Metrics Data Points - ES&T

PAZ

10/10/2018

Metrics Observations

This file prepares the data to be ploted onto the metrics figure. Some corrections where necessary to the assumed initial applications, which here are harmonized with the model.

Steps

1. The initial applications where revised in Book **09.3_OpenRayleigh.Rmd**. However, Dutt was considered to have applied S-metolachlor, which is incorrect, and here corrected.
2. The total mass applied is then merged to "time" and the cumulative computed.
3. The Outlet data from book **05_MassDischarged_outlet.Rmd** is used, df = *metrics_outlet_approx.csv*, and merged to plot cumulative export of dissolved SM. Cumulative min and max values are assumed to be the cumulative of the observed value minus and plus the SD of that day, respectively.
4. From **0.2_BEACH_SoilDateBulkCal.Rmd** use *metrics_part1.csv* which is based on *WeeklySoils_Rng.csv*. Here (*metrics_part1.csv*) bulk catchment masses are computed based on the latest approach.

Initial applications, correction to 09.3_OpenRayleigh.Rmd.

```
# Step 1
farm = read.csv2(file.path(path, "Data/FarmerPlots_model.csv"), header = TRUE, dec = ".", sep = ",")
# farm = read.csv2("Data/FarmerPlotsB.csv", header = TRUE, dec = ".") # Speich Late (May App)

farm$Farmer = as.character(farm$Farmer)
farm$onTrans = as.character(farm$onTrans)
farm$Crop = as.character(farm$Crop)
farm$Transect = as.character(farm$Transect)
farm$Date = as.POSIXct(farm$Date, "%d/%m/%Y", tz = "EST")

# Consider only plots on transect (and without Dutt)
farm = subset(farm, onTrans == T & Farmer != "Dutt")

# Model
# Applications Mass
# Product concentration (active ing.)
double = 2.0  # ~ Dosage for corn when growing beet
d_gold = 915  # g/L S-met #
m_gold = 960 # g/L S-met #

# Dosages # L/Ha * 1Ha/1000m2 = L/m2

d_corn = 2.1 * 1 / 10 ** 4  # 2.2 L/Ha * 2 Ha / 10000 m2
```

```r
m_beet = 0.6 * 1 / 10 ** 4
m_corn = 2.0 * 1 / 10 ** 4

m_beet_Friess = 0.6 * 1 / 10 ** 4 * (double) # 0.6 L/Ha * 2 Ha / 10000 m2 = L/m2
m_beet_Mathis = 0.6 * 1 / 10 ** 4 * (double)
m_beet_Burger = 0.6 * 1 / 10 ** 4 * (double)
m_beet_Kopp = 0.6 * 1 / 10 ** 4 * (double)

farm1 = farm
farm1$appMass = ifelse(farm$Farmer == "Friess" & farm$Crop == "Beet",
                       farm$m2*m_beet_Friess*m_gold,
                       ifelse(farm$Farmer == "Friess" & farm$Crop == "Corn",
                              farm$m2*m_corn*m_gold,
                              ifelse(farm$Farmer == "Speich" & farm$Crop == "Corn",
                                     farm$m2*m_corn*m_gold,
                                     ifelse(farm$Farmer == "Schmitt" & farm$Crop == "Corn",
                                            farm$m2*d_corn*d_gold,
                                            ifelse(farm$Farmer == "Burger" & farm$Crop == "Beet",
                                                   farm$m2*m_beet_Burger*m_gold,
                                                   ifelse(farm$Farmer == "Mathis" & farm$Crop == "Beet"


# Step 2
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
detach(package:plyr)
farm_tot = farm1 %>%
  group_by(Date) %>%
  summarize(totMass = sum(appMass))

tapp = merge(time, farm_tot, by.x = "DayMoYr", by.y = "Date", all = T)
tapp[1, c("totMass")] <- 0
tapp$totMass = ifelse(is.na(tapp$totMass), 0, tapp$totMass)
tapp$totMass = cumsum(tapp$totMass)
```

Merge outlet data

```
# Step 3

out = read.csv("BEACH_R/metrics_outlet_approx.csv")
#names(out)
out = out[, c("ti", "DissSmeto.g", "DissSmeto.g.SD")]

out$DayMoYr = as.POSIXct(strptime(out$ti, "%Y-%m-%d", tz="EST"))

out1 = out %>%
  group_by(DayMoYr) %>%
  summarize(outSM.g = sum(DissSmeto.g),
            outSM.g.SD = max(DissSmeto.g.SD))
out1$CumOutSM.g = cumsum(out1$outSM.g)

out1$CumOutSMmin = cumsum(out1$outSM.g - out1$outSM.g.SD)
out1$CumOutSMmax = cumsum(out1$outSM.g + out1$outSM.g.SD)

metrics = merge(tapp, out1, by = "DayMoYr", all = T)

metrics$PrctExp = (metrics$CumOutSM.g/metrics$totMass)*100
metrics$PrctExpMin = (metrics$CumOutSMmin/metrics$totMass)*100
metrics$PrctExpMax = (metrics$CumOutSMmax/metrics$totMass)*100

names(metrics)
```

```
##  [1] "DayMoYr"     "Date"        "Jdays"       "Days"        "totMass"
##  [6] "outSM.g"     "outSM.g.SD"  "CumOutSM.g"  "CumOutSMmin" "CumOutSMmax"
## [11] "PrctExp"     "PrctExpMin"  "PrctExpMax"
```

```
metrics = metrics[, c("DayMoYr", "Jdays", "Days", "totMass", "PrctExp", "PrctExpMin", "PrctExpMax")]
```

Merge soils

Will use Rayleigh (closed system, Elsner's notation), assuming a lab enrichment.

$$ln(\frac{1000 + \delta^{13}C_0 + \Delta\delta^{13}C}{1000 + \delta^{13}C_0}) = (\alpha - 1) \cdot lnf = \frac{\epsilon}{1000} \cdot lnf$$

$$Deg\ [\%] = 100 * [1 - \left(\frac{1000 + \delta^{13}C_0 + \Delta\delta^{13}C}{1000 + \delta^{13}C_0}\right)^{(1000/\epsilon)}]$$

```
soil = read.csv("BEACH_R/metrics_part1.csv")
names(soil)
```

```
##  [1] "ti"                  "WeekSubWeek"
##  [3] "Event"               "Duration.Hrs"
##  [5] "timeSinceApp"        "timeSinceApp.NoSo"
##  [7] "timeSinceApp.N"      "timeSinceApp.T"
##  [9] "timeSinceApp.S"      "diss.d13C"
## [11] "SD.d13C"             "CumOutDiss.g"
## [13] "CumOutFilt.g"        "TotSMout.g"
```

```
## [15] "TotSMout.g.SD"          "MELsm.g"
## [17] "MELsm.g.SD"             "Appl.Mass.g"
## [19] "Appl.Mass.g.OT"         "CumAppMass.g"
## [21] "CumAppMass.g.N"         "CumAppMass.g.T"
## [23] "CumAppMass.g.S"         "CumAppMass.g.OT"
## [25] "CumAppMass.g.N.OT"      "CumAppMass.g.T.OT"
## [27] "CumAppMass.g.S.OT"      "iniCo.ug.g.N"
## [29] "iniCo.ug.g.T"           "iniCo.ug.g.S"
## [31] "CumOutSmeto.g"          "CumOutMELsm.g"
## [33] "MassSoil.g.North"       "MassSoil.g.SD.North"
## [35] "Conc.mug.g.dry.soil.N" "Conc.North.SD"
## [37] "comp.d13C.North"        "comp.d13C.SD.North"
## [39] "ID.N"                   "Area.N"
## [41] "Area.T"                 "Area.S"
## [43] "MassSoil.g.Talweg"      "MassSoil.g.SD.Talweg"
## [45] "Conc.mug.g.dry.soil.T" "Conc.Talweg.SD"
## [47] "comp.d13C.Talweg"       "comp.d13C.SD.Talweg"
## [49] "ID.T"                   "MassSoil.g.South"
## [51] "MassSoil.g.SD.South"    "Conc.mug.g.dry.soil.S"
## [53] "Conc.South.SD"          "comp.d13C.South"
## [55] "comp.d13C.SD.South"     "ID.S"
## [57] "DD13C.North"            "DD13C.Talweg"
## [59] "DD13C.South"            "CatchMassSoil.g"
## [61] "CatchMassSoil.g.SD"     "BulkCatch.d13"
## [63] "BulkCatch.d13.SD"       "DD13.Bulk"
## [65] "Area.Catchment"         "BulkCatch.Conc"
## [67] "iniCo.Bulk"             "DayMoYr"
## [69] "Blk.Conc"               "Blk.Conc.mean"
## [71] "Blk.Conc.SD"            "Blk.d13C"
## [73] "Bulk.d13C.SDmean"       "Bulk.d13C.SD"
```

```
soil$DayMoYr = as.POSIXct(strptime(soil$ti, "%Y-%m-%d", tz="EST"))

soil = soil[4:nrow(soil), c("DayMoYr",
                            "Conc.mug.g.dry.soil.N", "Conc.mug.g.dry.soil.T", "Conc.mug.g.dry.soil.S",
                            "Conc.North.SD", "Conc.Talweg.SD", "Conc.South.SD",
#                            "Area.N", "Area.T", "Area.S",
                            "Blk.d13C", "Bulk.d13C.SD"
                            )]

soil = soil[complete.cases(soil$Conc.mug.g.dry.soil.N), ]

soil_density$pb_ave = soil_density$pb_ave*10^6 # soil density [g/m3]
soil_density$pb_min = soil_density$pb_min*10^6 # soil density [g/m3]
soil_density$pb_max = soil_density$pb_max*10^6 # soil density [g/m3]

soil = merge(soil, soil_density, by="DayMoYr")

# Areas from Book 0.2_BEACH_SoilDataBulkCal.Rmd
soil$Area.N = 139266.25
soil$Area.T = 43713.44
soil$Area.S = 133175.05

epsilon_model = -2.5
```

```r
soil$DD13C.comp = soil$Blk.d13C - d13Co
soil$Blab  <- (1- ((1000 + d13Co + soil$DD13C.comp)/(1000+ d13Co))^(1000/epsilon_lab) )*100
soil$Bmin  <- (1- ((1000 + d13Co + soil$DD13C.comp)/(1000+ d13Co))^(1000/epsilon_lab-0.5) )*100
soil$Bmax  <- (1- ((1000 + d13Co + soil$DD13C.comp)/(1000+ d13Co))^(1000/epsilon_lab+0.5) )*100
soil$Bmod  <- (1- ((1000 + d13Co + soil$DD13C.comp)/(1000+ d13Co))^(1000/epsilon_model) )*100




# Fresh tilled
rho1 = 0.98*10^6 # soil density [g/m3] Leaching experiment, "freshly tilled" soil
#depth1 = 0.007 # [m] # Fresh tilled
depth1 = 0.01

#soil$MassN.g = soil$Conc.mug.g.dry.soil.N/10**6*rho1*soil$Area.N*depth1
#soil$MassN.g.SD = soil$Conc.North.SD/10**6*rho1*soil$Area.N*depth1
soil$MassN.g = soil$Conc.mug.g.dry.soil.N/10**6*soil$pb_ave*soil$Area.N*depth1
soil$MassN.g.SD = soil$Conc.North.SD/10**6*soil$pb_ave*soil$Area.N*depth1

#soil$MassT.g = soil$Conc.mug.g.dry.soil.T/10**6*rho1*soil$Area.T*depth1
#soil$MassT.g.SD = soil$Conc.Talweg.SD/10**6*rho1*soil$Area.T*depth1
soil$MassT.g = soil$Conc.mug.g.dry.soil.T/10**6*soil$pb_ave*soil$Area.T*depth1
soil$MassT.g.SD = soil$Conc.Talweg.SD/10**6*soil$pb_ave*soil$Area.T*depth1

#soil$MassS.g = soil$Conc.mug.g.dry.soil.S/10**6*rho1*soil$Area.S*depth1
#soil$MassS.g.SD = soil$Conc.South.SD/10**6*rho1*soil$Area.S*depth1
soil$MassS.g = soil$Conc.mug.g.dry.soil.S/10**6*soil$pb_ave*soil$Area.S*depth1
soil$MassS.g.SD = soil$Conc.South.SD/10**6*soil$pb_ave*soil$Area.S*depth1

soil$Rem = soil$MassN.g + soil$MassT.g + soil$MassS.g


soil$RemMin = (soil$MassN.g - soil$MassN.g.SD) +
  (soil$MassT.g - soil$MassT.g.SD) +
  (soil$MassS.g - soil$MassS.g.SD)
soil$RemMax = (soil$MassN.g + soil$MassN.g.SD) +
  (soil$MassT.g + soil$MassT.g.SD) +
  (soil$MassS.g + soil$MassS.g.SD)

# Previously computed masses
if (FALSE){

old_masses = c("MassSoil.g.North", "MassSoil.g.SD.North",
               "MassSoil.g.Talweg", "MassSoil.g.SD.Talweg",
               "MassSoil.g.South", "MassSoil.g.SD.South")

soil$Rem = soil$MassSoil.g.North + soil$MassSoil.g.Talweg + soil$MassSoil.g.South
soil$MassSoil.g.SD.South = na.approx(soil$MassSoil.g.SD.South)

soil$RemMin = (soil$MassSoil.g.North - soil$MassSoil.g.SD.North) +
  (soil$MassSoil.g.Talweg - soil$MassSoil.g.SD.Talweg) +
  (soil$MassSoil.g.South - soil$MassSoil.g.SD.South)
soil$RemMax = (soil$MassSoil.g.North + soil$MassSoil.g.SD.North) +
```

```
  (soil$MassSoil.g.Talweg + soil$MassSoil.g.SD.Talweg) +
  (soil$MassSoil.g.South + soil$MassSoil.g.SD.South)
}

soil = soil[, c("DayMoYr", "Blab","Bmin", "Bmax", "Bmod", "Rem", "RemMin", "RemMax")]

metrics = merge(metrics, soil, by="DayMoYr", all = T)

metrics$PrcRem = (metrics$Rem/metrics$totMass)*100
metrics$PrcRemMin = (metrics$RemMin/metrics$totMass)*100
metrics$PrcRemMax = (metrics$RemMax/metrics$totMass)*100

names(metrics)
```

```
##  [1] "DayMoYr"    "Jdays"      "Days"       "totMass"    "PrctExp"
##  [6] "PrctExpMin" "PrctExpMax" "Blab"       "Bmin"       "Bmax"
## [11] "Bmod"       "Rem"        "RemMin"     "RemMax"     "PrcRem"
## [16] "PrcRemMin"  "PrcRemMax"
```

```
metrics = metrics[, c("DayMoYr", "Days", "totMass", "PrctExp", "PrctExpMin", "PrctExpMax",
                      "PrcRem", "PrcRemMin", "PrcRemMax", "Blab", "Bmin", "Bmax", "Bmod")]

metrics = subset(metrics, Days >= 0 & Days <= 115)

write.csv(metrics,
          'BEACH_R/Metrics_R.csv', row.names = F)

write.csv(metrics,
          file.path(model_obs, "Metrics_R.csv"), row.names = F)
```