# Observed Data Prep for Model Analysis - Soils

*PAZ*

*31/01/2018*

## Purpose

Generate BEACH soil calibration data with:

- Time.csv (Julian day marker)
- Composite soils (MonitoringScopeSoils_R.csv)
- Detailed soils (DetailConc.csv & DetailIsotopes.csv)

```
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"))
}
# Lab parameters and field constants
source(file.path(path, "global.R"))
```

## Packages

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

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

library("plyr")
library("dplyr")
```

### 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"
```

## Get composites and convert transect name

```
s = read.csv2(file.path(path, "Data/MonitoringScopeSoils_R.csv"))
s$Transect = as.character(s$Transect)
s$Transect = ifelse(s$Transect == 'T', 'V', s$Transect)
#q$Date = as.POSIXct(strptime(q$DateCheck, "%d/%m/%Y %H:%M", tz="EST"))
s$DayMoYr = as.POSIXct(strptime(s$Date.Soil, "%d/%m/%Y", tz="EST"))
s = subset(s, s$Wnum > 0)
```

### Merge Transects with Time (Julian Days), ouput as TSS for Nash computation

```
north = subset(s, Transect == "N")
valley = subset(s, Transect == "V")
south = subset(s, Transect == "S")

mean(north$Conc.mug.g.dry.soil, na.rm = T)
```

```
## [1] 2.090009
```

```
mean(valley$Conc.mug.g.dry.soil, na.rm = T)
```

```
## [1] 2.498216
```

```
mean(south$Conc.mug.g.dry.soil, na.rm = T)
```

```
## [1] 2.454024
```

```
mean(north$comp.d13C, na.rm = T)
```

```
## [1] -30.3303
```

```
mean(valley$comp.d13C, na.rm = T)
```

```
## [1] -29.98472
```

```
mean(south$comp.d13C, na.rm = T)
```

```
## [1] -29.86364
```

```
njd = merge(time, north, by = "DayMoYr", all = T)
vjd = merge(time, valley, by = "DayMoYr", all = T)
sjd = merge(time, south, by = "DayMoYr", all = T)

norConc_tss = njd[, c("Jdays", "Conc.mug.g.dry.soil")]
valConc_tss = vjd[, c("Jdays", "Conc.mug.g.dry.soil")]
souConc_tss = sjd[, c("Jdays", "Conc.mug.g.dry.soil")]
```

```
norConc_tss$Conc.mug.g.dry.soil = ifelse(is.na(norConc_tss$Conc.mug.g.dry.soil),
                                         -1e9, norConc_tss$Conc.mug.g.dry.soil)
valConc_tss$Conc.mug.g.dry.soil = ifelse(is.na(valConc_tss$Conc.mug.g.dry.soil),
                                         -1e9, valConc_tss$Conc.mug.g.dry.soil)
souConc_tss$Conc.mug.g.dry.soil = ifelse(is.na(souConc_tss$Conc.mug.g.dry.soil),
                                         -1e9, souConc_tss$Conc.mug.g.dry.soil)

norDelta_tss = njd[, c("Jdays", "comp.d13C")]
valDelta_tss = vjd[, c("Jdays", "comp.d13C")]
souDelta_tss = sjd[, c("Jdays", "comp.d13C")]

norDelta_tss$comp.d13C = ifelse(is.na(norDelta_tss$comp.d13C),
                                1e9, norDelta_tss$comp.d13C)
valDelta_tss$comp.d13C = ifelse(is.na(valDelta_tss$comp.d13C),
                                1e9, valDelta_tss$comp.d13C)
souDelta_tss$comp.d13C = ifelse(is.na(souDelta_tss$comp.d13C),
                                1e9, souDelta_tss$comp.d13C)

if (SAVE) {
  write.table(njd, "BEACH_R/north.tss", sep="\t", row.names = F)
  write.table(vjd, "BEACH_R/valley.tss", sep="\t", row.names = F)
  write.table(sjd, "BEACH_R/south.tss", sep="\t", row.names = F)
  write.table(norConc_tss, "BEACH_R/northConc.tss", sep="\t", row.names = F)
  write.table(valConc_tss, "BEACH_R/valleyConc.tss", sep="\t", row.names = F)
  write.table(souConc_tss, "BEACH_R/southConc.tss", sep="\t", row.names = F)
  write.table(norDelta_tss, "BEACH_R/northDelta.tss", sep="\t", row.names = F)
  write.table(valDelta_tss, "BEACH_R/valleyDelta.tss", sep="\t", row.names = F)
  write.table(souDelta_tss, "BEACH_R/southDelta.tss", sep="\t", row.names = F)
}
```

### Get all-composites dataframe for 1:1 calibration graphs

```
# Merge complete dataframe with time
st = merge(time, s, by = "DayMoYr", all = T)
st$IDcal = paste(st$Transect, st$Jdays, sep = "-")

conc_tss = st[, c("Jdays", "Conc.mug.g.dry.soil")]
delta_tss = st[, c("Jdays", "comp.d13C")]

conc_cal = st[, c("Jdays", "Transect", "IDcal", "Conc.mug.g.dry.soil", "Conc.ComSoil.SD")]
names(conc_cal) <- c("Jdays", "Transect", "IDcal", "ug.g", "ug.g.SD")
conc_cal = subset(conc_cal, !is.na(conc_cal$ug.g))
delta_cal = st[, c("Jdays", "Transect", "IDcal", "comp.d13C", "comp.d13C.SD")]
names(delta_cal) <- c("Jdays", "Transect", "IDcal", "d13C", "d13C.SD")
delta_cal = subset(delta_cal, !is.na(delta_cal$d13C))

if (F) {
  write.table(conc_cal, "BEACH_R/conc_comp_cal.tss", sep="\t", row.names = F)
  write.table(delta_cal, "BEACH_R/delta_comp_cal.tss", sep="\t", row.names = F)
}
```

# Detail concentration and isotopes

## Steps

- group isotope values by ID
- evaluate standard deviations
- remove unlikely delta values based on concentration
- compute means
- merge with concentrations

```
co = read.csv("DetailConc.csv")
is = read.csv("DetailIsotopes.csv")

if (length(is) == 1) {
  is = read.csv("DetailIsotopes.csv", sep = ";")
}

if (length(co) == 1) {
  co = read.csv("DetailConc.csv")
}

is = subset(is, !is.na(d13C))
```

### Group

```
# Same procedure as in Book 06_MassSoils_Composite
siso <- is[, c("ID", "d13C")]
sumIS = ddply(siso, c("ID"),
              summarise,
              N_detsoil = length(d13C),
              d13C.SD = sd(d13C),
              d13C = mean(d13C)
              )
sumIS = subset(sumIS, N_detsoil > 1 & d13C.SD < 0.8)
```

# Merge conc and isotopes, change Talweg's name, merge with time

```
ci = merge(co, sumIS, by = "ID", all = T)
ci$DayMoYr = as.POSIXct(strptime(ci$Sampled, "%d/%m/%Y", tz="EST"))
ci = subset(ci, !is.na(ci$DayMoYr))

ci$Transect = as.character(ci$Transect)
ci$Transect = ifelse(ci$Transect == 'T', 'V', ci$Transect)

ci$Plot = paste(ci$Transect, ci$Plot, sep = "-")
ci = ci[, c("DayMoYr", "ID", "Transect", "Plot", "ug.g", "d13C", "d13C.SD")]
detailed = merge(time, ci, by = "DayMoYr", all = T)
```

## Create plot subsets for plotting against model TSS

```
N1 = subset(detailed, is.na(Plot) | Plot == "N-1")
N2 = subset(detailed, is.na(Plot) | Plot == "N-2")
N3 = subset(detailed, is.na(Plot) | Plot == "N-3")
N4 = subset(detailed, is.na(Plot) | Plot == "N-4")
N5 = subset(detailed, is.na(Plot) | Plot == "N-5")
N7 = subset(detailed, is.na(Plot) | Plot == "N-7")
N8 = subset(detailed, is.na(Plot) | Plot == "N-8")

T4 = subset(detailed, is.na(Plot) | Plot == "T-4")
T5 = subset(detailed, is.na(Plot) | Plot == "T-5")
T7 = subset(detailed, is.na(Plot) | Plot == "T-7")
T8 = subset(detailed, is.na(Plot) | Plot == "T-8")
T9 = subset(detailed, is.na(Plot) | Plot == "T-9")
T10 = subset(detailed, is.na(Plot) | Plot == "T-10")

S11 = subset(detailed, is.na(Plot) | Plot == "S-11")
S12 = subset(detailed, is.na(Plot) | Plot == "S-12")
S13 = subset(detailed, is.na(Plot) | Plot == "S-13")

nor = list(N1, N2, N3, N4, N5, N7, N8)
val = list(T4, T5, T7, T8, T9, T10)
sou = list(S11, S12, S13)
```

### Save TSS detailed

```
if (SAVE) {

  counter = 0
  for (d in nor) {
    counter = counter + 1
    transect = "N"
    if (counter < 6) {
      name = paste(transect, as.character(counter), ".tss", sep = "")
    }
    else if (counter == 6) {
      counter = counter + 1
      name = paste(transect, as.character(counter), ".tss", sep = "")
    } else {
      name = paste(transect, as.character(counter), ".tss", sep = "")
    }
    write.table(d, paste("BEACH_R/", name, sep = ""), sep="\t", row.names = F)
  }

  counter = 3
  for (d in val) {
    counter = counter + 1
    transect = "V"
    if (counter < 6) {
      name = paste(transect, as.character(counter), ".tss", sep = "")
```

```
    }
    else if (counter == 6) {
      counter = counter + 1
      name = paste(transect, as.character(counter), ".tss", sep = "")
    } else {
      name = paste(transect, as.character(counter), ".tss", sep = "")
    }
    write.table(d, paste("BEACH_R/", name, sep = ""), sep="\t", row.names = F)
  }

  counter = 10
  for (d in sou) {
    counter = counter + 1
    transect = "S"
    name = paste(transect, as.character(counter), ".tss", sep = "")
    write.table(d, paste("BEACH_R/", name, sep = ""), sep="\t", row.names = F)
  }
}
```

## Get all-detailed dataframe for 1:1 calibration graphs

```
detailed$IDcal = paste(detailed$Plot, detailed$Jdays, sep = "-")

conc_det_cal = detailed[, c("Jdays", "Transect", "IDcal", "ug.g")]
conc_det_cal = subset(conc_det_cal, !is.na(conc_det_cal$ug.g))

delta_det_cal = detailed[, c("Jdays","Transect", "IDcal",  "d13C", "d13C.SD")]
delta_det_cal = subset(delta_det_cal, !is.na(delta_det_cal$d13C))

if (SAVE) {
  write.table(conc_det_cal, "BEACH_R/conc_det_cal.tss", sep="\t", row.names = F)
  write.table(delta_det_cal, "BEACH_R/delta_det_cal.tss", sep="\t", row.names = F)
}
```