

Pseudomonas Epiphytic Growth and Virulence Analysis

2024 Summer

Contents

Load Libraries	1
Goals	2
Virulence Analysis	2
Caculcate survival probabilities for each strain and create dataframe	2
Make Kaplan-Meier Plot	5
Epiphytic Growth Analysis	6
Calculate mean/variance epiphytic growth ability	6
Plot Epiphytic Growth	7
Combine epiphytic and virulence data	8
Plot it for different times	9
Session Information	11

Load Libraries

```
pacman::p_load(ggplot2, readxl, ggbeeswarm, dplyr, tidyverse, devtools, cowplot,
  knitr, survival, here, tibble, lubridate, formatR, gridExtra, ggsurvfit, gtsummary,
  tidycmprsk, install = FALSE)

# Load strain colors green = syringae, blue = fluorescences, brown = parallactic
strain_colors <- c(`194` = "sienna", `200` = "dodgerblue", `204` = "dodgerblue2",
  `205` = "dodgerblue3", `215` = "springgreen3", `216` = "dodgerblue4", `220` = "sienna4",
  `221` = "deepskyblue", `227` = "deepskyblue2", `228` = "deepskyblue3", B728a = "springgreen4",
  Cit7 = "springgreen2", Control = "black", pisi = "springgreen1")
```

```
pseud_epi_growth_2024summer_R <- read_excel("~/Desktop/Cornell/Hendry Lab/pseud-epi-growth/pseud_epi_gr
aphid_virulence_data <- read_csv("~/Desktop/Cornell/Hendry Lab/pseud-epi-growth/others_data/virulence_n
```

```
## Rows: 4973 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (3): date, treatment, replicate
## dbl (3): individual, censored, time
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Goals

- Create Kaplan-Meier curve for Pseud. virulence data
- Use stats (Wilcox?) to determine statistical significance of each strain
- Compare virulence data with epiphytic growth ability

Virulence Analysis

Note: *In order to help me with this analysis, I am using the following sites - Survival Analysis in R and Hazard Ratio: Interpretation & Definition.*

Calculate survival probabilities for each strain and create dataframe

```
# Fit the survival model
km_fit <- survfit(Surv(time, censored) ~ treatment, data = aphid_virulence_data)

# Extract survival probabilities at specific time points
time_points <- c(24, 48, 72)
km_summary <- summary(km_fit, times = time_points)

# Initialize empty lists to store the results
times_list <- list()
treatment_list <- list()
surv_prob_list <- list()

# Loop over each treatment group and extract survival probabilities at
# specified time points
for (i in 1:length(km_fit$strata)) {
  treatment_name <- names(km_fit$strata)[i]
  for (t in time_points) {
    idx <- which(km_summary$time == t & km_summary$strata == treatment_name)
    if (length(idx) > 0) {
      times_list <- c(times_list, t)
      treatment_list <- c(treatment_list, treatment_name)
      surv_prob_list <- c(surv_prob_list, km_summary$surv[idx])
    } else {
      times_list <- c(times_list, t)
      treatment_list <- c(treatment_list, treatment_name)
      surv_prob_list <- c(surv_prob_list, NA)
    }
  }
}
```

```

    }
}

# Create the data frame
surv_probs <- data.frame(time = unlist(times_list), treatment = unlist(treatment_list),
  surv_prob = unlist(surv_prob_list))

# Replace 'treatment=' with an empty string
surv_probs$treatment <- gsub("treatment=", "", surv_probs$treatment)

# Print the data frame
print(surv_probs)

```

```

##      time treatment  surv_prob
## 1     24        194 0.77891156
## 2     48        194 0.18707483
## 3     72        194 0.12244898
## 4     24        200 0.53159851
## 5     48        200 0.05576208
## 6     72        200 0.02230483
## 7     24        204 0.54929577
## 8     48        204 0.08450704
## 9     72        204 0.05281690
## 10    24        205 0.56949153
## 11    48        205 0.11525424
## 12    72        205 0.07118644
## 13    24        215 0.79513889
## 14    48        215 0.44791667
## 15    72        215 0.22222222
## 16    24        216 0.56250000
## 17    48        216 0.27430556
## 18    72        216 0.15625000
## 19    24        220 0.35135135
## 20    48        220 0.19256757
## 21    72        220 0.17229730
## 22    24        221 0.65543071
## 23    48        221 0.21722846
## 24    72        221 0.09363296
## 25    24        227 0.40000000
## 26    48        227 0.22033898
## 27    72        227 0.16949153
## 28    24        228 0.65671642
## 29    48        228 0.15298507
## 30    72        228 0.10447761
## 31    24      B728a 0.68041237
## 32    48      B728a 0.40549828
## 33    72      B728a 0.10996564
## 34    24       Cit7 0.75609756
## 35    48       Cit7 0.37630662
## 36    72       Cit7 0.17073171
## 37    24   Control 0.94777397
## 38    48   Control 0.87243151
## 39    72   Control 0.81678082

```

```
## 40    24      pisi 0.88250653
## 41    48      pisi 0.79634465
## 42    72      pisi 0.74151436
```

```
# If you need to save it to a file, you can use the following command
# write.csv(surv_probs, 'survival_probabilities.csv', row.names = FALSE)
```

Plot Survival Dataframe at different times

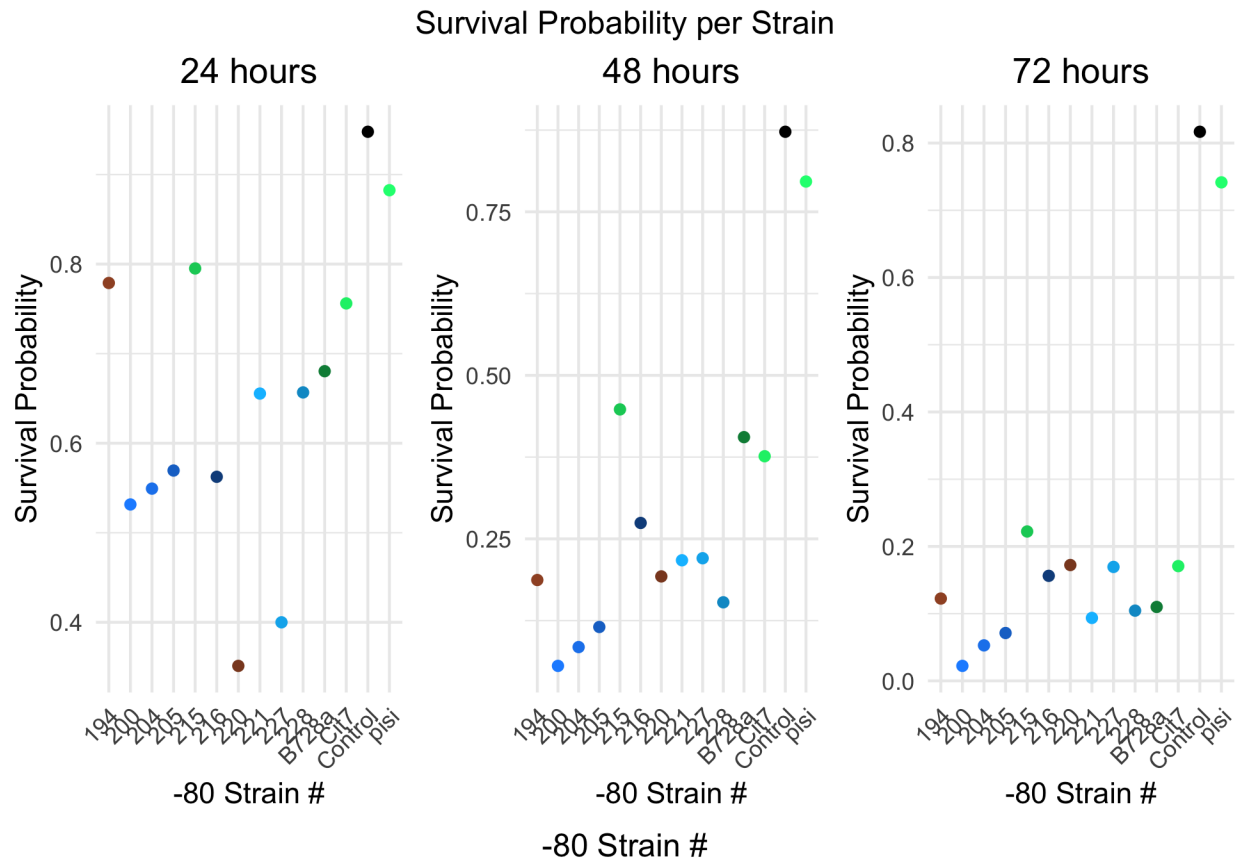
```
# subset data by time Subset data for three time points
surv_probs_24 <- surv_probs %>%
  filter(time == 24)
surv_probs_48 <- surv_probs %>%
  filter(time == 48)
surv_probs_72 <- surv_probs %>%
  filter(time == 72)

surv_plot_24 <- ggplot(data = surv_probs_24, aes(x = treatment, y = surv_prob, color = treatment,
  fill = treatment)) + geom_point() + labs(title = "24 hours", x = "-80 Strain #",
  y = "Survival Probability") + theme_minimal() + scale_fill_manual(values = strain_colors) +
  scale_color_manual(values = strain_colors) + theme(plot.title = element_text(hjust = 0.5),
  axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")

surv_plot_48 <- ggplot(data = surv_probs_48, aes(x = treatment, y = surv_prob, color = treatment,
  fill = treatment)) + geom_point() + labs(title = "48 hours", x = "-80 Strain #",
  y = "Survival Probability") + theme_minimal() + scale_fill_manual(values = strain_colors) +
  scale_color_manual(values = strain_colors) + theme(plot.title = element_text(hjust = 0.5),
  axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")

surv_plot_72 <- ggplot(data = surv_probs_72, aes(x = treatment, y = surv_prob, color = treatment,
  fill = treatment)) + geom_point() + labs(title = "72 hours", x = "-80 Strain #",
  y = "Survival Probability") + theme_minimal() + scale_fill_manual(values = strain_colors) +
  scale_color_manual(values = strain_colors) + theme(plot.title = element_text(hjust = 0.5),
  axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")

# Combine the plots
combined_virulence_plot <- grid.arrange(surv_plot_24, surv_plot_48, surv_plot_72,
  nrow = 1, top = "Survival Probability per Strain", bottom = "-80 Strain #")
```



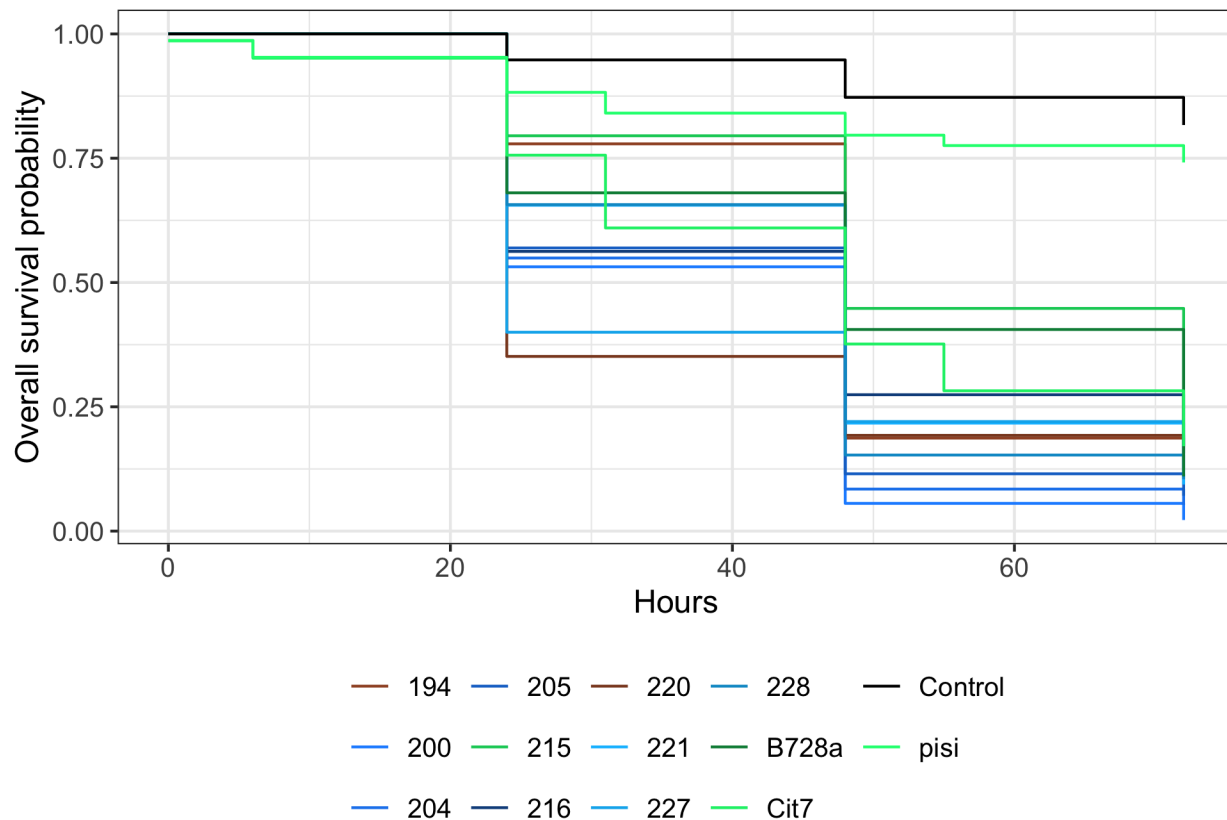
```
combined_virulence_plot
```

```
## TableGrob (3 x 3) "arrange": 5 grobs
##   z      cells    name      grob
## 1 1 (2-2,1-1) arrange  gtable[layout]
## 2 2 (2-2,2-2) arrange  gtable[layout]
## 3 3 (2-2,3-3) arrange  gtable[layout]
## 4 4 (1-1,1-3) arrange text[GRID.text.110]
## 5 5 (3-3,1-3) arrange text[GRID.text.111]
```

Make Kaplan-Meier Plot

Cohort Survival Curve

```
survfit2(Surv(time, censored) ~ treatment, data = aphid_virulence_data) %>%
  ggssurvfit() + labs(x = "Hours", y = "Overall survival probability") + scale_color_manual(values = s
```



Epiphytic Growth Analysis

Calculate mean/variance epiphytic growth ability

```
# Replace NA with a lower value or remove them for visualization For this
# example, I'll remove rows with NA in CFU_per_10_leafdiscs
cleaned_data <- pseud_epi_growth_2024summer_R %>%
  filter(!is.na(CFU_per_10_leafdiscs))

# Convert CFU_per_10_leafdiscs to numeric, handling scientific notation
cleaned_data$CFU_per_10_leafdiscs <- as.numeric(gsub("<", "", cleaned_data$CFU_per_10_leafdiscs))

## Warning: NAs introduced by coercion

# Subset data to remove unfinished strains
cleaned_data <- cleaned_data %>%
  filter(!strain %in% c("205", "pisi", "215"))

# Calculate mean and variance for each strain
strain_stats <- cleaned_data %>%
  group_by(strain) %>%
  summarise(mean_CFU = mean(CFU_per_10_leafdiscs, na.rm = TRUE), sd_CFU = sd(CFU_per_10_leafdiscs,
```

```

    na.rm = TRUE))

# Print the calculated statistics
print(strain_stats)

```

```

## # A tibble: 10 x 3
##   strain mean_CFU sd_CFU
##   <chr>      <dbl> <dbl>
## 1 194      156472. 142871.
## 2 200      261214. 180932.
## 3 204      186888. 189203.
## 4 216         6788.  7614.
## 5 220      172543. 189362.
## 6 221      259556. 193140.
## 7 227         82872. 153323.
## 8 228         22875.  23673.
## 9 B728a     71144. 135614.
## 10 Cit7      61906. 117046.

```

Plot Epiphytic Growth

```

# Create the box plot
ggplot(data = cleaned_data, aes(x = strain, y = CFU_per_10_leafdiscs, color = strain,
  fill = strain)) + geom_boxplot(outlier.shape = NA, alpha = 0.3) + geom_beeswarm(stroke = 0.5,
  size = 0.8, alpha = 0.8) + labs(title = "CFU per 10 Leaf Discs Across Samples",
  x = "-80 Strain #", y = "CFU per 10 Leaf Discs") + theme_minimal() + scale_fill_manual(values = strain_colors) +
  scale_color_manual(values = strain_colors) + theme(plot.title = element_text(hjust = 0.5),
  axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")

```

```

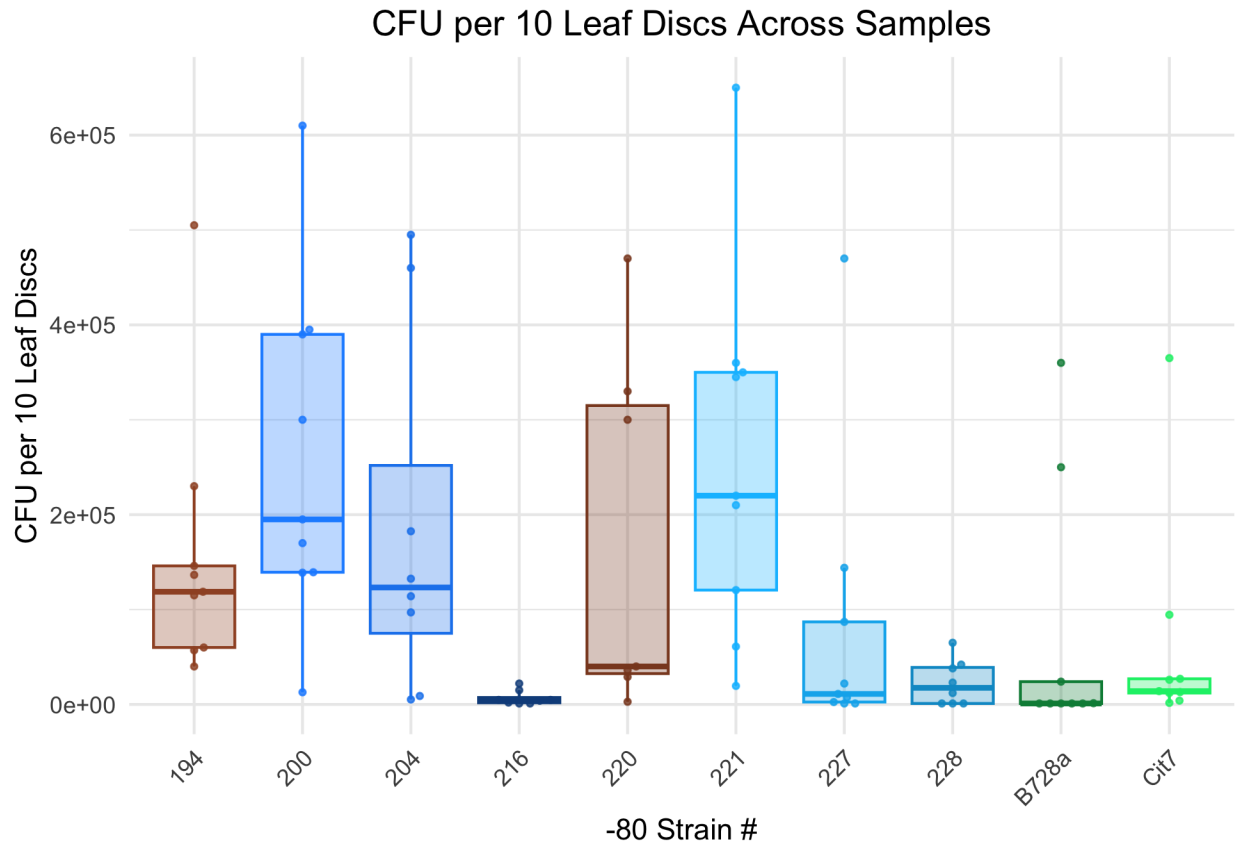
## Warning: Removed 8 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```

```

## Warning: Removed 8 rows containing missing values or values outside the scale range
## ('geom_point()').

```



Combine epiphytic and virulence data

```
# Assuming strain_stats has a column 'strain' and surv_probs has a column
# 'treatment' Rename columns if necessary to match the key for joining
strain_stats <- strain_stats %>%
  rename(treatment = strain)

# Combine strain_stats and surv_probs using left_join
epi_virulence_data <- left_join(strain_stats, surv_probs, by = "treatment")

# Subset data to remove unfinished strains
epi_virulence_data <- epi_virulence_data %>%
  filter(!treatment %in% c("205", "pisi", "215"))

# Print the combined data
print(epi_virulence_data)
```

```
## # A tibble: 30 x 5
##   treatment mean_CFU sd_CFU time surv_prob
##   <chr>         <dbl> <dbl> <dbl>    <dbl>
## 1 194         156472. 142871.    24    0.779
## 2 194         156472. 142871.    48    0.187
## 3 194         156472. 142871.    72    0.122
```



```
## 4 200      261214. 180932.    24    0.532
## 5 200      261214. 180932.    48    0.0558
## 6 200      261214. 180932.    72    0.0223
## 7 204      186888. 189203.    24    0.549
## 8 204      186888. 189203.    48    0.0845
## 9 204      186888. 189203.    72    0.0528
## 10 216      6788.   7614.     24    0.562
## # i 20 more rows
```

```
# Subset data for three time points
subset_data_24 <- epi_virulence_data %>%
  filter(time == 24)
subset_data_48 <- epi_virulence_data %>%
  filter(time == 48)
subset_data_72 <- epi_virulence_data %>%
  filter(time == 72)

# Calculate correlation coefficient between survival probability and epiphytic
# growth ability
correlation_24 <- cor(subset_data_24$surv_prob, subset_data_24$mean_CFU)
correlation_48 <- cor(subset_data_48$surv_prob, subset_data_48$mean_CFU)
correlation_72 <- cor(subset_data_72$surv_prob, subset_data_72$mean_CFU)
```

Plot it for different times

```
# Create scatter plot for 24 hours
p_24 <- ggplot(subset_data_24, aes(x = mean_CFU, y = surv_prob, color = treatment)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "24 Hours",
       x = "", # Remove x-axis label
       y = "Survival Probability",
       caption = paste("Correlation Coefficient:", round(correlation_24, 2))) +
  scale_color_manual(values = strain_colors) +
  theme_minimal() +
  theme(legend.position = "none", axis.text.x = element_blank()) # Remove x-axis labels

# Create scatter plot for 48 hours
p_48 <- ggplot(subset_data_48, aes(x = mean_CFU, y = surv_prob, color = treatment)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "48 Hours",
       x = "", # Remove x-axis label
       y = "", # Remove y-axis label
       caption = paste("Correlation Coefficient:", round(correlation_48, 2))) +
  scale_color_manual(values = strain_colors) +
  theme_minimal() +
  theme(legend.position = "none", axis.text.x = element_blank(), axis.title.y = element_blank()) # Remove y-axis title

# Create scatter plot for 72 hours
p_72 <- ggplot(subset_data_72, aes(x = mean_CFU, y = surv_prob, color = treatment)) +
```

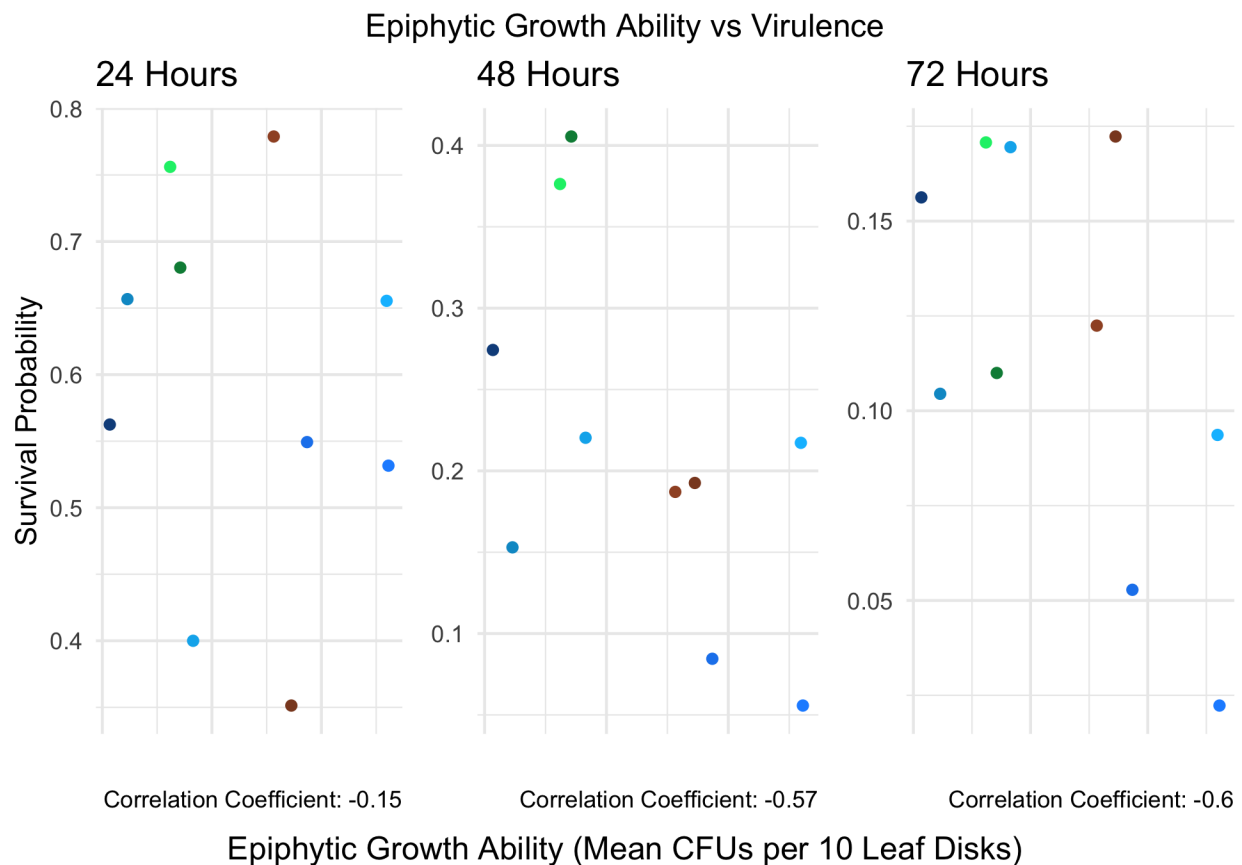
```

geom_point() +
geom_smooth(method = "lm", se = FALSE) +
labs(title = "72 Hours",
      x = "", # Remove x-axis label
      y = "", # Remove y-axis label
      caption = paste("Correlation Coefficient:", round(correlation_72, 2))) +
scale_color_manual(values = strain_colors) +
theme_minimal() +
theme(legend.position = "none", axis.text.x = element_blank(), axis.title.y = element_blank()) # Rem

# Combine the plots
combined_epi_virulence_plot <- grid.arrange(p_24, p_48, p_72, nrow = 1,
      top = "Epiphytic Growth Ability vs Virulence",
      bottom = "Epiphytic Growth Ability (Mean CFUs per 10 Leaf Disks)")

## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'

```



```
combined_epi_virulence_plot
```

```

## TableGrob (3 x 3) "arrange": 5 grobs
##   z      cells  name      grob
## 1 1 (2-2,1-1) arrange  gtable[layout]

```

```
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (2-2,3-3) arrange      gtable[layout]
## 4 4 (1-1,1-3) arrange text[GRID.text.433]
## 5 5 (3-3,1-3) arrange text[GRID.text.434]
```

Session Information

```
devtools::session_info()
```

```
## - Session info -----
##   setting      value
##   version      R version 4.4.0 (2024-04-24)
##   os           macOS Ventura 13.4
##   system       x86_64, darwin20
##   ui           X11
##   language     (EN)
##   collate      en_US.UTF-8
##   ctype        en_US.UTF-8
##   tz           America/New_York
##   date         2024-06-07
##   pandoc       3.1.11 @ /Applications/RStudio.app/Contents/Resources/app/quarto/bin/tools/x86_64/ (via rm
##
## - Packages -----
##   ! package      * version date (UTC) lib source
##   P backports     1.4.1   2021-12-13 [?] CRAN (R 4.4.0)
##   P beeswarm      0.4.0   2021-06-01 [?] CRAN (R 4.4.0)
##   P bit           4.0.5   2022-11-15 [?] CRAN (R 4.4.0)
##   P bit64         4.0.5   2020-08-30 [?] CRAN (R 4.4.0)
##   P broom         1.0.6   2024-05-17 [?] CRAN (R 4.4.0)
##   P broom.helpers 1.15.0  2024-04-05 [?] CRAN (R 4.4.0)
##   P cachem        1.0.8   2023-05-01 [?] CRAN (R 4.4.0)
##   P cellranger    1.1.0   2016-07-27 [?] CRAN (R 4.4.0)
##   P cli           3.6.2   2023-12-11 [?] CRAN (R 4.4.0)
##   P colorspace    2.1-0   2023-01-23 [?] CRAN (R 4.4.0)
##   P cowplot       * 1.1.3   2024-01-22 [?] CRAN (R 4.4.0)
##   P crayon        1.5.2   2022-09-29 [?] CRAN (R 4.4.0)
##   P devtools      * 2.4.5   2022-10-11 [?] RSPM
##   P digest        0.6.35  2024-03-11 [?] CRAN (R 4.4.0)
##   P dplyr         * 1.1.4   2023-11-17 [?] CRAN (R 4.4.0)
##   P ellipsis      0.3.2   2021-04-29 [?] RSPM
##   P evaluate      0.23    2023-11-01 [?] CRAN (R 4.4.0)
##   P fansi         1.0.6   2023-12-08 [?] CRAN (R 4.4.0)
##   P farver        2.1.2   2024-05-13 [?] CRAN (R 4.4.0)
##   P fastmap       1.1.1   2023-02-24 [?] CRAN (R 4.4.0)
##   P forcats       * 1.0.0   2023-01-29 [?] CRAN (R 4.4.0)
##   P formatR       * 1.14    2023-01-17 [?] RSPM
##   P fs            1.6.4   2024-04-25 [?] CRAN (R 4.4.0)
##   P generics      0.1.3   2022-07-05 [?] CRAN (R 4.4.0)
##   P ggbeeswarm    * 0.7.2   2023-04-29 [?] CRAN (R 4.4.0)
```

##	P ggplot2	* 3.5.1	2024-04-23	[?]	CRAN	(R 4.4.0)
##	P ggsurvfit	* 1.1.0	2024-05-08	[?]	CRAN	(R 4.4.0)
##	P glue	1.7.0	2024-01-09	[?]	CRAN	(R 4.4.0)
##	P gridExtra	* 2.3	2017-09-09	[?]	RSPM	
##	P gt	0.10.1	2024-01-17	[?]	CRAN	(R 4.4.0)
##	P gtable	0.3.5	2024-04-22	[?]	CRAN	(R 4.4.0)
##	P gtsummary	* 1.7.2	2023-07-15	[?]	CRAN	(R 4.4.0)
##	P here	* 1.0.1	2020-12-13	[?]	CRAN	(R 4.4.0)
##	P hms	1.1.3	2023-03-21	[?]	CRAN	(R 4.4.0)
##	P htmltools	0.5.8.1	2024-04-04	[?]	CRAN	(R 4.4.0)
##	P htmlwidgets	1.6.4	2023-12-06	[?]	CRAN	(R 4.4.0)
##	P httpuv	1.6.15	2024-03-26	[?]	RSPM	
##	P knitr	* 1.46	2024-04-06	[?]	CRAN	(R 4.4.0)
##	P labeling	0.4.3	2023-08-29	[?]	CRAN	(R 4.4.0)
##	P later	1.3.2	2023-12-06	[?]	RSPM	
##	P lattice	0.22-6	2024-03-20	[?]	CRAN	(R 4.4.0)
##	P lifecycle	1.0.4	2023-11-07	[?]	CRAN	(R 4.4.0)
##	P lubridate	* 1.9.3	2023-09-27	[?]	CRAN	(R 4.4.0)
##	P magrittr	2.0.3	2022-03-30	[?]	CRAN	(R 4.4.0)
##	P Matrix	1.7-0	2024-03-22	[?]	CRAN	(R 4.4.0)
##	P memoise	2.0.1	2021-11-26	[?]	CRAN	(R 4.4.0)
##	P mime	0.12	2021-09-28	[?]	CRAN	(R 4.4.0)
##	P miniUI	0.1.1.1	2018-05-18	[?]	RSPM	
##	P munsell	0.5.1	2024-04-01	[?]	CRAN	(R 4.4.0)
##	P pacman	0.5.1	2019-03-11	[?]	CRAN	(R 4.4.0)
##	P pillar	1.9.0	2023-03-22	[?]	CRAN	(R 4.4.0)
##	P pkgbuild	1.4.4	2024-03-17	[?]	RSPM	
##	P pkgconfig	2.0.3	2019-09-22	[?]	CRAN	(R 4.4.0)
##	P pkgload	1.3.4	2024-01-16	[?]	RSPM	
##	P profvis	0.3.8	2023-05-02	[?]	RSPM	
##	P promises	1.3.0	2024-04-05	[?]	RSPM	
##	P purrr	* 1.0.2	2023-08-10	[?]	CRAN	(R 4.4.0)
##	P R6	2.5.1	2021-08-19	[?]	CRAN	(R 4.4.0)
##	P Rcpp	1.0.12	2024-01-09	[?]	CRAN	(R 4.4.0)
##	P readr	* 2.1.5	2024-01-10	[?]	CRAN	(R 4.4.0)
##	P readxl	* 1.4.3	2023-07-06	[?]	CRAN	(R 4.4.0)
##	P remotes	2.5.0	2024-03-17	[?]	CRAN	(R 4.4.0)
##	P renv	1.0.7	2024-04-11	[1]	CRAN	(R 4.4.0)
##	P rlang	1.1.3	2024-01-10	[?]	CRAN	(R 4.4.0)
##	P rmarkdown	2.26	2024-03-05	[?]	CRAN	(R 4.4.0)
##	P rprojroot	2.0.4	2023-11-05	[?]	CRAN	(R 4.4.0)
##	P rstudioapi	0.16.0	2024-03-24	[?]	CRAN	(R 4.4.0)
##	P scales	1.3.0	2023-11-28	[?]	CRAN	(R 4.4.0)
##	P sessioninfo	1.2.2	2021-12-06	[?]	RSPM	
##	P shiny	1.8.1.1	2024-04-02	[?]	RSPM	
##	P stringi	1.8.3	2023-12-11	[?]	CRAN	(R 4.4.0)
##	P stringr	* 1.5.1	2023-11-14	[?]	CRAN	(R 4.4.0)
##	P survival	* 3.6-4	2024-04-24	[?]	CRAN	(R 4.4.0)
##	P tibble	* 3.2.1	2023-03-20	[?]	CRAN	(R 4.4.0)
##	P tidycmprsk	* 1.0.0	2023-10-30	[?]	CRAN	(R 4.4.0)
##	P tidyr	* 1.3.1	2024-01-24	[?]	CRAN	(R 4.4.0)
##	P tidyselect	1.2.1	2024-03-11	[?]	CRAN	(R 4.4.0)
##	P tidyverse	* 2.0.0	2023-02-22	[?]	CRAN	(R 4.4.0)
##	P timechange	0.3.0	2024-01-18	[?]	CRAN	(R 4.4.0)

```

## P tzdb          0.4.0    2023-05-12 [?] CRAN (R 4.4.0)
## P urlchecker    1.0.1    2021-11-30 [?] RSPM
## P usethis       * 2.2.3    2024-02-19 [?] RSPM
## P utf8          1.2.4    2023-10-22 [?] CRAN (R 4.4.0)
## P vctrs         0.6.5    2023-12-01 [?] CRAN (R 4.4.0)
## P vipor         0.4.7    2023-12-18 [?] CRAN (R 4.4.0)
## P vroom         1.6.5    2023-12-05 [?] CRAN (R 4.4.0)
## P withr         3.0.0    2024-01-16 [?] CRAN (R 4.4.0)
## P xfun          0.43     2024-03-25 [?] CRAN (R 4.4.0)
## P xml2          1.3.6    2023-12-04 [?] CRAN (R 4.4.0)
## P xtable        1.8-4    2019-04-21 [?] RSPM
## P yaml          2.3.8    2023-12-11 [?] CRAN (R 4.4.0)
##
## [1] /Users/zahavahrojer/Desktop/Cornell/Hendry Lab/pseud-epi-growth/analysis/renv/library/macos/R-4
## [2] /Users/zahavahrojer/Library/Caches/org.R-project.R/R/renv/sandbox/macos/R-4.4/x86_64-apple-darw
##
## P -- Loaded and on-disk path mismatch.
##
## -----

```