Wrangling Data from microplate OD600 growth curve in 96WP

El Park

09 March, 2025

R Setup

```
rm(list = ls())
setwd("./")
getwd()
```

```
## [1] "C:/Users/parke/OneDrive - Indiana University/GitHub/ASG-fitness-effects/microplate_fitness_assay
```

```
#install.packages("vegan")
#install.packages("tidyverse")
require("png")
```

```
## Loading required package: png
```

```
require("dplyr")
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.2.1
## v purrr     1.0.4     v tidyr     1.3.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
require("grid")
```

```
## Loading required package: grid
```

```
require("tibble")
require("knitr")
```

```
## Loading required package: knitr
```

```
require("extrafont")
```

```
## Loading required package: extrafont
## Registering fonts with R
```

```
require("ggrepel");
```

```
## Loading required package: ggrepel
```

```
require("gridExtra")
```

```
## Loading required package: gridExtra
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
require("contrast")
```

```
## Loading required package: contrast
```

```
require("vegan")
```

```
## Loading required package: vegan

## Warning: package 'vegan' was built under R version 4.4.3

## Loading required package: permute

## Warning: package 'permute' was built under R version 4.4.3

## Loading required package: lattice
```

```r
sem <- function(x) sqrt(var(x)/length(x))
cv <- function(x) (sd(x)/mean(x))*100
```

Load and wrangle data

```r
#load data
data<-read.csv("../data/20250228OD600.csv")
design<-read.csv("../reference/20250228OD600Design.csv", header=FALSE)

#subset design file, "wells" will be strain plus treatment but I can update design file to include a se
media<-design[1:8,1:12]
strain<-design[9:16,1:12]
infection<-design[17:24, 1:12]


# Create custom labels
rows <- LETTERS[1:8]  # A to H
cols <- 1:12  # 1 to 12
custom_labels <- unlist(lapply(rows, function(row) paste0(row, cols)))

#convert to long
med.long <- as.data.frame(pivot_longer(media, cols = everything(),
                                       names_to = "Well", values_to = "Media"))
stn.long <- as.data.frame(pivot_longer(strain, cols = everything(),
                                       names_to = "Well", values_to = "Strain"))
inf.long <- as.data.frame(pivot_longer(infection, cols = everything(),
                                       names_to = "Well", values_to = "Infection"))
OD.long <- data %>%
  pivot_longer(cols = -Time, names_to="Well", values_to="OD600")

#combine
OD.long$Media<-rep(med.long$Media, length.out=55392)
OD.long$Strain<-rep(stn.long$Strain, length.out=55392)
OD.long$Infection<-rep(inf.long$Infection, length.out=55392)


#filter out "Blanks"
summary<-OD.long[!apply(OD.long, 1, function(row) any(row=="Blank")), ]
```

Calculate fitness

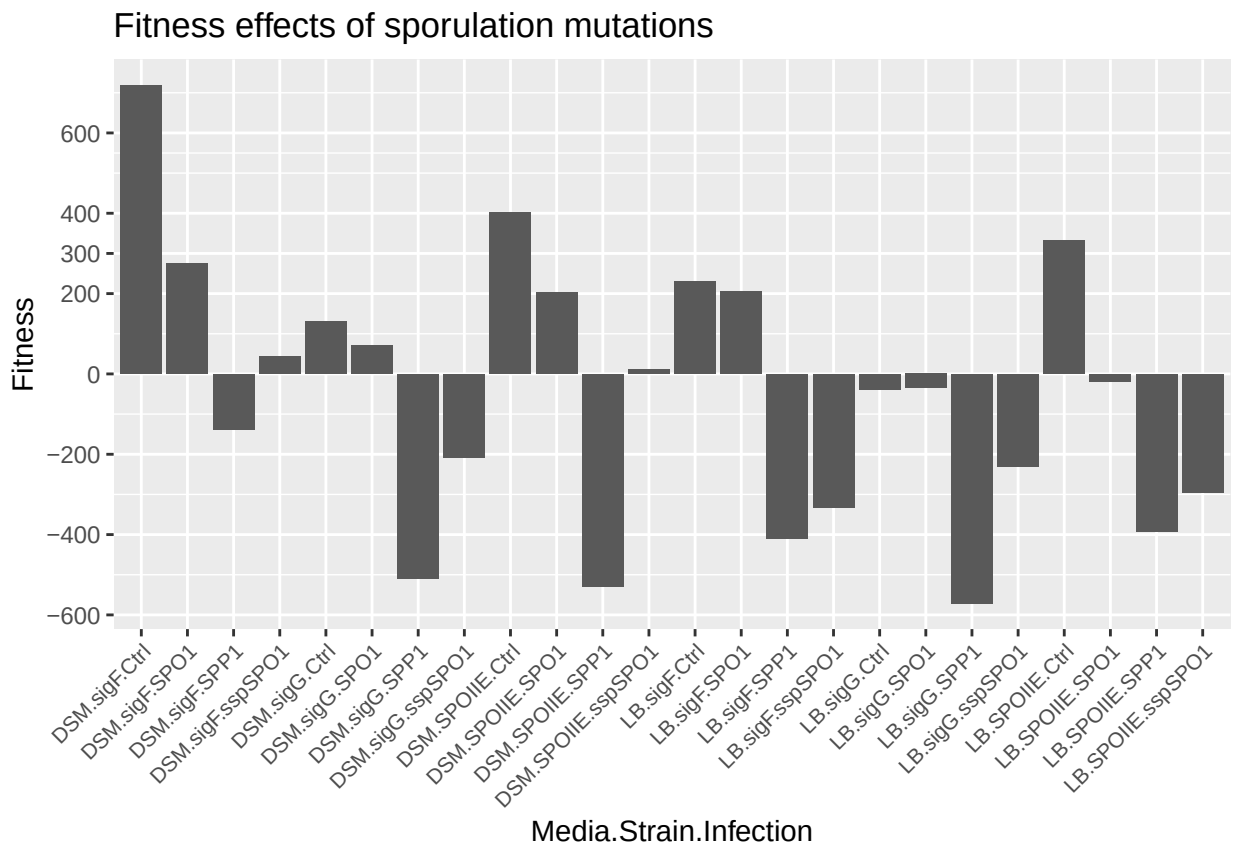
```r
#get unique combinations of Media, Infection, and Strain
combinations <- summary %>%
  distinct(Media, Infection, Strain)

#create data frames for each combination
for (i in 1:nrow(combinations)) {
  strain <- combinations$Strain[i]
  media <- combinations$Media[i]
  infection <- combinations$Infection[i]

  vector_name <- paste(strain, media, infection, sep = ".")
  assign(vector_name, summary %>%
```

```
    filter(Strain == strain, Media == media, Infection == infection) %>%
    select(Time, OD600))
}

#Calculate the average OD600 for each time point for wt strains
#allows us to obtain "fitness" values without unnecessarily creating power
wt_combinations <- combinations %>%
  filter(grepl("WT", Strain))

for (i in 1:nrow(wt_combinations)) {
  media <- wt_combinations$Media[i]
  infection <- wt_combinations$Infection[i]

  wt_vector_name <- paste("WT", media, infection, sep = ".")
  avg_vector_name <- paste("avg", media, infection, sep = ".")

  avg_vector <- get(wt_vector_name) %>%
    group_by(Time) %>%
    mutate(avg_OD600 = mean(OD600, na.rm = TRUE)) %>%
    ungroup()

  assign(avg_vector_name, avg_vector)
}

#get OD values of non wt combinations, subtract from corresponding wt OD avg, then take a sum of each v
non_wt_combinations <- combinations %>%
  filter(!grepl("WT", Strain))

fitness_list <- list()

for (i in 1:nrow(non_wt_combinations)) {
  strain <- non_wt_combinations$Strain[i]
  media <- non_wt_combinations$Media[i]
  infection <- non_wt_combinations$Infection[i]

  non_wt_vector_name <- paste(strain, media, infection, sep = ".")
  avg_vector_name <- paste("avg", media, infection, sep = ".")
  fitness_vector_name <- paste(media, strain, infection, "fitness", sep = ".")

  fitness_vector <- get(non_wt_vector_name)$OD600 - get(avg_vector_name)$avg_OD600

  assign(fitness_vector_name, fitness_vector)
  fitness_list[[fitness_vector_name]] <- fitness_vector
}

# Sum all fitness vectors into one data frame named pm
pm <- data.frame(
  media = character(),
  strain = character(),
  infection = character(),
  fitness_value = numeric()
)
```

```
for (name in names(fitness_list)) {
  parts <- strsplit(name, "\\.")[[1]]
  media <- parts[1]
  strain <- parts[2]
  infection <- parts[3]
  fitness_value <- sum(fitness_list[[name]])

  pm <- rbind(pm, data.frame(media = media, strain = strain, infection = infection, fitness_value = fitn
}

# Create the media.strain.infection variable manually
pm$media.strain.infection <- paste(pm$media, pm$strain, pm$infection, sep = ".")


# Plot fitness
plot.new()
```

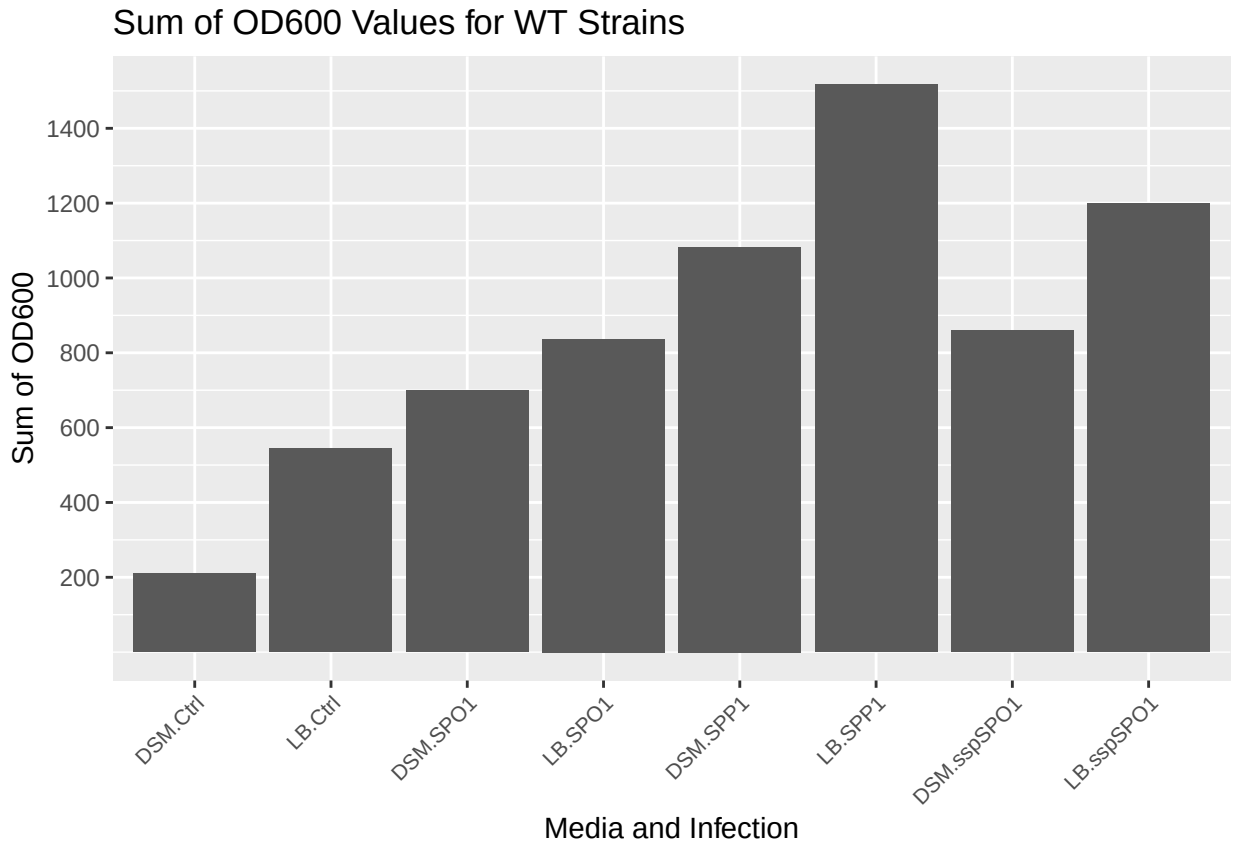
```
bg <- ggplot(pm, aes(x = media.strain.infection, y = fitness_value)) +
  geom_bar(stat = "identity") +
  labs(title = "Fitness effects of sporulation mutations", x = "Media.Strain.Infection", y = "Fitness") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8), # Rotate and adjust size of x-axis
        legend.key.size = unit(1, 'cm'),
        legend.position = "right") +
  scale_y_continuous(breaks = c(pretty(pm$fitness_value), 300)) # Add tick mark at 500
ggsave("20250228_fitness.png", plot = bg, width = 10, height = 12, dpi = 300)
```

```
print(bg)
```




```
#visualize OD sums of WT
wt_ft_list <- list()

for (i in 1:nrow(wt_combinations)) {
  media <- wt_combinations$Media[i]
  infection <- wt_combinations$Infection[i]

  wt_vector_name <- paste("WT", media, infection, sep = ".")

  sum_vector <- get(wt_vector_name) %>%
    group_by(Time) %>%
    summarise(sum_OD600 = sum(OD600, na.rm = TRUE)) %>%
    ungroup()

  wt_ft_list[[wt_vector_name]] <- sum_vector
}

# Sum all OD600 values into one data frame named wt.ft
wt.ft <- data.frame(
  media = character(),
  infection = character(),
  sum_OD600 = numeric()
)
```

```r
for (name in names(wt_ft_list)) {
  parts <- strsplit(name, "\\.")[[1]]
  media <- parts[2]
  infection <- parts[3]
  sum_OD600 <- sum(wt_ft_list[[name]]$sum_OD600)

  wt.ft <- rbind(wt.ft, data.frame(media = media, infection = infection, sum_OD600 = sum_OD600))
}

# Plot the sum of OD600 values
wtfit<-ggplot(wt.ft, aes(x = interaction(media, infection), y = sum_OD600)) +
  geom_bar(stat = "identity") +
  labs(title = "Sum of OD600 Values for WT Strains",
       x = "Media and Infection",
       y = "Sum of OD600") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8), # Rotate and adjust size of x-axis
        legend.key.size = unit(1, 'cm'),
        legend.position = "right") +
  scale_y_continuous(breaks = pretty(wt.ft$sum_OD600))
ggsave("20250228_wtfitness.png", plot = wtfit, width = 10, height = 6, dpi = 300)
print(wtfit)
```



Results/Interpretations:

Media-Maybe a linear regression?

DSM: 8 samples had an increase in fitness, 4 samples had a decrease. Out of the increases, half of them were moderately increased (>200); out of the decreases, 3 out of 4 were moderately decreased.

LB: 3 samples had an increase in fitness, 9 had a decrease. Out of the increases, all were moderately increased; out of the decreases, 6 were moderately decreased.

Results/Interpretations: Something about DSM increases survivability of mutants? Or maybe this could be due to occlusion of wells from lysis?

Infection

SPO1 (WT): 4 out of 6 samples infected with SPO1 were moderately increased in fitness, meaning a majority of the sporulation gene mutants faired better against SPO1 than WT.The other samples were either slightly increased or slightly decreased. Another trend is that in DSM, mutants are more likely to have an increase in fitness whereas they were less likely in LB.

sspSPO1 (Mutated virus): 4 out of 6 samples were moderately decreased, meaning this virus was more lethal in these strains than WT SPO1. Samples in DSM were less likely to have a difference in fitness from WT.

SPP1: All but one sample had the lowest fitness scores between both media. This sample still had a fitness score of less than -100. This virus was much more virulent in sporulation mutants regardless of media.

Other notes: I just remembered this is from 24 to 48 hours (I think) and checking the OD600 graphs, it seems like the growth curves are a little wonky. Looking back, most of the activity captured by the fitness assay occurs somewhere between 15-25 hours.

Strain

sig F: In LB The sigF strain is more fit than wt without infection, it does moderately better against SPO1, and there is a distinct decrease when infected with sspSPO1 and SPP1 In DSM sigF seems to grow much better than wt in general, moderately better when infected with SPO1, slightly worse when infected with SPP1, and very slightly better against sspSPO1.

sigG: In LB sigG does slightly worse/not much different from wt in general, but is much more susceptible to SPP1 infection and moderately more susceptible to sspSPO1. In DSM There's not much difference between LB and DSM, so perhaps something about the sigG mutation minimizes changes between media.

SPOIIE In LB SPOIIE seems to grow better than wt, which is both unexpected and different from previous experiments. It was more susceptible to SPP1 and sspSPO1 In DSM Also grows better in DSM, which was also unexpected. Something about DSM made it more resilient to SPO1 infection, which had been seen in the last experiment. It was more susceptible to SPP1 and less susceptible (same as WT) to sspSPO1 than in LB

Other notes: The last microplate experiment showed a slight decrease (slightly less than 100) in uninfected SPOIIE grown in DSM. This had been expected due to its inability to sporulate, so we hypothesized that the conditions in DSM are unfavorable for SPOIIE. However, in this experiment, we see a definite increase in fitness. Possibly due to the time frame captured by this run? Additionally, it seems most of the low fitness values of SPP1 are attributed by the resistance of wt to SPP1. It doesn't seem to change much (the mutants are still way more susceptible), but something to note?

Analysis

```
#
```

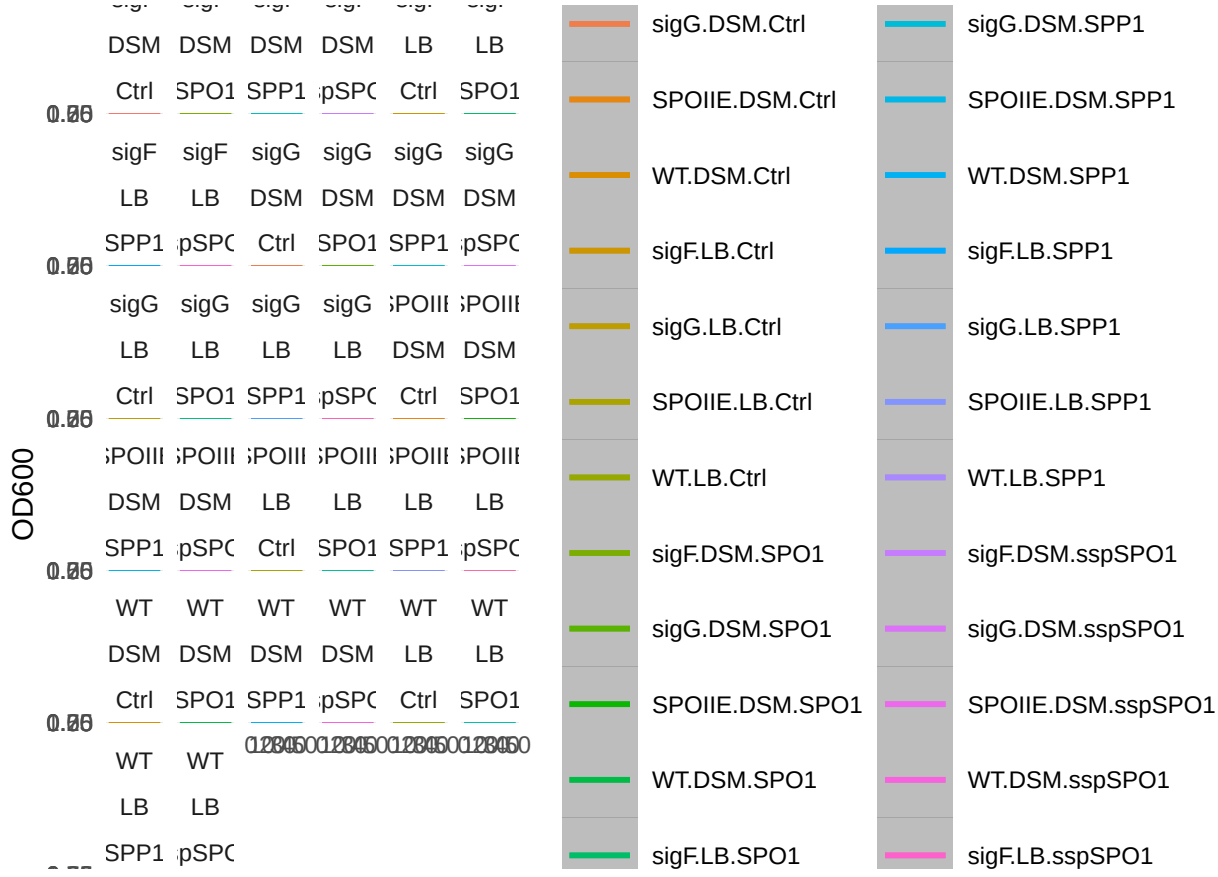
Plot data

```
setwd('C:/Users/parke/OneDrive - Indiana University/GitHub/ASG-fitness-effects/microplate_fitness_assay/

# Plot all growth over time in one plot, colored by design
p<-ggplot(summary, aes(x = Time, y = OD600, colour = interaction(Strain, Media, Infection)))+
  geom_smooth() +
  theme_minimal() + geom_smooth()+
  labs(title = "OD600 over Time for All Samples", x = "Time", y = "OD600", colour = "Media.Strain.Treatm
  theme(legend.key.size = unit(1, 'cm'), legend.position = "right")+
  facet_wrap(~Strain + Media + Infection)
ggsave("20250228_all.png", plot = p, width = 10, height = 12, dpi = 300)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
print(p)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

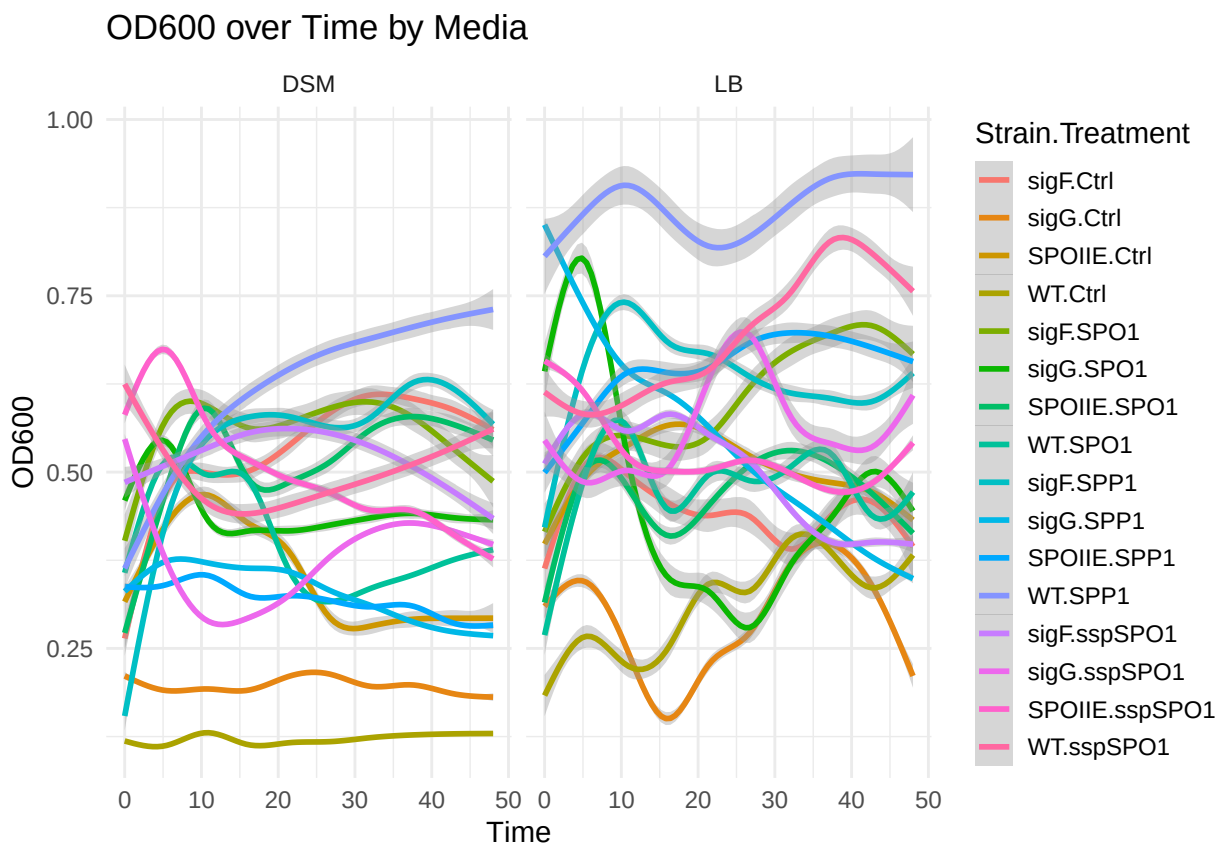


```
#plot and sort by Media
m<-ggplot(summary, aes(x = Time, y = OD600, colour = interaction(Strain, Infection)))+
  #geom_line() +
  theme_minimal() + geom_smooth()+
  labs(title = "OD600 over Time by Media", x = "Time", y = "OD600", colour = "Strain.Treatment") +
  theme(legend.key.size = unit(0.5, 'cm'), legend.position = "right")+
  facet_wrap(~Media)
ggsave("20250228_Media.png", plot = m, width = 8, height = 5, dpi = 300)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
print(m)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

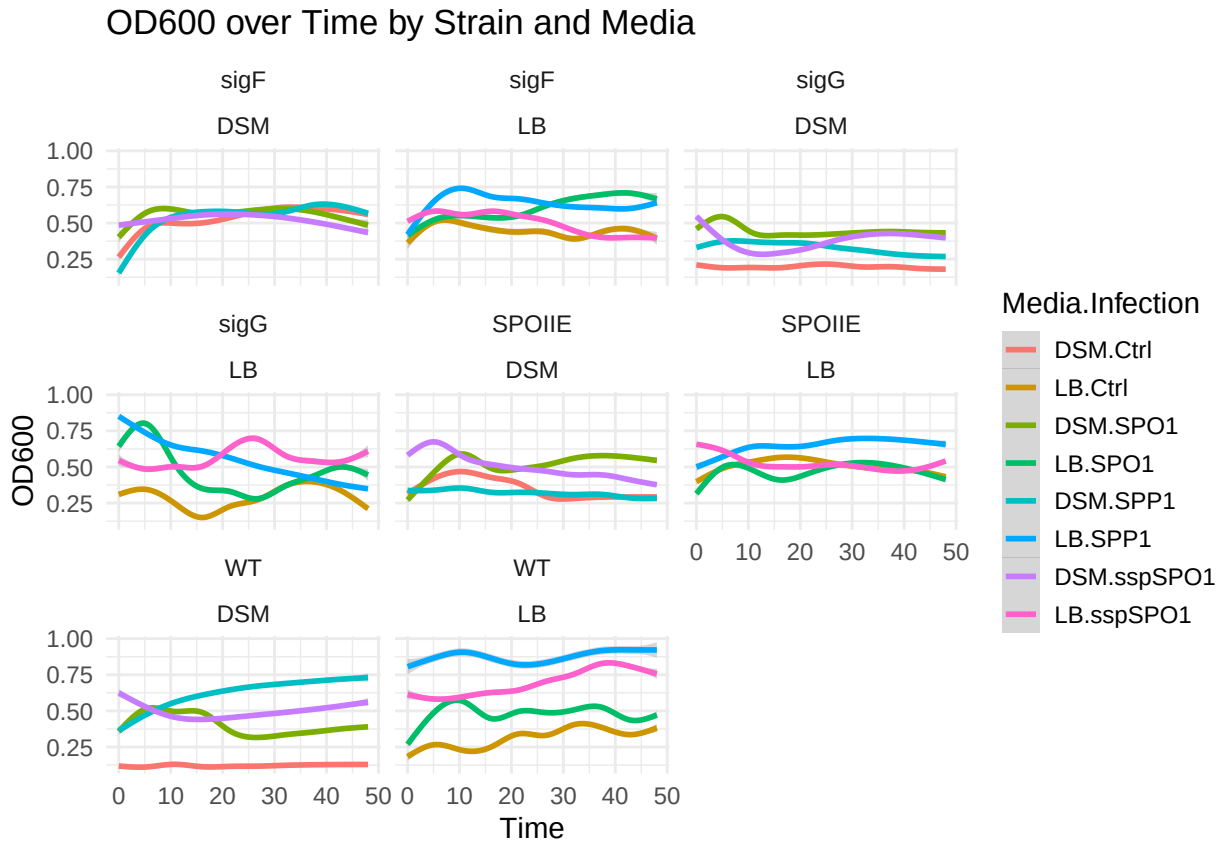



```
#plot and sort by treatment
d<-ggplot(summary, aes(x = Time, y = OD600, colour = interaction(Media, Infection))) +
  #geom_line() +
  theme_minimal() + geom_smooth()+
  labs(title = "OD600 over Time by Strain and Media", x = "Time", y = "OD600", colour = "Media.Infectio
  theme(legend.key.size = unit(0.5, 'cm'), legend.position = "right")+
  facet_wrap(~Strain+Media)
ggsave("20250228_Strains.png", plot = d, width = 8, height = 6, dpi = 300)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
print(d)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

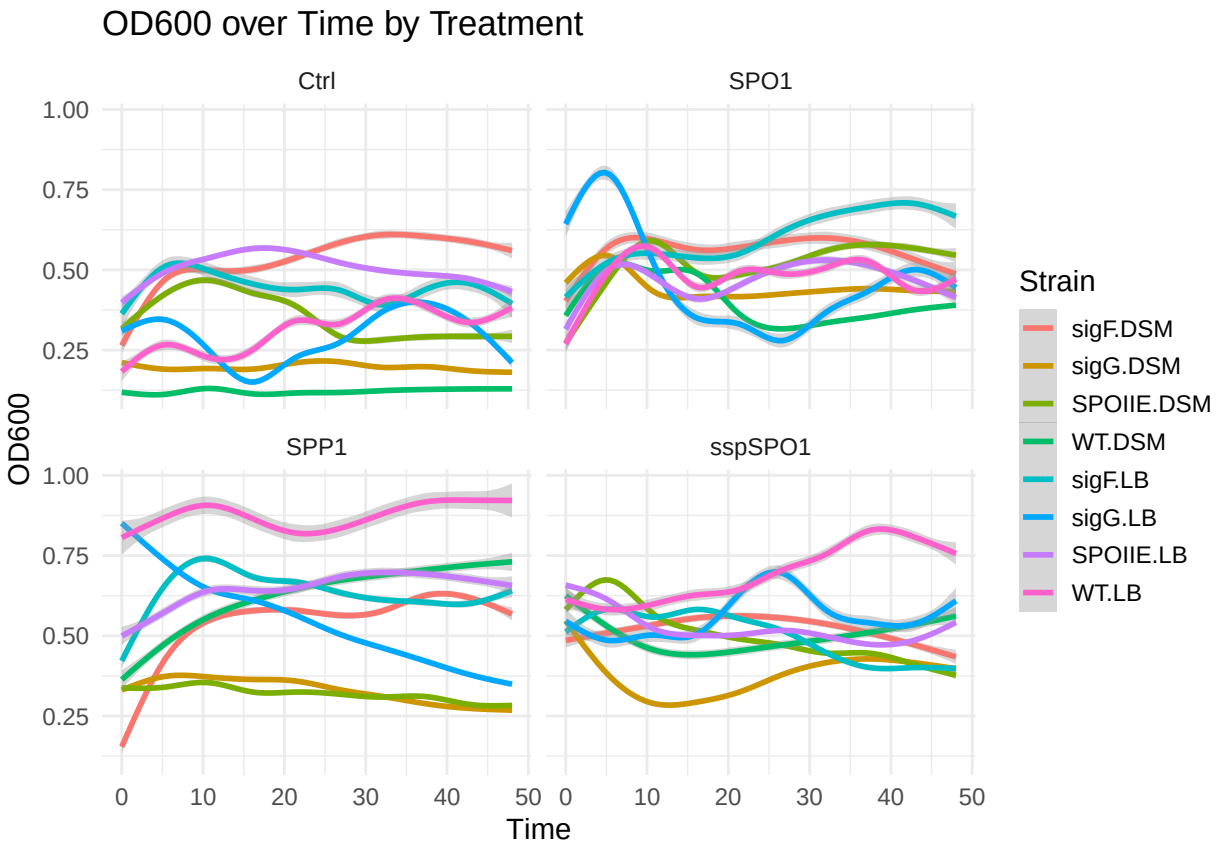



```
#plot and sort by infection
s <- ggplot(summary, aes(x = Time, y = OD600, colour = interaction(Strain, Media))) +
  theme_minimal() +
  geom_smooth() +
  labs(title = "OD600 over Time by Treatment", x = "Time", y = "OD600", colour = "Strain") +
  theme(legend.key.size = unit(0.5, 'cm'), legend.position = "right") +
  facet_wrap(~Infection)
ggsave("20250228_Infection.png", plot = d, width = 6, height = 4, dpi = 300)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
print(s)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```


OD600 over Time by Treatment
Ctrl
SPO1
SPP1
sspSPO1
OD600
1.00
0.75
0.50
0.25
0
10
20
30
40
50
Time
Strain
sigF.DSM
sigG.DSM
SPOIIE.DSM
WT.DSM
sigF.LB
sigG.LB
SPOIIE.LB
WT.LB

Results: >