

Biehler et al

Tricia

2024-08-05

load libraries

```
library(ggplot2)
library(dplyr)
```

Import CSV

```
data <- read.csv("../datafiles/BiehlerDataBKA.csv", header = TRUE, sep = ",")

# Convert 'Sex' and 'Species' to factors
data$Sex <- factor(data$Sex)
data$Species <- factor(data$Species)
```

Mean and Standard Devation for BKA

```
# Calculate mean and SD for each species
summary_stats <- data %>%
  group_by(Species) %>%
  summarise(
    Mean = mean(Bacterial_Killing_Percentage, na.rm = TRUE),
    SD = sd(Bacterial_Killing_Percentage, na.rm = TRUE)
  )

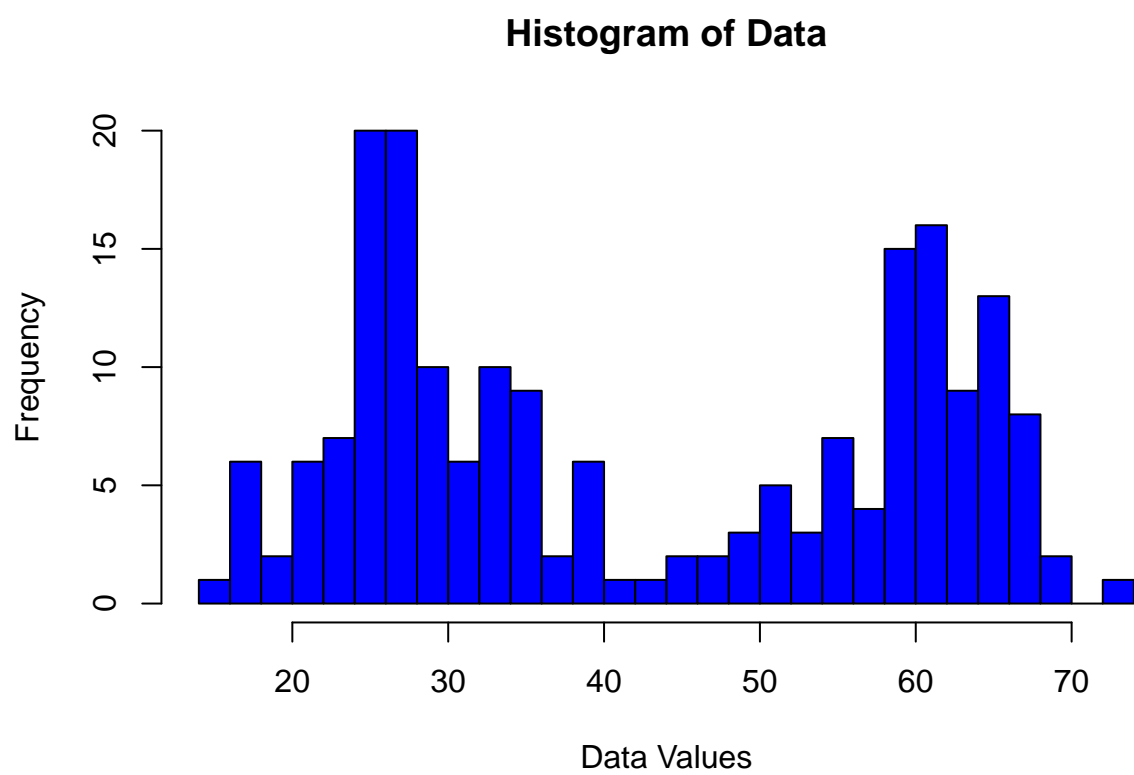
# Print the summary statistics
print(summary_stats)
```

```
## # A tibble: 4 x 3
##   Species Mean    SD
##   <fct>   <dbl> <dbl>
## 1 AMGO    26.3  4.76
## 2 HOFI    60.1  5.56
## 3 LEGO    27.6  6.12
## 4 PISI    33.4  7.38
```

normality for BKA

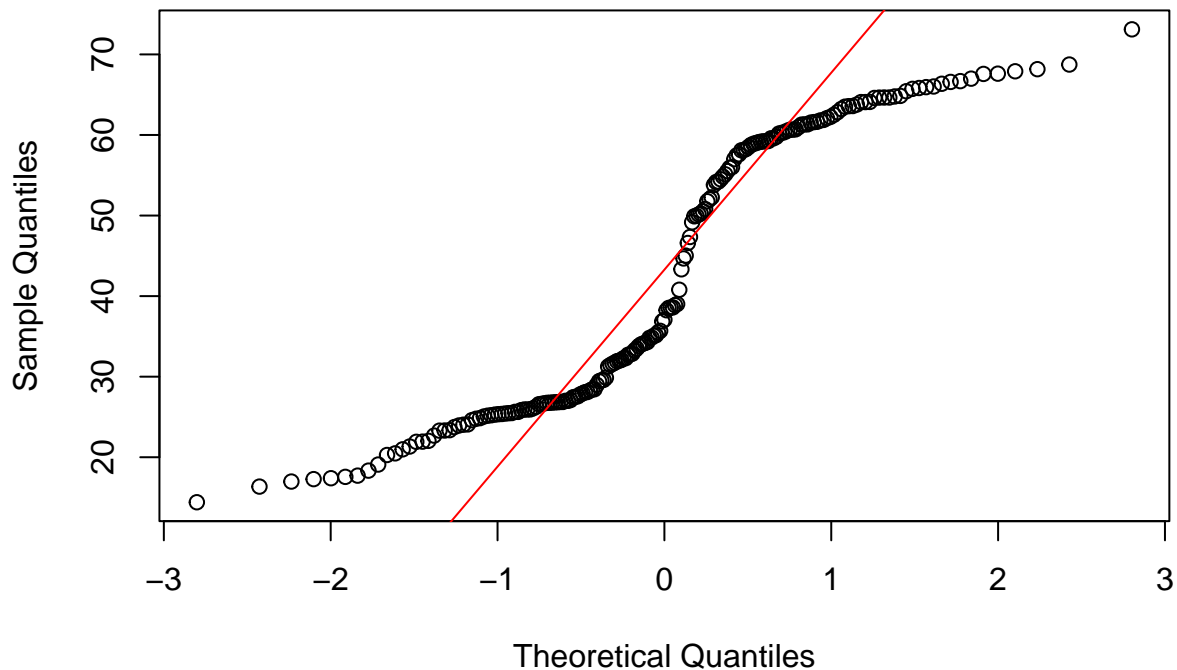
```
# histogram
hist(data$Bacterial_Killing_Percentage, main="Histogram of Data", xlab="Data Values",
```

```
breaks=30, col="blue")
```



```
# qqplot  
qqnorm(data$Bacterial_Killing_Percentage)  
qqline(data$Bacterial_Killing_Percentage, col = "red")
```

Normal Q-Q Plot



```
# shapiro-wilk  
shapiro.test(data$Bacterial_Killing_Percentage)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data$Bacterial_Killing_Percentage  
## W = 0.88674, p-value = 4.913e-11
```

Kruksal-Wallis of BKA

```
kruskal.test(data$Bacterial_Killing_Percentage, data$Species)
```

```
##  
## Kruskal-Wallis rank sum test  
##  
## data: data$Bacterial_Killing_Percentage and data$Species  
## Kruskal-Wallis chi-squared = 149.93, df = 3, p-value < 2.2e-16
```

Pairwise comparison with Bonferroni correction

```
bka_pairwise_results <- pairwise.wilcox.test(data$Bacterial_Killing_Percentage,  
      data$Species, p.adjust.method = "bonferroni")  
print(bka_pairwise_results)
```

```
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: data$Bacterial_Killing_Percentage and data$Species
##
##      AMGO      HOFI      LEGO
## HOFI 2.3e-15 -        -
## LEGO 1.00000 < 2e-16 -
## PISI 0.00039 2.2e-13 0.00462
##
## P value adjustment method: bonferroni
```

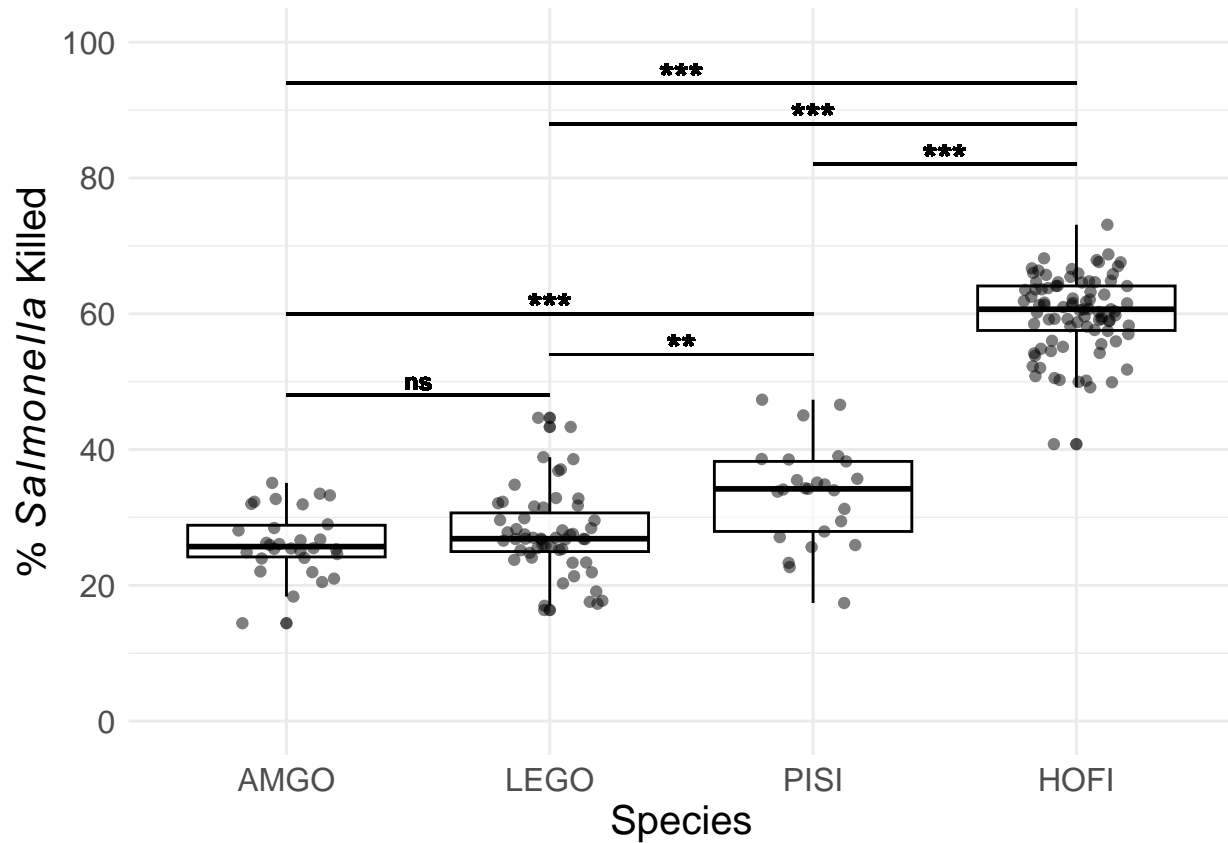
Boxplot of BKA

```
# Reorder the species factor levels
data$Species <- factor(data$Species, levels = c("AMGO", "LEGO", "PISI", "HOFI"))

# Create box plot with individual points
BKAAssay <- ggplot(data, aes(x=Species, y=Bacterial_Killing_Percentage)) +
  geom_boxplot(color="black", alpha=0.7) + # Box plot
  geom_jitter(width=0.2, size=1.5, alpha=0.5) + # Jittered points
  labs(x="Species",
       y=expression("% " * italic(Salmonella) * " Killed")) +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
        axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15),
        axis.text.x = element_text(size = 12), # Bold x-axis labels
        axis.text.y = element_text(size = 12) # Bold y-axis numbers
  ) +
  # Add line for significance between AMGO and HOFI
  geom_segment(aes(x=1, xend=4, y=94, yend=94), color="black", linewidth=.5) +
  # Add text for significance label "****"
  geom_text(aes(x=2.5, y=95, label="****"), color="black", size=5, fontface="bold") +
  # Add line for significance between LEGO and HOFI
  geom_segment(aes(x=2, xend=4, y=88, yend=88), color="black", linewidth=.5) +
  # Add text for significance label "****"
  geom_text(aes(x=3, y=89, label="****"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and HOFI
  geom_segment(aes(x=3, xend=4, y=82, yend=82), color="black", linewidth=.5) +
  # Add text for significance label "****"
  geom_text(aes(x=3.5, y=83, label="****"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and AMGO
  geom_segment(aes(x=1, xend=3, y=60, yend=60), color="black", linewidth=.5) +
  # Add text for significance label "****"
  geom_text(aes(x=2, y=61, label="****"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and LEGO
  geom_segment(aes(x=2, xend=3, y=54, yend=54), color="black", linewidth=.5) +
  # Add text for significance label "****"
  geom_text(aes(x=2.5, y=55, label="****"), color="black", size=5, fontface="bold") +
  # Add line for significance between AMGO and LEGO
  geom_segment(aes(x=1, xend=2, y=48, yend=48), color="black", linewidth=.5) +
```

```
# Add text for significance label "***"
geom_text(aes(x=1.5, y=50, label="ns"), color="black", size=3, fontface="bold")

print(BKAAssay)
```



```
# export tiff with 300dpi
ggsave(
  filename="../figures/BKAAssay.tiff",
  plot = BKAAssay,
  width = 200,
  height = 200,
  units = c("mm"),
  dpi = 300,
  bg = "white"
)
```

Test effect of sex

```
clean_data <- data[!is.na(data$Species) & !is.na(data$Sex), ]
wilcox.test(data$Bacterial_Killing_Percentage ~ data$Sex)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: data$Bacterial_Killing_Percentage by data$Sex
```

```
## W = 5033.5, p-value = 0.1569
## alternative hypothesis: true location shift is not equal to 0
```

Kruskal-Wallis test for interaction

```
kw_test <- kruskal.test(Bacterial_Killing_Percentage ~ interaction(Species, Sex), data = data)
print(kw_test)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: Bacterial_Killing_Percentage by interaction(Species, Sex)
## Kruskal-Wallis chi-squared = 149.79, df = 7, p-value < 2.2e-16
```

Pairwise Results

```
pairwise_results <- pairwise.wilcox.test(data$Bacterial_Killing_Percentage,
                                         interaction(data$Species, data$Sex),
                                         p.adjust.method = "bonferroni")
print(pairwise_results)
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: data$Bacterial_Killing_Percentage and interaction(data$Species, data$Sex)
##
##          AMGO.Female LEGO.Female PISI.Female HOFI.Female AMGO.Male LEGO.Male
## LEGO.Female 1.000          -          -          -          -          -
## PISI.Female 0.196          1.000          -          -          -          -
## HOFI.Female 2.5e-09        1.6e-10        6.1e-10        -          -          -
## AMGO.Male   1.000          1.000          0.119          1.3e-13        -          -
## LEGO.Male   1.000          1.000          0.074          < 2e-16        1.000          -
## PISI.Male   0.157          1.000          1.000          6.1e-10        0.076          0.190
## HOFI.Male   7.3e-06        1.1e-06        3.6e-06        1.000          5.2e-09        5.8e-15
##
##          PISI.Male
## LEGO.Female -
## PISI.Female -
## HOFI.Female -
## AMGO.Male   -
## LEGO.Male   -
## PISI.Male   -
## HOFI.Male   2.7e-06
##
## P value adjustment method: bonferroni
```

Pairwise Results with Test Statistic

```
# Load necessary library
library(dplyr)

# Define the pairwise combinations of Species
```

```

pairs <- combn(levels(data$Species), 2, simplify = FALSE)

# Initialize lists to store results
test_stats <- list()
p_values <- list()

# Loop through each pair and perform the Wilcoxon rank-sum test
for (pair in pairs) {
  species1 <- pair[1]
  species2 <- pair[2]
  subset_data <- data %>% filter(Species %in% c(species1, species2))

  test_result <- wilcox.test(Bacterial_Killing_Percentage ~ Species, data = subset_data)

  test_stats[[paste(species1, species2, sep = " vs ")] <- test_result$statistic
  p_values[[paste(species1, species2, sep = " vs ")] <- test_result$p.value
}

# Convert lists to data frames
test_stats_df <- as.data.frame(do.call(rbind, test_stats), stringsAsFactors = FALSE)
p_values_df <- as.data.frame(do.call(rbind, p_values), stringsAsFactors = FALSE)

# Apply Bonferroni correction
adjusted_p_values <- p.adjust(unlist(p_values_df), method = "bonferroni")

# Combine results into a single data frame
results_df <- data.frame(
  Comparison = rownames(test_stats_df),
  Test_Statistic = unlist(test_stats_df),
  P_Value = unlist(p_values_df),
  Adjusted_P_Value = adjusted_p_values
)

# Print the results
print(results_df)

```

##	Comparison	Test_Statistic	P_Value	Adjusted_P_Value
##	W1 AMGO vs LEGO	712	3.008786e-01	1.000000e+00
##	W2 AMGO vs PISI	147	6.444665e-05	3.866799e-04
##	W3 AMGO vs HOFI	0	3.862104e-16	2.317262e-15
##	W4 LEGO vs PISI	363	7.706813e-04	4.624088e-03
##	W5 LEGO vs HOFI	2	1.394992e-23	8.369951e-23
##	W6 PISI vs HOFI	3	3.606098e-14	2.163659e-13

Import Growth Curve Data

```

growthcurvedata <- read.csv("../datafiles/GrowthCurve.csv", header = TRUE, sep = ",")

```

Growth Curve

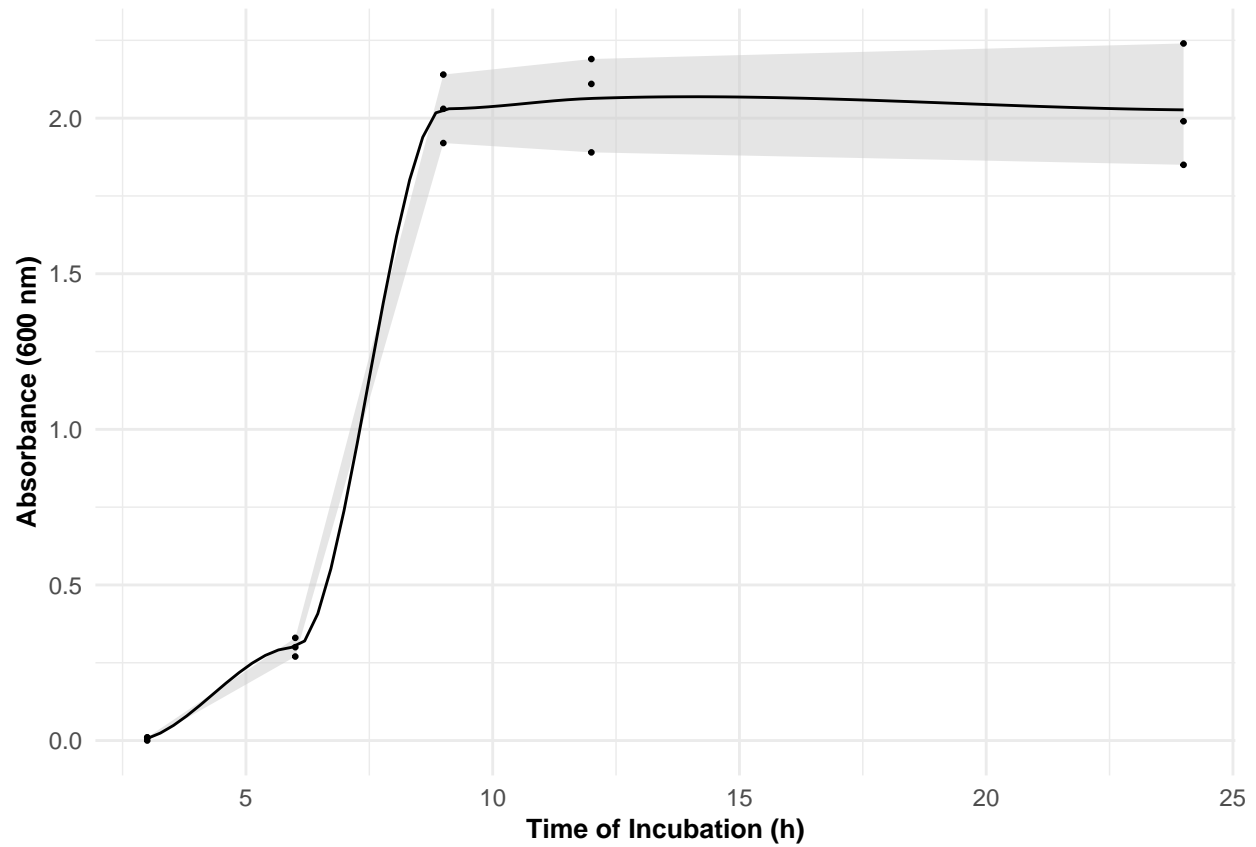
```

# Calculate mean absorbance for each time point
average_data <- growthcurvedata %>%
  group_by(Time.h.) %>%
  summarise(
    Mean = mean(A600),
    SD = sd(A600),
    Min = min(A600),
    Max = max(A600)
  )

# Plot with smoothed line and shaded area representing the standard deviation
growthcurve <- ggplot() +
  geom_ribbon(data = average_data, aes(x = Time.h., y = Mean, ymin = Min, ymax = Max),
    fill = "grey80", alpha = 0.5) +
  geom_smooth(data = average_data, aes(x = Time.h., y = Mean), method = "loess",
    formula = y ~ x, color = "black", se = FALSE,
    size = 0.5) +
  geom_point(data = growthcurvedata, aes(x = Time.h., y = A600), color = "black", size = 0.5) +
  labs(x = "Time of Incubation (h)",
    y = "Absorbance (600 nm)") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 10),
    axis.title.x = element_text(face = "bold", size = 10),
    axis.title.y = element_text(face = "bold", size = 10)
  )

print(growthcurve)

```

export tiff with 300dpi

```
ggsave(  
  filename="../figures/growthcurve.tiff",  
  plot = growthcurve,  
  width = 200,  
  height = 100,  
  units = c("mm"),  
  dpi = 300,  
  bg = "white"  
)
```