BEACH Hydro Data Preparation

PAZ

31/01/2018

Purpose

Generate BEACH calibration data with:

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

Lab parameters and field constants

```
if (MAC) {
  path = file.path("/Users/DayTightChunks/Documents/PhD/HydrologicalMonitoring")
} else {
  path = file.path("D:/Documents/these_pablo/Alteckendorf2016/HydrologicalMonitoring")
}
source(file.path(path, "global.R"))
```

Packages

```
# Plotting functions
library("scales")
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 3.3.3
```

```
library("dplyr")
library("reshape")
library("zoo") # na.approx()
```

Working directory

```
# setwd("D:/Documents/these_pablo/Alteckendorf2016/R")
# setwd("/Users/DayTightChunks/Documents/PhD/Routput/Alteck/R")
getwd()
```

```
## [1] "D:/Documents/these_pablo/Models/BEACH2016/DataInput/Tables/DataSource"
```

Rainfall

Note: Not for TSS This was not the source for BEACH input, this was only to analyse Alteck's Pluviometer, which has only rainfall from March 2016 onward.

```
rain = read.csv2(file.path(path, "Data/sixMinutePluvioAlteck2016.csv"), header = F)

head(rain)
```

```
##                  V1  V2
## 1 25/03/2016 05:38 0.2
## 2 25/03/2016 05:44 0.0
## 3 25/03/2016 05:50 0.0
## 4 25/03/2016 05:56 0.0
## 5 25/03/2016 06:02 0.0
## 6 25/03/2016 06:08 0.0
```

```
rain$V1 <- as.character(rain$V1)
rain$Date = as.POSIXct(strptime(rain$V1,
                                "%d/%m/%Y %H:%M", tz="EST") )

rain$DayMoYr = as.POSIXct(strptime(rain$V1,
                                   "%d/%m/%Y", tz="EST")  )

# Check number of NA values
CHECKO = FALSE
if (CHECKO){
  sum(is.na(rain$Date))
  naDates = rain[is.na(rain$Date == TRUE),]
}

rainDay <- rain %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(Rain.mm = sum(V2))
```

Prepare Rainfall Time Series (TSS)

```
if (FALSE) {
  rainDay$time = seq.int(nrow(rainDay))
  rain_tss = rainDay[,c("time", "Rain.mm")]
  #rain_tss = rbind(c("2016-03-25 to 2016-07-11", NA), rain_tss)
  write.table(rain_tss, "BEACH_R/rain_mmday.tss", sep="\t", row.names = F)
}
```

Analyse Rainfall Monthly Values

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

```
                      )
         )

rainDay$Wet = ifelse(rainDay$Rain.mm > 0, 1, 0)
rainDay$Dry = ifelse(rainDay$Rain.mm == 0, 1, 0)

rainSumm <- rainDay %>%
  group_by(Month) %>%
  dplyr::summarize(WetDays = sum(Wet),
                   DryDays = sum(Dry),
                   MeanP = mean(Rain.mm),
                   StdP = sd(Rain.mm),
                   TotP = sum(Rain.mm))

rainSumm$Prct = rainSumm$WetDays/(rainSumm$WetDays+rainSumm$DryDays)
```

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", "We
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"))

CHECKO = F
if (CHECKO){
  sum(is.na(q$Date))
  naDates = q[is.na(q$Date == TRUE),]

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

Merge 2-min disch with sub-weekly markers

- Convert to mass (and corresponding SD) 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] "Conc.mug.L"              "Conc.SD"
## [13] "OXA_mean"                "OXA_SD"
## [15] "ESA_mean"                "ESA_SD"
## [17] "N.x"                     "diss.d13C"
## [19] "SD.d13C"                 "MES.mg.L"
## [21] "MES.sd"                 "MO.mg.L"
## [23] "Conc.Solids.mug.gMES"   "Conc.Solids.ug.gMES.SD"
## [25] "SmetOut_ug.obs"         "SmetOut_ug.sd"
## [27] "OxaOut_ug.obs"          "OxaOut_ug.sd"
## [29] "EsaOut_ug.obs"          "EsaOut_ug.sd"
## [31] "SmetSS_ug.obs"          "SmetSS_ug.sd"
## [33] "MassDelta.obs"          "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
qmDay <- qm %>%
  group_by(DayMoYr, SubWeeks) %>%
  dplyr::summarize(Volday.L = sum(Vol.L),
                   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
allDup = qmDay %>%
  group_by(DayMoYr) %>%
  filter(n()>1)

# Assume same delta on the same day
```

```
deltasDup = allDup %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(delta.obs = mean(delta.obs, na.rm = T),
                   delta.sd = mean(delta.sd, na.rm = T))

deltasDup$delta.obs = ifelse(deltasDup$delta.obs == "NaN", NA, deltasDup$delta.obs)
deltasDup$delta.sd = ifelse(deltasDup$delta.sd == "NaN", NA, deltasDup$delta.sd)

# Delete delta columns on allDup
cols = ncol(allDup)-2
allDup = allDup[, c(1:cols)]

# Invert 1st two rows for na.approx
allDup = allDup[c(2,1:nrow(allDup)), ]

allDup$ConcSmOut_ugL.obs = na.approx(allDup$ConcSmOut_ugL.obs)
allDup$ConcSmOut_ugL.sd = na.approx(allDup$ConcSmOut_ugL.sd)

allDup$ConcOxOut_ugL.obs= na.approx(allDup$ConcOxOut_ugL.obs)
allDup$ConcOxOut_ugL.sd = na.approx(allDup$ConcOxOut_ugL.sd)

allDup$ConcEsOut_ugL.obs = na.approx(allDup$ConcEsOut_ugL.obs)
allDup$ConcEsOut_ugL.sd = na.approx(allDup$ConcEsOut_ugL.sd)

allDup = merge(allDup, deltasDup, by = "DayMoYr", all = T)

ndup = qmDay %>%
  group_by(DayMoYr) %>%
  filter(n()==1)


qmDay = rbind.data.frame(ndup, allDup)

qmDay = qmDay[order(qmDay$DayMoYr), ]

# head(dupQm)
```

3. Apply eq. 1 while grouping by day

```
qmBlk = qmDay %>%
  group_by(DayMoYr) %>%
  dplyr::summarize(VolTot.L = sum(Volday.L),
                   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)
                   )
```

4. Merge markers

```
m <- q %>%
  group_by(DayMoYr) %>%
  dplyr::summarise(SubWeeks = SubWeeks[1])

qmBlk = merge(qmBlk, m, by = "DayMoYr")
```

Prepare Volume Discharged Time Series (TSS)

```
names(qmBlk)
```

```
##  [1] "DayMoYr"          "VolTot.L"          "ConSmOut_ugL.blk"
##  [4] "ConSmOut_ugL.sd"  "ConOxOut_ugL.blk" "ConOxOut_ugL.sd"
##  [7] "ConEsOut_ugL.blk" "ConEsOut_ugL.sd"  "deltaOut.blk"
## [10] "deltaOut.sd"      "SubWeeks"
```

```
# Vol.L/day
qmBlk$time = seq.int(nrow(qmBlk))

qmBlk$VolTot.m3 = qmBlk$VolTot.L/10^3
Volm3_tss = qmBlk[,c("time", "VolTot.m3")]
write.table(Volm3_tss, "BEACH_R/Vol_m3day.tss", sep="\t", row.names = F)




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)

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

}
```