# Purpose: Air Microbiome project- Script to plot OPC data-

#Authors: Enzo Acerbi, Caroline Chenard

#Created: 24 Nov 2014

#Last updated: 26 Nov 2014

if (!file.exists("data")) dir.create("data")

if (!file.exists("src")) dir.create("src")

if (!file.exists("results")) dir.create("results")

if (!file.exists("doc")) dir.create("doc")

library(ggplot2)

library(stringr)

# open the data file (remove first 10 rows)

OPC\_down1 <- read.csv("data/2014\_11\_07\_down\_sec.csv", header=T, skip=10)

OPC\_down2 <- read.csv("data/2014\_11\_14\_down\_sec.csv", header=T, skip=10)

OPC\_down3 <- read.csv("data/2014\_11\_21\_down\_sec.csv", header=T, skip=10)

OPC\_up1 <- read.csv("data/2014\_11\_07\_up\_sec.csv", header=T, skip=10)

OPC\_up2 <- read.csv("data/2014\_11\_14\_up\_sec.csv", header=T, skip=10)

OPC\_up3 <- read.csv("data/2014\_11\_21\_up\_sec.csv", header=T, skip=10)

#NOTE: all this in not necessary with the function strptime

#Date.and.Time column is a string, need to convert it to numeric

#date\_time\_down = OPC\_down$Date.and.Time

#date\_time\_up = OPC\_up$Date.and.Time

#date\_time\_down = levels(date\_time\_down)

#date\_time\_up = levels(date\_time\_up)

#Last part is Date, I am only interested in time. Select the last 8 characters of each fields

#time\_down = vector(length=length(date\_time\_down))

#for (i in 1:length(date\_time\_down)){

# time\_down[i] <- substr(date\_time\_down[i], nchar(date\_time\_down[i])-8+1, nchar(date\_time\_down[i])

# time\_down[i] <- str\_replace\_all(time\_down[i],":", "")

#}

#time\_up = vector(length=length(date\_time\_up))

#for (i in 1:length(date\_time\_up)){

# time\_up[i] <- substr(date\_time\_up[i], nchar(date\_time\_up[i])-8+1, nchar(date\_time\_up[i]))

# time\_up[i] <- str\_replace\_all(time\_up[i],":", "")

#}

#Replace the old Date\_time string with the new string (as numeric)

#OPC\_down$Date.and.time <- as.numeric(time\_down)

#OPC\_up$Date.and.time <- as.numeric(time\_up)

#Convert data.and.time in a format that ggplot2 can understand

tmp = OPC\_down1$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_down1$Date.and.Time = tmp

tmp = OPC\_down2$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_down2$Date.and.Time = tmp

tmp = OPC\_down3$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_down3$Date.and.Time = tmp

tmp = OPC\_up1$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_up1$Date.and.Time = tmp

tmp = OPC\_up2$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_up2$Date.and.Time = tmp

tmp = OPC\_up3$Date.and.Time

tmp = levels(tmp)

tmp = strptime(tmp, "%m/%d/%Y %H:%M:%S")

OPC\_up3$Date.and.Time = tmp

OPC\_down <- rbind(OPC\_down1, OPC\_down2,OPC\_down3)

OPC\_up <- rbind(OPC\_up1, OPC\_up2,OPC\_up3)

# If normalize data needed please check script OPC\_particle\_over\_time.R

# in this case particle reads for upfile will need to change to multiply updata with the following:

1.1942813 0.8311830 1.2458125 0.6759423 0.4685215 0.6162582

# Plot the particle number based on size into 6 different plots

ch1 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y = Ch1.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(1.1942813\*Ch1.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("0.3" \* mu ~ "m") + xlab("") +

theme(legend.position="top") +

scale\_colour\_brewer(palette="Set1")

ch1

ch2 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y =Ch2.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(0.8311830\*Ch2.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("0.5" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#ch2

ch3 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y =Ch3.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(1.2458125\*Ch3.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("1.0" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#ch3

ch4 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y =Ch4.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(0.6759423\*Ch4.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("2.5" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#ch4

ch5 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y =Ch5.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(0.4685215\*Ch5.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("5.0" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#ch5

ch6 <- ggplot() + geom\_point(data = OPC\_down, aes(x=Date.and.Time, y =Ch6.Diff...., colour="down")) + geom\_point(data = OPC\_up, aes(x=Date.and.Time, y =(0.6162582\*Ch6.Diff....), colour="up"))+

theme\_bw(base\_size = 14) +

ylab ("10.0" \* mu ~ "m") + xlab("Time") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#ch6

#arrange plot into 2 rows and 3 columns

library(gridExtra)

grid.arrange(ch1, ch2, ch3, ch4,ch5, ch6, ncol=1, nrow = 6)

# I have a couple of outliers so let's remove them

OPC\_down\_sub <- subset(OPC\_down, OPC\_down$Ch1.Diff....<1500000 & OPC\_down$Ch2.Diff....<100000 & OPC\_down$Ch3.Diff....<10000 & OPC\_down$Ch4.Diff....<4000 & OPC\_down$Ch5.Diff....<200 & OPC\_down$Ch6.Diff....<100)

summary (OPC\_down\_sub)

# Repeat on the OPC\_up data.

OPC\_up\_sub <- subset(OPC\_up, OPC\_up$Ch1.Diff....<1500000 & OPC\_up$Ch2.Diff....<100000 & OPC\_up$Ch3.Diff....<10000 & OPC\_up$Ch4.Diff....<4000 & OPC\_up$Ch5.Diff....<200 & OPC\_up$Ch6.Diff....<100)

summary (OPC\_up\_sub)

#check that the data still look the same

sub\_ch1 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch1.Diff...., colour="down")) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch1.Diff...., colour="up"))+

theme\_bw(base\_size = 12) +

ylab ("0.3" \* mu ~ "m") + xlab("") +

theme(legend.position="top") +

scale\_colour\_brewer(palette="Set1")

sub\_ch1

sub\_ch2 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch2.Diff...., colour="down")) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch2.Diff...., colour="up"))+

theme\_bw(base\_size = 12) +

#scale\_size c(0.1, 0.1)+

ylab ("0.5" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

sub\_ch2

sub\_ch3 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch3.Diff...., colour="down" )) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch3.Diff...., colour="up"))+

theme\_bw(base\_size = 12) +

ylab ("1.0" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#sub\_ch3

sub\_ch4 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch4.Diff...., colour="down")) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch4.Diff...., colour="up"))+

theme\_bw(base\_size = 12) +

ylab ("2.5" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#sub\_ch4

sub\_ch5 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch5.Diff...., colour="down")) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch5.Diff...., colour="up"))+

theme\_bw(base\_size = 12) +

ylab ("5.0" \* mu ~ "m") + xlab("") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

#sub\_ch5

sub\_ch6 <- ggplot() + geom\_point(data = OPC\_down\_sub, aes(x=Date.and.Time, y =Ch6.Diff...., colour="down")) + geom\_point(data = OPC\_up\_sub, aes(x=Date.and.Time, y =Ch6.Diff...., colour="up" ))+

theme\_bw(base\_size = 12) +

ylab ("10.0" \* mu ~ "m") + xlab("Time") +

theme(legend.position="none" ) +

scale\_colour\_brewer(palette="Set1")

sub\_ch6

#arrange plot into 2 rows and 3 columns

library(gridExtra)

grid.arrange(sub\_ch1, sub\_ch2,sub\_ch3,sub\_ch4, sub\_ch5, sub\_ch6, ncol=2, nrow = 3)

geom\_rect(data= rain, aes(xmi, xmax=Trough, ymin=-Inf, ymax=+Inf), fill='pink', alpha=0.2)