SHIME_Cellcounts_Processing

Load Packages

```
# install packages & load libraries

library(dplyr)
library(plyr)
library(ggplot2)
library(scales)
library(tidyr)
library(knitr)
library(stringr)
library(ggpubr)

library("Phenoflow") # for fingerprinting
library("flowViz") # for plotting
library("ggplot2") # for plotting
library("flowAI") # for denoising

library(scales)
library(pwr)
library(readxl)
library(xlsx)
library(vegan)
library(car)
library(mvnormtest)
library(DescTools)
library(ggsignif)

set.seed(777)
```

Load PBS-controls

```
# path = "PBS"
# flowDataPBS <- read.flowSet(path = path, transformation = FALSE, pattern=".fcs")

path <- "PBS"
flowDataPBS <- read.flowSet(path = path, transformation = FALSE, pattern=".fcs")
fcsfiles <- list.files(path = path, recursive = TRUE, pattern = ".fcs",
                       full.names = TRUE)
flowDataPBS <- flowCore::read.flowSet(files = fcsfiles, pattern = ".fcs")

flowData_transformedPBS <- transform(flowDataPBS,`FL1-H`=asinh(`FL1-H`),
                                     `SSC-H`=asinh(`SSC-H`),
```

```
                                    `FL3-H`=asinh(`FL3-H`),
                                    `FSC-H`=asinh(`FSC-H`))
param=c("FL1-H", "FL3-H","SSC-H","FSC-H")
remove(flowDataPBS)
```

Load All FCS-files

```
path = "FCS"
#reads FCS-files
flowData <- read.flowSet(path = path, transformation = FALSE, pattern=".fcs")

### Denoise data => define gates

# Select phenotypic features of interest and transform parameters
flowData_transformed <- transform(flowData,`FL1-H`=asinh(`FL1-H`),
                                    `SSC-H`=asinh(`SSC-H`),
                                    `FL3-H`=asinh(`FL3-H`),
                                    `FSC-H`=asinh(`FSC-H`))
param=c("FL1-H", "FL3-H","SSC-H","FSC-H")
remove(flowData)
```

Gating

```
### Create a PolygonGate for denoising the dataset
### Define coordinates for gate in sqrcut1 in format: c(x,x,x,x,y,y,y,y)

# Intact Cells
#matrix(c(9.5,9,16,16,   5,8.2,15,10),ncol=2, nrow=4)  nrow=4)
    sqrcut1 <- matrix(c(9.5,9.5,16,16,   6,9,15,10),ncol=2, nrow =4)
    colnames(sqrcut1) <- c("FL1-H","FL3-H")
    polyGate1 <- polygonGate(.gate=sqrcut1, filterId = "Intact Cells")

plot1 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[1], filter=polyGate1,
       scales=list(y=list(limits=c(0,20)),
                   x=list(limits=c(6,16))),
       #axis = axis.default, nbin=125,
       par.strip.text=list(col="white", size = 8, font=2, cex=2), smooth=FALSE)
plot1
```

```
# Damaged Cells
    # matrix(c(9,8.25,16,16,   8.2,12.5,18,15),ncol=2, nrow=4)
sqrcut2 <- matrix(c(8.5,8.5,14,14,10.2,   10,13,17,15,10),ncol=2, nrow=5)
colnames(sqrcut2) <- c("FL1-H","FL3-H")
polyGate2 <- polygonGate(.gate=sqrcut2, filterId = "Damaged Cells")

plot2 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[155], filter=polyGate2,
       scales=list(y=list(limits=c(0,20)),
                   x=list(limits=c(6,16))),
       axis = axis.default, nbin=125,
```

```
       par.strip.text=list(col="white", font=2, cex=2), smooth=FALSE)
plot2
```

```
# All Cells
sqrcut3 <- matrix(c(9.5,8.25,16,16,   5,12.5,18,10),ncol=2, nrow=4)
colnames(sqrcut3) <- c("FL1-H","FL3-H")
polyGate3 <- polygonGate(.gate=sqrcut3, filterId = "Total Cells")

###  Gating quality check
plot3 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[16], filter=polyGate3,
       scales=list(y=list(limits=c(0,20)),
                   x=list(limits=c(6,16))),
       axis = axis.default, nbin=125,
       par.strip.text=list(col="white", font=2, cex=2), smooth=FALSE)
plot3
```

```
### Check gating quality using PBS files

plotPBS <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformedPBS[20],
                  filter=polyGate1,scales=list(y=list(limits=c(0,20)),
                                               x=list(limits=c(6,16))),
      # axis = axis.default, nbin=125,
       par.strip.text=list(col="white", font=2, cex=2), smooth=FALSE,
       xlab = "FITC - H",
       ylab = "PerCP-H")
plotPBS
```

```
### Isolate only the cellular information based on the polyGate3
flowData_Intact <- Subset(flowData_transformed, polyGate1)
# flowData_Damaged <- Subset(flowData_transformed, polyGate2)
flowData_Total <- Subset(flowData_transformed, polyGate3)
```

Plot Gates

```
### Make plots for article
panel.background <- trellis.par.get("panel.background")
panel.background$col <- "white"
trellis.par.set("panel.background", panel.background)
trellis.par.set(strip.background=list(col="lightgrey"))

flowViz.par.set(theme = trellis.par.get(), reset = TRUE)

plotPBSArt <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformedPBS[20],
                     filter=polyGate1,scales=list(y=list(limits=c(0,20)),
                                                  x=list(limits=c(6,16))),
      # axis = axis.default, nbin=125,
       par.strip.text=list(col="Black", font=1, cex=1), smooth=FALSE,
       xlab = "FITC - H",
       ylab = "PerCP-H",strip=strip.custom(factor.levels = c("PBS")),
      par.settings = list(strip.background=list(col="lightgrey")))
plotPBSArt
```

```
#png ("Densityplot_Batch_PBS.png", height = 2500, width = 3250, res = 300)
#plotPBSArt
#dev.off()

dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/Densityplot_SHIME_PBS.tiff",width=4*dpi,
     height=4*dpi,res=dpi,compression="lzw")
plotPBSArt
dev.off()

## Gate 1: Intact cells
plotSampleArt1 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[171],
                         filter=polyGate1,scales=list(y=list(limits=c(0,20)),
                                                      x=list(limits=c(6,16))),
      # axis = axis.default, nbin=125,
       par.strip.text=list(col="black", font=1, cex=1), smooth=FALSE,
       xlab = "FITC - H",
       ylab = "PerCP-H",strip=strip.custom(factor.levels = c("P80 - 0,5% - D2 - T12")),
      par.settings = list(strip.background=list(col="lightgrey")))
plotSampleArt1
```

```
dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/Densityplot_SHIME_Sample170.tiff",width=4*dpi,
     height=4*dpi,res=dpi,compression="lzw")
plotSampleArt1
dev.off()


plotSampleArt2 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[1],
                         filter=polyGate1,
                         scales=list(y=list(limits=c(0,20)),
                                     x=list(limits=c(6,16))),
      # axis = axis.default, nbin=125,
       par.strip.text=list(col="black", font=1, cex=1), smooth=FALSE,
       xlab = "FITC - H",
       ylab = "PerCP-H",strip=strip.custom(factor.levels = c("Control - D1 - T2")),
       par.settings = list(strip.background=list(col="lightgrey")))
plotSampleArt2
```

```
dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/Densityplot_SHIME_Sample1.tiff",width=4*dpi,
     height=4*dpi,res=dpi,compression="lzw")
plotSampleArt2
dev.off()


plotSampleArt3 <- xyplot(`FL3-H` ~ `FL1-H`, data=flowData_transformed[100],
                         filter=polyGate1,scales=list(y=list(limits=c(0,20)),
                                                      x=list(limits=c(6,16))),
      # axis = axis.default, nbin=125,
       par.strip.text=list(col="black", font=1, cex=1), smooth=FALSE,
       xlab = "FITC - H",
       ylab = "PerCP-H",strip=strip.custom(factor.levels = c("RL - 0,05 - D1 - T14")),
       par.settings = list(strip.background=list(col="lightgrey")))
```

```
plotSampleArt3
```

```
dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/Densityplot_SHIME_Sample100.tiff",width=4*dpi,
     height=4*dpi,res=dpi,compression="lzw")
plotSampleArt3
dev.off()
```

Extract Cellcounts

```
### Extract the cell counts
a <-flowCore::filter(flowData_Intact, polyGate1)
Intact <- summary(a);Intact <- toTable(Intact)
# b <- flowCore::filter(flowData_Damaged, polyGate2)
# Damaged <- summary(b);Damaged <- toTable(Damaged)
s <- flowCore::filter(flowData_Total, polyGate3)
TotalCount <- summary(s);TotalCount <- toTable(TotalCount)

### Extract the volume
vol <- as.numeric(flowCore::fsApply(flowData_transformed,
                                    FUN = function(x) x@description$`$VOL`))/1000

### Store the data
results_counts <- data.frame(Samples=flowCore::sampleNames(flowData_transformed),
                             Total.cells = TotalCount$true, Intact = Intact$true)
#absolute counts

results_counts_vol <- data.frame(Samples=flowCore::sampleNames(flowData_transformed),
                             Total.cells = TotalCount$true/vol, Intact = Intact$true/vol)
#contains conentrations per µL
colnames(results_counts_vol) <- c("Filename","Total","Intact")

Filename <- data.frame(do.call(rbind, lapply(strsplit(as.character(
  results_counts_vol$Filename),c(".f")), rbind)))
results_counts_vol$Filename <- Filename[,1]
### Exporting cell counts to .csv file to working directory
write.csv2(file="results.counts.csv", results_counts)
write.csv2(file="results.counts_vol.csv", results_counts_vol)
```

```
results_S3_T14_T16 <- readxl::read_excel("SHIME_FC-data.xlsx",sheet = "S3-T14-T16")
colnames(results_S3_T14_T16) <- c("Filename","Total","Intact")
results_counts_vol <- rbind(results_counts_vol,results_S3_T14_T16)
```

Load Metadata

```
metadataFC <- readxl::read_excel("Metadata_FC.xlsx",sheet = "Metadata")

metadataFC <- as.data.frame(metadataFC) #to avoid warnings/errors with rownames

metadataFC$SHIME <- as.factor(metadataFC$SHIME)
```

```
metadataFC$Emulsifier <- as.factor(metadataFC$Emulsifier)
metadataFC$EM_Conc <- as.factor(metadataFC$EM_Conc)
metadataFC$Timepoint <- as.numeric(metadataFC$Timepoint)
```

```
#paste metadata together with Intact and Damaged cellcounts
Cells <- merge(metadataFC, results_counts_vol, by = "Filename")
#extract only the Intact and Damaged cell columns

#calculating concentrations in original samples: 4 and 5 times dilution
#    - calculation to ml
Cells$Intact.cells.conc <- Cells$Intact*10000*1000
Cells$Intact.cells.conc.log <- log10(Cells$Intact*10000*1000)

Cells$Total.cells.conc <-  Cells$Total*10000*1000
Cells$Total.cells.conc.log <- log10(Cells$Total.cells.conc)


Cells <- unite(Cells, Treatment, c(Emulsifier, EM_Conc), remove=FALSE, sep =' - ')
Cells$SHIME <- as.factor(Cells$SHIME)
Cells$Emulsifier <- as.factor(Cells$Emulsifier)
Cells$EM_Conc <- as.factor(Cells$EM_Conc)
Cells$Timepoint <- as.numeric(Cells$Timepoint)

write.csv2(file="SHIME_ExtractedCells.csv", Cells)
```

Plotting

```
Cells_S2 <- dplyr::filter(Cells, Donor == 1)

plotTotal_S2 <- ggplot(Cells_S2, aes(x=Timepoint,y=Total.cells.conc.log,
                                     color = Treatment))+
  facet_grid(Donor~.)+
  theme_bw()+
  #geom_rect(aes(xmin=9.5,xmax=18,ymin=8.6,ymax=11),fill="grey",alpha=0.6,colour = "grey")+
   annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 11.5, xmax = 18,
        ymin = -Inf, ymax = Inf) +
  geom_line() + geom_point()+
  xlab("Timepoint")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Total cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  scale_color_manual(values = c("Black","#d28eed", "#844da8","#74dff7", "#125087",
                                "#f5e2a2","#e3d029", "#F781BF","#f2b385", "#d97914"))+
  scale_x_continuous(breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
```

```
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotTotal_S2
```

```
Cells_S3 <- dplyr::filter(Cells, Donor == 2)

plotTotal_S3 <- ggplot(Cells_S3, aes(x=Timepoint,y=Total.cells.conc.log,
                                     color = Treatment))+
  facet_grid(Donor~.)+
  theme_bw()+
  #geom_rect(aes(xmin=9.5,xmax=18,ymin=8.6,ymax=11),fill="grey",alpha=0.6,colour="grey")+
   annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 9.5, xmax = 16,
        ymin = -Inf, ymax = Inf) +
  geom_line() + geom_point()+
  xlab("Timepoint")+

  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Total cell concentration " *mL^-1)) +
  labs(fill = "EM-conc. (%)") +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  scale_color_manual(values = c("Black","#999999","#d28eed","#844da8","#74dff7","#125087",
                                "#f5e2a2", "#e3d029", "#F781BF","#f2b385", "#d97914"))+
  # scale_x_continuous(breaks=sort(unique(Cells_S2$Timepoint)))+
   scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotTotal_S3
```

```
plotTotal <- ggarrange(plotTotal_S2, plotTotal_S3,heights = c(2,2.2),
          labels = c("A", "B"),
```

```
          ncol = 1, nrow = 2, align = "v")

dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/SHIME_TotalCellcounts1.tiff",width=8*dpi,
     height=6*dpi,res=dpi,compression="lzw")
plotTotal
dev.off()



plotIntact_S2 <- ggplot(Cells_S2, aes(x=Timepoint,y=Intact.cells.conc.log,
                                      color = Treatment))+

    facet_grid(Donor~.)+
    annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 11.5, xmax = 18,
        ymin = -Inf, ymax = Inf) +
    geom_line() + geom_point()+
  xlab("Timepoint")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  theme_bw()+
  scale_color_manual(values = c("Black","#d28eed","#844da8","#74dff7","#125087","#f5e2a2",
                                "#e3d029", "#F781BF","#f2b385", "#d97914"))+
  scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotIntact_S2
```

```
plotIntact_S3 <- ggplot(Cells_S3, aes(x=Timepoint,y=Intact.cells.conc.log,
                                      color = Treatment)) +
  facet_grid(Donor~.)+
    annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 9.5, xmax = 16,
        ymin = -Inf, ymax = Inf) +
    geom_line() + geom_point()+
  xlab("Timepoint")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
```

```
  ylab(bquote("Cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  theme_bw()+
  scale_color_manual(values = c("Black","#999999","#d28eed","#844da8","#74dff7","#125087",
                                "#f5e2a2", "#e3d029", "#F781BF","#f2b385","#d97914"))+
  scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotIntact_S3
```

```
plotIntact <- ggarrange(plotIntact_S2 +rremove("x.text"), plotIntact_S3,heights = c(2,2.2),
          labels = c("A", "B"),
          ncol = 1, nrow = 2, align = "v")

dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/SHIME_IntactCellcounts1.tiff",width=8*dpi,
     height=6*dpi,res=dpi,compression="lzw")
plotIntact
dev.off()
```

Plotting Rhamnolipids P80

```
Cells_S2 <- dplyr::filter(Cells, Donor == 1)
Cells_S2_RL.P80 <- dplyr::filter(Cells_S2, Emulsifier %in% c("Control","RL","P80"))

plotTotal_S2_RL.P80 <- ggplot(Cells_S2_RL.P80, aes(x=Timepoint,y=Total.cells.conc.log,
                                                   color = Treatment)) +
  facet_grid(Donor~.)+
  theme_bw()+
  #geom_rect(aes(xmin=9.5,xmax=18,ymin=8.6,ymax=11),fill="grey",alpha=0.6,colour="grey")+
   annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 11.5, xmax = 18,
        ymin = -Inf, ymax = Inf) +
  geom_line() + geom_point()+
  xlab("Timepoint (Day)")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
```

```
  ylab(bquote("Total cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  ylim(8.8,10.2)+
  scale_color_manual(values = c("Black","#d28eed","#844da8","#74dff7","#125087","#f5e2a2",
                                "#e3d029", "#F781BF","#f2b385", "#d97914"))+
  scale_x_continuous(breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotTotal_S2_RL.P80
```

```
Cells_S3 <- dplyr::filter(Cells, Donor == 2)
Cells_S3_RL.P80 <- dplyr::filter(Cells_S3, Emulsifier %in% c("Control","RL","P80"))

plotTotal_S3_RL.P80 <- ggplot(Cells_S3_RL.P80, aes(x=Timepoint,y=Total.cells.conc.log,
                                                   color = Treatment)) +
  facet_grid(Donor~.)+
  theme_bw()+
  #geom_rect(aes(xmin=9.5,xmax=18,ymin=8.6,ymax=11),fill="grey",alpha=0.6,colour ="grey")+
   annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 9.5, xmax = 16,
        ymin = -Inf, ymax = Inf) +
  geom_line() + geom_point()+
  xlab("Timepoint (Day)")+
  ylim(8.8,10.2)+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Total cell concentration " *mL^-1)) +
  labs(fill = "EM-conc. (%)") +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  scale_color_manual(values = c("Black","#d28eed","#844da8","#74dff7","#125087","#f5e2a2",
                                "#e3d029","#F781BF","#f2b385", "#d97914"))+
  # scale_x_continuous(breaks=sort(unique(Cells_S2$Timepoint)))+
  scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
```

```
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotTotal_S3_RL.P80
```

```
plotTotal_RL.P80 <- ggarrange(plotTotal_S2_RL.P80, plotTotal_S3_RL.P80,heights = c(2,2.2),
          labels = c("A", "B"),
          ncol = 1, nrow = 2, align = "v")

dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/SHIME_TotalCellcounts_RL.P80.tiff",width=8*dpi,
     height=6*dpi,res=dpi,compression="lzw")
plotTotal_RL.P80
dev.off()



plotIntact_S2_RL.P80 <- ggplot(Cells_S2_RL.P80, aes(x=Timepoint,y=Intact.cells.conc.log,
                                                    color = Treatment)) +
  facet_grid(Donor~.)+
    annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 11.5, xmax = 18,
        ymin = -Inf, ymax = Inf) +
    geom_line() + geom_point()+
  xlab("Timepoint (Day)")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  theme_bw()+
  ylim(8.8,10.2)+
  scale_color_manual(values = c("Black","#d28eed","#844da8","#74dff7","#125087","#f5e2a2",
                                "#e3d029","#F781BF","#f2b385", "#d97914"))+
  scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_blank(),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
```

```
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotIntact_S2_RL.P80
```

```
plotIntact_S3_RL.P80 <- ggplot(Cells_S3_RL.P80, aes(x=Timepoint,y=Intact.cells.conc.log,
                                                    color = Treatment)) +
  facet_grid(Donor~.)+
    annotate("rect", fill = "grey", alpha = 0.5,
        xmin = 9.5, xmax = 16,
        ymin = -Inf, ymax = Inf) +
    geom_line() + geom_point()+
  xlab("Timepoint (Day)")+
  # xlab(bquote('Assimilation ('*mu~ 'mol' ~CO[2]~ m^-2~s^-1*')'))+
  ylab(bquote("Cell concentration " *mL^-1)) +
  #ggtitle("Total cell concentrations (/ml)") +
  #scale_fill_manual(values =c("#F4685CFF", "#AB337CFF","#FDE4A6FF","#681D81FF"))+
  # scale_fill_brewer(palette="Set1")+
  labs(fill = "EM-conc. (%)") +
  theme_bw()+
  ylim(8.8,10.2)+
  scale_color_manual(values = c("Black", "#d28eed","#844da8","#74dff7","#125087","#f5e2a2",
                                "#e3d029","#F781BF","#f2b385","#d97914"))+
  scale_x_continuous(limits=c(NA,18),breaks=c(0,2,4,6,8,10,12,14,16,18))+
  theme(axis.text.x=element_text(angle=90, hjust=1, vjust = 0,size = 8),
        axis.text.y = element_text(size = 10,family='sans',angle=0),
        axis.title.x = element_text(size =12,family='sans',angle=0),
        axis.title.y = element_text(size=12,family='sans',angle=90),
        legend.text = element_text(size=10,family='sans',angle=0),
        legend.title = element_text(size=10,family='sans',angle=0),
        strip.text = element_text(size=10,angle=0,family='sans'),
        legend.position = "right",
        legend.key.size = unit(0.3,"cm"),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  theme(axis.line.y=element_line(colour="black"),axis.line=element_line(colour="black"))
  # coord_cartesian(ylim=c(7.5,10))
# scale_fill_brewer(palette="spectral")
plotIntact_S3_RL.P80
```

```
plotIntact_RL.P80 <- ggarrange(plotIntact_S2_RL.P80 +rremove("x.text"), plotIntact_S3_RL.P80,
                               heights = c(2,2.2),
          labels = c("A", "B"),
          ncol = 1, nrow = 2, align = "v")

dpi=300
tiff("/media/projects2/LisaM/SHIME_EM/FlowCyto/SHIME_IntactCellcounts_RL.P80.tiff",width=8*dpi,
     height=6*dpi,res=dpi,compression="lzw")
plotIntact_RL.P80
dev.off()
```