```
################################################
#
#RESEARCH ARTICLE:
#
#A  high-throughput microbial growth data assembly and analysis protocol
#to elucidate the strain variability of Listeria monocytogenes in NaCl stress
#
#Mariella Aalto-Araneda, Anna Pöntinen, Maiju Pesonen, Jukka Corander,
#Annukka Markkula, Taurai Tasara, Roger Stephan, Hannu Korkeala
#
#Supplement S10. THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
#
################################################
#
#VOLUME 1: INSPECTING CONTROL STRAINS
#
################################################

###SETTING WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola/controls")

###UPLOADING FILE
salt_cont1 = read.csv2("controls_all_orig.csv",strip.white =
TRUE,stringsAsFactors=F)

# inspecting file
str(salt_cont1)
summary(salt_cont1)
names(salt_cont1)

###REMOVING EXTRA-ODs TO AVOID ARTIFACT FROM ABSORBANCE MEASUREMENT OSCILLATION

drop <- c("OD0.1","OD0.6"  ,          "OD0.8"       ,    "OD1.1",
"OD1.6"   ,         "OD1.8"          , "OD2.1",
 "OD2.6"      ,      "OD2.8"        ,    "OD3.1",
 "OD3.6"      ,      "OD3.8"       ,     "OD4.1",
"OD4.6"        ,    "OD4.8"         ,    "OD5.1",
"OD5.6"         ,   "OD5.8"          ,   "OD6.1" ,
"OD6.6"          ,  "OD6.8"          ,  "OD7.1",
 "OD7.6"          ,  "OD7.8"          ,  "OD8.1",
"OD8.6"     ,       "OD8.8"   ,          "OD9.1" ,
 "OD9.6"     ,       "OD9.8"  ,          "OD10.1",
"OD10.6"      ,     "OD10.8"   ,        "OD11.1",
"OD11.6"       ,    "OD11.8"    ,       "OD12.1" ,
 "OD12.6"       ,    "OD12.8"    ,       "OD13.1",
"OD13.6"         ,  "OD13.8"      ,     "OD14.1",
"OD14.6"    ,       "OD14.8"       ,    "OD15.1",
"OD15.6"     ,      "OD15.8"        ,   "OD16.1",
 "OD16.6"     ,      "OD16.8"        ,  "OD17.1",  "OD17.6"      ,    "OD17.8" )

salt_cont2 = salt_cont1[,!(names(salt_cont1) %in% drop)]
names(salt_cont2)

# renaming OD-variables
colnames(salt_cont2)[colnames(salt_cont2)=="OD0.3"] <- "OD0"
colnames(salt_cont2)[colnames(salt_cont2)=="OD1.3"] <- "OD1"
colnames(salt_cont2)[colnames(salt_cont2)=="OD2.3"] <- "OD2"
colnames(salt_cont2)[colnames(salt_cont2)=="OD3.3"] <- "OD3"
colnames(salt_cont2)[colnames(salt_cont2)=="OD4.3"] <- "OD4"
colnames(salt_cont2)[colnames(salt_cont2)=="OD5.3"] <- "OD5"
colnames(salt_cont2)[colnames(salt_cont2)=="OD6.3"] <- "OD6"
colnames(salt_cont2)[colnames(salt_cont2)=="OD7.3"] <- "OD7"
colnames(salt_cont2)[colnames(salt_cont2)=="OD8.3"] <- "OD8"
colnames(salt_cont2)[colnames(salt_cont2)=="OD9.3"] <- "OD9"
colnames(salt_cont2)[colnames(salt_cont2)=="OD10.3"] <- "OD10"
colnames(salt_cont2)[colnames(salt_cont2)=="OD11.3"] <- "OD11"
colnames(salt_cont2)[colnames(salt_cont2)=="OD12.3"] <- "OD12"
colnames(salt_cont2)[colnames(salt_cont2)=="OD13.3"] <- "OD13"
colnames(salt_cont2)[colnames(salt_cont2)=="OD14.3"] <- "OD14"
```

```
colnames(salt_cont2)[colnames(salt_cont2)=="OD15.3"] <- "OD15"
colnames(salt_cont2)[colnames(salt_cont2)=="OD16.3"] <- "OD16"
colnames(salt_cont2)[colnames(salt_cont2)=="OD17.3"] <- "OD17"

#Saving new dataset
write.csv2(salt_cont2, "controls_all_OD017.csv", row.names = F)

###PLOT ALL CONTROLS BY LABORATORY TECHNICIAN

#Labtech A or B (all repls)
#A: run 1 - 24 (ctrl 1) & 1 - 24 (ctrl 2)
#B: run 25 - 150 (ctrl 1) & 25 - 150 (ctrl 2)

setwd("Z:/Desktop/R_juttuja/suola/controls")

# Upload csv file where time as rows and replicates as columns. Replicates named
by run number (above).
salt_cont = read.csv2("salt_cont_noLabTech_allrepls.csv",strip.white =
TRUE,stringsAsFactors=F)

# data from wide to long format
library(tidyr)
salt_long <- gather(salt_cont, ctrl_strain, growth, ctrl1_1:ctrl2_150,
factor_key = T)
salt_long$time_num <- rep(0:17, 300)
salt_long$ctrl <- gsub( "_.*$", "", salt_long$ctrl_strain)
salt_long$run <- as.factor(sub('.*_', '', salt_long$ctrl_strain))

# add well info
salt_long$lab_tech <- NA
salt_long$run <- as.numeric((as.character(salt_long$run)))

for(i in 1:nrow(salt_long)){
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] < 25){
    salt_long$lab_tech[i] <- "A"
  }
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] > 24){
    salt_long$lab_tech[i] <- "B"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] < 25){
    salt_long$lab_tech[i] <- "A"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] > 24){
    salt_long$lab_tech[i] <- "B"
  }
}

salt_long$run <- as.factor(salt_long$run)
salt_long$lab_tech <- as.factor(salt_long$lab_tech)

library(ggplot2)
library(directlabels)
library(ggrepel)
p <- ggplot(salt_long, aes(time_num, growth, group = ctrl_strain, colour =
lab_tech), legend = FALSE) + geom_line(size = 1, show.legend=F)
p <- p + facet_grid(. ~ ctrl)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, replicates of
control strains from each run, color-coded by laboratory technician: A (red), B
(blue)",x="Time (h)", y = "OD")
p <- p + geom_text(data = salt_long[salt_long$time_num == 17,], aes(label =
lab_tech), hjust = -0.3, size = 2)

p <- p + theme(legend.position="none",
               axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))
```

```
#pdf("salt_control_curves_inout.pdf", width = 10 , height = 5)
p
dev.off()

###PLOT ALL CONTROLS BY WELLS

#Inner or outer well
#Inner well: run 1 - 100 (ctrl 1) & 1 - 76 (ctrl 2)
#Outer well: run 101 - 150 (ctrl 1) & 77 - 150 (ctrl 2)

setwd("Z:/Desktop/R_juttuja/suola/controls")

# Upload csv file where time as rows and replicates as columns. Replicates named
by run number (above).
salt_cont = read.csv2("salt_cont_noInOut.csv",strip.white =
TRUE,stringsAsFactors=F)

# data from wide to long format
library(tidyr)
salt_long <- gather(salt_cont, ctrl_strain, growth, ctrl1_1:ctrl2_150,
factor_key = T)
salt_long$time_num <- rep(0:17, 300)
salt_long$ctrl <- gsub( "_.*$", "", salt_long$ctrl_strain)
salt_long$run <- as.factor(sub('.*_', '', salt_long$ctrl_strain))

# add well info
salt_long$well <- NA
salt_long$run <- as.numeric((as.character(salt_long$run)))

for(i in 1:nrow(salt_long)){
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] < 101){
    salt_long$well[i] <- "inner"
  }
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] > 100){
    salt_long$well[i] <- "outer"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] < 77){
    salt_long$well[i] <- "inner"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] > 76){
    salt_long$well[i] <- "outer"
  }
}

salt_long$run <- as.factor(salt_long$run)
salt_long$well <- as.factor(salt_long$well)

library(ggplot2)
library(directlabels)
library(ggrepel)
p <- ggplot(salt_long, aes(time_num, growth, group = ctrl_strain, colour =
well), legend = FALSE) + geom_line(size = 1, show.legend=F)
p <- p + facet_grid(. ~ ctrl)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, replicates of
control strains from each run, color-coded by inner (red) and outer (blue)
honeycomb plate wells",x="Time (h)", y = "OD")
p <- p + geom_text(data = salt_long[salt_long$time_num == 17,], aes(label =
well), hjust = -0.3, size = 2)

p <- p + theme(legend.position="none",
               axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))

#pdf("salt_control_curves_inout.pdf", width = 10 , height = 5)
```

```
p
dev.off()

###REMOVE OUTER WELLS

salt_cont1 = read.csv2("controls_all_od017.csv",strip.white =
TRUE,stringsAsFactors=F)
names(salt_cont1)

salt_cont1 <- salt_cont1[!(salt_cont1$outer_no0_yes1==1),]

write.csv2(salt_cont1, "controls_inner_OD017.csv", row.names = F)

################################################
#THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
#AALTO-ARANEDA ET AL.
#
################################################
#
#VOLUME 2: BIOSCREEN RAWDATA HANDLING AND EXPLORATION
#MODIFIED FROM CONNELLY
http://bconnelly.net/2014/04/analyzing-microbial-growth-with-r/
#
################################################

###INSTALLING PACKAGES
install.packages(c('reshape2', 'dplyr', 'ggplot2', 'grofit'))

###SETTING WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola")

###UPLOADING FILE
#csv where strains in rows and variables in columns (serotype, ODs, replicate
numbers)
salt_repls = read.csv2("all_repls.csv",strip.white = TRUE,stringsAsFactors=F)

str(salt_repls)
summary(salt_repls)
names(salt_repls)

###REMOVE EXTRA-ODs TO AVOID ARTIFACT FROM ABSORBANCE MEASUREMENT OSCILLATION

drop <- c("OD0.1","OD0.6"  ,          "OD0.8"       ,    "OD1.1",
"OD1.6"   ,         "OD1.8"          , "OD2.1",
 "OD2.6"      ,      "OD2.8"        ,     "OD3.1",
 "OD3.6"      ,      "OD3.8"       ,      "OD4.1",
"OD4.6"        ,    "OD4.8"         ,    "OD5.1",
"OD5.6"          ,  "OD5.8"          ,   "OD6.1" ,
"OD6.6"           , "OD6.8"          ,  "OD7.1",
 "OD7.6"           , "OD7.8"          ,  "OD8.1",
"OD8.6"     ,       "OD8.8"  ,          "OD9.1" ,
 "OD9.6"     ,       "OD9.8"  ,          "OD10.1",
"OD10.6"      ,     "OD10.8"   ,        "OD11.1",
"OD11.6"       ,    "OD11.8"    ,       "OD12.1" ,
 "OD12.6"       ,    "OD12.8"    ,       "OD13.1",
"OD13.6"          , "OD13.8"       ,    "OD14.1",
"OD14.6"    ,       "OD14.8"        ,   "OD15.1",
"OD15.6"     ,      "OD15.8"         ,  "OD16.1",
 "OD16.6"     ,     "OD16.8"         ,  "OD17.1",  "OD17.6"      ,    "OD17.8" )

salt_cont2 = salt_repls[,!(names(salt_repls) %in% drop)]
names(salt_cont2)

# rename OD variables
colnames(salt_cont2)[colnames(salt_cont2)=="OD0.3"] <- "OD0"
colnames(salt_cont2)[colnames(salt_cont2)=="OD1.3"] <- "OD1"
colnames(salt_cont2)[colnames(salt_cont2)=="OD2.3"] <- "OD2"
colnames(salt_cont2)[colnames(salt_cont2)=="OD3.3"] <- "OD3"
colnames(salt_cont2)[colnames(salt_cont2)=="OD4.3"] <- "OD4"
```

```
colnames(salt_cont2)[colnames(salt_cont2)=="OD5.3"] <- "OD5"
colnames(salt_cont2)[colnames(salt_cont2)=="OD6.3"] <- "OD6"
colnames(salt_cont2)[colnames(salt_cont2)=="OD7.3"] <- "OD7"
colnames(salt_cont2)[colnames(salt_cont2)=="OD8.3"] <- "OD8"
colnames(salt_cont2)[colnames(salt_cont2)=="OD9.3"] <- "OD9"
colnames(salt_cont2)[colnames(salt_cont2)=="OD10.3"] <- "OD10"
colnames(salt_cont2)[colnames(salt_cont2)=="OD11.3"] <- "OD11"
colnames(salt_cont2)[colnames(salt_cont2)=="OD12.3"] <- "OD12"
colnames(salt_cont2)[colnames(salt_cont2)=="OD13.3"] <- "OD13"
colnames(salt_cont2)[colnames(salt_cont2)=="OD14.3"] <- "OD14"
colnames(salt_cont2)[colnames(salt_cont2)=="OD15.3"] <- "OD15"
colnames(salt_cont2)[colnames(salt_cont2)=="OD16.3"] <- "OD16"
colnames(salt_cont2)[colnames(salt_cont2)=="OD17.3"] <- "OD17"

# save new file
#write.csv2(salt_cont2, "repls_all_OD017.csv", row.names = F)

###REMOVE OUTER WELLS
salt_repls_inner <- salt_cont2[!(salt_cont2$outer1_inner0==1),]

str(salt_repls_inner)
summary(salt_repls_inner)
names(salt_repls_inner)

write.csv2(salt_repls_inner, "repls_inner_OD017.csv", row.names = F)
#from this file, create saltrepls_final.csv by transposing replicates as columns

# importing data in csv format: save your raw data in excel into csv
# dataset: column 1: time (h), other columns: each replicate (absorbance value,
OD600)
#i.e. file where replicates in columns and time in rows
rawdata = read.csv2("saltrepls_final.csv",strip.white = TRUE,stringsAsFactors=F)

# inspecting the file
names(rawdata) #column headings
str(rawdata) #variables
summary(rawdata) #structure

# create new file in order to get the replicate names for saltreplcolumns-file
#write.csv2(rawdata, "saltrepls_final_v2.csv", row.names = F)

# tidying data: organizes each OD measurement to a single row and names the
variables
library(reshape2)
reshaped <- melt(rawdata, id=c("Time"), variable.name="replicate",
                 value.name="OD600")

# inspect tidied data
summary(reshaped) #structure
head(reshaped, n=200) #first n rows of reshaped file

# creating the file that describes how to annotate
# NB! have a row corresponding to each pairing variable in reshaped!
newcolumns <- read.csv2("saltreplcolumns.csv",strip.white =
TRUE,stringsAsFactors=T)
head(newcolumns, n=10)

class(newcolumns$replicate)
class(reshaped$replicate)
levels(newcolumns$replicate)
levels(reshaped$replicate)

# annotating the data
library(dplyr)

# combine the reshaped data with the new additional columns/variables, pairing
them by replicate value
annotated <- inner_join(reshaped, newcolumns, by="replicate")
```

```
# take a peek at the records in annotated
head(annotated, n=200)
tail(annotated, n=200)
summary(annotated)

# save new annotated data set
#write.csv2(annotated, "saltrepl-annotated.csv", row.names = F)

# OR start from the annotated file
#annotated = read.csv2("saltrepl-annotated.csv")

###GROUPING THE DATA
grouped <- group_by(annotated, strain, Time)

# calculating averages for the grouped data
stats <- summarise(grouped, N=length(OD600), Average=mean(OD600),
StDev=sd(OD600), CV=sd(OD600)/mean(OD600))

str(stats)
summary(stats)
a = sum(stats$N == 1)/18
b = sum(stats$N == 2)/18
c = sum(stats$N == 3)/18
d = sum(stats$N == 4)/18
e = sum(stats$N == 5)/18
f = sum(stats$N == 6)/18

g = a+b+c+d+e+f

a/g*100 #%of strains with 1 replicate
b/g*100 #%of strains with 2 replicates
c/g*100 #%of strains with 3 replicates
d/g*100 #%of strains with 4 replicates
e/g*100 #%of strains with 5 replicates
f/g*100 #%of strains with 6 replicates

median(stats$N)
mean(stats$N)
median(stats$StDev, na.rm=T)
mean(stats$StDev, na.rm=T)
median(stats$CV, na.rm=T)
mean(stats$CV, na.rm=T)

# transforming data back to "wide" format, time as rows and strains as columns
library(reshape2)
stats_wide <- dcast(stats, Time ~ strain, value.var="Average")
write.csv2(stats_wide, "saltstrains_avgODs.csv", row.names = F)

# transforming data back to "wide" format, time as colums and strains as rows
library(reshape2)
stats_wide <- dcast(stats, strain ~ Time, value.var="Average")
#stats_wide <- dcast(stats, strain ~ Time, value.var="StDev")
#stats_wide <- dcast(stats, strain ~ Time, value.var="CV")
write.csv2(stats_wide, "saltstrains_avgODs_v2.csv", row.names = F) #from this
file, create salt_data.csv where strains in rows and labtech + ODs in columns
#write.csv2(stats_wide, "saltstrains_stdevODs.csv", row.names = F)
#write.csv2(stats_wide, "saltstrains_cvODs.csv", row.names = F)

################################################
# THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
# AALTO-ARANEDA ET AL.
#
################################################
#
# VOLUME 3: NORMALISE BATCH-EFFECTS FROM THE RESPONSE VARIABLES
#
################################################

# lab technician changed
```

```
# lab_tech a (A): run 1-16 (ctrl 1) & 1-13 (ctrl 2)
# lab_tech e (B): run 17-100 (ctrl 1) & 14-76 (ctrl 2)

###WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola/controls")

###UPLOADING FILE
# Upload csv file where time as rows and replicates as columns. Replicates named
by run number (above).
salt_cont <- read.csv2("salt_cont_noLabTech.csv",strip.white =
TRUE,stringsAsFactors=F)

###PLOT CONTROL CURVES

# data from wide to long format
library(tidyr)
salt_long <- gather(salt_cont, ctrl_strain, growth, ctrl1_1:ctrl2_76, factor_key
= T)
salt_long$time_num <- rep(0:17, 176)
salt_long$ctrl <- gsub( "_.*$", "", salt_long$ctrl_strain)
salt_long$run <- as.factor(sub('.*_', '', salt_long$ctrl_strain))

# add lab tech info
salt_long$lab_tech <- NA
#salt_long$run <- as.numeric(salt_long$run) #This original code is flawed, mixes
up the run numbers
salt_long$run <- as.numeric((as.character(salt_long$run)))

for(i in 1:nrow(salt_long)){
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] < 17){
    salt_long$lab_tech[i] <- "A"
  }
  if(salt_long$ctrl[i] == "ctrl1" & salt_long$run[i] > 16){
    salt_long$lab_tech[i] <- "B"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] < 14){
    salt_long$lab_tech[i] <- "A"
  }
  if(salt_long$ctrl[i] == "ctrl2" & salt_long$run[i] > 13){
    salt_long$lab_tech[i] <- "B"
  }
}

salt_long$run <- as.factor(salt_long$run)
salt_long$lab_tech <- as.factor(salt_long$lab_tech)

library(ggplot2)
library(directlabels)
library(ggrepel)
p <- ggplot(salt_long, aes(time_num, growth, group = ctrl_strain, colour =
lab_tech), legend = FALSE) + geom_line(size = 1, show.legend=F)
p <- p + facet_grid(. ~ ctrl)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, replicates of
control strains from inner wells only, color-coded by laboratory technician: A
(red), B (blue)",x="Time (h)", y = "OD")
p <- p + geom_text(data = salt_long[salt_long$time_num == 17,], aes(label =
lab_tech), hjust = -0.3, size = 2)

p <- p + theme(legend.position="none",
               axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))

#pdf("salt_control_curves.pdf", width = 10 , height = 5)
p
dev.off()
```

```
###GROUPING DATA BY LABTECH, CALCULATING MEAN CONTROL CURVES

# salt data
salt_data <- read.csv2("salt_data_notnorm.csv")

###MEAN OF OD OF EACH LABTECH GROUP IN EACH TIME POINT FOR ALL STRAINS
salt_lab1_strains <- salt_data[which(salt_data$laborant == "A"),c(1,3,4:21)]
salt_lab2_strains <- salt_data[which(salt_data$laborant == "B"),c(1,3,4:21)]

lab1_means <- apply(salt_lab1_strains[,3:20], 2, mean)
lab2_means <- apply(salt_lab2_strains[,3:20], 2, mean)

###MEAN OF OD OF EACH LABTECH GROUP IN EACH TIME POINT FOR CONTROL 1
controls_names <- colnames(salt_cont[,-1])
t_controls <- t(salt_cont[,-1])
transpose_controls <- as.data.frame(cbind(controls_names, t_controls))
colnames(transpose_controls) <- c("ctrl_strain", as.character(salt_cont[,1]))
transpose_controls$run <- as.numeric(sub('.*_', '',
transpose_controls$ctrl_strain))

salt.ctrl.lab <- unique(salt_long[,c(2,7)])
transpose_controls_lab <- merge(salt.ctrl.lab, transpose_controls, by =
"ctrl_strain")

salt_lab1_ctrl_strains <-
transpose_controls[which(transpose_controls_lab$lab_tech == "A"),]
salt_lab2_ctrl_strains <-
transpose_controls[which(transpose_controls_lab$lab_tech == "B"),]

salt_lab1_ctrl_strains$ctrl <- gsub( "_.*$", "",
salt_lab1_ctrl_strains$ctrl_strain)
salt_lab2_ctrl_strains$ctrl <- gsub( "_.*$", "",
salt_lab2_ctrl_strains$ctrl_strain)

salt_lab1_ctrl1_strains <-
salt_lab1_ctrl_strains[which(salt_lab1_ctrl_strains$ctrl == "ctrl1"),]
salt_lab2_ctrl1_strains <-
salt_lab2_ctrl_strains[which(salt_lab2_ctrl_strains$ctrl == "ctrl1"),]
salt_lab1_ctrl2_strains <-
salt_lab1_ctrl_strains[which(salt_lab1_ctrl_strains$ctrl == "ctrl2"),]
salt_lab2_ctrl2_strains <-
salt_lab2_ctrl_strains[which(salt_lab2_ctrl_strains$ctrl == "ctrl2"),]

salt_lab1_ctrl1_strains.num <- apply(salt_lab1_ctrl1_strains[,2:19], 2,
function(x) as.numeric(as.character(x)))
salt_lab2_ctrl1_strains.num <- apply(salt_lab2_ctrl1_strains[,2:19], 2,
function(x) as.numeric(as.character(x)))
salt_lab1_ctrl2_strains.num <- apply(salt_lab1_ctrl2_strains[,2:19], 2,
function(x) as.numeric(as.character(x)))
salt_lab2_ctrl2_strains.num <- apply(salt_lab2_ctrl2_strains[,2:19], 2,
function(x) as.numeric(as.character(x)))

lab1_ctrl1_means <- apply(salt_lab1_ctrl1_strains.num, 2, mean)
lab2_ctrl1_means <- apply(salt_lab2_ctrl1_strains.num, 2, mean)
lab1_ctrl2_means <- apply(salt_lab1_ctrl2_strains.num, 2, mean)
lab2_ctrl2_means <- apply(salt_lab2_ctrl2_strains.num, 2, mean)

ctrl_means <- as.data.frame(cbind(lab1_ctrl1_means, lab2_ctrl1_means,
lab1_ctrl2_means, lab2_ctrl2_means))
ctrl_means$Time <- rownames(ctrl_means)

library(tidyr)
ctrl_means_long <- gather(ctrl_means, Time, mean_growth,
lab1_ctrl1_means:lab2_ctrl2_means, factor_key = T)
ctrl_means_long$time_num <- rep(0:17, 4)
ctrl_means_long$ctrl <- as.factor(rep(c("ctrl1", "ctrl2"), each = 36))
ctrl_means_long$lab_tech <- as.factor(rep(c("A", "B"), each = 18, times = 2))
```

```
### PLOT
salt_long$run <- as.numeric(as.character(salt_long$run))
salt_long$lab_tech <- as.factor(salt_long$lab_tech)
salt_long$run <- as.factor(salt_long$run)

library(ggplot2)
library(directlabels)

p <- ggplot(salt_long, aes(time_num, growth, group = ctrl_strain, colour =
lab_tech), legend = FALSE) + geom_line(size = 1, show.legend=F)
p <- p + facet_grid(. ~ ctrl)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, replicates of
control strains from inner wells only, color-coded by laboratory technician A
(red) & B (blue), dashed line represents mean",x="Time (h)", y = "OD")
p <- p + geom_text(data = salt_long[salt_long$time_num == 17,], aes(label =
lab_tech), hjust = -0.3)
p <- p + theme(legend.position="none",
               axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))
p <- p + stat_summary(aes(group = lab_tech), geom = "line", linetype = 'dashed',
fun.y = mean, size = 1, colour = "black")

#pdf("salt_control_curves_with_means.pdf", width = 10 , height = 5)
p
dev.off()

###DIFFERENCES BETWEEN LABORANTS IN CTRL 1 & CTRL 2 IN EACH TIMEPOINT

ctrl1_lab_diff <- lab2_ctrl1_means - lab1_ctrl1_means
ctrl2_lab_diff <- lab2_ctrl2_means - lab1_ctrl2_means

ctrl_diff <- as.data.frame(cbind(ctrl1_lab_diff, ctrl2_lab_diff))
ctrl_diff$Time <- rownames(ctrl_diff)

mean_ctrl_diff <- apply(ctrl_diff[,1:2], 1, mean)

library(tidyr)
ctrl_diff_long <- gather(ctrl_diff, Time, lab_diff,
ctrl1_lab_diff:ctrl2_lab_diff, factor_key = T)
ctrl_diff_long$time_num <- rep(0:17, 2)
ctrl_diff_long$ctrl <- as.factor(rep(c("ctrl1", "ctrl2"), each = 18))

p <- ggplot(ctrl_diff_long, aes(time_num, lab_diff, group = ctrl, colour =
ctrl)) + geom_line(size = 1)
p <- p + labs(title="Difference between lab technicians (A-B) by hour, for both
salt control curves and their mean (dashed line)",x="Time (h)", y = "OD")
p <- p + theme(axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(-0.3, -0.15, 0, 0.15, 0.3))
p <- p + stat_summary(aes(group = 1), geom = "line", linetype = 'dashed', fun.y
= mean, size = 1, colour = "black")

#pdf("/Users/maijukujala/Documents/POSTDOC/Listeria/plots/salt_control_lab_diff.
pdf", width = 10 , height = 5)
p
dev.off()

###PLOT CURVES FOR REAL STRAINS, IS BATCH EFFECT VISIBLE THERE AS WELL?
salt_data <- read.csv2("salt_data.csv")
salt_data_ods <- salt_data[,c(1,3,4:21)]

library(tidyr)
```

```
salt_data_long <- gather(salt_data_ods, Time, growth, OD0:OD17, factor_key = T)
salt_data_long$time_num <- as.numeric(as.character(sub('.*D', '',
salt_data_long$Time)))

salt_data_long$laborant <- as.factor(salt_data_long$laborant)

library(ggplot2)
library(directlabels)
p <- ggplot(salt_data_long, aes(time_num, growth, group = strain, colour =
laborant)) + geom_line(size = 1, show.legend=T)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, not corrected
for bacth effect, 388 strains color-coded by lab technician (A-B)",x="Time (h)",
y = "OD")
p <- p + theme(axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))

#pdf("/Users/maijukujala/Documents/POSTDOC/Listeria/plots/salt_data_curves_by_br
oth.pdf", width = 7 , height = 5)
p
dev.off()

###ADD CALCULATED DIFFERENCE TO ALL lab_tech A STRAINS IN THE DATA
corr_lab1_strains <- cbind(salt_lab1_strains[,1:2], salt_lab1_strains[,-(1:2)] +
t(replicate(nrow(salt_lab1_strains), mean_ctrl_diff)))
corr_strains <- rbind(corr_lab1_strains, salt_lab2_strains)

library(tidyr)
corr_strains_long <- gather(corr_strains, Time, growth, OD0:OD17, factor_key =
T)
corr_strains_long$time_num <- as.numeric(as.character(sub('.*D', '',
corr_strains_long$Time)))
corr_strains_long$laborant <- as.factor(corr_strains_long$laborant)

library(ggplot2)
library(directlabels)
p <- ggplot(corr_strains_long, aes(time_num, growth, group = strain, colour =
laborant)) + geom_line(size = 1, show.legend=T)
p <- p + labs(title="NaCl 9.0%, L. monocytogenes growth in hours, corrected for
batch effect, 388 strains color-coded by lab technician (A-B)",x="Time (h)", y =
"OD")
p <- p + theme(axis.line = element_line(colour = "grey"),
               panel.grid.major = element_line(size = 0.1, linetype = 'dotted',
colour = "black"),
               panel.background = element_blank())
p <- p + scale_x_continuous(breaks = c(0:17))
p <- p + scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1))

#pdf("salt_data_curves_by_broth_corrected_for_batch.pdf", width = 7 , height =
5)
p
dev.off()

###SAVE DATASET WITH CORRECTED OD VALUES
write.csv2(corr_strains, "saltstrains_norm.csv", row.names = F) #add serotypes
here and use this to create input files for grofit

################################################
# THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
# AALTO-ARANEDA ET AL.
#
################################################
#
# VOLUME 4: CALCULATING GROWTH PARAMETERS
# grofit: Fitting Biological Growth Curves with R
# Matthias Kahm, Guido HasenbrinK, Hella Lichtenberg-Frat´e, Jost Ludwig, Maik
```

```
Kschischo
#
################################################

###SETTING WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola/grofit")

###INSTALING GROFIT FROM ARCHIVE
url <-
"http://cran.r-project.org/src/contrib/Archive/grofit/grofit_1.1.1-1.tar.gz"
pkgFile <- "grofit_1.1.1-1.tar.gz"
download.file(url = url, destfile = pkgFile)
install.packages(pkgs=pkgFile, type="source", repos=NULL)

# creating grofit input from an organized csv file (see format in article)
library(grofit)

# time table
time <- read.csv2("salttimegrofit.csv", header = FALSE, dec = ",")
head(time)

#  growth table
data <- read.csv2("saltdatagrofit.csv", header = FALSE, dec = ",")
head(data)

# setting options
# here you can also set smoothness on spline with smooth.gc
MyOpt1 <- grofit.control(suppress.messages = TRUE, interactive = FALSE)

# perform grofit, only gcFit

Fitall <- gcFit(time, data, MyOpt1)

# PLOTTING CURVES INDIVIDUALLY
#plot(Fitall)

# save a text file with the best fitting model & spline growth parameters

summary.gcFit(Fitall$gcTable)
sink(file = "Fitall_strains.txt")
Fitall$gcTable
sink()

###FITTING EACH MODEL AT A TIME

## gompertz
MyOpt2 <- grofit.control(suppress.messages = TRUE, interactive = FALSE, fit.opt
= "m", model.type = c("gompertz"))
Fitall2 <- gcFit(time, data, MyOpt2)

#plot(Fitall2)

# save file with parameters
summary.gcFit(Fitall2)
sink(file = "Fitall_strains_gompertz.txt")
Fitall2
sink()

## logistic
MyOpt3 <- grofit.control(suppress.messages = TRUE, interactive = FALSE, fit.opt
= "m", model.type = c("logistic"))
Fitall3 <- gcFit(time, data, MyOpt3)

#plot(Fitall3)

# save file with parameters
summary.gcFit(Fitall3)
sink(file = "Fitall_strains_logistic.txt")
Fitall3
```

```
sink()

## richards
MyOpt4 <- grofit.control(suppress.messages = TRUE, interactive = FALSE, fit.opt
= "m", model.type = c("richards"))
Fitall4 <- gcFit(time, data, MyOpt4)

#plot(Fitall4)

# save file with parameters
summary.gcFit(Fitall4)
sink(file = "Fitall_strains_richards.txt")
Fitall4
sink()

## modified gompertz
MyOpt5 <- grofit.control(suppress.messages = TRUE, interactive = FALSE, fit.opt
= "m", model.type = c("gompertz.exp"))
Fitall5 <- gcFit(time, data, MyOpt5)

#plot(Fitall5)

# save file with parameters
summary.gcFit(Fitall5)
sink(file = "Fitall_strains_gompertz_exp.txt")
Fitall5
sink()

## spline
MyOpt6 <- grofit.control(suppress.messages = TRUE, interactive = FALSE, fit.opt
= "s")
Fitall6 <- gcFit(time, data, MyOpt6)

#plot(Fitall6)

# save file with parameters
summary.gcFit(Fitall6)
sink(file = "Fitall_strains_spline.txt")
Fitall6
sink()

################################################
# THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
# AALTO-ARANEDA ET AL.
#
################################################
#
# VOLUME 5: SUMMARIZING GROWTH PARAMETERS
#
################################################

### SETTING WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola/grofit")

### UPLOADING FILE
#csv file with strains as rows and all different growth parameters as columns
sada = read.csv2("saltdata_norm_ALLparam.csv",strip.white =
TRUE,stringsAsFactors=T)

# inspecting variables
str(sada)

###CALCULATING PERCENTAGES OF FITTED CURVES AND BEST MODEL
summary(sada) ##gives the number of NA's, the percentage of which is calculated
and subtracted below
#Gompertz, curves fitted
100-48/388*100
#Mod-Gompertz, curves fitted
100-230/388*100
```

```
#Richards, curves fitted
100-88/388*100
#Logistic, curves fitted
100-13/388*100
#Spline, curves fitted
100-0/388*100

##numbers of variable categories obtained by summary() command
#Gompertz, best model
2/388*100
#Mod-Gompertz, best model
150/388*100
#Richards, best model
163/388*100
#Logistic, best model
73/388*100

###CREATING NEW VARIABLES TO DETERMINE THE PERCENTAGE OF NEGATIVE LAGS
sada$lambda_spline_zero0_1 = NA
for(i in 1:nrow(sada)){
   if(sada$lambda_spline[i] >= 0) {
      sada$lambda_spline_zero0_1[i] = 1
    } else { sada$lambda_spline_zero0_1[i] = 0}
}
sada$lambda_gomp_zero0_1 = ifelse(sada$lambda_gomp >= 0, c(1), c(0))
sada$lambda_modgomp_zero0_1 = ifelse(sada$lambda_modgomp >= 0, c(1), c(0))
sada$lambda_log_zero0_1 = ifelse(sada$lambda_log >= 0, c(1), c(0))
sada$lambda_rich_zero0_1 = ifelse(sada$lambda_rich >= 0, c(1), c(0))

sada$lambda_spline_zero0_1 = as.factor(sada$lambda_spline_zero0_1)
sada$lambda_gomp_zero0_1 = as.factor(sada$lambda_gomp_zero0_1)
sada$lambda_modgomp_zero0_1 = as.factor(sada$lambda_modgomp_zero0_1)
sada$lambda_log_zero0_1 = as.factor(sada$lambda_log_zero0_1)
sada$lambda_rich_zero0_1 = as.factor(sada$lambda_rich_zero0_1)

summary(sada) ##gives the number of "0" for the above-created variables, the
percentage of which is calculated below for the curves fitted by each model
#Gompertz, neglag
45/(388-48)*100
#Mod-Gompertz,  neglag
0/(388-230)*100
#Richards,  neglag
25/(388-88)*100
#Logistic,  neglag
57/(388-13)*100
#Spline,  neglag
12/(388-0)*100

###BOXPLOTTING GROWTH PARAMETERS
#UPLOADING FILE
#csv file with strains as rows (x 5) and the four growth parameters as columns,
one additional column specifying the fitting method
sada1 = read.csv2("saltdata_norm_ALLparam_concatenated.csv",strip.white =
TRUE,stringsAsFactors=T)

#FOR PUBLICATION, Drawing all growth parameters by method
par(mfrow=c(2,2),oma=c(3,1,3,1),mar=c(2.5,4,0,1))
boxplot(mu~method,data=sada1,boxwex=0.5,cex.axis=1.5,ylim=c(0, 0.2))#removes
outliers, goes up to 1.5)
mtext(expression(mu),2,cex=2.2,line=2.8, las=1)
boxplot(lambda~method,data=sada1,boxwex=0.5,cex.axis=1.5,ylim=c(-5, 15))#removes
outliers, goes up to 50
mtext(expression(lambda),2,cex=2.2,line=2.8, las=1)
boxplot(A~method,data=sada1,boxwex=0.5,cex.axis=1.5,ylim=c(0, 6)) #removes
outliers, goes up to 250
mtext("MaxOD",2,cex=2.2,line=2.8, las=3)
boxplot(integral~method,data=sada1,boxwex=0.5,cex.axis=1.5)
mtext("AUC",2,cex=2.2,line=2.8, las=3)
mtext("Method",1,outer=T,cex=2,line=1)
```

```
mtext("Growth parameters by fitting method",3,cex=2,outer=T,line=1)

################################################
# THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
# AALTO-ARANEDA ET AL.
#
################################################
#
# VOLUME 6: BOXPLOTS FOR GROWTH PARAMETERS
#
################################################

### SETTING WORKING DIRECTORY

setwd("Z:/Desktop/R_juttuja/suola/figstables")

### IMPORT DATA IN CSV
#csv file where strains in rows and other variables (incl. OD, spline
parameters, clusters) in columns
data = read.csv2("saltdata_norm_spline_param_OD.csv",strip.white =
TRUE,stringsAsFactors=T)

# inspect file
names(data) #column headings
str(data) #variables
summary(data) #structure

# changing cluster-variable to factor
data$CLU3_AUC=factor(data$CLU3_AUC)

### BOXPLOTTING BY BEST FITTING PARAMETERS
# Drawing boxplots of spline growth parameters in serotype groups

par(mfrow=c(2,2),oma=c(3,1,3,1),mar=c(2.5,4,0,1))
boxplot(mu_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)
mtext(expression(mu),2,cex=2.1,line=2.8, las=1)
boxplot(lambda_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)
mtext(expression(lambda),2,cex=2.1,line=2.8, las=1)
boxplot(A_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)
mtext("MaxOD",2,cex=2.1,line=2.8, las=3)
boxplot(integral_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)
mtext("AUC",2,cex=2.1,line=2.8, las=3)
mtext("Serotype",1,outer=T,cex=2,line=1)
mtext("Spline growth parameters by serotype",3,cex=2,outer=T,line=1)

# drawing boxplots of spline growth parameters in cluster groups

par(mfrow=c(2,2),oma=c(3,1,3,1),mar=c(2.5,4,0,1))
boxplot(mu_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Good")
)
mtext(expression(mu),2,cex=2.1,line=2.8, las=1)
boxplot(lambda_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Go
od"))
mtext(expression(lambda),2,cex=2.1,line=2.8, las=1)
boxplot(A_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Good"))
mtext("MaxOD",2,cex=2.1,line=2.8, las=3)
boxplot(integral_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","
Good"))
mtext("AUC",2,cex=2.1,line=2.8, las=3)
mtext("Growth cluster",1,outer=T,cex=2,line=1)
mtext("Spline fit growth parameters by growth cluster",3,cex=2,outer=T,line=1)

# FOR PUBLICATION
# drawing boxplots of spline growth parameters in serotype and cluster groups

par(mfrow=c(4,2),oma=c(1,1,3,1),mar=c(2.5,4,0,1))
boxplot(mu_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Good")
,cex.axis=1.5, col=c(2,4,5))
mtext(expression(mu),2,cex=2,line=2.5, las=1)
```

```
mtext("Growth cluster",3,cex=1.6,line=1)
boxplot(mu_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)
mtext("Serotype",3,cex=1.6,line=1)

boxplot(lambda_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Go
od"),cex.axis=1.5, col=c(2,4,5))
mtext(expression(lambda),2,cex=2,line=2.5, las=1)
boxplot(lambda_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)

boxplot(A_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","Good"),
cex.axis=1.5, col=c(2,4,5))
mtext("MaxOD",2,cex=2,line=2.5, las=3)
boxplot(A_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)

boxplot(integral_spline~CLU3_AUC,data=data,boxwex=0.5,names=c("Poor","Average","
Good"),cex.axis=1.5, col=c(2,4,5))
mtext("AUC",2,cex=2,line=2.5, las=3)
boxplot(integral_spline~Serotype,data=data,boxwex=0.5,cex.axis=1.5)

################################################
# THE R PROTOCOL FOR LARGE-SCALE BIOSCREEN GROWTH EXPERIMENTS
# AALTO-ARANEDA ET AL.
#
################################################
#
# VOLUME 7: DRAWING STRAIN GROWTH CURVE GRAPH BY SEROTYPES AND CLUSTERS
#
################################################

### SETTING WORKING DIRECTORY
setwd("Z:/Desktop/R_juttuja/suola/figstables")

### ANNOTATING STRAIN OD-FILE
# importing data in csv format: save your raw data in excel into csv
# dataset: column 1: time (h), columns 2-389: each strain (absorbance value,
OD600): i.e. file where strains in columns and time in rows

rawdata = read.csv2("saltstrains_final_norm.csv",strip.white =
TRUE,stringsAsFactors=F)

# inspecting the file
names(rawdata) #column headings
str(rawdata) #variables
summary(rawdata) #structure

# tidying data: organizes each OD measurement to a single row and names the
variables
library(reshape2)
reshaped <- melt(rawdata, id=c("Time"), variable.name="strain",
                 value.name="OD600")

# inspect tidied data
summary(reshaped) #structure
head(reshaped, n=20) #first n rows of reshaped file

# Creating the file that describes how to annotate, NB! have a row corresponding
to each pairing variable in reshaped!
# csv file with strains as rows and serotypes, lab techs and clusters as columns

newcolumns <- read.csv2("Strain_clucolumns.csv",strip.white =
TRUE,stringsAsFactors=T)
head(newcolumns, n=20)
str(newcolumns)
summary(newcolumns)
newcolumns$CLU3_AUC=factor(newcolumns$CLU3_AUC)

class(newcolumns$strain)
class(reshaped$strain)
levels(newcolumns$strain)
```

```
levels(reshaped$strain)

#Annotating the data
library(dplyr)

# Combine the reshaped data with the new additional columns/variables, pairing
them by replicate value
annotated <- inner_join(reshaped, newcolumns, by="strain")

# Take a peek at the records in annotated
summary(annotated)
str(annotated)

### DRAW GROWTH CURVES
# loop to draw growth curves: this draws all of them (initial check that the
loop works)
repls=levels(annotated$strain)
plot(0:17,annotated$OD600[which(annotated$strain==repls[1])],ann=F,type="o",pch=
1,ylim=c(0,1))

for(i in 2:length(repls)){
  lines(0:17,annotated$OD600[which(annotated$strain==repls[i])],type="o",pch=1)

}

mtext("Time (h)",1,line=3)
mtext("OD600",2,line=3)
mtext("All strains in NaCl 9.0%",3,line=2)

### PLOT ALL STRAINS COLOR-CODED BY AUC 3-CLUSTER
# First creating separate datasets for each cluster

A=which(annotated$CLU3_AUC=="1")
B=which(annotated$CLU3_AUC=="2")
C=which(annotated$CLU3_AUC=="3")

dataA=annotated[A,]
dataB=annotated[B,]
dataC=annotated[C,]

dataA #strains in CLU1 "Poor"
dataB #strains in CLU2 "Average"
dataC #strains in CLU3 "Good"

replsA=levels(factor(dataA$strain))
replsB=levels(factor(dataB$strain))
replsC=levels(factor(dataC$strain))

# drawing plot
plot(0:17,dataA$OD600[which(dataA$strain==replsA[1])],ann=F,type="l",ylim=c(0,1)
,col="2")

for(i in 2:length(replsA)){
  lines(0:17,dataA$OD600[which(dataA$strain==replsA[i])],col="2")

}
for(i in 1:length(replsB)){
  lines(0:17,dataB$OD600[which(dataB$strain==replsB[i])],col="4")

}
for(i in 1:length(replsC)){
  lines(0:17,dataC$OD600[which(dataC$strain==replsC[i])],col="5")

}
legend(0,1,legend=c("Good","Average","Poor"),col=c(5,4,2),lty=c(rep(1,3)))
mtext("Time (h)",1,line=2.5)
mtext("Optical density (OD600)",2,line=2.5)
mtext("All strains in NaCl 9.0% by AUC growth clusters",3,line=1.5)
```

```
# drawing plot, greyscale
plot(0:17,dataA$OD600[which(dataA$strain==replsA[1])],ann=F,type="l",ylim=c(0,1)
,col="1")

for(i in 2:length(replsA)){
  lines(0:17,dataA$OD600[which(dataA$strain==replsA[i])],col="1")

}
for(i in 1:length(replsB)){
  lines(0:17,dataB$OD600[which(dataB$strain==replsB[i])],col="grey40")

}
for(i in 1:length(replsC)){
  lines(0:17,dataC$OD600[which(dataC$strain==replsC[i])],col="grey80")

}
legend(0,1,legend=c("Good","Average","Poor"),col=c("grey80","grey40",1),lty=c(re
p(1,3)))
mtext("Time (h)",1,line=2.5)
mtext("Optical density (OD600)",2,line=2.5)
mtext("All strains in NaCl 9.0% by AUC growth clusters",3,line=1.5)

### MULTIPANEL PLOTS BY SEROTYPE AND CLUSTER
# create separate datasets

a1=which(annotated$Serotype=="1/2a"&annotated$CLU3_AUC=="1")
a2=which(annotated$Serotype=="1/2a"&annotated$CLU3_AUC=="2")
a3=which(annotated$Serotype=="1/2a"&annotated$CLU3_AUC=="3")
b1=which(annotated$Serotype=="1/2b"&annotated$CLU3_AUC=="1")
b2=which(annotated$Serotype=="1/2b"&annotated$CLU3_AUC=="2")
b3=which(annotated$Serotype=="1/2b"&annotated$CLU3_AUC=="3")
c1=which(annotated$Serotype=="1/2c"&annotated$CLU3_AUC=="1")
c2=which(annotated$Serotype=="1/2c"&annotated$CLU3_AUC=="2")
c3=which(annotated$Serotype=="1/2c"&annotated$CLU3_AUC=="3")
d1=which(annotated$Serotype=="3a"&annotated$CLU3_AUC=="1")
d2=which(annotated$Serotype=="3a"&annotated$CLU3_AUC=="2")
d3=which(annotated$Serotype=="3a"&annotated$CLU3_AUC=="3")
e1=which(annotated$Serotype=="4b"&annotated$CLU3_AUC=="1")
e2=which(annotated$Serotype=="4b"&annotated$CLU3_AUC=="2")
e3=which(annotated$Serotype=="4b"&annotated$CLU3_AUC=="3")

dataa1=annotated[a1,]
dataa2=annotated[a2,]
dataa3=annotated[a3,]
datab1=annotated[b1,]
datab2=annotated[b2,]
datab3=annotated[b3,]
datac1=annotated[c1,]
datac2=annotated[c2,]
datac3=annotated[c3,] #empty
datad1=annotated[d1,]
datad2=annotated[d2,]
datad3=annotated[d3,] #empty
datae1=annotated[e1,] #empty
datae2=annotated[e2,]
datae3=annotated[e3,]

replsa1=levels(factor(dataa1$strain))
replsa2=levels(factor(dataa2$strain))
replsa3=levels(factor(dataa3$strain))
replsb1=levels(factor(datab1$strain))
replsb2=levels(factor(datab2$strain))
replsb3=levels(factor(datab3$strain))
replsc1=levels(factor(datac1$strain))
replsc2=levels(factor(datac2$strain))
replsc3=levels(factor(datac3$strain))
replsd1=levels(factor(datad1$strain))
replsd2=levels(factor(datad2$strain))
replsd3=levels(factor(datad3$strain))
```

```
replse1=levels(factor(datae1$strain))
replse2=levels(factor(datae2$strain))
replse3=levels(factor(datae3$strain))

# draw multipanel plot by serotypes
par(mfrow=c(3,2),oma=c(4,3,3,1),mar=c(2,3,1,1))

# 1/2a (a)
plot(0:17,dataa1$OD600[which(dataa1$strain==replsa1[1])],ann=F,type="l",cex.axis
=1.5,ylim=c(0,1),col="2")

for(i in 2:length(replsa1)){
  lines(0:17,dataa1$OD600[which(dataa1$strain==replsa1[i])],col="2")

}
for(i in 1:length(replsa2)){
  lines(0:17,dataa2$OD600[which(dataa2$strain==replsa2[i])],col="4")

}
for(i in 1:length(replsa3)){
  lines(0:17,dataa3$OD600[which(dataa3$strain==replsa3[i])],col="5")

}
legend(-3,1,legend="1/2a",cex=3,box.lty=0)

# 1/2b (b)
plot(0:17,datab1$OD600[which(datab1$strain==replsb1[1])],ann=F,type="l",
cex.axis=1.5,ylim=c(0,1),col="2")

for(i in 2:length(replsb1)){
  lines(0:17,datab1$OD600[which(datab1$strain==replsb1[i])],col="2")

}
for(i in 1:length(replsb2)){
  lines(0:17,datab2$OD600[which(datab2$strain==replsb2[i])],col="4")

}
for(i in 1:length(replsb3)){
  lines(0:17,datab3$OD600[which(datab3$strain==replsb3[i])],col="5")

}
legend(-3,1,legend="1/2b",cex=3,box.lty=0)

# 1/2c (c) NB! empty c3 left out
plot(0:17,datac1$OD600[which(datac1$strain==replsc1[1])],ann=F,cex.axis=1.5,type
="l",ylim=c(0,1),col="2")

for(i in 2:length(replsc1)){
  lines(0:17,datac1$OD600[which(datac1$strain==replsc1[i])],col="2")

}
for(i in 1:length(replsc2)){
  lines(0:17,datac2$OD600[which(datac2$strain==replsc2[i])],col="4")

}
legend(-3,1,legend="1/2c",cex=3,box.lty=0)

# 3a (d) NB! empty d3 left out
plot(0:17,datad1$OD600[which(datad1$strain==replsd1[1])],ann=F,cex.axis=1.5,type
="l",ylim=c(0,1),col="2")

for(i in 2:length(replsd1)){
  lines(0:17,datad1$OD600[which(datad1$strain==replsd1[i])],col="2")

}
for(i in 1:length(replsd2)){
  lines(0:17,datad2$OD600[which(datad2$strain==replsd2[i])],col="4")

}
```

```
legend(-3,1,legend="3a",cex=3,box.lty=0)

# 4b (e) NB! empty e1 left out
plot(0:17,datae2$OD600[which(datae2$strain==replse2[1])],ann=F,cex.axis=1.5,type
="l",ylim=c(0,1),col="4")

for(i in 2:length(replse2)){
  lines(0:17,datae2$OD600[which(datae2$strain==replse2[i])],col="4")

}
for(i in 1:length(replse3)){
  lines(0:17,datae3$OD600[which(datae3$strain==replse3[i])],col="5")

}
legend(-3,1,legend="4b",cex=3,box.lty=0)

# all serotypes
plot(0:17,dataA$OD600[which(dataA$strain==replsA[1])],ann=F,cex.axis=1.5,type="l
",ylim=c(0,1),col="2")

for(i in 2:length(replsA)){
  lines(0:17,dataA$OD600[which(dataA$strain==replsA[i])],col="2")

}
for(i in 1:length(replsB)){
  lines(0:17,dataB$OD600[which(dataB$strain==replsB[i])],col="4")

}
for(i in 1:length(replsC)){
  lines(0:17,dataC$OD600[which(dataC$strain==replsC[i])],col="5")

}
legend(-3,1.1,legend="All",cex=3,box.lty=0)
legend(0,0.7,legend=c("Good","Average","Poor"),col=c(5,4,2),lty=c(rep(1,3)))

#axis labels
mtext("Time (h)",1,outer=T,line=1.5, cex=1.5)
mtext("Optical density (OD600)",2,outer=T,line=0.1, cex=1.5)
#mtext("Strains by serotype and AUC cluster in NaCl 9.0%",3,outer=T,line=0.1,
cex=2)

################################################
END
################################################
```