Rayleigh Calculations

PAZ

06/04/2017

Introduction

Degradation extent is calculated based on closed and open system Rayleigh equations

Packages

```r
library(sm)
library(vioplot)

library(dplyr)
library(tidyr)
library(zoo)
library(reshape)
library(ggplot2)
library("ggrepel")

library("plotly")
library("cowplot")
library("gridExtra")
library("Cairo")
library("GGally")
library("scales")

library("plotKML")

# Stats
library("vegan")
library("cluster")

# Saving a xlxs file
# library(xlsx)
```

Lab parameters

```r
# Initial signature measured in tank
initialDelta = d13Co = -32.253

# Define initial concentration (for Raleigh plots)
#Co <- 8 # ug/g dry soil (based on Corn applications)
# Co <- 6.53 # ug/g dry soil (based on Max conc. measured in soils)
# Note: Each transect now has individual starting concentration
# Reference values
# Ehssan values:
epsilon_max = -1.8
epsilon_min = -2.6
epsilon_mean= -2.2 # ± 0.4

epsilon_lab = epsilon_mean

# Field values, after dilution correction (Van Breukelen 2008):
# Calculated in Book 9.1
epsilonField_max = -1.7 + 0.33
epsilonField_min = -1.7 - 0.33
epsilonField_mean = -1.7 # ± 0.33
```

Soils

```
VERTICAL = F
if (VERTICAL) {
  # Mixed transects in one column (NO bulk)
  soils = read.csv2("Data/WeeklySoils_Rng.csv",
                        na.strings=c('#DIV/0!', '', 'NA'), header = TRUE)
  soils$Date.ti <- as.POSIXct(strptime(soils$Date.ti,
                                          "%Y-%m-%d %H:%M", tz="EST")) # csv typos, option 1
  sum(is.na(soils$Date.ti)) == 0

  names(soils)

  keepSoils <- c(
    "ID","Transect","Wnum",  "Date.ti",
    "Conc.mug.g.dry.soil", "Conc.ComSoil.SD" ,
    "comp.d13C", "comp.d13C.SD", "comp.IMP.d13C",
    "DD13C.comp"
  )
} else {
  soils = read.csv2("Data/MassBalance_R.csv",
                      na.strings=c('#DIV/0!', '', 'NA'), header = TRUE)
  names(soils)
  soils$ti <- as.POSIXct(strptime(soils$ti, "%Y-%m-%d %H:%M", tz="EST"))
  colnames(soils)[colnames(soils) == "ti"] <- "Date.ti"

  keepSoils <- c(
    "Date.ti", "WeekSubWeek", "ID.N",
    "diss.d13C", "SD.d13C",
    "CumAppMass.g" ,
    "CatchMassSoil.g", "CatchMassSoil.g.SD",
    "comp.d13C.North" , "comp.d13C.SD.North",
    "comp.d13C.Talweg", "comp.d13C.SD.Talweg",
    "comp.d13C.South" , "comp.d13C.SD.South",
    "BulkCatch.d13", "BulkCatch.d13.SD",
    "Conc.mug.g.dry.soil.N", "Conc.mug.g.dry.soil.T", "Conc.mug.g.dry.soil.S",
    "BulkCatch.Conc"
  )
}
# Test
sum(is.na(soils$Date.ti)) == 0
```

```
## [1] TRUE
```

```
soils <- soils[, colnames(soils) %in% keepSoils]
```

Rayleigh (closed system, Elsner's notation)

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

were,

$$f = \frac{C_t}{C_0}$$

Accounting for dilution

The Rayleigh equation above assumes that f reflects solely reduction in concentrations due to degradation and should thus be expressed as $f_{degradation}$. Accounting for dilution processes, the remaining fraction that is measured in the field sample becomes then f_{total}, where:

$$f_{total} = f_{degradation} \cdot f_{dilution}$$

Following Van Breukelen (2007),

$$f_{degradation} = f_{total} \cdot F$$

where the dilution factor F (i.e. the number of times the source volume has become diluted at the observation location) can be calculated if ϵ_{lab} is known:

$$F = e^{(\Delta/\epsilon_{lab} - lnf_{total})}$$

were,

$$\Delta = 1000 \cdot ln\Big(\frac{10^{-3}\delta_t^{13}C + 1}{10^{-3}\delta_0^{13}C + 1}\Big)$$

# Soils

Initial soil concentration(s)

```
# soils$iniCo <- ifelse(soils$Transect == "N", 4.82,
#                          ifelse(soils$Transect == "T", 5.32,
#                                 ifelse(soils$Transect == "S", 6.53, NA)))
iniCo <- 12.17
soils$iniCo <- 12.17 # Average for plots with application 1+ day
# soils$iniCo <- 9.96 # Median for plots with application 1+ day
# soils$iniCo <- 22.7 # MAX for plots with application 1+ day

# Based on mean plot Day +1, by Transect
# soils$iniCo <- ifelse(soils$Transect == "N", 16.2,
#                        ifelse(soils$Transect == "T", 8.9,
#                               ifelse(soils$Transect == "S", 10.0, NA)))
```

Dilution factor (F)

```
if (VERTICAL) {
  soils$ftot <- soils$Conc.mug.g.dry.soil/soils$iniCo

  # Van Breukelen notation
  soils$Delta <- 1000*log( (10^-3*soils$comp.d13C+1)/(10^-3*d13Co+1) )

  # Elsner notation
  # soils$DeltaX <- log(1000+d13Co+soils$DD13C.comp)/(1000+d13Co)

  soils$Fdil =
    exp( soils$Delta/epsilon_lab -log(soils$ftot) )
  median(soils$Fdil, na.rm = T)

  # Fdil < 1, otherwise this
  soils$Fdil <- ifelse(soils$Fdil < 1, NA, soils$Fdil)

  # In case lab E is not appropriate, an adjustment may be necessary
  adj_epsilon = -3.5

  #soils$FdilAdj =
  #  exp( soils$Delta/adj_epsilon -log(soils$ftot) )
} else {
  soils$ftot.Bulk <- soils$BulkCatch.Conc/soils$iniCo
  # Van Breukelen notation
  soils$Delta <- 1000*log( (10^-3*soils$BulkCatch.d13+1)/(10^-3*d13Co+1) )
  soils$Fdil =
    exp( soils$Delta/epsilon_lab -log(soils$ftot.Bulk) )
  median(soils$Fdil, na.rm = T)
  soils$Fdil <- ifelse(soils$Fdil < 1, NA, soils$Fdil)
}
```

We can now obtain $f_{dilution}$ and $f_{degradation}$:

```
if (VERTICAL){
  soils$fdil <- 1/soils$Fdil
  soils$fdeg <- soils$ftot * soils$Fdil

  #soils$fdilAdj <- 1/soils$FdilAdj
  #soils$fdegAdj <- soils$ftot * soils$FdilAdj

  soils$Dprct <- (1- soils$fdil)*100
  #soils$DprctAdj <- (1- soils$fdilAdj)*100

  soils$Bprct <- (1-soils$fdeg)*100
  #soils$BprctAdj <- (1-soils$fdegAdj)*100

  soils$Tprct <- (1-soils$ftot)*100
} else {
  soils$fdil.Bulk <- 1/soils$Fdil
  soils$fdeg.Bulk <- soils$ftot.Bulk * soils$Fdil

  #soils$fdilAdj <- 1/soils$FdilAdj
  #soils$fdegAdj <- soils$ftot * soils$FdilAdj

  soils$Dprct.Bulk <- (1- soils$fdil.Bulk)*100
  #soils$DprctAdj <- (1- soils$fdilAdj)*100

  soils$Bprct.Bulk <- (1-soils$fdeg.Bulk)*100
  #soils$BprctAdj <- (1-soils$fdegAdj)*100

  soils$Tprct.Bulk <- (1-soils$ftot.Bulk)*100
}
```

As wel as the breakdown ($B*$) and dilution factors ($D*$):

$$B^* = \frac{log(f_{dil})}{log(f_{tot})}$$

```
if (VERTICAL){
  soils$Dstar = log(soils$fdil)/log(soils$ftot)
  soils$Bstar = log(soils$fdeg)/log(soils$ftot)

  soils$Dstar  <- ifelse(soils$Dstar< 0, NA, soils$Dstar)
  soils$Bstar  <- ifelse(soils$Bstar> 1, NA, soils$Bstar)

  #soils$DstarAdj = log(soils$fdilAdj)/log(soils$ftot)
  #soils$BstarAdj = log(soils$fdegAdj)/log(soils$ftot)
} else {
  soils$Dstar = log(soils$fdil.Bulk)/log(soils$ftot.Bulk)
  soils$Bstar = log(soils$fdeg.Bulk)/log(soils$ftot.Bulk)

  soils$Dstar  <- ifelse(soils$Dstar< 0, NA, soils$Dstar)
  soils$Bstar  <- ifelse(soils$Bstar> 1, NA, soils$Bstar)
}
```

The relationship $D * /B*$ related the extent of dilution relative to degradation, and can be obtained by:

```
soils$DB = soils$Dstar/soils$Bstar
soils$DB <- ifelse(soils$DB < 0, NA, soils$DB)

#soils$DBAdj = soils$DstarAdj/soils$BstarAdj

median(soils$DB, na.rm = T)
```

```
## [1] 0.8329408
```

```
#median(soils$DBAdj)
```

Calculating a field enrichment after correcting for via breakdown factor (B^*),

$$\epsilon_{field} = B^* \cdot \epsilon_{lab} = \frac{\Delta}{ln f_{total}}$$

```
soils$Efield <- soils$Bstar * epsilon_lab
# soils$EfieldAdj <- soils$BstarAdj * epsilon_lab

Efield <- median(soils$Efield, na.rm = T)

Efield
```

```
## [1] -1.200451
```

```
sd(soils$Efield, na.rm = T)
```

```
## [1] 0.3881709
```

```
# median(soils$EfieldAdj)
```

Save soils for Bar Plots

```
# Check correct:
soils$Etrue =  soils$Efield/soils$Bstar
median(soils$Etrue, na.rm = T) # Should be close to -2.2, YES.
```

```
## [1] -2.2
```

```
#write.csv2(soils,
#           'Data/Rayleigh_Soils.csv', row.names = F)

write.csv2(soils,
           'Data/Rayleigh_Soils.csv', row.names = F)
```

In Van Breuklen, the degraded and diluted fractions are plotted against each other. They find that in the fringe of the plume, where more dilution occurs, also more degradation occurs, likely associated to oxidant availability and lower toxicity levels.

```
if (VERTICAL){
  # Van Breuklen plot this
  soils$degY = -log(soils$fdeg)
  soils$dilX = -log(soils$fdil)


  DBmodel<-lm( degY ~ dilX , data= soils, subset=(!is.na(Etrue) ))
  cof_DB <- as.numeric(coef(DBmodel)[2])
  # se_DB <- summary(DBmodel)$coef[[4]]*1000
  summary(DBmodel)

  ggplot(data = subset(soils, ( !is.na(fdil) & dilX>0 )), aes(x=dilX, y=degY)) +
    geom_point()

}
```

In contrast, in top soils, a slightly negative but not significant relationship between extent of dilution and degradation was found. This is to be expected as concentration in tops soils are lower than in aquifer systems for legacy contaminants. At lower concentrations, lack of sufficient substrate may be associated to lower bacterial communities capable of degradation.

Waters

Conversion of initial concentration in soils to pore water, assuming all S-met mass is available, may lead to an Fdil_w.

For waters, no dilution factor can be applied. The degraded fraction of off-site transport must be equivalent to catchment soils, as degradation is negligible within an event. However, it may be of interest to obtain the degradation extent that would be computed if only outlet observations were conducted, without knowledge of dilution extent in catchment soils. As such, the fraction degraded in outlet waters will be obtained via ϵ_{lab} and with the closed system Rayleigh equation, without making use of concentration data.

```r
waters = read.csv2("Data/WeeklyHydroContam_R.csv")
waters$ti <- as.POSIXct(strptime(waters$ti, "%Y-%m-%d %H:%M", tz="EST"))
colnames(waters)[colnames(waters) == "ti"] <- "Date.ti"
waters$Events <- factor(waters$Events, levels = unique(waters$Events))
waters$Event <- factor(waters$Event, levels = unique(waters$Event))

names(waters)
```

```
##  [1] "Date.ti"                "WeekSubWeek"
##  [3] "tf"                     "iflux"
##  [5] "fflux"                  "changeflux"
##  [7] "maxQ"                   "minQ"
##  [9] "dryHrs"                 "Duration.Hrs"
## [11] "chExtreme"              "Peak"
## [13] "Markers"                "TimeDiff"
## [15] "AveDischarge.m3.h"      "Volume.m3"
## [17] "Sampled.Hrs"            "Sampled"
## [19] "Conc.mug.L"             "Conc.SD"
## [21] "OXA_mean"               "OXA_SD"
## [23] "ESA_mean"               "ESA_SD"
## [25] "N.x"                    "diss.d13C"
## [27] "SD.d13C"                "se.d13C"
## [29] "N_ngC.diss"             "ngC.mean.diss"
## [31] "ngC.SD.diss"            "MES.mg.L"
## [33] "MES.sd"                 "MO.mg.L"
## [35] "Conc.Solids.mug.gMES"   "Conc.Solids.ug.gMES.SD"
## [37] "N.y"                    "filt.d13C"
## [39] "filt.SD.d13C"           "filt.se.d13C"
## [41] "N_ngC.fl"               "ngC.mean.fl"
## [43] "ngC.SD.fl"              "DD13C.diss"
## [45] "DD13C.filt"             "NH4.mM"
## [47] "TIC.ppm.filt"           "Cl.mM"
## [49] "NO3...mM"               "PO4..mM"
## [51] "NPOC.ppm"               "TIC.ppm.unfilt"
## [53] "TOC.ppm.unfilt"         "ExpMES.Kg"
## [55] "Appl.Mass.g"            "timeSinceApp"
## [57] "Appl.Mass.g.NoSo"       "timeSinceApp.NoSo"
## [59] "CumAppMass.g"           "DissSmeto.mg"
## [61] "DissSmeto.mg.SD"        "DissSmeto.g"
## [63] "DissSmeto.g.SD"         "DissOXA.mg"
## [65] "DissOXA.mg.SD"          "DissOXA.g"
## [67] "DissOXA.g.SD"           "DissESA.mg"
## [69] "DissESA.mg.SD"          "DissESA.g"
## [71] "DissESA.g.SD"           "FiltSmeto.mg"
## [73] "FiltSmeto.mg.SD"        "FiltSmeto.g"
## [75] "FiltSmeto.g.SD"         "TotSMout.mg"
## [77] "TotSMout.mg.SD"         "TotSMout.g"
## [79] "TotSMout.g.SD"          "FracDiss"
## [81] "FracFilt"               "MELsm.g"
## [83] "MELsm.g.SD"             "CumOutDiss.g"
## [85] "CumOutFilt.g"           "CumOutSmeto.g"
## [87] "CumOutMELsm.g"          "BalMassDisch.g"
## [89] "prctMassOut"            "FracDeltaOut"
## [91] "Events"                 "Weeks"
## [93] "Event"
```

```
keepWater <- c(
  "Date.ti", "WeekSubWeek", "Sampled" , "Volume.m3", "AveDischarge.m3.h",
  "Markers" , "TimeDiff", "Duration.Hrs",
  "Conc.mug.L", "Conc.SD" ,
  "OXA_mean", "OXA_SD",
  "ESA_mean", "ESA_SD",
  "diss.d13C", "SD.d13C",
  "filt.d13C", "filt.SD.d13C",
  "DD13C.diss", "DD13C.filt",
  "Appl.Mass.g",  "CumAppMass.g",
  "DissSmeto.g", "DissSmeto.g.SD",
  "DissOXA.g", "DissOXA.g.SD",
  "DissESA.g", "DissESA.g.SD",
  "FiltSmeto.g", "FiltSmeto.g.SD",
  "TotSMout.g",  "TotSMout.g.SD",
  "MELsm.g", "MELsm.g.SD",
  "CumOutDiss.g", "CumOutFilt.g",
  "CumOutSmeto.g", "CumOutMELsm.g",
  "Events"
)
waters <- waters[ , colnames(waters) %in% keepWater]
```

Field Assumptions

Converting soil to volumetric concentrations in soils,

$$C_{soil}\ [\mu g/L_{soil}] = C_{soil}[\frac{\mu g\ _{S-met}}{g_{soil}}] \cdot \rho_{soil}[\frac{g_{soil}}{m^3}] \cdot \frac{1m^3}{10^3 L_{soil}}$$

$$C_{soil}\ [\mu g/L_{H_2O}] = \frac{C_{soil}}{\theta_{sat} + \rho_{soil} \cdot K_d}$$

```
# S-metolachlor Mass [g]
# Conc. [ug/g dry soil] * [g/10^6 ug] * density [g/m3] * depth [m]* A [m2]
# Soil bulk density: 2200 or 0.99? -> Leaching experiments: 0.99 [g/cm3]
rho = 0.99*10^6 # soil density [g/m3]
depth = 0.01 # [m]
theta = 0.4
# Kd = 2.397/10^6 # m3/g (aged)
Kd = 3.99/10^6 #m3/g (fresh)
```

```
# Cosed vs. Open system Rayleigh
OPEN = T

if (OPEN) {
  waters$f.diss <-
  ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_lab))

  waters$f.diss.min <-
    ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_max))

  waters$f.diss.max <-
    ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_min))

  waters$B.diss <- (1-waters$f.diss)*100
  waters$B.diss.max <- (1-waters$f.diss.min)*100
  waters$B.diss.min <- (1-waters$f.diss.max)*100
} else if (!VERTICAL){
  ws <- merge(waters, soils, by = "Date.ti", all = T )

  # Assume Bulk soils conc. doesn't evolve close to event
  ws$BulkCatch.Conc <- na.locf(ws$BulkCatch.Conc)

  # Convert concentrations from mass to vol H2O, assuming linear sorption
  ws$poolCo_w <- (ws$BulkCatch.Conc*rho/10^3)/(theta + rho*Kd)
  ws$iniCo_w <- (iniCo*rho/10^3)/(theta + rho*Kd)

  # f_tot
  # Problem here is initial Co is not applied but, available at time of discharge
  ws$ftot_w <- ws$Conc.mug.L/ws$poolCo_w

  # Van Breukelen notation
  ##########################
  ##########################
  # Do we need to change d13Co to initial at event or initial product ??
  ws$Delta_w <- 1000*log( (10^-3*ws$diss.d13C.x + 1)/(10^-3*d13Co+1) )

  ws$Fdil_w =
    exp( ws$Delta_w/epsilon_lab -log(ws$ftot_w) )
  median(ws$Fdil_w, na.rm = T)

  # Fdil < 1, otherwise this
  ws$Fdil_w <- ifelse(ws$Fdil_w < 1, NA, ws$Fdil_w)

  ws$fdil_w <- 1/ws$Fdil_w
  ws$fdeg_w <- ws$ftot_w * ws$Fdil_w

  ws$Dprct_w <- (1- ws$fdil_w)*100
  #ws$DprctAdj <- (1- ws$fdilAdj)*100

  ws$Bprct_w <- (1-ws$fdeg_w)*100
  #ws$BprctAdj <- (1-ws$fdegAdj)*100

  ws$Tprct_w <- (1-ws$ftot_w)*100
  ws$Dstar_w = log(ws$fdil_w)/log(ws$ftot_w)
  ws$Bstar_w = log(ws$fdeg_w)/log(ws$ftot_w)

  ws$Dstar_w  <- ifelse(ws$Dstar_w< 0, NA, ws$Dstar_w)
  ws$Bstar_W  <- ifelse(ws$Bstar_w> 1, NA, ws$Bstar_w)

  ws$DB_w = ws$Dstar_w/ws$Bstar_w
}
```

Save Waters for Bar Plots

```
write.csv2(waters,
           'Data/Rayleigh_Waters.csv', row.names = F)
```

Van Breukelen, Boris M. 2007. "Quantifying the degradation and dilution contribution to natural attenuation of contaminants by means of an open system Rayleigh equation." *Environ. Sci. Technol.* 41 (14): 4980–5.