

step4_virulence_test

lihui xiang

2025-04-10

As we've already localize 8 lactamase-related gene, we then use GFP tag labeled the accessory chromosome that contained those 8 genes.(those 8 genes are in the same cluster of one accessory unique chromosome), through confocol-microscope we filtered those A GFP tag was introduced into this chromosome, and two single-spore isolates carrying the GFP-tagged chromosome were obtained.

I.A hydroponic test tube virulence assay confirmed the GFP insertion did not affect the virulence of strain 89-1A.

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(readxl)
library(ggpubr)

# Read the CSV file using a relative path from your project root
df <- read.csv("gfp-tag isolates virulence test.csv")

# Rename the first column to 'variety' if needed
colnames(df)[1] <- "variety"

# Reshape the data into long format for ggplot compatibility
df_long <- df %>%
  pivot_longer(cols = -variety, names_to = "replicate", values_to = "score")

# Set factor levels to ensure plotting order
df_long$variety <- factor(df_long$variety, levels = c("89-1A", "1.11-3", "1.11-6", "1.12-4", "1.12-10",

# Summarize mean score and standard error by variety
df_summary <- df_long %>%
  group_by(variety) %>%
  summarise(
    mean_score = mean(score, na.rm = TRUE),
    se = sd(score, na.rm = TRUE) / sqrt(n())
  )

# Conduct pairwise t-tests comparing each strain with the control (89-1A)
control_group <- df_long %>% filter(variety == "89-1A")
comparison_varieties <- levels(df_long$variety)[-1]

# Initialize significance results dataframe
sig_results <- data.frame(variety = character(), p_value = numeric())
```

```

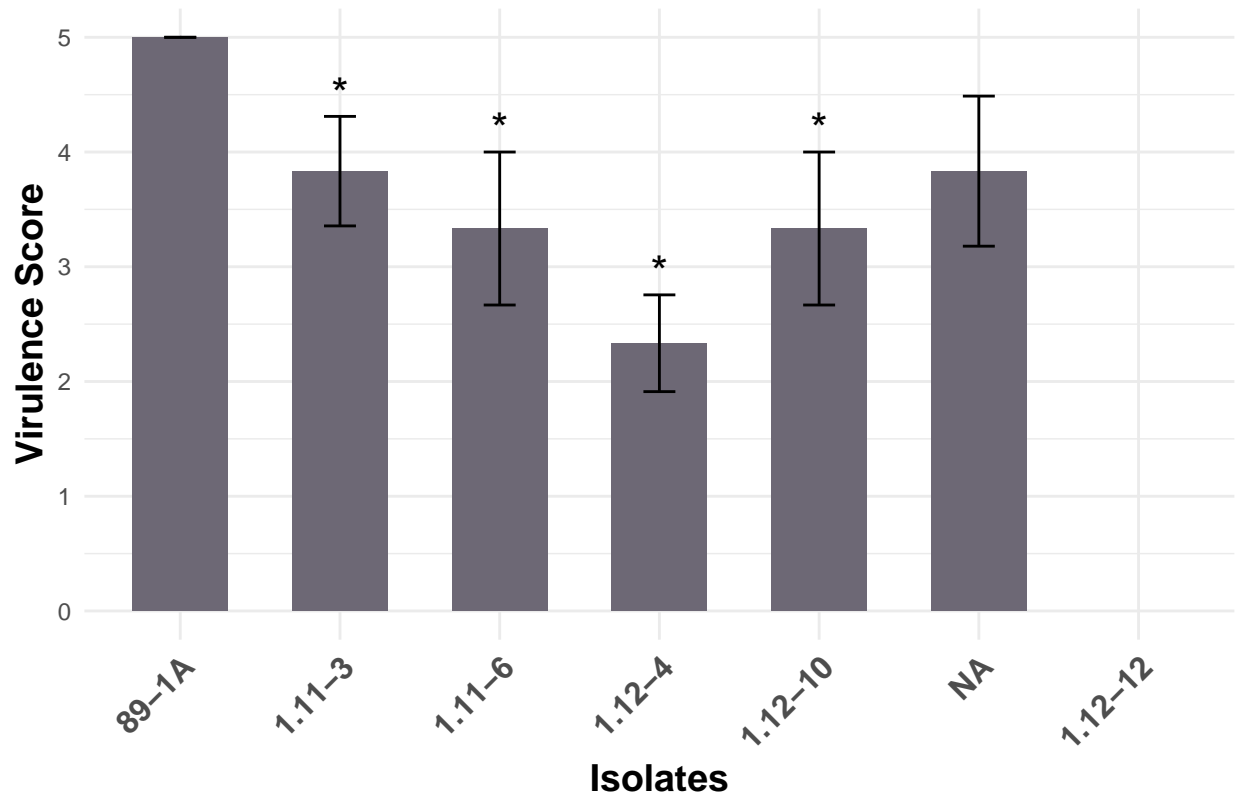
# Loop over varieties and perform t-tests
for (variety in comparison_varieties) {
  test_group <- df_long %>% filter(variety == variety)
  t_test_result <- t.test(control_group$score, test_group$score, var.equal = TRUE)
  sig_results <- rbind(sig_results, data.frame(variety = variety, p_value = t_test_result$p.value))
}

# Assign significance labels based on p-value
sig_results$label <- ifelse(sig_results$p_value < 0.001, "***",
                           ifelse(sig_results$p_value < 0.01, "**",
                                   ifelse(sig_results$p_value < 0.05, "*", "ns")))

# Merge with summary data to position labels above bars
sig_results <- sig_results %>%
  left_join(df_summary, by = "variety") %>%
  mutate(y_position = mean_score + se + 0.2)

# Create bar plot with error bars and significance annotations
ggplot(df_summary, aes(x = variety, y = mean_score)) +
  geom_bar(stat = "identity", fill = "#6D6875", width = 0.6) +
  geom_errorbar(aes(ymin = mean_score - se, ymax = mean_score + se),
               width = 0.2, color = "black") +
  labs(x = "Isolates", y = "Virulence Score") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, face = "bold", size = 12),
    axis.title = element_text(face = "bold", size = 14),
    plot.caption = element_text(size = 12, face = "italic", hjust = 0.5)
  ) +
  geom_text(data = sig_results, aes(x = variety, y = y_position, label = label),
           size = 6, color = "black") +
  labs(caption = "* p < 0.05, ** p < 0.01, *** p < 0.001, ns = not significant")

```



* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, ns = not significant

##II. Using one of the GFP-tagged isolates, we performed flow cytometry screening and successfully isolated 44 conidia lacking GFP fluorescence, indicating they may have spontaneously lost the target chromosome. Subsequent virulence assays revealed isolates with a marked reduction in virulence.

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(ggpubr)

# Read CSV using relative path from project root
df <- read.csv("3-17 chro-lost virulence test.csv")

# Rename the first column to 'variety' (if unnamed)
colnames(df)[1] <- "variety"

# Convert wide format to long format
df_long <- df %>%
  pivot_longer(cols = -variety, names_to = "replicate", values_to = "score")

# Set factor levels for ordering
df_long$variety <- factor(df_long$variety, levels = unique(df_long$variety))

# Calculate mean and standard error
df_summary <- df_long %>%
  group_by(variety) %>%
  summarise(
    mean_score = mean(score, na.rm = TRUE),
```

```

    se = sd(score, na.rm = TRUE) / sqrt(n())
  )

# Prepare control and test varieties for t-tests
control_group <- df_long %>% filter(variety == levels(df_long$variety)[1]) # Use first variety as control
comparison_varieties <- levels(df_long$variety)[-1]

# Run t-tests and collect significance
sig_results <- data.frame(variety = character(), p_value = numeric())

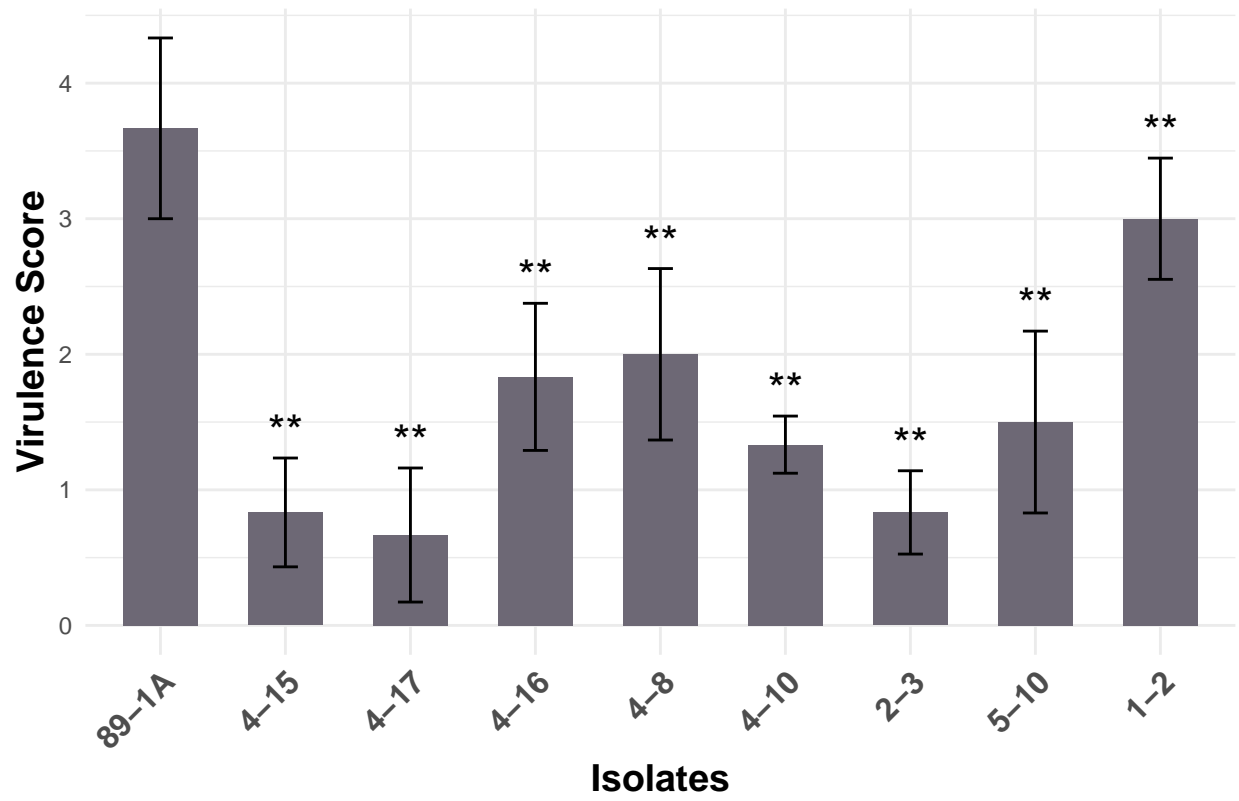
for (variety in comparison_varieties) {
  test_group <- df_long %>% filter(variety == variety)
  t_test_result <- t.test(control_group$score, test_group$score, var.equal = TRUE)
  sig_results <- rbind(sig_results, data.frame(variety = variety, p_value = t_test_result$p.value))
}

# Add significance label
sig_results$label <- ifelse(sig_results$p_value < 0.001, "***",
                           ifelse(sig_results$p_value < 0.01, "**",
                                   ifelse(sig_results$p_value < 0.05, "*", "ns")))

# Merge position info for annotations
sig_results <- sig_results %>%
  left_join(df_summary, by = "variety") %>%
  mutate(y_position = mean_score + se + 0.2)

# Draw bar plot with error bars and significance labels
ggplot(df_summary, aes(x = variety, y = mean_score)) +
  geom_bar(stat = "identity", fill = "#6D6875", width = 0.6) +
  geom_errorbar(aes(ymin = mean_score - se, ymax = mean_score + se),
               width = 0.2, color = "black") +
  labs(x = "Isolates", y = "Virulence Score") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, face = "bold", size = 12),
    axis.title = element_text(face = "bold", size = 14),
    plot.caption = element_text(size = 12, face = "italic", hjust = 0.5)
  ) +
  geom_text(data = sig_results, aes(x = variety, y = y_position, label = label),
           size = 6, color = "black") +
  labs(caption = "* p < 0.05, ** p < 0.01, *** p < 0.001, ns = not significant")

```



* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, ns = not significant