Observed Data Prep for Model Analysis

PAZ

31/01/2018

Purpose

Generate BEACH calibration data with:

- **groupAlteck2016_R.csv** (Book 04)

Lab parameters and field constants

```
if (MAC) {
  if (WIN){
    path = file.path("C:/Users/DayTimeChunks/Documents/PhD/HydrologicalMonitoring")

  } else {
    # path = file.path("/Users/DayTightChunks/Documents/PhD/HydrologicalMonitoring")
    path = file.path("/Users/DayTightChunks/Documents/PhD/hydrological-monitoring")
    time = read.csv2("/Users/DayTightChunks/Documents/PhD/Models/phd-model-master/Analysis/Data/Time.csv
    time$DayMoYr = as.POSIXct(strptime(time$Date, "%d/%m/%Y", tz="EST"))
  }
} else {
  path = file.path("D:/Documents/these_pablo/Alteckendorf2016/HydrologicalMonitoring")
  time = read.csv2("D:/Documents/these_pablo/Models/BEACH2016/Analysis/Data/Time.csv")
  time$DayMoYr = as.POSIXct(strptime(time$Date, "%d/%m/%Y", tz="EST"))
}
source(file.path(path, "global.R"))
```

Packages

```
# Plotting functions
library("scales")
library("tidyr")
library("dplyr")
library("reshape")
library("zoo") # na.approx()
```

Working directory

```
# setwd("D:/Documents/these_pablo/Alteckendorf2016/R")

# MAC
# setwd("/Users/DayTightChunks/Documents/PhD/Routput/Alteck/R")
```

```
# Mac-WIN
# setwd("C:/Users/DayTightChunks/Documents/Models/pesti-beach16/Analysis/Data")
getwd()
```

```
## [1] "/Users/DayTightChunks/Documents/PhD/Models/phd-model-master/Analysis/Data"
```

Discharge & Response Variables (with markers)

- Ignoring δ in filters (for now)

```
q = read.csv2(file.path(path, "Data/groupAlteck2016_R.csv"))
q$Vol.L = q$Vol2min * 1000

q = q[ , c("Date", "DateCheck", "Q.HW1", "DayMoYr", "Vol.L", "sampleQ", "Type", "SubWeeks", "Weeks", "W
names(q)
```

```
##  [1] "Date"      "DateCheck" "Q.HW1"     "DayMoYr"   "Vol.L"
##  [6] "sampleQ"   "Type"      "SubWeeks"  "Weeks"     "WeekNo"
```

```
mark = read.csv(file.path(path, "Data/MarkerResponse_R05.csv"))
mark = mark[, c("WeekSubWeek",
                # "AveDischarge.m3.h", "Volume.m3",  "Sampled.Hrs",
                # "Sampled",
                "Conc.mug.L" , "Conc.SD",
                # "Vol.SPE.L", "Conc.in500uL",
                "OXA_mean", "OXA_SD", "ESA_mean", "ESA_SD",
                "N.x", "diss.d13C", "SD.d13C",
                "MES.mg.L", "MES.sd", "MO.mg.L", "Conc.Solids.mug.gMES", "Conc.Solids.ug.gMES.SD" #,
                #"N.y",  "filt.d13C",  "filt.SD.d13C" #,
                #"DD13C.diss", "DD13C.filt"
                )]
names(mark)
```

```
##  [1] "WeekSubWeek"            "Conc.mug.L"
##  [3] "Conc.SD"                "OXA_mean"
##  [5] "OXA_SD"                 "ESA_mean"
##  [7] "ESA_SD"                 "N.x"
##  [9] "diss.d13C"              "SD.d13C"
## [11] "MES.mg.L"               "MES.sd"
## [13] "MO.mg.L"                "Conc.Solids.mug.gMES"
## [15] "Conc.Solids.ug.gMES.SD"
```

```
# Delete repeated W6 observation, or with NA in week markers
# mark = mark[mark$WeekSubWeek != as.character("W6-3j7") & !is.na(mark$WeekSubWeek), ]



q$Date = as.POSIXct(strptime(q$DateCheck, "%d/%m/%Y %H:%M", tz="EST"))
q$DayMoYr = as.POSIXct(strptime(q$DateCheck, "%d/%m/%Y", tz="EST"))
q$Min = 2.0

CHECKO = F
if (CHECKO){
  sum(is.na(q$Date))
```

```
  naDates = q[is.na(q$Date == TRUE),]

  duplicateAlteck <- q[duplicated(q$DateCheck),]
  head(duplicateAlteck)
}
```

Prepare Volume Discharged Time Series (TSS)

```
qDay <- q %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(Volday.L = sum(Vol.L))

qDay$VolTot.m3 = round(qDay$Volday.L/10^3, 3)

qTime = merge(time, qDay, by = "DayMoYr", all = T)

qTime_cal = subset(qTime, !is.na(VolTot.m3))
qTime_cal = qTime_cal[, c("Jdays", "VolTot.m3")]
names(qTime_cal) = c("Jdays", "Qm3")

mean(qTime$VolTot.m3, na.rm = T)
```

```
## [1] 259.9841
```

```
sd(qTime$VolTot.m3, na.rm = T)
```

```
## [1] 453.5581
```

```
Volm3_tss = qTime[,c("Jdays", "VolTot.m3")]

Volm3_tss$VolTot.m3 = ifelse(is.na(Volm3_tss$VolTot.m3), -1.0, Volm3_tss$VolTot.m3)

if (F) {
  write.table(Volm3_tss, "BEACH_R/q_obs_m3day.tss", sep="\t", row.names = F, col.names = F)
  write.table(qTime_cal, "BEACH_R/q_out_cal.tss", sep="\t", row.names = F, col.names = T) # m3day
}

if (F) {

  ## Convert m3.h -> m3
  qDay <- q %>%
    group_by(DayMoYr) %>%
    dplyr::summarize(Q.m3 = sum(Vol2min))

  qDay$Q.mm = (qDay$Q.m3/catchment_area)*10^3

  qDay$time = seq.int(nrow(qDay))

  # Qm3/day
  DischQm3_tss = qDay[,c("time", "Q.m3")]
  write.table(DischQm3_tss, "BEACH_R/disch_m3day.tss", sep="\t", row.names = F, col.names = F)

  # Qmm/day
```

```
  DischQmm_tss = qDay[,c("time", "Q.mm")]
  write.table(DischQmm_tss, "BEACH_R/disch_mmday.tss", sep="\t", row.names = F)


}
```

Merge 2-min disch with sub-weekly markers

- Convert to mass discharged

```
qm = merge(q, mark, by.x = "SubWeeks", by.y = "WeekSubWeek", all = T)

# Dissolved
qm$SmetOut_ug.obs = qm$Vol.L*qm$Conc.mug.L
qm$SmetOut_ug.sd = qm$Vol.L*qm$Conc.SD
qm$OxaOut_ug.obs =  qm$Vol.L*qm$OXA_mean
qm$OxaOut_ug.sd =  qm$Vol.L*qm$OXA_SD
qm$EsaOut_ug.obs =  qm$Vol.L*qm$ESA_mean
qm$EsaOut_ug.sd =  qm$Vol.L*qm$ESA_SD

# Suspended Solids (SS)
# Smet.ug in SS = ug/g * (MES [mg/L] * [1g/10^3mg])  * Vol [L]
qm$SmetSS_ug.obs = qm$Conc.Solids.mug.gMES * (qm$MES.mg.L*1/10^3)  * qm$Vol.L
qm$SmetSS_ug.sd = qm$Conc.Solids.ug.gMES.SD * (qm$MES.sd*1/10^3)  * qm$Vol.L

qm$MassDelta.obs = qm$SmetOut_ug.obs*qm$diss.d13C
qm$MassDelta.sd = qm$SmetOut_ug.sd*qm$SD.d13C


names(qm)
```

```
##  [1] "SubWeeks"               "Date"
##  [3] "DateCheck"              "Q.HW1"
##  [5] "DayMoYr"                "Vol.L"
##  [7] "sampleQ"                "Type"
##  [9] "Weeks"                  "WeekNo"
## [11] "Min"                    "Conc.mug.L"
## [13] "Conc.SD"                "OXA_mean"
## [15] "OXA_SD"                 "ESA_mean"
## [17] "ESA_SD"                 "N.x"
## [19] "diss.d13C"              "SD.d13C"
## [21] "MES.mg.L"               "MES.sd"
## [23] "MO.mg.L"                "Conc.Solids.mug.gMES"
## [25] "Conc.Solids.ug.gMES.SD" "SmetOut_ug.obs"
## [27] "SmetOut_ug.sd"          "OxaOut_ug.obs"
## [29] "OxaOut_ug.sd"           "EsaOut_ug.obs"
## [31] "EsaOut_ug.sd"           "SmetSS_ug.obs"
## [33] "SmetSS_ug.sd"           "MassDelta.obs"
## [35] "MassDelta.sd"
```

Compute by-day & by-subweek

- ignoring solids (for now)

We need to obtain a bulk concentration for the day, not by taking the mean concentration, but rather the proportional mass contribution of each sub-sample, determined by the relative volume such that the concentration of each daily sample (C_i) is :

$$C_{bulksample} = \frac{C_1 \cdot V_1 + C_2 \cdot V_2}{V_{tot}} \tag{1}$$

To obtain the bluk concentration (and because there are overlapps of some days across subsamples) we need to first to:

1. Convert discharge mass to concentration
2. Interpolate concentrations (and deltas) of only the duplicate days, where no sample was possible

```
# Step 1
# Sum masses by day, without mixing days from different samples (the latter will be 'duplicate' days)
qmDay <- qm %>%
  group_by(DayMoYr, SubWeeks) %>%
  dplyr::summarize(Volday.L = sum(Vol.L),
                   SmpHrs = sum(Min)/60,
                   SmOut_ug.obs = sum(SmetOut_ug.obs),
                   SmOut_ug.sd = (sum(SmetOut_ug.sd^2))^0.5, # Cumulative SD
                   OxOut_ug.obs = sum(OxaOut_ug.obs),
                   OxOut_ug.sd = (sum(OxaOut_ug.sd^2))^0.5, # Cumulative SD
                   EsOut_ug.obs = sum(EsaOut_ug.obs),
                   EsOut_ug.sd = (sum(EsaOut_ug.sd^2))^0.5, # Cumulative SD
                   ConcSmOut_ugL.obs = SmOut_ug.obs/Volday.L, # Smet
                   ConcSmOut_ugL.sd = SmOut_ug.sd/Volday.L,
                   ConcOxOut_ugL.obs = OxOut_ug.obs/Volday.L, # Oxa
                   ConcOxOut_ugL.sd = OxOut_ug.sd/Volday.L,
                   ConcEsOut_ugL.obs = EsOut_ug.obs/Volday.L, # Esa
                   ConcEsOut_ugL.sd = EsOut_ug.sd/Volday.L,
                   delta.obs = sum(MassDelta.obs)/(sum(SmetOut_ug.obs)),
                   delta.sd = (sum(MassDelta.sd^2))^0.5/(sum(SmetOut_ug.sd^2))^0.5
                   )

# Step 2
# Get all duplicated days with data
allDup = qmDay %>%
  group_by(DayMoYr) %>%
  filter(n()>1 & !is.na(SmOut_ug.obs))

# Do flow proportional average
qmBlkDup = allDup %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(VolTot.L = sum(Volday.L),
                   TotHrs = sum(SmpHrs),
                   ConSmOut_ugL.blk = sum(ConcSmOut_ugL.obs * Volday.L)/sum(Volday.L),
                   ConSmOut_ugL.sd = sum(ConcSmOut_ugL.sd * Volday.L)/sum(Volday.L),
                   ConOxOut_ugL.blk = sum(ConcOxOut_ugL.obs * Volday.L)/sum(Volday.L),
                   ConOxOut_ugL.sd = sum(ConcOxOut_ugL.sd * Volday.L)/sum(Volday.L),
                   ConEsOut_ugL.blk = sum(ConcEsOut_ugL.obs * Volday.L)/sum(Volday.L),
```

```
                    ConEsOut_ugL.sd = sum(ConcEsOut_ugL.sd * Volday.L)/sum(Volday.L),
                    deltaOut.blk = sum(delta.obs * Volday.L)/sum(Volday.L),
                    deltaOut.sd =  sum(delta.sd * Volday.L)/sum(Volday.L)
                    )

# Removed rows - Only for reference (not used)
removed = qmDay %>%
  group_by(DayMoYr) %>%
  filter(n()>1 & is.na(SmOut_ug.obs))

removedBlk = removed %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(VolTot.L = sum(Volday.L),
                    TotHrs = sum(SmpHrs),
                    ConSmOut_ugL.blk = sum(ConcSmOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConSmOut_ugL.sd = sum(ConcSmOut_ugL.sd * Volday.L)/sum(Volday.L),
                    ConOxOut_ugL.blk = sum(ConcOxOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConOxOut_ugL.sd = sum(ConcOxOut_ugL.sd * Volday.L)/sum(Volday.L),
                    ConEsOut_ugL.blk = sum(ConcEsOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConEsOut_ugL.sd = sum(ConcEsOut_ugL.sd * Volday.L)/sum(Volday.L),
                    deltaOut.blk = sum(delta.obs * Volday.L)/sum(Volday.L),
                    deltaOut.sd =  sum(delta.sd * Volday.L)/sum(Volday.L)
                    )


# Get all non-duplicated days. do average to get same variables
ndup = qmDay %>%
  group_by(DayMoYr) %>%
  filter(n()==1)

qmBlk_nDup = ndup %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(VolTot.L = sum(Volday.L),
                    TotHrs = sum(SmpHrs),
                    ConSmOut_ugL.blk = sum(ConcSmOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConSmOut_ugL.sd = sum(ConcSmOut_ugL.sd * Volday.L)/sum(Volday.L),
                    ConOxOut_ugL.blk = sum(ConcOxOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConOxOut_ugL.sd = sum(ConcOxOut_ugL.sd * Volday.L)/sum(Volday.L),
                    ConEsOut_ugL.blk = sum(ConcEsOut_ugL.obs * Volday.L)/sum(Volday.L),
                    ConEsOut_ugL.sd = sum(ConcEsOut_ugL.sd * Volday.L)/sum(Volday.L),
                    deltaOut.blk = sum(delta.obs * Volday.L)/sum(Volday.L),
                    deltaOut.sd =  sum(delta.sd * Volday.L)/sum(Volday.L)
                    )


# Bind Bulk samples
qmBlk = rbind(qmBlkDup, qmBlk_nDup)

qmBlk90 = qmBlk %>%
  filter(TotHrs >= 21.6)

qmBlk50 = qmBlk %>%
  filter(TotHrs >= 12.0)
```

```
# Get dates/markers column
m <- q %>%
  group_by(DayMoYr) %>%
  dplyr::summarise(SubWeeks = SubWeeks[1])

qmBlk90_obs = merge(qmBlk90, m, by = "DayMoYr", all = T)
qmBlk50_obs = merge(qmBlk50, m, by = "DayMoYr", all = T)

qmBlk90_obs = qmBlk90_obs[order(qmBlk90_obs$DayMoYr), ]
qmBlk50_obs = qmBlk50_obs[order(qmBlk50_obs$DayMoYr), ]

# Natural log (ln) for low concentrations
qmBlk90$lnConSmOut_ugL.blk = log(qmBlk90$ConSmOut_ugL.blk)

mean(qmBlk90$ConSmOut_ugL.blk, na.rm = T)  # = 1.9897
```

```
## [1] 1.989693
```

```
mean(qmBlk90$lnConSmOut_ugL.blk, na.rm = T)  # -0.644
```

```
## [1] -0.644245
```

```
mean(qmBlk90$deltaOut.blk, na.rm = T)  # -30.52
```

```
## [1] -30.519
```

3. Obtain approximate total mass discharged (without interpolation)

```
qmBlk_mass = merge(qmBlk, m, by = "DayMoYr", all = T)
# Interpolate bulk concentrations to obtained a cumulative mass discharged
# Att:!!! not to use concentration as observation point, only as approximate loading totals
# Convert to grams (to match model output)
qmBlk_mass$MassSmOut_g = ((qmBlk_mass$ConSmOut_ugL.blk)/10**6)*qmBlk_mass$VolTot.L

qmBlk_mass$MassSmOut_g = ifelse(is.na(qmBlk_mass$MassSmOut_g), 0,
                                qmBlk_mass$MassSmOut_g)
qmBlk_mass$CumMassSmOut_g = cumsum(qmBlk_mass$MassSmOut_g)



### CALIBRATION DATA FRAME here!!
# Data frame for calibration (removed NAs)
qmBlk_cal = merge(qmBlk90, m, by = "DayMoYr", all = T)
qmBlk_cal$MassSmOut_g = ((qmBlk_cal$ConSmOut_ugL.blk)/10**6)*qmBlk_cal$VolTot.L
qmBlk_cal = subset(qmBlk_cal, !is.na(MassSmOut_g))
```

4. Incorporate Julian days before saving

```
qmBlk90 = merge(time, qmBlk90, by = "DayMoYr", all = T)
qmBlk_mass = merge(time, qmBlk_mass, by = "DayMoYr", all = T)

# Calibration Sests
qmBlk90cal = merge(time, qmBlk_cal, by = "DayMoYr", all = T)

# Loads
loads_g_cal = subset(qmBlk90cal, !is.na(MassSmOut_g))
```

```
loads_g_cal = loads_g_cal[, c("Jdays", "MassSmOut_g")]
names(loads_g_cal) =  c("Jdays", "smloads.g")

# Concentration (outlet)
conc_out_cal = subset(qmBlk90cal, !is.na(ConSmOut_ugL.blk))
conc_out_cal = conc_out_cal[, c("Jdays", "ConSmOut_ugL.blk")]
names(conc_out_cal) = c("Jdays", "ug.L")

# Delta outlet
d13c_out_cal = subset(qmBlk90cal, !is.na(deltaOut.blk))
d13c_out_cal = d13c_out_cal[, c("Jdays", "deltaOut.blk")]
names(d13c_out_cal) = c("Jdays", "d13C")

if (F) {
  # write.csv(qmBlk, "qmBlk_R.csv", row.names = F) # , sep = ";", dec = ".")
  write.table(loads_g_cal, "BEACH_R/lds_out_cal.tss", sep="\t", row.names = F, col.names = T)
  write.table(conc_out_cal, "BEACH_R/conc_out_cal.tss", sep="\t", row.names = F, col.names = T)
  write.table(d13c_out_cal, "BEACH_R/d13c_out_cal.tss", sep="\t", row.names = F, col.names = T)
}
```

Get S-met Concentrations ($\geq$ 90% of day samples) AND Mass export Time Series (TSS)

```
Conc_ugL_tss = qmBlk90[,c("Jdays", "ConSmOut_ugL.blk")]
Conc_ugL_tss$ConSmOut_ugL.blk = ifelse(is.na(Conc_ugL_tss$ConSmOut_ugL.blk), -1e9, Conc_ugL_tss$ConSmOu

Mass_g_tss = qmBlk_mass[,c("Jdays", "MassSmOut_g")]
Mass_g_tss$MassSmOut_g = ifelse(is.na(Mass_g_tss$MassSmOut_g), -1e9, Mass_g_tss$MassSmOut_g)

CumMass_g_tss = qmBlk_mass[,c("Jdays", "CumMassSmOut_g")]
CumMass_g_tss$CumMassSmOut_g = ifelse(is.na(CumMass_g_tss$CumMassSmOut_g), -1e9, CumMass_g_tss$CumMassS
```

Get $\delta^{13}C$ time series (TSS)

```
Delta_out_tss = qmBlk90[,c("Jdays", "deltaOut.blk")]
Delta_out_tss$deltaOut.blk = ifelse(is.na(Delta_out_tss$deltaOut.blk), 1e9, Delta_out_tss$deltaOut.blk)
```

Save time series

```
if (F){
  write.table(Conc_ugL_tss, "BEACH_R/Conc_ugL.tss", sep="\t", row.names = F, col.names = F)
  write.table(Delta_out_tss, "BEACH_R/Delta_out.tss", sep="\t", row.names = F, col.names = F)
}

if (F){
  write.table(Mass_g_tss, "BEACH_R/MassOut_g.tss", sep="\t", row.names = F, col.names = F)
  write.table(CumMass_g_tss, "BEACH_R/CumMass_g.tss", sep="\t", row.names = F, col.names = F)
}
```

Analyse Discharge Monthly Values

```
if (F) {
  qDay$Month <-
  ifelse(qDay$DayMoYr >= as.POSIXct("2016-03-24 00:30:00", tz = "EST") &
           qDay$DayMoYr < as.POSIXct("2016-04-01 00:00:00", tz = "EST"), "March",
         ifelse(qDay$DayMoYr >= as.POSIXct("2016-04-01 00:00:00", tz = "EST") &
                  qDay$DayMoYr < as.POSIXct("2016-05-01 00:00:00", tz = "EST"), "April",
                ifelse(qDay$DayMoYr >= as.POSIXct("2016-05-01 00:00:00", tz = "EST") &
                         qDay$DayMoYr < as.POSIXct("2016-06-01 00:00:00", tz = "EST"), "May",
                       ifelse(qDay$DayMoYr >= as.POSIXct("2016-06-01 00:00:00", tz = "EST") &
                                qDay$DayMoYr < as.POSIXct("2016-07-01 00:00:00", tz = "EST"), "June", "
                              )
                       )
                )
         )

dischSumm <- qDay %>%
  group_by(Month) %>%
  dplyr::summarize(MeanQmm = mean(Q.mm),
                   SdevQmm = sd(Q.mm),
                   MeanQm3 = mean(Q.m3))

}
```