

GC_data_transformation.R

Ross

Mon Aug 24 09:56:33 2015

```
#Script for automating data transformation from S&W Lab's GC  
#Created by Keiran Cantilina  
#Lasted edited by Keiran Cantilina on 8/20/2015
```

```
#This script should magically take raw GC files and format them correctly for processing  
#by Allison's Septic_R_code.R script, provided you fill out the description file correctly.
```

```
#It also spits out an error message if a given set from a given day and owner is incomplete (ie. you br  
#If that happens, check the batch descriptions output file to see how far you got/where things got  
#stuck and replace the missing data with NAs or Os.
```

```
#See example file to learn how to format description file correctly
```

```
#Chamber measurements will also be automatically moved to the correct directory if you have  
#them all stored in one folder (chdir).
```

```
#NOTE: DO NOT have an incomplete set as the last entry in the description file!!
```

```
#      The program can even work with out-of-order entries, as long as the last entry isn't a fragment.
```

```
# Clear workspace  
rm(list=ls())
```

```
#Load libraries if just starting  
library(reshape2)  
library(plyr)  
library(ggplot2)  
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.2
```

```
#Set working R directory  
#Keiran: "C://Users//Ross//Desktop//R_code//GC//"  
dir <- "C://Users//Ross//Desktop//R_code//GC//"  
setwd(dir)
```

```
#Set working batch directory  
#Keiran: "C://Users//Ross//Desktop//Septic 2015//Septic 2015//2015_data//Raw data"  
bdir <- "C://Users//Ross//Desktop//Septic 2015//Septic 2015//2015_data//Raw data"
```

```

#Set working template directory
#Keiran: "C://Users//Ross//Desktop//Septic 2015//Septic 2015//2015_data//Templates"
tdir <- "C://Users//Ross//Desktop//Septic 2015//Septic 2015//2015_data//Templates"

#Set working chamber heights directory
#Keiran: "C://Users//Ross//Desktop//R_code//GC//Chamber Height//"
chdir <- "C://Users//Ross//Desktop//R_code//GC//Chamber Height//"

#Load description file
batch_descript <- (read.csv(paste(dir,"\\Batch_descriptions.csv",sep="")))
batch_descript$Transcribed <- "NO"

#Load Headers
headers <- read.csv(paste(bdir,"//Headers.csv", sep=""))

#Load templates
template <- read.csv(paste(tdir,"//Template-yyyyymmdd_septic-homeowner_GAS.csv",sep=""))
template_co2 <- read.csv(paste(tdir,"//Template-yyyyymmdd_septic-homeowner_CO2_standard.csv",sep=""))

## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : incomplete final line found by readTableHeader on 'C://Users//
## Ross//Desktop//Septic 2015//Septic 2015//2015_data//Templates//Template-
## yyyyymmdd_septic-homeowner_CO2_standard.csv'

template_n2o <- read.csv(paste(tdir,"//Template-yyyyymmdd_septic-homeowner_N2O_standard.csv",sep=""))

## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : incomplete final line found by readTableHeader on 'C://Users//
## Ross//Desktop//Septic 2015//Septic 2015//2015_data//Templates//Template-
## yyyyymmdd_septic-homeowner_N2O_standard.csv'

template_ch4 <- read.csv(paste(tdir,"//Template-yyyyymmdd_septic-homeowner_CH4_standard.csv",sep=""))

#Load Batch and add headers
gas <- 6 #sets first gas to n2o
k <- 1 #Counts iterations of the big loop
i <- 1 #Indices
r<- nrow(batch_descript)
while (i<=r&gas!=15){#Loop to evaluate and execute batch description instructions until the number of p
  h <- 1
  while (batch_descript$Transcribed[i]=="YES"&i<r){#Skip entry if it has already been transcribed
    i<- i+1
  }
  batch_ID <- batch_descript$Batch[i]#Read batch name
  batch_ID <- paste(batch_ID, ".csv", sep="")
  batch_rawdata <- (read.csv(paste(bdir,batch_ID,sep="//"),header=FALSE))#Open batch file of given name
  colnames(batch_rawdata) <- headers$Value #Import headers

```

```

#batch_rawdata[is.na(batch_rawdata)] <- 0 #Use if you need to turn NAs into 0s

#Move area data into vector
date <- batch_descript$Collection_date[i]
owner <- batch_descript$House_owner[i]
begin1 <- batch_descript$Range_begin[i]
end1 <- batch_descript$Range_end[i]
set_quant1 <- (end1-begin1+1) #keeps track of how many data points of the set are in the batch (the +
gas_area1 <- batch_rawdata[begin1:end1,gas] #Creates vector containing gas values
std_curve_index <- rep(h,set_quant1) #Sets standard curve index and creates vector
j <- i #sets placeholder index with i's value (because i can change and j keeps track of what i was b
gas_area <- gas_area1 #n2o_area is a common variable between loops
set_quant <- set_quant1

#Housekeeping (Directory creation, file moving, chamber heights file management)
if (dir.exists(paste(dir,"Data-",owner,sep=""))==FALSE) {dir.create(paste(dir, "Data-", owner,"/",sep
if (dir.exists(paste(dir,"Data-",owner,"/",date,sep=""))==FALSE) {dir.create(paste(dir, "Data-", own
if(gas==12){
  batch_descript$Transcribed[i] <- "YES" #Notates file was used so that it doesn't get reused
}
y<-paste(date,"_septic-",owner,"_chamber-heights.csv", sep = "")
w<-list.files(path=chdir, pattern = y) #Searches for applicable chamber heights file
if(is.null(w)==FALSE){#moves chamber heights data to correct directory if it's in the chdir
  file.copy(from=(paste(chdir,w, sep="")),to=(paste(dir,"Data-", owner,"/",date, "/",w,sep="")))
}

#Go to next item with same owner if data isn't complete (remember, this is still within the previous wh
while(set_quant<24) {#there are 24 data points for each set
  h <- h+1 #advance standard curve index
  i <- i+1 #advance search index by one so we don't search the already read entry
  while (owner!=batch_descript$House_owner[i]) {#advances search index until a match for owner name i
    i <- (i+1)
    if(date!=batch_descript$Collection_date[i]) {
      stop(paste("ERROR: Missing samples for", owner, "on date", date)) #Gives error if the next entry
    }
    while (batch_descript$Transcribed[i]=="YES"&i<r){#Skip entry if it has already been transcribed
      i<- i+1
    }
  }
  batch_ID <- batch_descript$Batch[i]
  batch_ID <- paste(batch_ID,".csv",sep="")
  batch_rawdata <- (read.csv(paste(bdir,batch_ID,sep="/"),header=FALSE))
  colnames(batch_rawdata) <- headers$Value
  #batch_rawdata[is.na(batch_rawdata)] <- 0 #Use if you need to turn NAs into 0s
  begin2 <- batch_descript$Range_begin[i]
  end2 <- batch_descript$Range_end[i]
  set_quant2 <- (end2-begin2+1)
  std_curve_index2 <- rep(h,set_quant2)
  set_quant <- (set_quant2+set_quant)
  gas_area2 <- batch_rawdata[begin2:end2,gas]

```

```

std_curve_index <- c(std_curve_index,std_curve_index2) #Adds Std_Curve_Num together
if(batch_descript$Order[i]>batch_descript$Order[j]){
  gas_area <- c(gas_area,gas_area2) #adds next data onto incomplete previous data after checking to
}
if(batch_descript$Order[j]>batch_descript$Order[i]){
  gas_area <- c(gas_area2,gas_area) #adds next data onto incomplete previous data
}
gas_area3 <- gas_area
}

#Move reconstituted data to file and increment/reset indices
template$Area <- gas_area #Put combined vectors into template
template$Std_Curve_Num <- std_curve_index
if (gas==6){gasname <- "N2O"}
if (gas==9){gasname <- "CH4"}
if (gas==12){gasname <- "CO2"}
if (dir.exists(paste(dir,"Data-",owner,sep=""))==FALSE) {dir.create(paste(dir, "Data-", owner,"/",sep=""))}
if (dir.exists(paste(dir,"Data-",owner,"/",date,sep=""))==FALSE) {dir.create(paste(dir, "Data-", owner,"/",date,"_septic-",owner,"_",gasname,sep=""))}
write.csv(template, file=paste(dir, "Data-", owner,"/",date, "/",date,"_septic-",owner,"_",gasname,sep=""),as.is=TRUE)
if(gas==12){
  batch_descript$Transcribed[i] <- "YES" #Notates file was used so that it doesn't get reused
}
gascounter <- gas #Helps the program know if the gas type was changed from the last run of the loop
if(i==nrow(batch_descript)) {gas <- gas+3}{g <- 1}
if(j==(i-1)){g <- i+1} #If an experiment was split sequentially, continue on
if(i>(j+1)){g <- j+1} #If an experiment was not split sequentially, go back to the gap
if(i==j){g <- i+1} #If an experiment was not split, continue on
if(gas>gascounter) {
  g <- 1
}
i <- g #g is a buffer so that all of the above if statements integrate into i at once
k <- k+1 #Counter for multiple sets for a given owner on a given date
}

#Standards data transformation
i <- 1 #Indices
while (i<=nrow(batch_descript)){#Loop to find batch IDs until there are no more
  batch_ID <- batch_descript$Batch[i]#Read batch name
  batch_ID <- paste(batch_ID,".csv",sep="")
  batch_rawdata <- (read.csv(paste(bdir,batch_ID,sep="/"),header=FALSE))#Open batch file of given name
  colnames(batch_rawdata) <- headers$Value #Import headers
  #batch_rawdata[is.na(batch_rawdata)] <- 0 #Use if you need to turn NAs into 0s
  date <- batch_descript$Collection_date[i]
  owner <- batch_descript$House_owner[i]
  if (dir.exists(paste(dir, "Data-", owner,"/",date,sep=""))==FALSE) {dir.create(paste(dir, "Data-", owner,"/",date,"_septic-",owner,"_",date,sep=""))}
  template_n2o$Area1 <- c(batch_rawdata$Area1[1],batch_rawdata$Area1[2],batch_rawdata$Area1[3]) #Move data to template
  template_ch4$Area1 <- c(batch_rawdata$Area2[1],batch_rawdata$Area2[2],batch_rawdata$Area2[3])
  template_co2$Area1 <- c(batch_rawdata$Area3[1],batch_rawdata$Area3[2],batch_rawdata$Area3[3])
  write.csv(template_n2o, file=paste(dir,"Data-", owner,"/",date, "/",date,"_septic-", owner,"_N2O_",date,sep=""),as.is=TRUE)
  write.csv(template_ch4, file=paste(dir,"Data-", owner,"/",date, "/",date,"_septic-",owner,"_CH4_",date,sep=""),as.is=TRUE)
  write.csv(template_co2, file=paste(dir,"Data-", owner,"/",date, "/",date,"_septic-",owner,"_CO2_",date,sep=""),as.is=TRUE)
  i <- i+1
}

```

```
i <- i+1 #Advance index
}

write.csv(batch_descript,row.names=FALSE,file=paste(dir,"Batch_descriptions_output.csv", sep="")) #Help
#You can set the above line to overwrite the batch description file for keeping track of large jobs.

print("DONE") #If you don't see this, something went wrong or I done goofed.

## [1] "DONE"
```