Biological Station. Rainfall is collected at 15-minute intervals, and there data here are aggregated to hourly sums. Discharge is collected at 15-minute intervals, and is shown as both mean discharge for the hour (m3/s) and sum of the 4 contributing measurements (m3/hr). Groundwater discharge (GW_Q) is described in the text. Mean depth is taken from the weir.

```r
glimpse(Tac_all)
```

```
## Rows: 4,324
## Columns: 16
## $ Timestamp  <dttm> 2013-04-01 00:00:00, 2013-04-01 01:00:00, 2013-04-01 02:00~
## $ sumQ_m3_hr <dbl> 0.244, 0.535, 0.617, 0.710, 0.773, 0.834, 0.840, 0.812, 0.7~
## $ meanQ_m3_s <dbl> 0.002033333, 0.002229167, 0.002570833, 0.002958333, 0.00322~
## $ mean_depth <dbl> 0.24300, 0.24575, 0.25025, 0.25500, 0.25800, 0.26075, 0.261~
## $ GW_Q       <dbl> 0.02196, 0.04815, 0.05553, 0.06390, 0.06957, 0.07506, 0.075~
## $ RH         <dbl> 92.850, 93.100, 92.900, 92.600, 93.300, 93.400, 93.350, 93.~
## $ Rainfall   <dbl> 0.254, 0.000, 1.778, 0.000, 1.524, 0.508, 1.016, 1.524, 0.0~
## $ bp_mmHg    <dbl> 755.6687, 755.8967, 756.2283, 756.0831, 755.5855, 755.3016,~
## $ CO2_ppm    <dbl> 6287.696, 6042.019, 6058.962, 6086.072, 6087.767, 6055.574,~
## $ temp.water <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ pH         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ DO.obs     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ DO.sat     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ cond       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ wellCO2    <dbl> 41074.82, 42818.21, 44179.08, 45055.86, 45396.08, 46122.21,~
## $ PAR        <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0~
```

We will use this dataset build on with the following calculations at the hourly resolution and aggregate to daily scale at the end of the document.

Inorganic carbon chemistry

Here, we calculate the necessary parameters for $[DIC]$ partitioning and flux calculations.

From pH measured from the sonde, we calculate $[H^+]$ and use the carbonate equilibrium constants, K_1 and K_2 to calculate $[DIC]$ partitioning coefficients to estimate $[H_2CO_3^*]$, $[HCO_3^-]$, and $[CO_3^{2-}]$ (which we ultimately don't consider due to the pH). The partioning coefficients, α_0, α_1, and α_2 correspond to the fraction of $[H_2CO_3^*]$, $[HCO_3^-]$, and $[CO_3^{2-}]$, respectively.

$$\alpha_0 = (1 + \frac{K_1}{[H^+]} + \frac{K_1 K_2}{[H^+]^2})^{-1}$$
$$\alpha_1 = (1 + \frac{[H^+]}{K_1} + \frac{K_2}{[H^+]})^{-1}$$
$$\alpha_2 = (1 + \frac{[H^+]}{K_2} + \frac{[H^+]^2}{K_1 K_2})^{-1}$$

Since we calculate CO_2 (as $[H_2CO_3^*]$) using Henry's Law, we can use the partitioning coefficients to estimate $[DIC]$:

$$[DIC] = \frac{[H_2CO3^*]}{\alpha_0}$$

From $[DIC]$, we calculate $[HCO3^-]$ as

$$[HCO3^-] = [DIC] * \alpha_1$$

```
# define carbonate equilibrium constants (reference)
k1 = 10^-6.3  # k1 = [H+][HCO3-]/[H2CO3]
k2 = 10^-10.3 # k2 = [H+][CO32-]/[HCO3-]

Tac_all = Tac_all %>%
  mutate(

    # calculate [H+] from pH
    H = 10^(-1*pH),

    # DIC dissociation constants
    alpha0 = (1 + (k1/H) + ((k1*k2)/(H^2)))^-1,
    alpha1 = (1 + (H/k1) + (k2/H))^-1,
    alpha2 = (1 + (H/k2) + ((H^2)/(k1*k2)))^-1,

    # temperature-corrected Henry's Law constant (atm*L/mol, Plummer and Busenberg 1982)
    kH = 29.41*exp(-2400*(1/(273 + temp.water) - (1/298))),

    # CO2 saturation (mol m-3), assuming 400 ppm
    CO2_sat = ((400/1e6)/kH)*1000,

    # CO2 in the stream and well (mol m-3)
    CO2_aq = ((CO2_ppm/1e6)/kH)*1000,
    CO2_well = ((wellCO2/1e6)/kH)*1000,

    # DIC partitioning; since pH was < 7, CO3 was <0.2% of DIC, and not calculated
    H2CO3 = (1/kH)*(CO2_ppm/1000),    # H2CO3* (mol m-3)
    TotDIC = H2CO3/alpha0,            # total DIC (mol m-3)
    HCO3 = TotDIC * alpha1            # HCO3 (mol m-3)
  )
```

CO_2 flux calculations

CO_2 evasion (F_{CO2})

```
## seasonal gas exchange coefficients from Oviedo-Vargas et al. (2015)
K_prop_wet = 10.2   # gas exchange coefficient for wet season propane (d-1)
K_prop_dry = 27.2   # gas exchange coefficient for dry season propane (d-1)

# calculate Schmidt numbers for propane, CO2, and O2, using mean temperature (Raymond et al. 2012)
temp_mean = mean(Tac_all$temp.water, na.rm = TRUE)

Sc_prop = 3545.60 - (203.41*temp_mean) + (4.78*(temp_mean^2)) - (0.0404*(temp_mean^3))
Sc_CO2 = 1686.08 - (89.66*temp_mean) + (2.07*(temp_mean^2)) - (0.018*(temp_mean^3))
Sc_O2 = 1568 - (86.04*temp_mean) + (2.142*(temp_mean^2)) - (0.0216*(temp_mean^3))

# Convert to units for CO2 (to calculate CO2 evasion)
K_CO2_dry = K_prop_dry*(Sc_prop/Sc_CO2)^0.7 # gas exchange coefficient for CO2, d-1
K_CO2_wet = K_prop_wet*(Sc_prop/Sc_CO2)^0.7 # gas exchange coefficient for CO2, d-1

# Convert to units for O2 (to calculate net ecosystem production later on)
K_O2_dry = K_prop_dry*(Sc_prop/Sc_O2)^0.7 # gas exchange coefficient for O2, d-1
K_O2_wet = K_prop_wet*(Sc_prop/Sc_O2)^0.7 # gas exchange coefficient for O2, d-1

# calculate evasion for dry season (April) and wet season (May-September), g C m-2 h-1
# gas exchange coefficients are scaled from d^-1 to hr^-1
# multiply by stream depth to scale to area of the stream
# multiply by 12.01 to convert mol C to g C

Tac_all = Tac_all %>%
  mutate(
    CO2_efflux = ifelse(month(Timestamp) == 'April',
                        ((CO2_aq - CO2_sat)*(K_CO2_dry/24))*mean_depth,  # mol C m-2 h-1
                        ((CO2_aq - CO2_sat)*(K_CO2_wet/24))*mean_depth)  # mol C m-2 h-1
    )
```

Hydrologic fluxes: GW_{CO2}, up- and down-reach exports

```
# Dimensions of the Taconazo
Tac_length = 1587.4       # total length of the main-stem Taconazo
Tac_reach = 75            # length of the study reach, m
Tac_width_upper_dry = 1.4 # m
Tac_width_upper_wet = 1.8 # m
Tac_width_lower_dry = 1.5 # m
Tac_width_lower_wet = 2.8 # m

Tac_all = Tac_all %>%
  mutate(
    # groundwater velocity, or the volume of GW entering the benthic area in the reach (m h-1)
    GW_velocity = ifelse(month(Timestamp) == 'April',
                         GW_Q/(Tac_reach * Tac_width_lower_dry),
                         GW_Q/(Tac_reach * Tac_width_lower_wet)
```

```
                              ),

    # GWCO2 flux (mol CO2 m-2 h-1)
    GWCO2 = GW_velocity * CO2_well,

    # upreach discharge (m3 h-1); subtract out groundwater discharge from discharge at the weir
    Q_upreach = sumQ_m3_hr - GW_Q,

    # upreach DIC
    upreach_DIC = Q_upreach * (CO2_aq + HCO3),

    # hydrologic export, hourly CO2_aq and HCO3 * stream discharge
    CO2_aq_export = ((CO2_aq) * (sumQ_m3_hr)),      # aqueous CO2 export (mol h-1)
    HCO3_aq_export = (HCO3 * (sumQ_m3_hr)),         # HCO3 export (mol h-1)
  )
```

Net ecosystem production

This approach is outlined in Hall and Hothkiss (2017)

```
Tac_all = Tac_all %>%
  mutate(
    ts = log((298.15 - temp.water)/(298.15 + temp.water)), # function as part of O2 saturation, Hall an
    u = 10^(8.10765 - (1750.3/(235 + temp.water))), # function as part of O2 saturation

    # calculate O2 saturation; we will also use this for Figure 3
    Osat = exp(2.00907 +
                 (3.22014*ts) +
                 (4.0501*ts^2) +
                 (4.94457*ts^3) -
                 (0.256847*ts^4) +
                 (3.88767*ts^5)) * ((bp_mmHg-u)/(760-u))*1.42905,

    # this calculation is in terms of mol C m-2 h-1
    # we convert to terms of C by assuming a 1:1 molar respiratory quotient and switching the sign; NEP
    direct_met_mol = ifelse(month(Timestamp) == 'April',
                            (((diff(DO.obs))-((K_O2_dry/24)*(Osat - DO.obs)))*(1/32))*mean_depth*-1,
                            (((diff(DO.obs))-((K_O2_wet/24)*(Osat - DO.obs)))*(1/32))*mean_depth*-1),
    )
```

Reach-scale mass balance

$$\frac{d[DIC]}{dt} = inputs - outputs = [((Q_d - Q_{GW}) * [DIC_u]) + (Q_{GW} * [DIC_{GW}]) + (NEP * Lwz)] - [(F_{CO2} * Lw) + (Q_d * [DIC_d])]$$

```
Tac_scale = Tac_all %>%

  # select necessary columns and fluxes
  dplyr::select(Timestamp,
                Rainfall,
                pH,
                alpha0,
```

```r
              Q_upreach,
              upreach_DIC,        # mol DIC h-1
              GWCO2,              # mol CO2 m-2 h-1
              CO2_efflux,         # mol CO2 m-2 h-1
              direct_met_mol,     # mol CO2 m-2 h-1
              CO2_aq_export, # mol CO2 h-1
              HCO3_aq_export # mol HCO3 h-1
              ) %>%

  # convert to mol DIC by dividing by the first partitioning coefficient
  mutate(GW_DIC = GWCO2/alpha0,                             # convert to: mol DIC m-2 h-1
         F_DIC = CO2_efflux/alpha0,                         # convert to: mol DIC m-2 h-1
         NEP_DIC = direct_met_mol/alpha0,                   # convert to: mol DIC m-2 h-1
         down_DIC = HCO3_aq_export + CO2_aq_export     # convert to: mol DIC h-1
         ) %>%

  # calculate mass balance terms (equation 4)
  mutate(DIC_in = upreach_DIC,                                  # mol DIC h-1
         GW_in = ifelse(month(Timestamp) == 'April',
                        GW_DIC*Tac_reach*Tac_width_lower_dry,
                        GW_DIC*Tac_reach*Tac_width_lower_wet),       # mol DIC h-1
         NEP_in = ifelse(month(Timestamp) == 'April',
                         NEP_DIC*Tac_reach*Tac_width_lower_dry,
                         NEP_DIC*Tac_reach*Tac_width_lower_wet),     # mol DIC h-1

         DIC_out = down_DIC,                                    # mol DIC h-1
         F_out = ifelse(month(Timestamp) == 'April',
                        F_DIC*Tac_reach*Tac_width_lower_dry,
                        F_DIC*Tac_reach*Tac_width_lower_wet),        # mol DIC h-1

         dC_dt = (DIC_in + NEP_in + GW_in) -
           (F_out + DIC_out)
         )
```

Figures

Fig 2

```r
# panel a) daily rainfall
rain_plot = ggplot(Tac_all %>%
                     group_by(Date = date(Timestamp)) %>%
                     summarise(dayRain = sum(Rainfall, na.rm = TRUE)),
                   aes(x = as.POSIXct(Date)))+
  geom_bar(aes(y = dayRain), stat = 'identity', fill = 'black')+
  ylim(120, 0)+
  ylab('Daily Rainfall (mm)')+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(axis.title.x = element_blank())

# panel b) mean hourly discharge
Q_plot = ggplot(Tac_all, aes(x = Timestamp))+
```

```r
  geom_line(aes(y = meanQ_m3_s),
            color = 'black')+
  geom_line(aes(y = GW_Q/60),
            color = 'red')+
  ylab(expression(paste("Discharge (", m^3," ", s^-1,")")))+
  scale_color_manual(values = c('black',
                                'red'))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  theme(axis.title.x = element_blank())+
  annotate('text',
           x = as.POSIXct(c('2013-04-6 00:00',
                 '2013-06-17 00:00',
                 '2013-08-21 00:00')),
           y = 0.15,
           label = "*",
           size = 8)

# panel c) dissolved oxygen
DO = ggplot(data = Tac_all, aes(x = Timestamp))+
  geom_point(aes(y = DO.obs))+
  labs(x = "Time", y = expression(paste("DO (mg " ,L^{-1},")")))+
  #ylim(0,8.5)+
  theme(axis.title.x = element_blank())+
  scale_x_datetime(date_breaks = "1 month", labels = date_format("%b"))

# panel d) stream pCO2
strCO2_plot = ggplot(Tac_all, aes(x = Timestamp))+
  geom_point(aes(y = CO2_ppm), color = 'black')+
  ylab(expression(paste("Stream ", CO[2]," ", "(ppmv)")))+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(axis.title.x = element_blank())

# panel e) well pCO2
wellCO2_plot = ggplot(Tac_all, aes(x = Timestamp))+
  geom_point(aes(y = wellCO2), color = 'black')+
  ylab(expression(paste("Well ", CO[2]," ", "(ppmv)")))+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(axis.title.x = element_blank())+
  ylim(30000, 50000)

# combine into Fig 2
plot_grid(rain_plot, Q_plot, DO, strCO2_plot, wellCO2_plot,
          nrow = 5,
          align = 'hv',
          labels = 'auto', hjust = -9, vjust = 2)
```

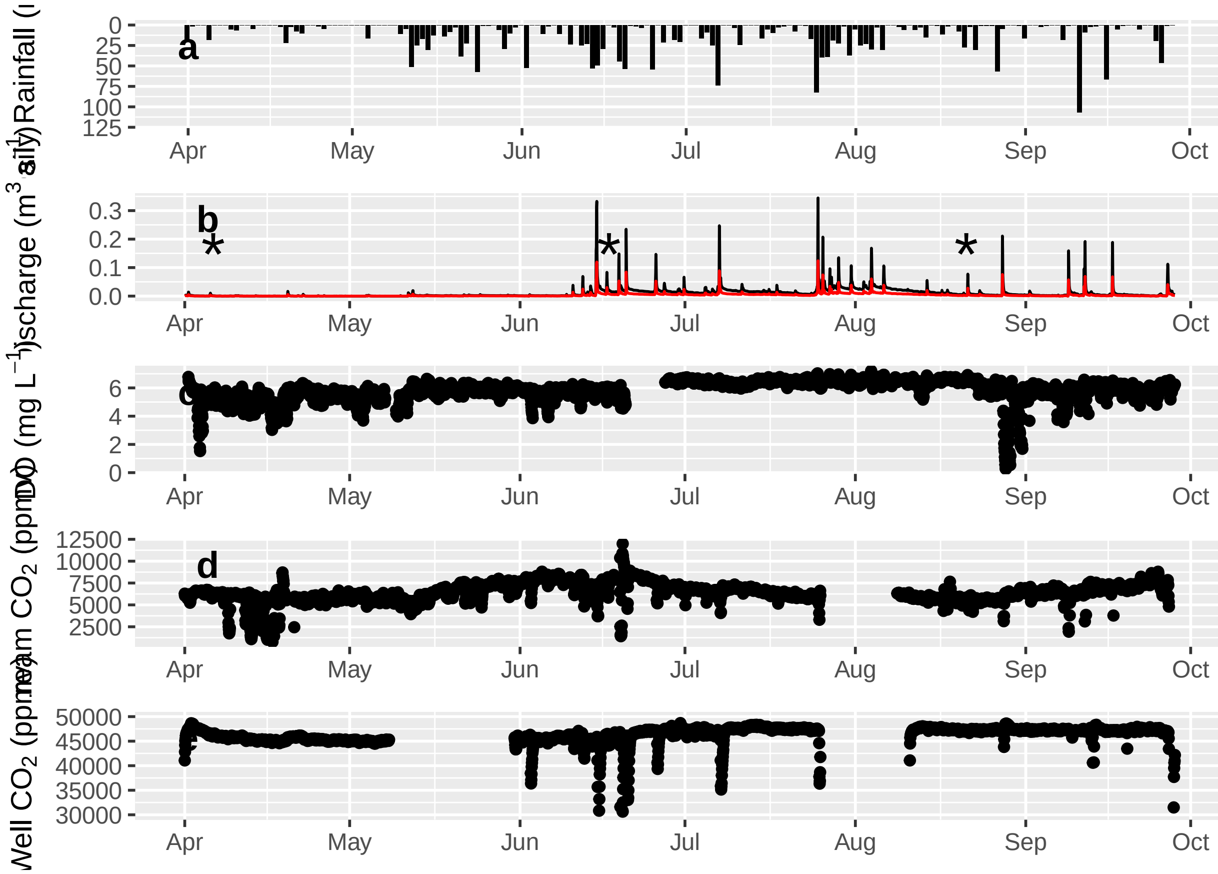


Fig 3

```
# subset the data and calculate the departure in umol L-1
Tac_departure = Tac_all %>%
  dplyr::select(Timestamp, Osat, DO.obs, CO2_aq, CO2_sat) %>%
  mutate(CO2_dep = (CO2_aq - CO2_sat)*1000,           # umol/L
         O2_dep = ((DO.obs - Osat)/32)*1000)          # umol/L

# calculate ellipsis for all data
Tac_dep_clean = na.omit(Tac_departure)
Tac_dep_clean$month = lubridate::month(Tac_dep_clean$Timestamp,
                                       abbr = TRUE, label = TRUE)

# calculate info for Table 2
Tac_mu = c(mean(Tac_dep_clean$CO2_dep, na.rm = TRUE),
           mean(Tac_dep_clean$O2_dep, na.rm = TRUE))
Tac_corMat = cor(cbind(Tac_dep_clean$CO2_dep,
                       Tac_dep_clean$O2_dep))
Tac_covMat = cov(cbind(Tac_dep_clean$CO2_dep,
                       Tac_dep_clean$O2_dep))
Tac_evals = eigen(Tac_covMat)$values
Tac_ell_length = 2*sqrt(5.991*Tac_evals)
```

```r
Tac_reg = lmodel2(data = Tac_dep_clean,
                  O2_dep ~ CO2_dep, nperm = 99)
```

```
## RMA was not requested: it will not be computed.
```

```r
Tac_inter = Tac_reg$regression.results[2,2]
Tac_slope = Tac_reg$regression.results[2,3]
Tac_corr = Tac_reg$r

Tac_metrics = data.frame(meanCO2dep = Tac_mu[1],
                         meanO2dep = Tac_mu[2],
                         offset = Tac_mu[1] + Tac_mu[2],
                         EQ = 1/abs(Tac_slope),
                         width = Tac_ell_length[2],
                         stretch = Tac_ell_length[1])
Tac_metrics
```

```
##   meanCO2dep meanO2dep   offset       EQ    width  stretch
## 1   204.2567 -31.77724 172.4794 9.913518 120.6049 185.9119
```

```r
# plot for all data points with 95% CI ellipsis
all_dep = ggplot()+
  geom_point(alpha = 0.2,
             data = Tac_departure,
             aes(x = CO2_dep, y = O2_dep,
                 color = factor(month(Timestamp))))+
  geom_hline(yintercept = 0)+
  geom_vline(xintercept = 0)+
  geom_abline(intercept = 0, slope = -1, linetype = 2)+
  xlim(-20, 400)+
  ylim(-400, 20)+
  xlab(expression(paste(CO[2], " Departure (µM)")))+
  ylab(expression(paste(O[2], " Departure (µM)")))+
  guides(colour = guide_legend(override.aes = list(alpha = 1)))+
  scale_color_jco(name = "Month",
                  labels = c("Apr", "May", "Jun", "Jul", "Aug", "Sep"))
all_dep
```

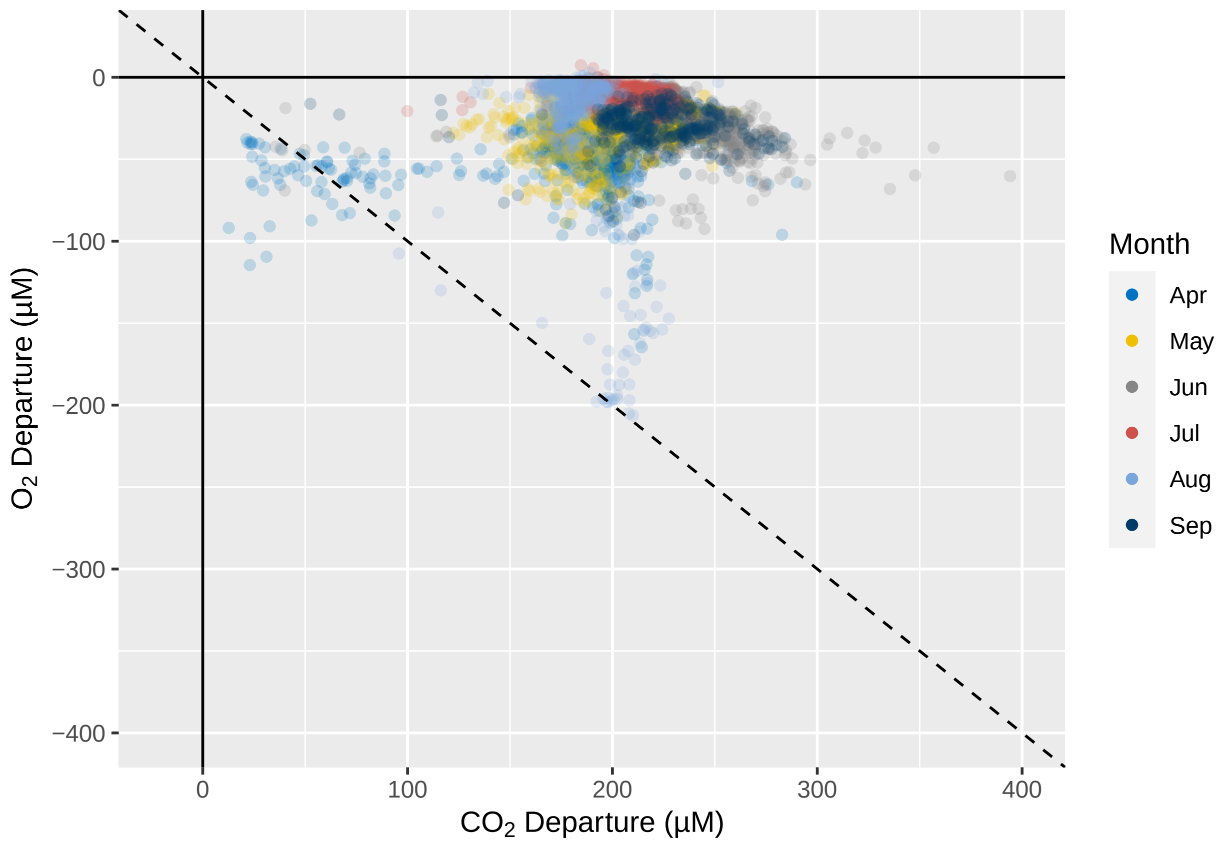
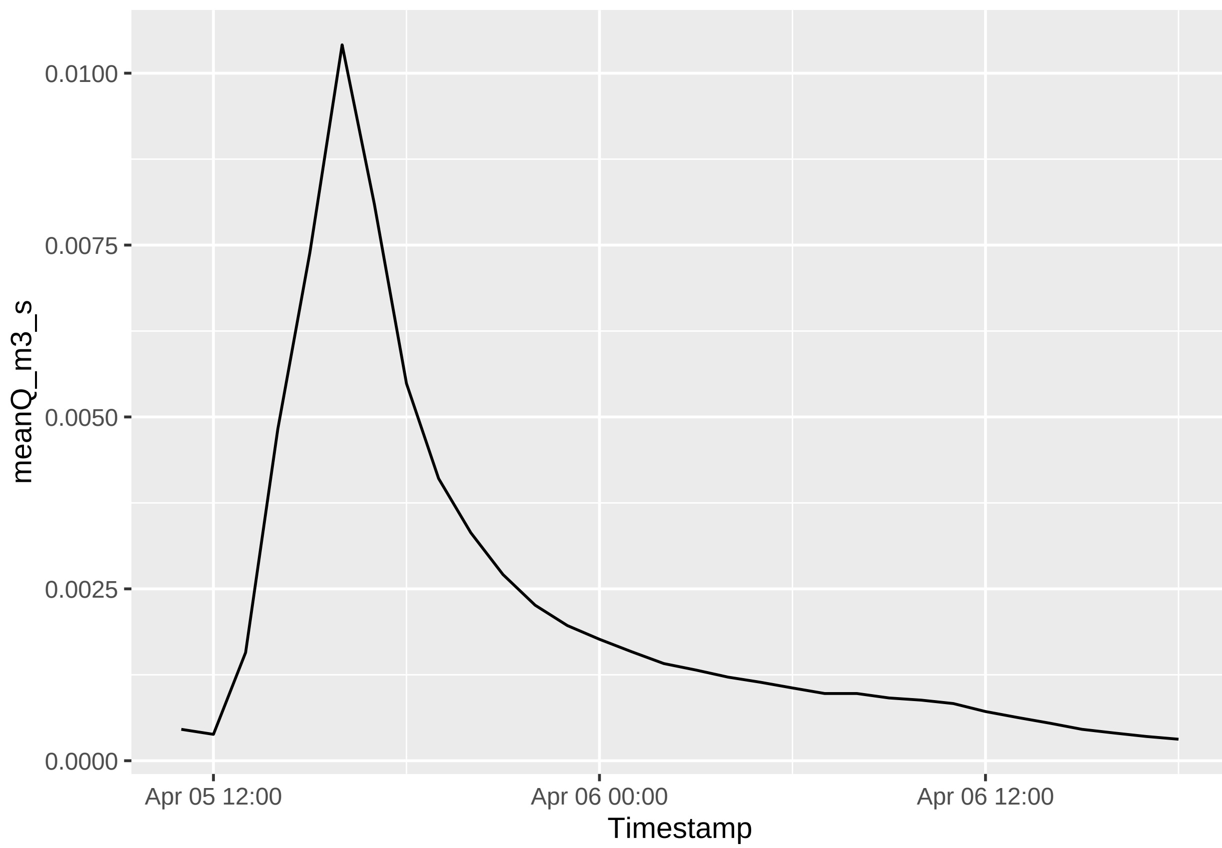


Fig 4

```
# subset the dry season rainfall event
event_dry = Tac_all %>%
  dplyr::filter(Timestamp > "2013-04-05 06:00:00",
                Timestamp < "2013-04-06 15:00:00") %>%
  dplyr::select(Timestamp, meanQ_m3_s, Rainfall, CO2_ppm, DO.obs, wellCO2) %>%
  dplyr::mutate(obs = 1:dplyr::n())

# time series of the event
ggplot(event_dry, aes(Timestamp, meanQ_m3_s))+
  geom_line()
```

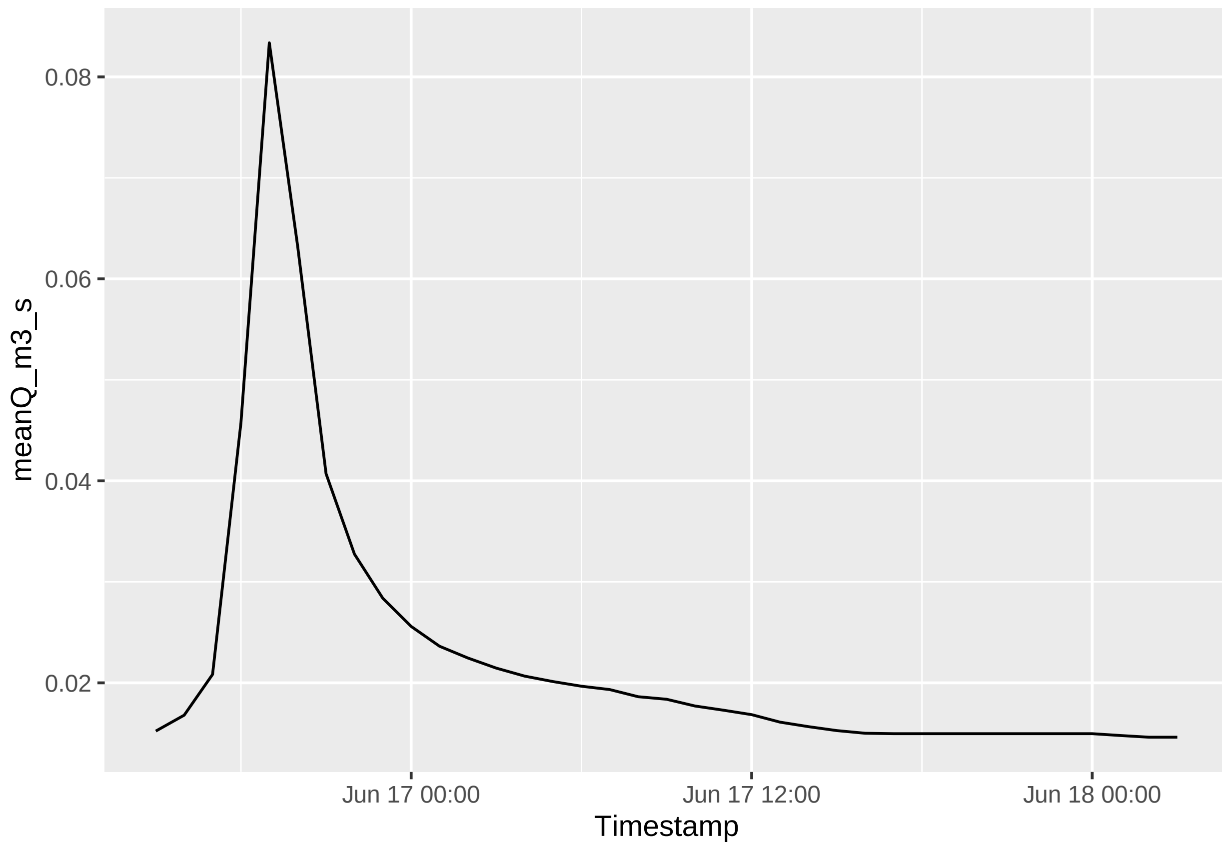



```
hyst_dry = ggplot(event_dry,
                  aes(x = meanQ_m3_s, y = CO2_ppm,color = obs))+
  geom_point(size = 1)+
  geom_path(linejoin = "round", lineend = "square",
            arrow = arrow(angle = 30,
                          ends = 'last',
                          type = 'closed',
                          length = unit(0.1, 'inch')))+
  scale_color_viridis_c(name = "Hour")+
  ylab(expression(paste(pCO[2], " (ppmv)")))+
  xlab(expression(paste("Discharge (", m^3," ", s^{-1},")")))+
  scale_y_continuous(limits = c(3000, 9000),
                     breaks = c(3000, 6000, 9000),
                     labels = c(3000, 6000, 9000))


# early wet season event
## subset data
event_earlyWet = Tac_all %>%
  dplyr::filter(Timestamp > "2013-06-16 10:00:00",
                Timestamp < "2013-06-18 00:00:00") %>%
  dplyr::select(Timestamp, meanQ_m3_s, Rainfall, CO2_ppm, DO.obs, wellCO2) %>%
  dplyr::mutate(obs = 1:dplyr::n())

# plot the hydrograph
ggplot(event_earlyWet, aes(Timestamp, meanQ_m3_s))+
```

```
  geom_line()
```

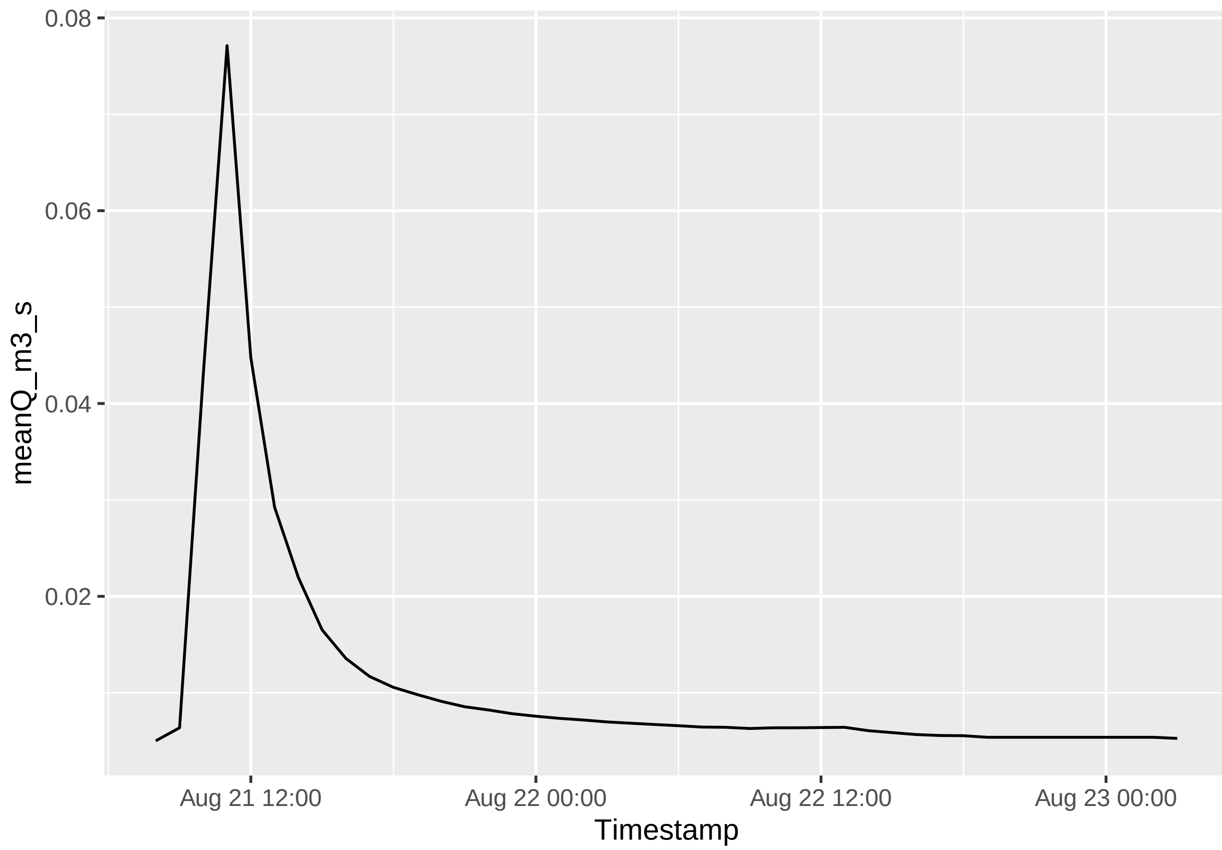



```
# Hysteresis plot
hyst_earlyWet = ggplot(event_earlyWet, aes(x = meanQ_m3_s, y = CO2_ppm,
                                           color = obs))+
  geom_point(size = 1)+
  geom_path(linejoin = "round", lineend = "square",
            arrow = arrow(angle = 30,
                          ends = 'last',
                          type = 'closed',
                          length = unit(0.1, 'inch')))+
  scale_color_viridis_c(name = "Hour")+
  ylab(expression(paste(pCO[2], " (ppmv)")))+
  xlab(expression(paste("Discharge (", m^3," ", s^{-1},")")))+
  scale_y_continuous(limits = c(3000, 9000),
                     breaks = c(3000, 6000, 9000),
                     labels = c(3000, 6000, 9000))

# late wet season event ----

event_lateWet = Tac_all %>%
  dplyr::filter(Timestamp > "2013-08-21 03:00:00",
                Timestamp < "2013-08-23 00:00:00") %>%
  dplyr::select(Timestamp, meanQ_m3_s, Rainfall, CO2_ppm, DO.obs, wellCO2) %>%
  dplyr::mutate(obs = 1:dplyr::n())
```

```r
# hydrograph
ggplot(event_lateWet, aes(Timestamp, meanQ_m3_s))+
  geom_line()
```




```r
# Hysteresis plot
hyst_lateWet = ggplot(event_lateWet, aes(x = meanQ_m3_s, y = CO2_ppm,
                             color = obs))+
  geom_point(size = 1)+
  geom_path(linejoin = "round", lineend = "square",
            arrow = arrow(angle = 30,
                          ends = 'last',
                          type = 'closed',
                          length = unit(0.1, 'inch')))+
  scale_color_viridis_c(name = "Hour",
                        labels = c(0, 10, 20, 30, 40, 50),
                        breaks = c(0, 10, 20, 30, 40, 50))+
  ylab(expression(paste(pCO[2], " (ppmv)")))+
  xlab(expression(paste("Discharge (", m^3," ", s^{-1},")")))+
  scale_y_continuous(limits = c(3000, 9000),
                     breaks = c(3000, 6000, 9000),
                     labels = c(3000, 6000, 9000))

Fig4 = plot_grid(hyst_dry, hyst_earlyWet, hyst_lateWet,
          ncol = 3, nrow = 1,
          labels = 'auto', label_fontface = 'plain',
```

```
          hjust = c(-8, -8, -9))
Fig4
```

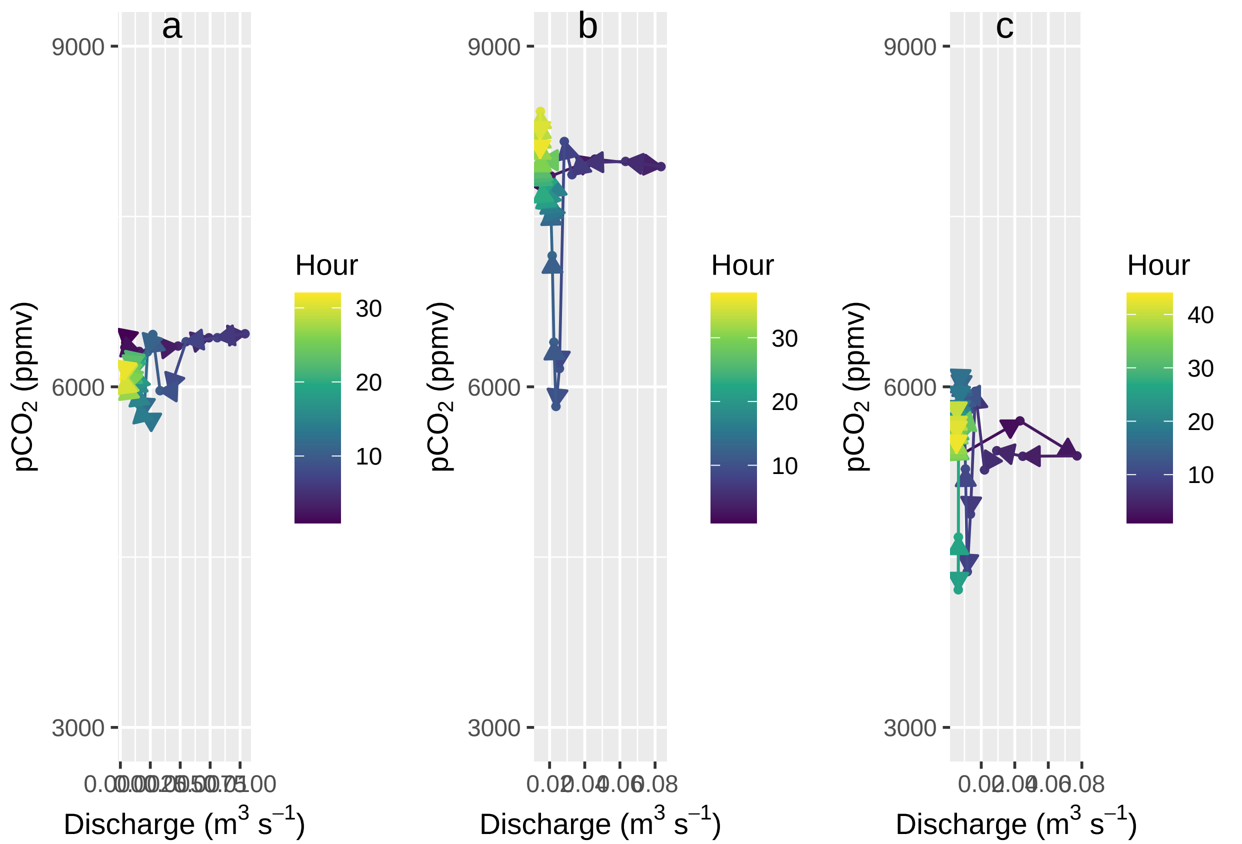
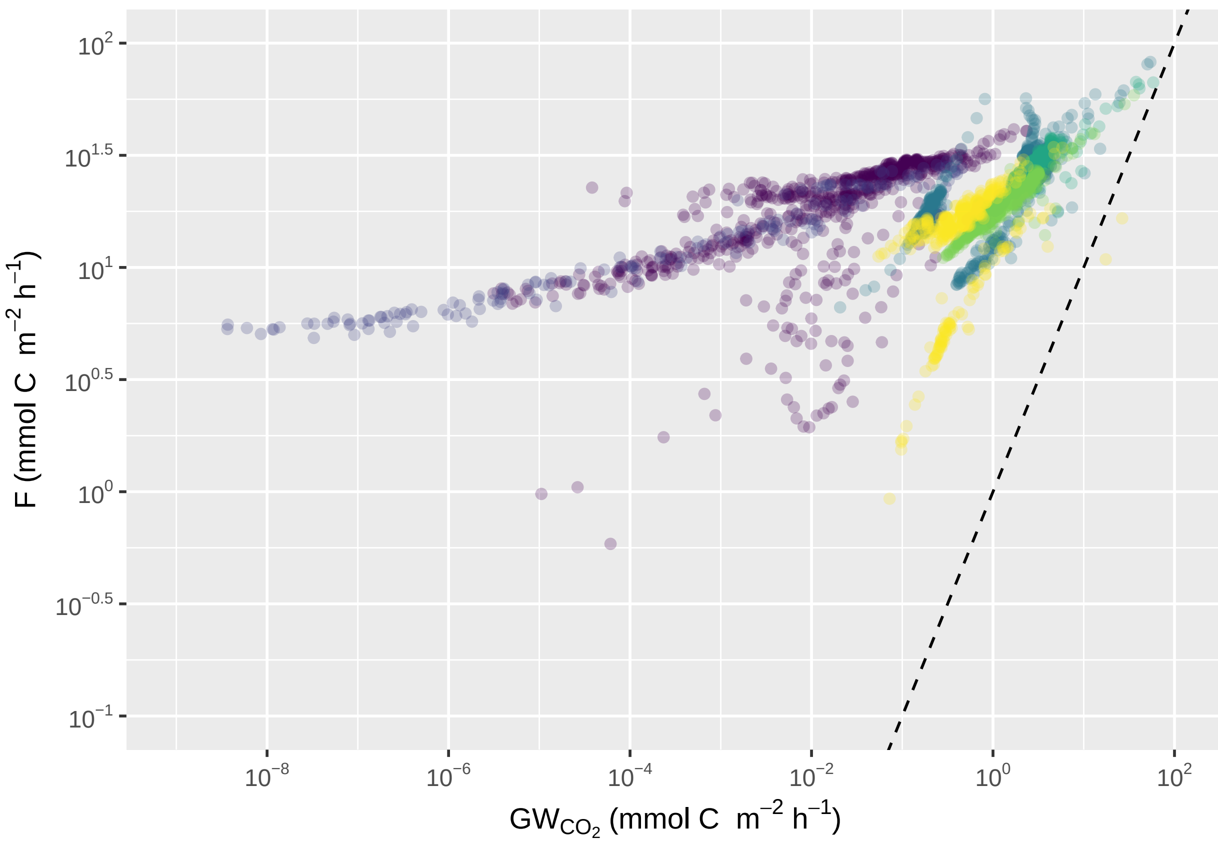
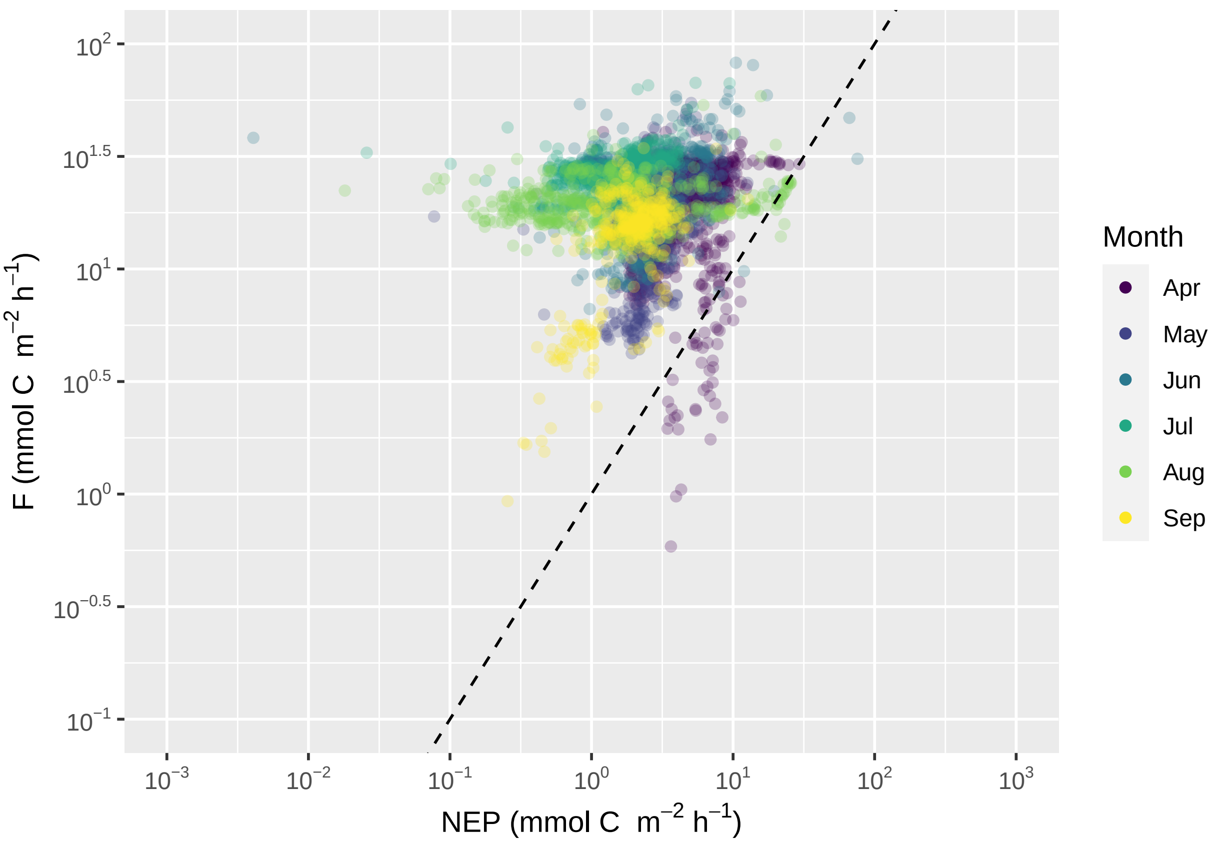


Fig 5

```
F_from_GW = ggplot(Tac_scale,
                  aes(x = GWCO2*1000,
                      y = CO2_efflux*1000))+                    # convert to mmol
  geom_point(aes(color = factor(month(Timestamp,
                                      label = TRUE, abbr = TRUE))),
             alpha = 0.25)+
  geom_abline(aes(slope = 1, intercept = 0),
              linetype = 2)+
  scale_x_log10(limits = c(1e-9, 1e2),
                breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_y_log10(limits = c(1e-1, 1e2),
                breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_color_viridis_d(name = 'Month')+
  xlab(expression(paste(GW[CO[2]], ' (mmol C ', " ", m^-2," ", h^-1,")")))+
  ylab(expression(paste('F (mmol C ', " ", m^-2," ", h^-1,")")))+
  theme(legend.position = 'none')
F_from_GW
```




```
F_from_NEP = ggplot(Tac_scale ,
                    aes(x = direct_met_mol*1000,
                        y = CO2_efflux*1000))+
  geom_point(aes(color = factor(month(Timestamp,
                                      label = TRUE, abbr = TRUE))),
             alpha = 0.25)+
  geom_abline(aes(slope = 1, intercept = 0),
              linetype = 2)+
  scale_x_log10(limits = c(1e-3, 1e3),
                breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_y_log10(limits = c(1e-1, 1e2),
                breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_color_viridis_d(name = 'Month')+
  xlab(expression(paste('NEP (mmol C ', " ", m^-2," ", h^-1,")")))+
  ylab(expression(paste('F (mmol C ', " ", m^-2," ", h^-1,")")))+
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
F_from_NEP
```




```r
Fig5 = plot_grid(F_from_GW, F_from_NEP,
                 ncol = 2,
                 rel_widths = c(0.75, 1),
                 labels = 'auto', label_fontface = 'plain',
                 hjust = -8.5, vjust = 1.5)
Fig5
```

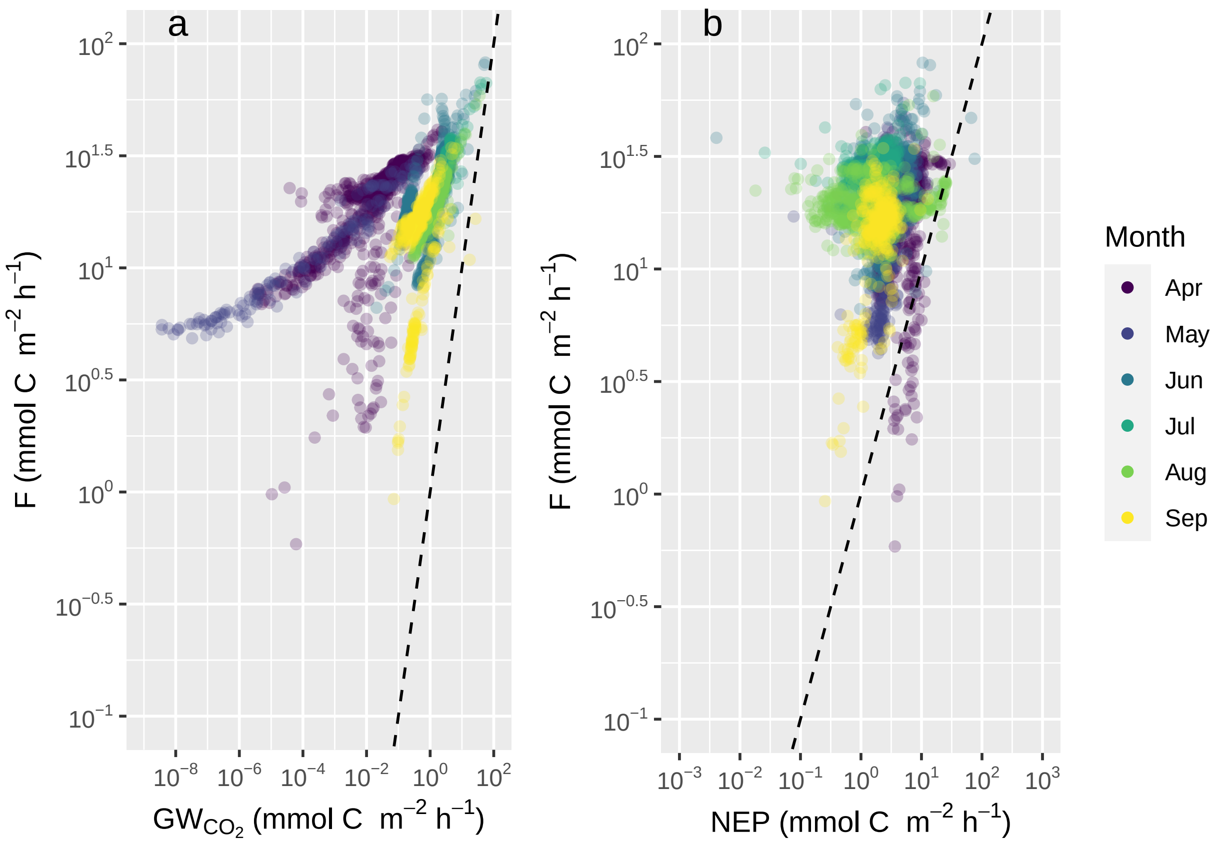
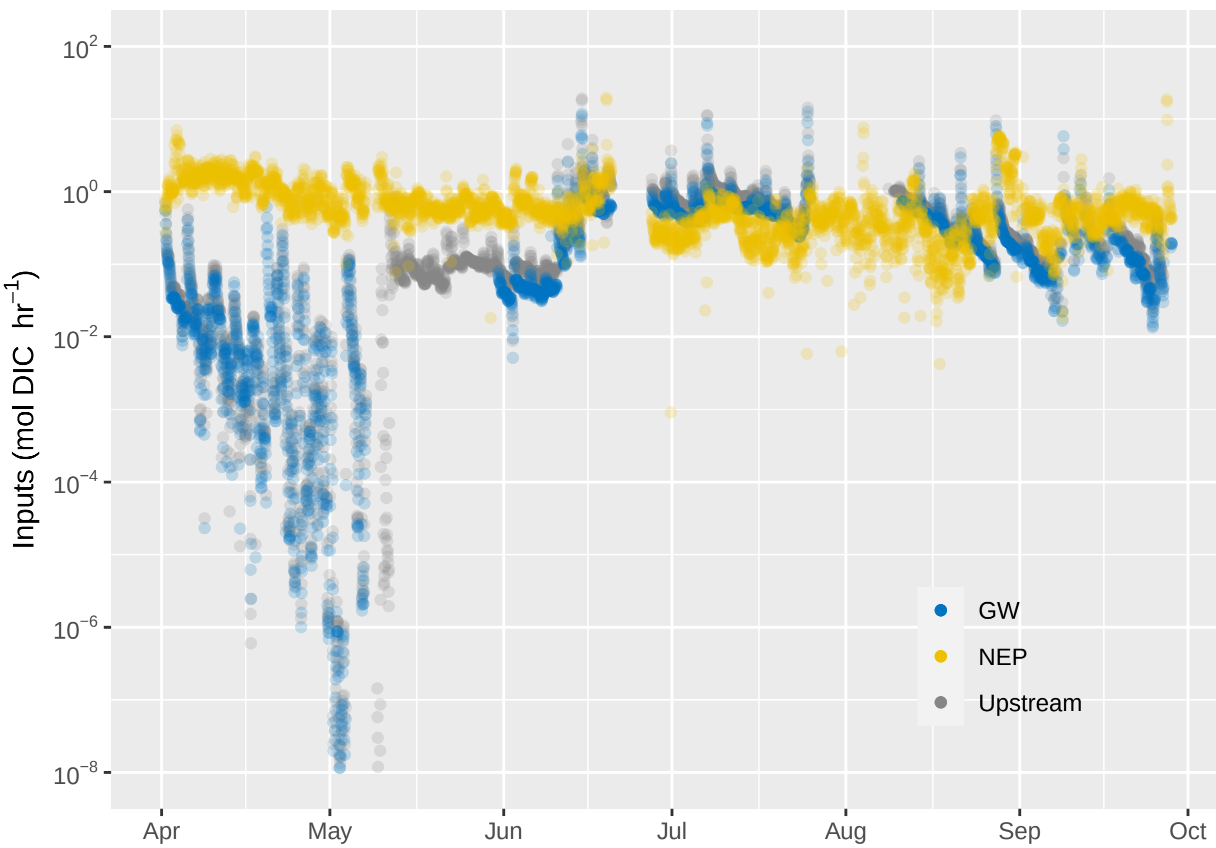
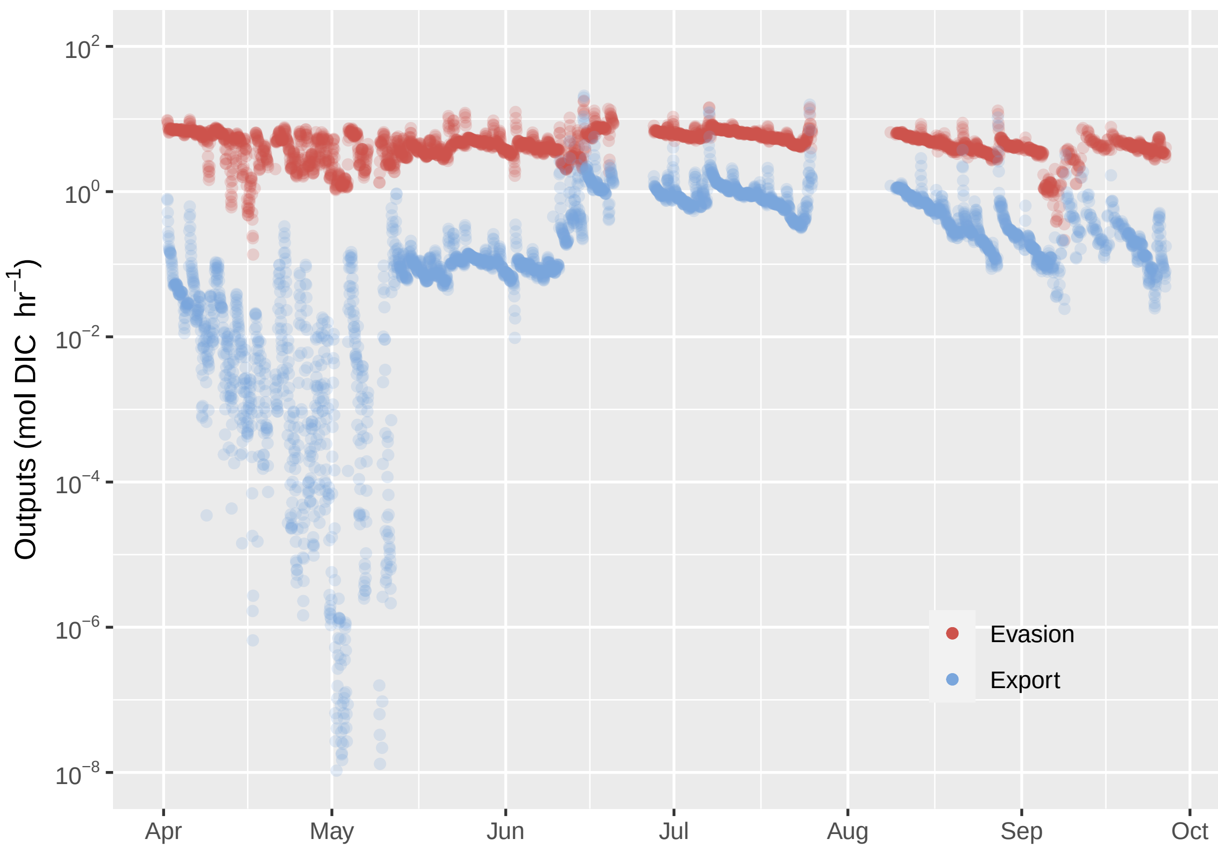


Fig 6

```
Tac_scaled_in = ggplot(Tac_scale,
       aes(x = Timestamp))+
  geom_point(aes(y = DIC_in, color = 'Upstream'), alpha = 0.2)+
  geom_point(aes(y = GW_in, color = 'GW'), alpha = 0.2)+
  geom_point(aes(y = NEP_in, color = 'NEP'), alpha = 0.2)+
  scale_color_manual(values = c('#0073C2FF', '#EFC000FF', '#868686FF'))+
  ylab(expression(paste("Inputs (mol DIC ","  ",hr^-1,")")))+
  scale_y_log10(limits = c(.00000001, 1e2),
                breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(legend.position = c(0.8, 0.2),
        legend.title = element_blank(),
        legend.background = element_blank(),
        axis.title.x = element_blank())+
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
Tac_scaled_in
```




```
Tac_scaled_out = ggplot(Tac_scale,
       aes(x = Timestamp))+
  geom_point(aes(y = F_out, color = 'Evasion'), alpha = 0.2)+
  geom_point(aes(y = DIC_out, color = 'Export'), alpha = 0.2)+
  scale_color_manual(values = c("#CD534CFF", "#7AA6DCFF"))+
  ylab(expression(paste("Outputs (mol DIC "," ",hr^-1,")")))+
  scale_y_log10(limits = c(.00000001, 1e2),
              breaks = trans_breaks("log10", function(x) 10^x),
              labels = trans_format("log10", math_format(10^.x)))+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(legend.position = c(0.8, 0.2),
        legend.title = element_blank(),
        legend.background = element_blank(),
        axis.title.x = element_blank())+
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
Tac_scaled_out
```




```
Tac_scaled_dCdt = ggplot(Tac_scale)+
  geom_density(aes(x = dC_dt), fill = 'grey')+
  xlab(expression(paste("dDIC ", dt^-1, ' (mol DIC)')))+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank())+
  coord_flip()
Tac_scaled_dCdt
```


```
Fig6 = plot_grid(plot_grid(Tac_scaled_in, Tac_scaled_out,
                    nrow = 2, align = 'hv',
                    labels = 'auto', label_fontface = 'plain',
                    hjust = -8, vjust = 1.75),
          Tac_scaled_dCdt,
          ncol = 2,
          rel_widths = c(3,1),
          labels = c('', 'c'), label_fontface = 'plain',
          hjust = -8,
          align = 'hv')
Fig6
```

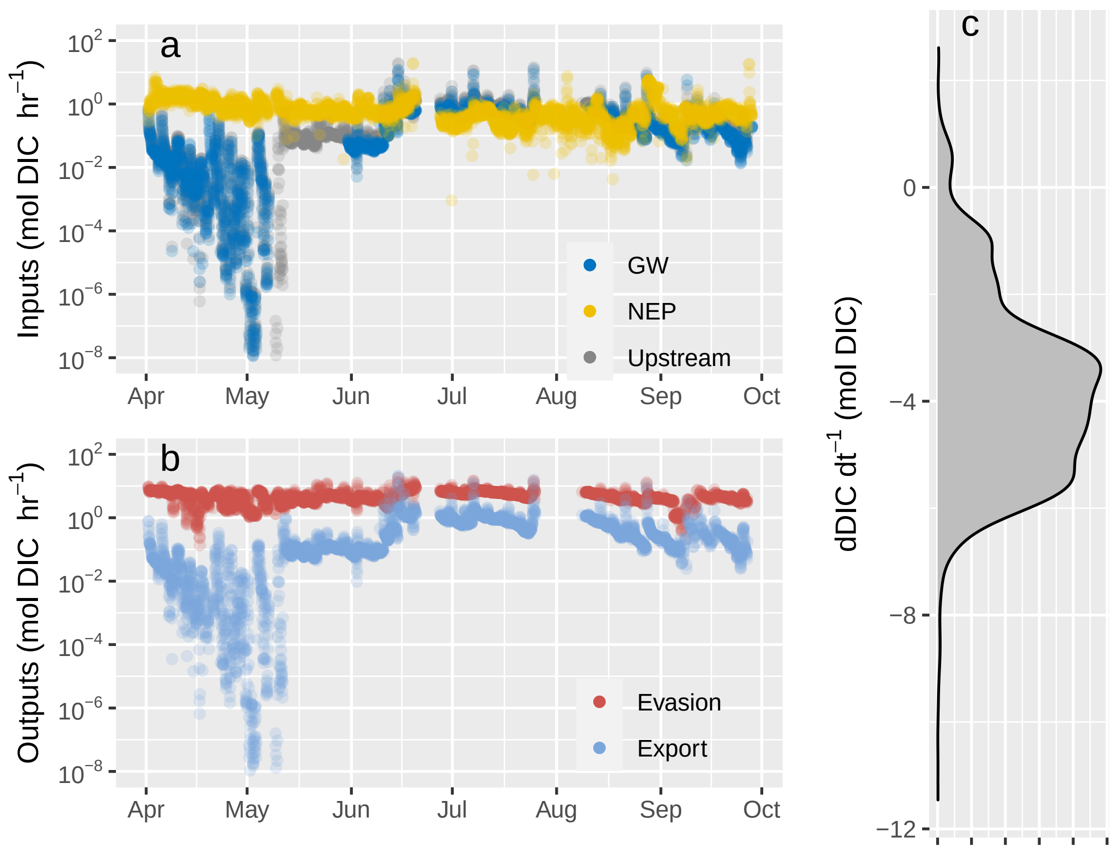


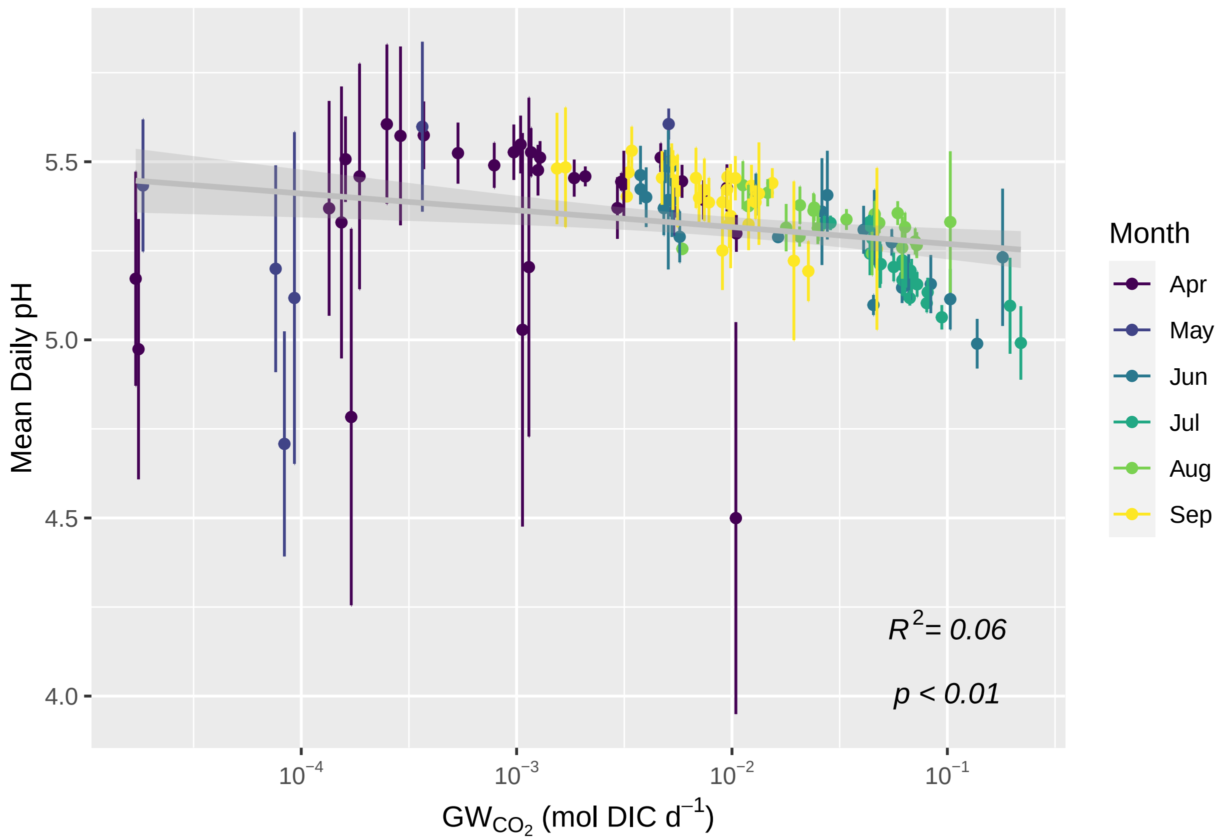
Fig 7

```
Tac_corr = Tac_scale %>%
  group_by(Timestamp = date(Timestamp)) %>%
  summarise(sumGW = sum(GWCO2, na.rm = TRUE),
            sumF = sum(CO2_efflux, na.rm = TRUE),
            sumNEP = sum(direct_met_mol, na.rm = TRUE),
            sumDICin = sum(DIC_in, na.rm = TRUE),
            sumDICout = sum(DIC_out, na.rm = TRUE),
            sumrain = sum(Rainfall, na.rm = TRUE),
            meanpH = mean(pH, na.rm = TRUE),
            sdpH = sd(pH, na.rm = TRUE))

GW_cor_pH = ggplot(Tac_corr %>%
         filter(sumGW > 1e-5),
       aes(x = sumGW,
           y = meanpH))+
  geom_point(aes(color = month(Timestamp,
                               label = TRUE, abbr = TRUE)))+
  geom_errorbar(aes(ymin = meanpH - sdpH,
                    ymax = meanpH + sdpH,
                    color = month(Timestamp,
                                  label = TRUE, abbr = TRUE)))+
  geom_smooth(method = 'lm',
```

```
              color = 'grey',
              alpha = 0.25)+
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)))+
  scale_color_viridis_d(name = 'Month')+
  xlab(expression(paste(GW[CO[2]], ' (mol DIC ', d^-1, ")")))+
  ylab('Mean Daily pH')+
  annotate(x = 0.1, y = 4, 'text', label = expression(italic("p < 0.01")))+
  annotate(x = 0.1, y = 4.2, 'text', label = expression(italic(paste(R^2,"= 0.06"))))
GW_cor_pH
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Supplemental Figure

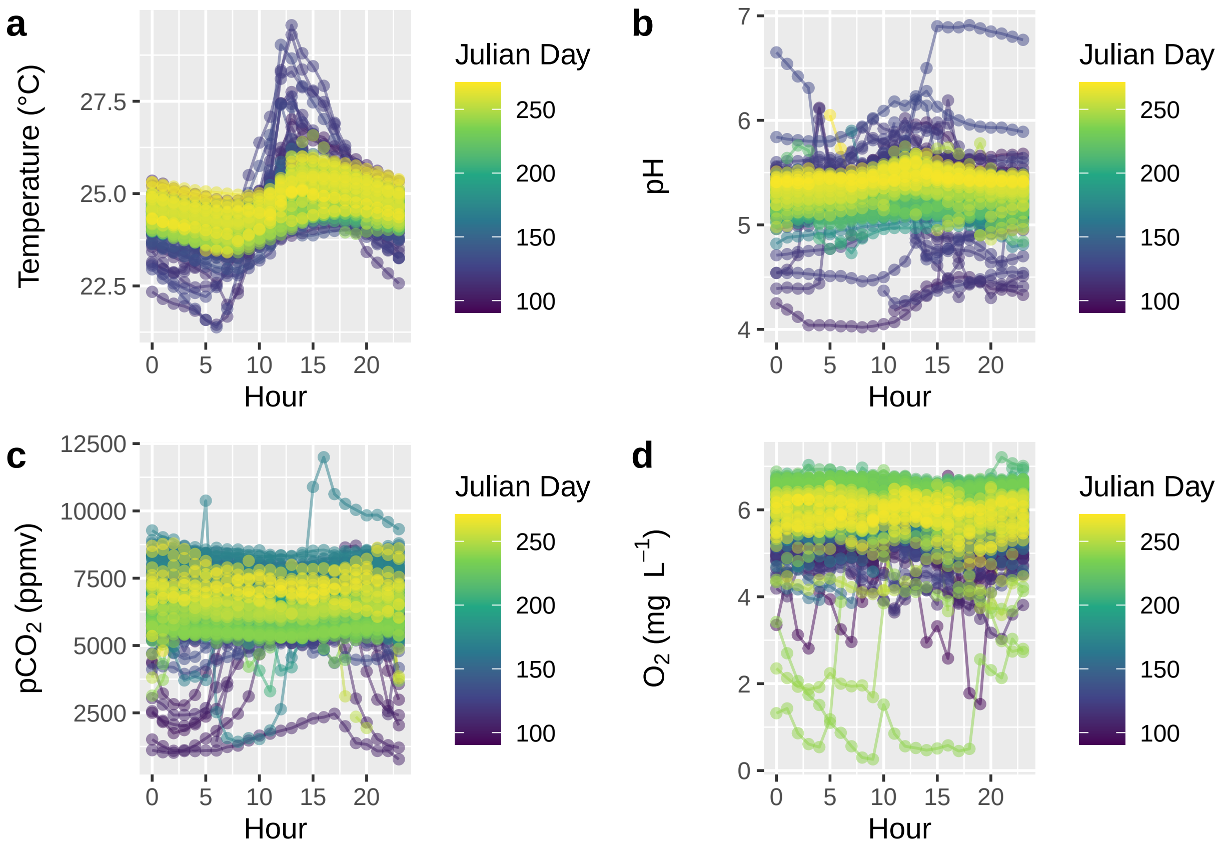
```
# temperature
tempHour = ggplot(Tac_all, aes(x = hour(Timestamp), y = temp.water))+
  geom_line(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  xlab('Hour')+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))+
  ylab("Temperature (°C)")
```

```r
# pH
pHHour = ggplot(Tac_all, aes(x = hour(Timestamp), y = pH))+
  geom_line(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  labs(x = 'Hour', y = "pH")+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))

# stream CO2, ppm
CO2Hour = ggplot(Tac_all, aes(x = hour(Timestamp), y = CO2_ppm))+
  geom_line(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  xlab('Hour')+
  ylab(expression(paste(pCO[2], ' (ppmv)')))+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))

# dissolved oxygen
DOHour = ggplot(Tac_all, aes(x = hour(Timestamp), y = DO.obs))+
  geom_line(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5, aes(group = yday(Timestamp), color = yday(Timestamp)))+
  xlab('Hour')+
  ylab(expression(paste(O[2], " (mg "," ",L^{-1},")")))+
  scale_colour_viridis_c(name = 'Julian Day')+
  theme(legend.title = element_text(face = 'plain'))

FigS1 = plot_grid(tempHour, pHHour, CO2Hour, DOHour,
                  ncol = 2, nrow = 2, align = 'hv',
                  labels = 'auto')
FigS1
```



Session info

```r
pander(sessionInfo())
```

R version 4.0.3 (2020-10-10)

Platform: x86_64-w64-mingw32/x64 (64-bit)

locale: *LC_COLLATE=English_United States.1252*, *LC_CTYPE=English_United States.1252*, *LC_MONETARY=English_United States.1252*, *LC_NUMERIC=C* and *LC_TIME=English_United States.1252*

attached base packages: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: *purrr(v.0.3.4)*, *ggsci(v.2.9)*, *ellipse(v.0.4.2)*, *lmodel2(v.1.7-3)*, *scales(v.1.1.1)*, *tidyquant(v.1.0.3)*, *quantmod(v.0.4.18)*, *TTR(v.0.24.2)*, *PerformanceAnalytics(v.2.0.4)*, *xts(v.0.12.1)*, *zoo(v.1.8-9)*, *pander(v.0.6.3)*, *ggrepel(v.0.9.1)*, *cowplot(v.1.1.1)*, *ggplot2(v.3.3.3)*, *lubridate(v.1.7.10)*, *tidyr(v.1.1.3)* and *dplyr(v.1.0.6)*

loaded via a namespace (and not attached): *tidyselect(v.1.1.0)*, *xfun(v.0.23)*, *splines(v.4.0.3)*, *lattice(v.0.20-41)*, *colorspace(v.2.0-0)*, *vctrs(v.0.3.8)*, *generics(v.0.1.0)*, *viridisLite(v.0.4.0)*, *htmltools(v.0.5.1.1)*, *mgcv(v.1.8-35)*, *yaml(v.2.2.1)*, *utf8(v.1.1.4)*, *rlang(v.0.4.10)*, *pillar(v.1.6.0)*, *glue(v.1.4.2)*, *withr(v.2.4.2)*, *DBI(v.1.1.1)*, *lifecycle(v.1.0.0)*, *stringr(v.1.4.0)*, *Quandl(v.2.10.0)*, *munsell(v.0.5.0)*, *gtable(v.0.3.0)*, *evaluate(v.0.14)*, *labeling(v.0.4.2)*, *knitr(v.1.32)*, *curl(v.4.3)*, *fansi(v.0.4.1)*, *highr(v.0.9)*, *Rcpp(v.1.0.5)*, *jsonlite(v.1.7.2)*, *farver(v.2.1.0)*, *digest(v.0.6.27)*, *stringi(v.1.5.3)*, *grid(v.4.0.3)*, *quadprog(v.1.5-8)*, *cli(v.2.4.0)*, *tools(v.4.0.3)*, *magrittr(v.2.0.1)*, *tibble(v.3.0.4)*, *crayon(v.1.4.1)*, *pkgconfig(v.2.0.3)*, *Matrix(v.1.3-2)*, *ellipsis(v.0.3.1)*, *assertthat(v.0.2.1)*, *rmarkdown(v.2.7)*, *httr(v.1.4.2)*, *rstudioapi(v.0.13)*, *R6(v.2.5.0)*, *nlme(v.3.1-152)* and *compiler(v.4.0.3)*