T6SSs in E. coli

Kristina Nesporova

2024-10-24

## R Markdown T6-20241024

## Import libraries

library(ggplot2)  
library(tidyr)  
library(dplyr)

library(corrplot)

library(viridis)

library(grid)  
library(ggplotify)  
library(RColorBrewer)  
library(polycor)

## Merge results with metadata

EB\_PG\_v.8\_20241010 <- read.csv("~/T6-20241024/Intermediate tables/EB\_PG\_v.8\_20241010.csv", sep=";", stringsAsFactors=TRUE)  
   
T6SS\_complete\_results\_v.3\_20240719 <- read.csv("~/T6-20241024/Intermediate tables/T6SS\_complete\_results\_v.3\_20240719.csv", sep=";", stringsAsFactors=TRUE)  
   
EB\_T6\_20241024 <- merge(EB\_PG\_v.8\_20241010, T6SS\_complete\_results\_v.3\_20240719, by = "Barcode", all = TRUE)   
  
EB\_T6\_20241024 <- subset(EB\_T6\_20241024, !is.na(Coverage)) #Filter out the genomes without metadata (the metadata sheet has been already filtered for low coverage and other issues)

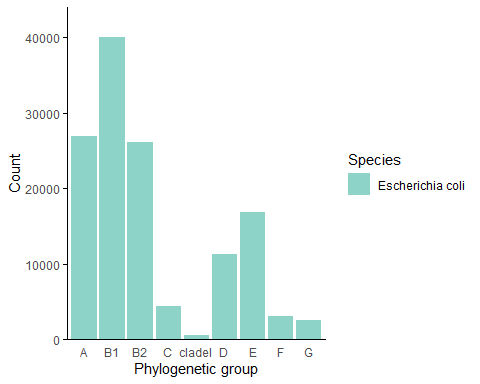
## Creating Figures

The figures are generatad in automatic size in Rmarkdown - the wanted size and resolution can be reached by using ggsave and prefered format (.png, .pfd,…)

#ggsave(“plotxx.png”, plot = xx, dpi = 1000) #example - object “xx” needs to be created first e.g. xx <- ggplot(EB\_T6\_20241024, aes(x = PG, fill = Species)) + geom\_bar()

### Figure 1A

ggplot(EB\_T6\_20241024, aes(x = PG, fill = Species)) +   
 geom\_bar() +   
 scale\_fill\_brewer(palette = "Set3") +   
 theme\_minimal() +   
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black") # Add y-axis ticks  
 ) +   
 labs(x = "Phylogenetic group", y = "Count") +   
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) # Add space at the top

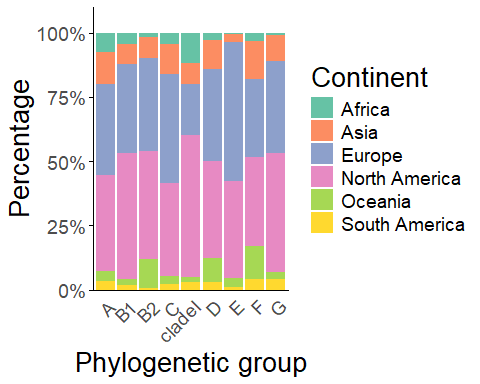


### Figure 1B-1\_Continent

proportions\_EB\_PG\_Continent <- EB\_T6\_20241024 %>%  
 group\_by(PG, Continent) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_EB\_PG\_Continent, aes(x = PG, y = Percentage, fill = Continent)) +   
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "Continent") +  
 scale\_fill\_brewer(palette = "Set2") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Ensure space at the top  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(), # Remove grey background  
 panel.grid = element\_blank(), # Remove gridlines  
 axis.line = element\_line(color = "black"), # Highlight x and y axes  
 axis.text.x = element\_text(size = 14, angle = 45, hjust = 1),   
 axis.text.y = element\_text(size = 14),   
 axis.title.x = element\_text(size = 20),   
 axis.title.y = element\_text(size = 20),  
 axis.ticks.y = element\_line(color = "black"), # Add y-axis ticks  
 legend.title = element\_text(size = 20),   
 legend.text = element\_text(size = 14)   
 )

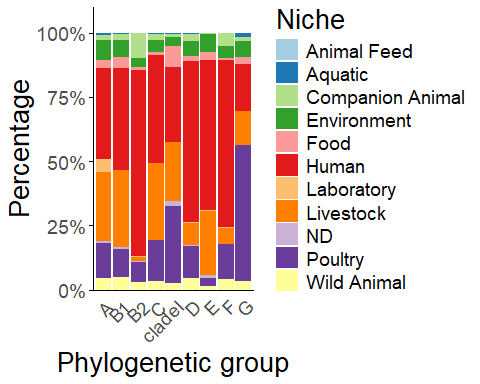


### Figure 1B-2\_Niche

knitr::opts\_chunk$set(echo = TRUE)  
proportions\_EB\_PG\_Niche <- EB\_T6\_20241024 %>%  
 group\_by(PG, Source.Niche) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_EB\_PG\_Niche, aes(x = PG, y = Percentage, fill = Source.Niche)) +   
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "Niche") +  
 scale\_fill\_brewer(palette = "Paired") + # Use Paired color palette from RColorBrewer  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 14, angle = 45, hjust = 1),   
 axis.text.y = element\_text(size = 14),   
 axis.title.x = element\_text(size = 20),   
 axis.title.y = element\_text(size = 20),  
 axis.ticks.y = element\_line(color = "black"),  
 legend.title = element\_text(size = 20),   
 legend.text = element\_text(size = 14)   
 )

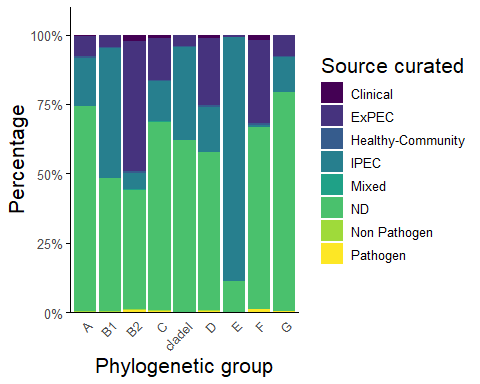


### Figure 1C Source curated

proportions\_EB\_PG\_Clinical <- EB\_T6\_20241024 %>%  
 group\_by(PG, Source.curated) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_EB\_PG\_Clinical, aes(x = PG, y = Percentage, fill = Source.curated)) +   
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "Source curated") +  
 scale\_fill\_viridis\_d() + # Use the viridis color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 10, angle = 45, hjust = 1),   
 axis.text.y = element\_text(size = 10),   
 axis.title.x = element\_text(size = 16),   
 axis.title.y = element\_text(size = 16),  
 axis.ticks.y = element\_line(color = "black"),  
 legend.title = element\_text(size = 16),   
 legend.text = element\_text(size = 10)   
 )



### Figure 2A Normalized histograms for number of tss genes in for each subclass

Example is for T6SSi1 - can be adapted for the other two subclass

hist\_plot <- function() {  
 hist(EB\_T6\_20241024[,"NT6SSi1"],   
 breaks = seq(-0.5, 13.5, 1), # Bins corresponding to integer values  
 main = "T6SSi1",   
 xlab = "Number of core tss genes",   
 ylab = "Frequency",   
 col = "#8DD3C7", # Set3 color from ggplot  
 freq = FALSE, # Use probability densities instead of frequencies  
 xaxt = 'n', # Suppress the default x-axis  
 ylim = c(0, 1))  
   
 # Customize the x-axis and y-axis  
 axis(1, at = 0:13, labels = 0:13, col.axis = "black", col = "black") # X-axis only  
 axis(2, col.axis = "black", col = "black") # Y-axis only  
   
 # Add box outline for axis lines  
 box(col = "black")  
}

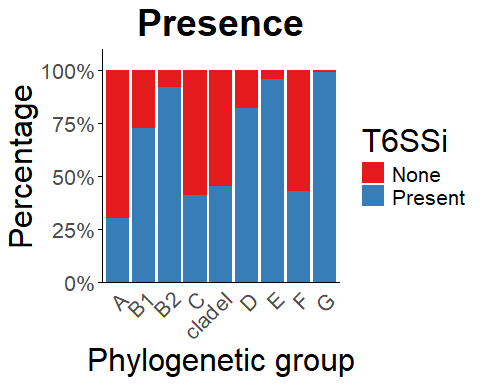
Fig2a\_1 <- as.ggplot(hist\_plot)

### Figure 2B-1 Presence of minimally 1 T6SSi in respective phylogenetic groups

proportions\_Pmin1 <- EB\_T6\_20241024 %>%  
 group\_by(PG, Pmin1) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_Pmin1, aes(x = PG, y = Percentage, fill = Pmin1)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(  
 x = "Phylogenetic group",  
 title = "Presence",  
 y = "Percentage",  
 fill = "T6SSi"  
 ) +  
 scale\_fill\_brewer(palette = "Set1") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(), # Remove grey background  
 panel.grid = element\_blank(), # Remove gridlines  
 axis.line = element\_line(color = "black"), # Highlight x and y axes  
 axis.text.x = element\_text(size = 16, angle = 45, hjust = 1),  
 axis.text.y = element\_text(size = 16),  
 axis.title.x = element\_text(size = 24),  
 axis.title.y = element\_text(size = 24),  
 legend.title = element\_text(size = 24),  
 legend.text = element\_text(size = 16),  
 axis.ticks.y = element\_line(color = "black"),  
 plot.title = element\_text(size = 28, face = "bold", hjust = 0.5) # Customize the title  
 )

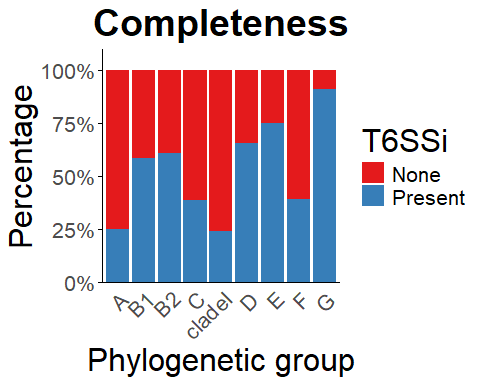


### Figure 2B-2 Presence of minimally 1 “Complete” T6SSi in respective phylogenetic groups

proportions\_Cmin1 <- EB\_T6\_20241024 %>%  
 group\_by(PG, Cmin1) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_Cmin1, aes(x = PG, y = Percentage, fill = Cmin1)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 title = "Completeness",  
 y = "Percentage",  
 fill = "T6SSi") +   
 scale\_fill\_brewer(palette = "Set1") +   
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(), # Remove grey background  
 panel.grid = element\_blank(), # Remove gridlines  
 axis.line = element\_line(color = "black"), # Highlight x and y axes  
 axis.text.x = element\_text(size = 16, angle = 45, hjust = 1),   
 axis.text.y = element\_text(size = 16),   
 axis.title.x = element\_text(size = 24),   
 axis.title.y = element\_text(size = 24),   
 legend.title = element\_text(size = 24),  
 legend.text = element\_text(size = 16),  
 axis.ticks.y = element\_line(color = "black"),  
 plot.title = element\_text(size = 28, face = "bold", hjust = 0.5) # Customize the title  
 )

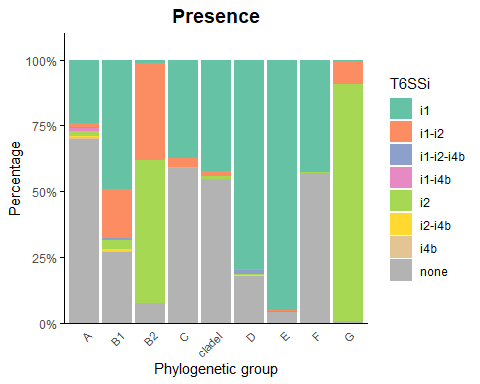


### Figure 2C-1 Presence of multiple T6SSi in respective phylogenetic groups

proportions\_Pmultiple <- EB\_T6\_20241024 %>%  
 group\_by(PG, Pmultiple) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

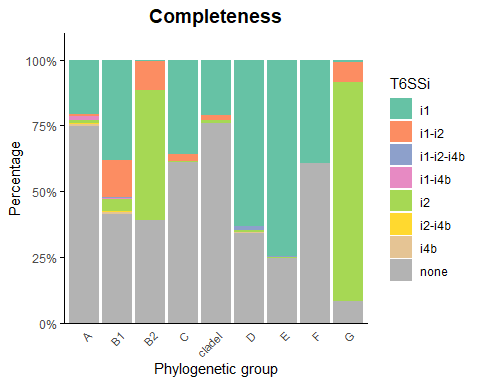
ggplot(proportions\_Pmultiple, aes(x = PG, y = Percentage, fill = Pmultiple)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(  
 x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "T6SSi",  
 title = "Presence" # Add the title here  
 ) +  
 scale\_fill\_brewer(palette = "Set2") +   
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 axis.ticks.y = element\_line(color = "black"),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )



### Figure 2C-2 Presence of multiple complete T6SSi in respective phylogenetic groups

proportions\_Cmultiple <- EB\_T6\_20241024 %>%  
 group\_by(PG, Cmultiple) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

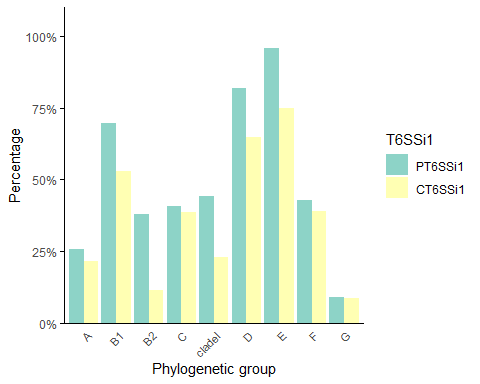
## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.



### Figure 2D\_1 T6SSi1

summary\_data\_i1 <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PT6SSi1 = mean(PT6SSi1 == 1) \* 100,  
 CT6SSi1 = mean(CT6SSi1 == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PT6SSi1", "CT6SSi1"))) # Reorder levels

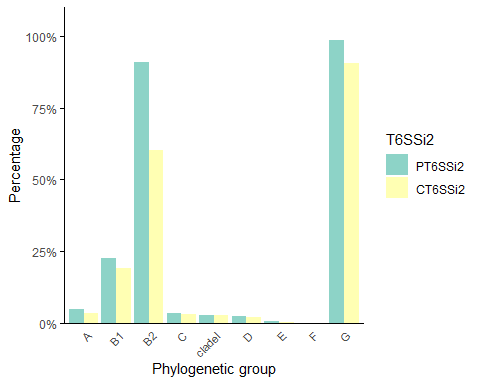
ggplot(summary\_data\_i1, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi1") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure 2D\_2 T6SSi2

summary\_data\_i2 <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PT6SSi2 = mean(PT6SSi2 == 1) \* 100,  
 CT6SSi2 = mean(CT6SSi2 == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PT6SSi2", "CT6SSi2"))) # Reorder levels

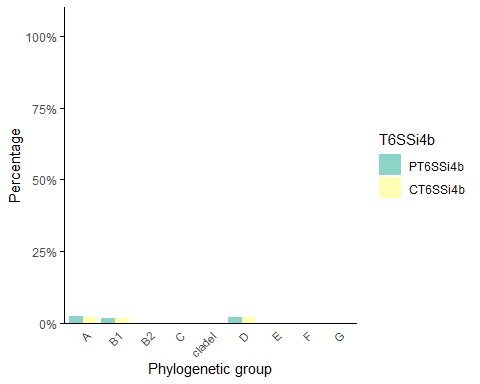
ggplot(summary\_data\_i2, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi2") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure 2D\_3 T6SSi4b

summary\_data\_i4b <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PT6SSi4b = mean(PT6SSi4b == 1) \* 100,  
 CT6SSi4b = mean(CT6SSi4b == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PT6SSi4b", "CT6SSi4b"))) # Reorder levels

ggplot(summary\_data\_i4b, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi4b") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure 3A\_2 T6SS Completeness in Clinical groups

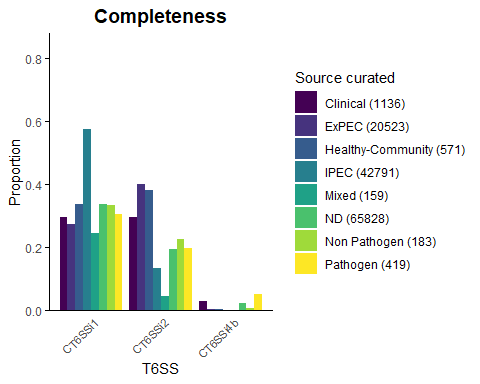
Clinical\_CT6 <- EB\_T6\_20241024 %>%  
 select(Source.curated, CT6SSi1, CT6SSi2, CT6SSi4b)  
  
counts\_Source\_curated <- Clinical\_CT6 %>% #Calculate counts for each category in Source.curated  
 group\_by(Source.curated) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
print(counts\_Source\_curated) #Display the counts

## # A tibble: 8 × 2  
## Source.curated Count  
## <fct> <int>  
## 1 Clinical 1136  
## 2 ExPEC 20523  
## 3 Healthy-Community 571  
## 4 IPEC 42791  
## 5 Mixed 159  
## 6 ND 65828  
## 7 Non Pathogen 183  
## 8 Pathogen 419

data\_long\_Clinical\_T6C <- Clinical\_CT6 %>% #Reshape the data to long format  
 pivot\_longer(cols = -Source.curated, names\_to = "T6SS", values\_to = "Value")

proportions\_Clinical\_T6C <- data\_long\_Clinical\_T6C %>% #Calculate proportions  
 group\_by(Source.curated, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels <- c( #Create a named vector for custom labels  
 "Clinical" = "Clinical (1136)",  
 "ExPEC" = "ExPEC (20523)",  
 "Healthy-Community" = "Healthy-Community (571)",  
 "IPEC" = "IPEC (42791)",  
 "Mixed" = "Mixed (159)",  
 "ND" = "ND (65828)",  
 "Non Pathogen" = "Non Pathogen (183)",  
 "Pathogen" = "Pathogen (419)"  
)

ggplot(proportions\_Clinical\_T6C, aes(x = T6SS, y = Proportion, fill = Source.curated)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "Source curated", title = "Completeness", ) + # Add a label for fill legend  
 scale\_fill\_manual(values = viridis::viridis(length(custom\_labels)),   
 labels = custom\_labels) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )



### Figure 3A\_1 T6SS in Clinical groups

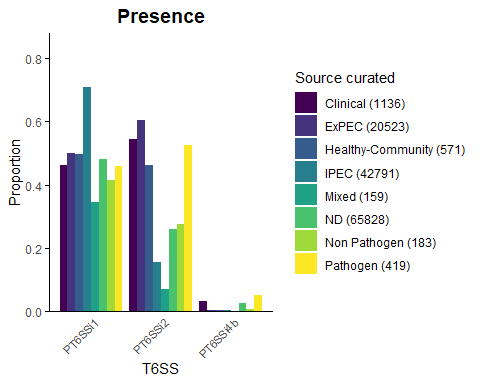
Clinical\_PT6 <- EB\_T6\_20241024 %>%  
 select(Source.curated, PT6SSi1, PT6SSi2, PT6SSi4b)  
  
counts\_Source\_curated <- Clinical\_PT6 %>% #Calculate counts for each category in Source.curated  
 group\_by(Source.curated) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
print(counts\_Source\_curated) #Display the counts

## # A tibble: 8 × 2  
## Source.curated Count  
## <fct> <int>  
## 1 Clinical 1136  
## 2 ExPEC 20523  
## 3 Healthy-Community 571  
## 4 IPEC 42791  
## 5 Mixed 159  
## 6 ND 65828  
## 7 Non Pathogen 183  
## 8 Pathogen 419

data\_long\_Clinical\_T6P <- Clinical\_PT6 %>% #Reshape the data to long format  
 pivot\_longer(cols = -Source.curated, names\_to = "T6SS", values\_to = "Value")

proportions\_Clinical\_T6P <- data\_long\_Clinical\_T6P %>% #Calculate proportions  
 group\_by(Source.curated, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels <- c( #Create a named vector for custom labels  
 "Clinical" = "Clinical (1136)",  
 "ExPEC" = "ExPEC (20523)",  
 "Healthy-Community" = "Healthy-Community (571)",  
 "IPEC" = "IPEC (42791)",  
 "Mixed" = "Mixed (159)",  
 "ND" = "ND (65828)",  
 "Non Pathogen" = "Non Pathogen (183)",  
 "Pathogen" = "Pathogen (419)"  
)

ggplot(proportions\_Clinical\_T6P, aes(x = T6SS, y = Proportion, fill = Source.curated)) +   
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "Source curated", title = "Presence",) + # Add a label for fill legend  
 scale\_fill\_manual(values = viridis::viridis(length(custom\_labels)),   
 labels = custom\_labels) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )



### Figure 3B\_1 T6SS in ExPEC type

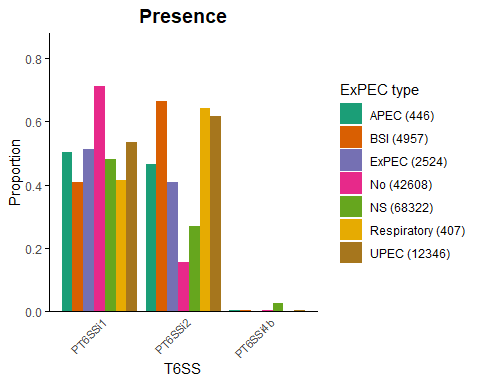
ExPECtype\_PT6 <- EB\_T6\_20241024 %>%  
 select(ExPEC.type, PT6SSi1, PT6SSi2, PT6SSi4b)  
  
counts\_ExPECtype <- ExPECtype\_PT6 %>% #Calculate counts for each category in Source.curated  
 group\_by(ExPEC.type) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
print(counts\_ExPECtype) #Display the counts

## # A tibble: 7 × 2  
## ExPEC.type Count  
## <fct> <int>  
## 1 APEC 446  
## 2 BSI 4957  
## 3 ExPEC 2524  
## 4 No 42608  
## 5 NS 68322  
## 6 Respiratory 407  
## 7 UPEC 12346

data\_long\_ExPECtype\_T6P <- ExPECtype\_PT6 %>% #Reshape the data to long format  
 pivot\_longer(cols = -ExPEC.type, names\_to = "T6SS", values\_to = "Value")

proportions\_ExPECtype\_T6P <- data\_long\_ExPECtype\_T6P %>% #Calculate proportions  
 group\_by(ExPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels\_ExPEC <- c( #Create a named vector for custom labels  
 "APEC" = "APEC (446)",  
 "BSI" = "BSI (4957)",  
 "ExPEC" = "ExPEC (2524)",  
 "No" = "No (42608)",  
 "NS" = "NS (68322)",  
 "ND" = "ND (65828)",  
 "Respiratory" = "Respiratory (407)",  
 "UPEC" = "UPEC (12346)"  
)

ggplot(proportions\_ExPECtype\_T6P, aes(x = T6SS, y = Proportion, fill = ExPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "ExPEC type", title = "Presence") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_ExPEC), "Dark2"),  
 labels = custom\_labels\_ExPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )

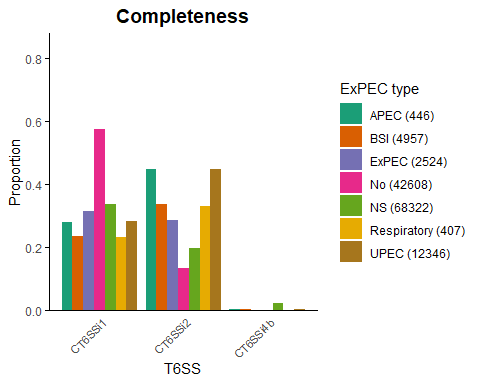


### Figure 3B\_2 T6SS (Complete) in ExPEC type

ExPECtype\_CT6 <- EB\_T6\_20241024 %>%  
 select(ExPEC.type, CT6SSi1, CT6SSi2, CT6SSi4b)  
  
data\_long\_ExPECtype\_T6C <- ExPECtype\_CT6 %>% #Reshape the data to long format  
 pivot\_longer(cols = -ExPEC.type, names\_to = "T6SS", values\_to = "Value")

proportions\_ExPECtype\_T6C <- data\_long\_ExPECtype\_T6C %>% #Calculate proportions  
 group\_by(ExPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels\_ExPEC <- c(  
 "APEC" = "APEC (446)",  
 "BSI" = "BSI (4957)",  
 "ExPEC" = "ExPEC (2524)",  
 "No" = "No (42608)",  
 "NS" = "NS (68322)",  
 "ND" = "ND (65828)",  
 "Respiratory" = "Respiratory (407)",  
 "UPEC" = "UPEC (12346)" #Create a named vector for custom labels  
)

ggplot(proportions\_ExPECtype\_T6C, aes(x = T6SS, y = Proportion, fill = ExPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "ExPEC type", title = "Completeness") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_ExPEC), "Dark2"),  
 labels = custom\_labels\_ExPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )



### Figure 3c\_1 T6SS proportion in IPEC.type

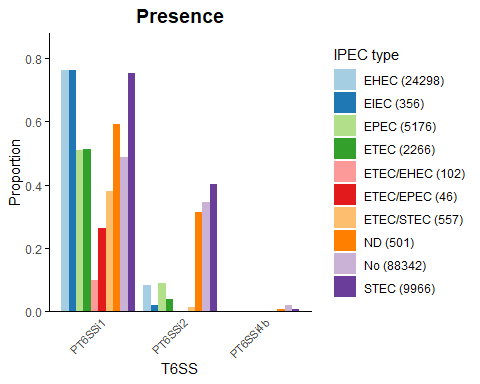
IPECtype\_PT6 <- EB\_T6\_20241024 %>%  
 select(IPEC.type, PT6SSi1, PT6SSi2, PT6SSi4b)  
  
counts\_IPECtype <- IPECtype\_PT6 %>% #Calculate counts for each category in Source.curated  
 group\_by(IPEC.type) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
print(counts\_IPECtype) #Display the counts

## # A tibble: 10 × 2  
## IPEC.type Count  
## <fct> <int>  
## 1 EHEC 24298  
## 2 EIEC 356  
## 3 EPEC 5176  
## 4 ETEC 2266  
## 5 ETEC.EHEC 102  
## 6 ETEC.EPEC 46  
## 7 ETEC.STEC 557  
## 8 ND 501  
## 9 No 88342  
## 10 STEC 9966

data\_long\_IPECtype\_T6P <- IPECtype\_PT6 %>%  
 pivot\_longer(cols = -IPEC.type, names\_to = "T6SS", values\_to = "Value") #Reshape the data to long format

proportions\_IPECtype\_T6P <- data\_long\_IPECtype\_T6P %>% #Calculate proportions  
 group\_by(IPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels\_IPEC <- c( #Create a named vector for custom labels  
 "EHEC" = "EHEC (24298)",  
 "EIEC" = "EIEC (356)",  
 "EPEC" = "EPEC (5176)",  
 "ETEC" = "ETEC (2266)",  
 "ETEC.EHEC" = "ETEC/EHEC (102)",  
 "ETEC.EPEC" = "ETEC/EPEC (46)",  
 "ETEC.STEC" = "ETEC/STEC (557)",  
 "ND" = "ND (501)",  
 "No" = "No (88342)",  
 "STEC" = "STEC (9966)"  
)

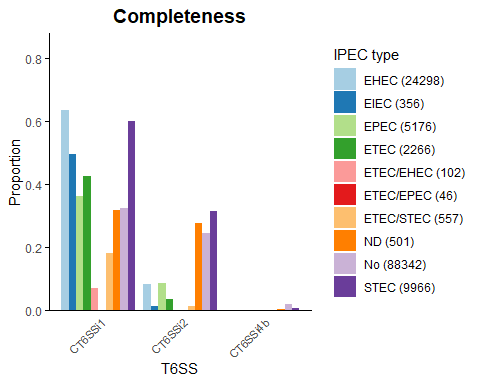
ggplot(proportions\_IPECtype\_T6P, aes(x = T6SS, y = Proportion, fill = IPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "IPEC type", title = "Presence") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_IPEC), "Paired"),  
 labels = custom\_labels\_IPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )



### Figure 3c\_2 T6SS proportion (Complete) in IPEC type

IPECtype\_CT6 <- EB\_T6\_20241024 %>%  
 select(IPEC.type, CT6SSi1, CT6SSi2, CT6SSi4b)  
  
data\_long\_IPECtype\_T6C <- IPECtype\_CT6 %>% #Reshape the data to long format  
 pivot\_longer(cols = -IPEC.type, names\_to = "T6SS", values\_to = "Value")

ggplot(proportions\_IPECtype\_T6C, aes(x = T6SS, y = Proportion, fill = IPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "IPEC type", title = "Completeness") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_IPEC), "Paired"),  
 labels = custom\_labels\_IPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5) # Customize the title  
 )

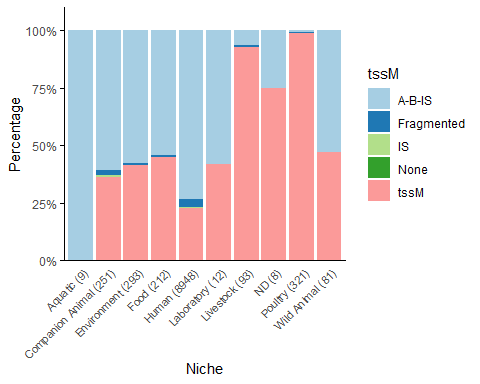


### Figure 5a ST131 only - tssM for Niche

ST131 <- subset(EB\_T6\_20241024, ST == "ST131")  
  
counts\_Niche\_ST131 <- ST131 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST131\_counted\_Niche <- ST131 %>%  
 left\_join(counts\_Niche\_ST131, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST131\_Niche\_tssM <- ST131\_counted\_Niche %>%  
 group\_by(Source.Niche, tssM\_frag) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

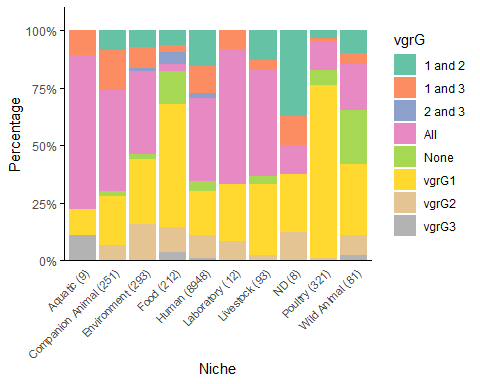
## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

ggplot(proportions\_ST131\_Niche\_tssM, aes(x = Source.Niche, y = Percentage, fill = tssM\_frag)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "tssM") +  
 scale\_fill\_brewer(palette = "Paired") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure 5b ST131 only - vgrG for Niche

ggplot(proportions\_ST131\_Niche\_vgrG, aes(x = Source.Niche, y = Percentage, fill = vgrG\_KO178)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "vgrG") +  
 scale\_fill\_brewer(palette = "Set2") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



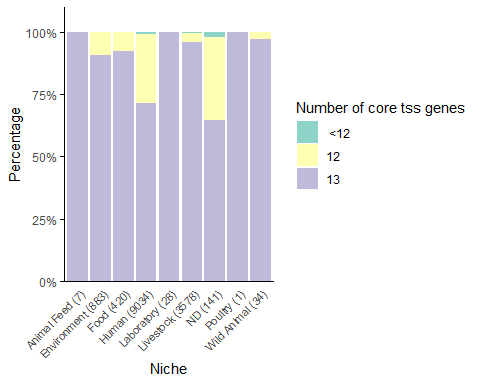
### Figure 5c ST11 completeness

ST11 <- subset(EB\_T6\_20241024, ST == "ST11")  
  
ST11 <- subset(EB\_T6\_20241024, ST == "ST11") %>%  
 mutate(ST11 = case\_when(  
 NT6SSi1 >= 0 & NT6SSi1 <= 11 ~ "Below 12",  
 NT6SSi1 == 12 ~ "Twelf",  
 NT6SSi1 == 13 ~ "Complete",  
 TRUE ~ NA\_character\_  
 ),  
 # Ensure the factor levels are ordered as desired  
 ST11 = factor(ST11, levels = c("Below 12", "Twelf", "Complete")))  
  
counts\_Niche\_ST11 <- ST11 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST11\_counted\_Niche <- ST11 %>%  
 left\_join(counts\_Niche\_ST11, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST11\_Niche <- ST11\_counted\_Niche %>%  
 group\_by(Source.Niche, ST11) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

custom\_labels <- c("Below 12" = " <12", "Twelf" = "12", "Complete" = "13")

ggplot(proportions\_ST11\_Niche, aes(x = Source.Niche, y = Percentage, fill = ST11)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "Number of core tss genes") +  
 scale\_fill\_manual(values = RColorBrewer::brewer.pal(3, "Set3"), labels = custom\_labels) +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure 6A - correlations

T6\_CP\_vfdb\_resfin\_20241017\_no\_tss <- read.csv("~/T6-20241024/Intermediate tables/Correlations/T6\_CP\_vfdb\_resfin\_20241017\_no\_tss.csv", sep=";", stringsAsFactors=TRUE)

polychoric\_coef\_T6\_CP\_notss <- hetcor(T6\_CP\_vfdb\_resfin\_20241017\_no\_tss, ML = TRUE)$correlations  
  
write.table(polychoric\_coef\_T6\_CP\_notss, file = "Supplementary\_table\_8.csv", append = FALSE, quote = TRUE, sep = ";")

cor\_matrix\_T6\_CP\_no\_tss <- as.matrix(polychoric\_coef\_T6\_CP\_notss)  
  
dist\_matrix\_T6\_CP\_no\_tss <- dist(cor\_matrix\_T6\_CP\_no\_tss)  
  
hclust\_res <- hclust(dist\_matrix\_T6\_CP\_no\_tss, method = "complete")  
  
groups <- cutree(hclust\_res, k = 2)   
  
ordered\_indices <- order(groups)  
cor\_matrix\_T6\_CP\_no\_tss\_reordered <- cor\_matrix\_T6\_CP\_no\_tss[ordered\_indices, ordered\_indices]  
  
col\_palette <- colorRampPalette(c("blue", "white", "red"))(200)

pdf("Fig6.pdf", width = 10, height = 10) # Adjust width and height as needed  
corrplot(cor\_matrix\_T6\_CP\_no\_tss\_reordered, method = "color", tl.cex = 0.2, tl.col = "black", col = col\_palette)  
dev.off() # Close the PDF device and save the plot

## png   
## 2

### Figure 6B - correlations T6SSi1/i2 selected VAGs

T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigD <- read.csv("~/T6-20241024/Intermediate tables/Correlations/T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigD.csv", sep=";", stringsAsFactors=TRUE)  
  
polychoric\_coef\_T6\_CP\_notss\_D <- hetcor(T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigD, ML = TRUE)$correlations  
  
cor\_matrix\_T6\_CP\_no\_tss\_D <- as.matrix(polychoric\_coef\_T6\_CP\_notss\_D)

pdf("Fig6D.pdf", width = 10, height = 10) # Adjust width and height as needed  
corrplot(cor\_matrix\_T6\_CP\_no\_tss\_D, method = "color", tl.cex = 2.2, tl.col = "black", col = col\_palette)  
dev.off() # Close the PDF device and save the plot

## png   
## 2

### Figure 6C - correlations T6SSi1/i2, MDR, selected VAGs

T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigE <- read.csv("~/T6-20241024/Intermediate tables/Correlations/T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigD.csv", sep=";", stringsAsFactors=TRUE)  
  
polychoric\_coef\_T6\_CP\_notss\_E <- hetcor(T6\_CP\_vfdb\_resfin\_20241017\_no\_tss\_FigE, ML = TRUE)$correlations  
  
cor\_matrix\_T6\_CP\_no\_tss\_E <- as.matrix(polychoric\_coef\_T6\_CP\_notss\_E)

pdf("Fig6E.pdf", width = 10, height = 10) # Adjust width and height as needed  
corrplot(cor\_matrix\_T6\_CP\_no\_tss\_E, method = "color", tl.cex = 2.0, tl.col = "black", col = col\_palette)  
dev.off() # Close the PDF device and save the plot

## png   
## 2

## Suplementary figures

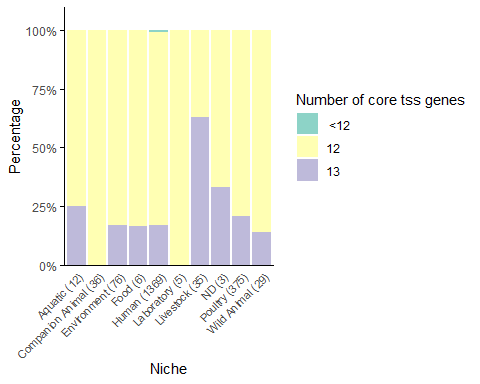
### Figure S8a ST95 niche

ST95 <- subset(EB\_T6\_20241024, ST == "ST95")  
  
ST95 <- ST95 %>%  
 mutate(ST95 = case\_when(  
 NT6SSi1 >= 0 & NT6SSi1 <= 11 ~ "Below 12",  
 NT6SSi1 == 12 ~ "Twelf",  
 NT6SSi1 == 13 ~ "Complete",  
 TRUE ~ NA\_character\_  
 ),  
 # Ensure the factor levels are ordered as desired  
 ST95 = factor(ST95, levels = c("Below 12", "Twelf", "Complete")))  
  
counts\_Niche\_ST95 <- ST95 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST95\_counted\_Niche <- ST95 %>%  
 left\_join(counts\_Niche\_ST95, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST95\_Niche <- ST95\_counted\_Niche %>%  
 group\_by(Source.Niche, ST95) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

custom\_labels <- c("Below 12" = " <12", "Twelf" = "12", "Complete" = "13")

ggplot(proportions\_ST95\_Niche, aes(x = Source.Niche, y = Percentage, fill = ST95)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "Number of core tss genes") +  
 scale\_fill\_manual(values = RColorBrewer::brewer.pal(3, "Set3"), labels = custom\_labels) +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



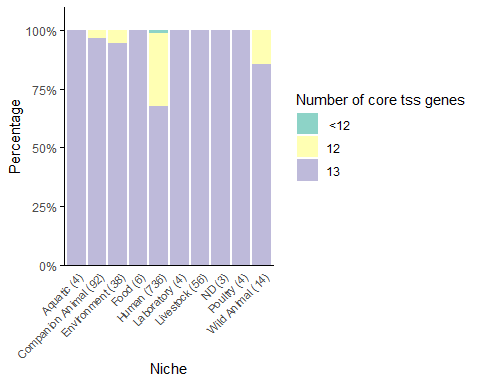
### Figure S8b ST167 niche

ST167 <- subset(EB\_T6\_20241024, ST == "ST167")  
  
ST167 <- ST167 %>%  
 mutate(ST167 = case\_when(  
 NT6SSi1 >= 0 & NT6SSi1 <= 11 ~ "Below 12",  
 NT6SSi1 == 12 ~ "Twelf",  
 NT6SSi1 == 13 ~ "Complete",  
 ),  
 # Ensure the factor levels are ordered as desired  
 ST167 = factor(ST167, levels = c("Below 12", "Twelf", "Complete")))  
  
  
counts\_Niche\_ST167 <- ST167 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST167\_counted\_Niche <- ST167 %>%  
 left\_join(counts\_Niche\_ST167, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST167\_Niche <- ST167\_counted\_Niche %>%  
 group\_by(Source.Niche, ST167) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

custom\_labels <- c("Below 12" = " <12", "Twelf" = "12", "Complete" = "13")

ggplot(proportions\_ST167\_Niche, aes(x = Source.Niche, y = Percentage, fill = ST167)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "Number of core tss genes") +  
 scale\_fill\_manual(values = RColorBrewer::brewer.pal(3, "Set3"), labels = custom\_labels) +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



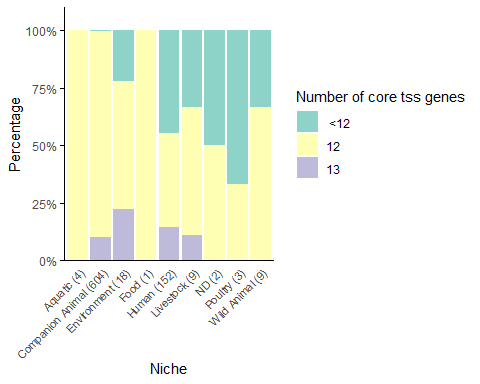
### Figure S8d ST372 niche

ST372 <- subset(EB\_T6\_20241024, ST == "ST372")  
  
ST372 <- ST372 %>%  
 mutate(ST372 = case\_when(  
 NT6SSi1 >= 0 & NT6SSi1 <= 11 ~ "Below 12",  
 NT6SSi1 == 12 ~ "Twelf",  
 NT6SSi1 == 13 ~ "Complete",  
 TRUE ~ NA\_character\_  
 ),  
 # Ensure the factor levels are ordered as desired  
 ST372 = factor(ST372, levels = c("Below 12", "Twelf", "Complete")))  
  
counts\_Niche\_ST372 <- ST372 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST372\_counted\_Niche <- ST372 %>%  
 left\_join(counts\_Niche\_ST372, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST372\_Niche <- ST372\_counted\_Niche %>%  
 group\_by(Source.Niche, ST372) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

custom\_labels <- c("Below 12" = " <12", "Twelf" = "12", "Complete" = "13")

ggplot(proportions\_ST372\_Niche, aes(x = Source.Niche, y = Percentage, fill = ST372)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "Number of core tss genes") +  
 scale\_fill\_manual(values = RColorBrewer::brewer.pal(3, "Set3"), labels = custom\_labels) +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



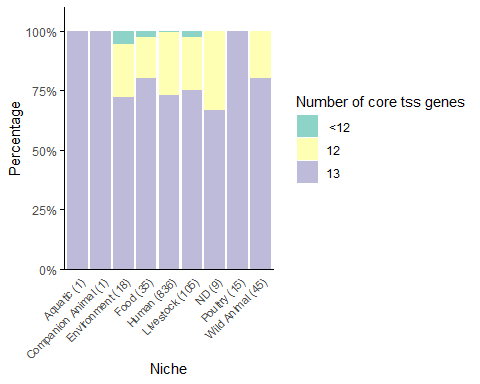
### Figure S8c ST442 niche

ST442 <- subset(EB\_T6\_20241024, ST == "ST442")  
  
ST442 <- ST442 %>%  
 mutate(ST442 = case\_when(  
 NT6SSi1 >= 0 & NT6SSi1 <= 11 ~ "Below 12",  
 NT6SSi1 == 12 ~ "Twelf",  
 NT6SSi1 == 13 ~ "Complete",  
 TRUE ~ NA\_character\_ # Optional: for cases outside the specified ranges  
 ),  
 # Ensure the factor levels are ordered as desired  
 ST442 = factor(ST442, levels = c("Below 12", "Twelf", "Complete")))  
  
counts\_Niche\_ST442 <- ST442 %>%  
 group\_by(Source.Niche) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST442\_counted\_Niche <- ST442 %>%  
 left\_join(counts\_Niche\_ST442, by = "Source.Niche") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(Source.Niche = paste0(Source.Niche, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST442\_Niche <- ST442\_counted\_Niche %>%  
 group\_by(Source.Niche, ST442) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'Source.Niche'. You can override using the  
## `.groups` argument.

custom\_labels <- c("Below 12" = " <12", "Twelf" = "12", "Complete" = "13")

ggplot(proportions\_ST442\_Niche, aes(x = Source.Niche, y = Percentage, fill = ST442)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "Number of core tss genes") +  
 scale\_fill\_manual(values = RColorBrewer::brewer.pal(3, "Set3"), labels = custom\_labels) +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )

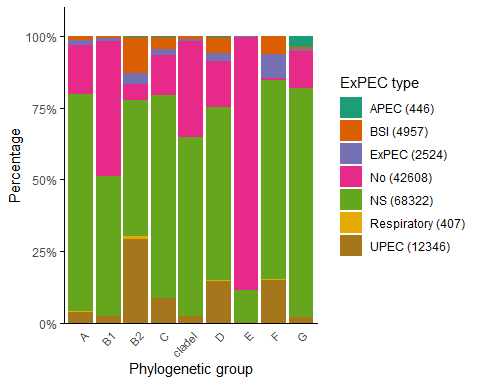


### Figure S2a ExPEC type vs PGs

proportions\_PG\_ExPEC.type <- EB\_T6\_20241024 %>%  
 group\_by(PG, ExPEC.type) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_PG\_ExPEC.type, aes(x = PG, y = Percentage, fill = ExPEC.type)) +   
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "ExPEC type") +  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_ExPEC), "Dark2"),  
 labels = custom\_labels\_ExPEC) + # Use custom labels  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )

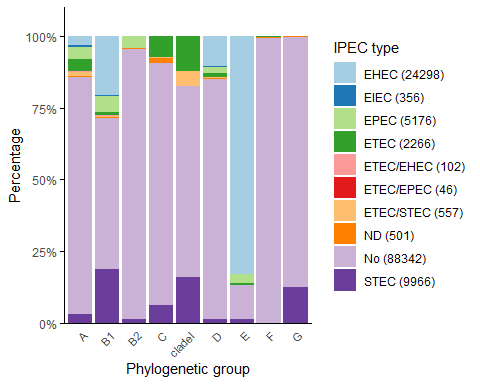


### Figure S2b IPEC type vs PGs

proportions\_PG\_IPEC.type <- EB\_T6\_20241024 %>%  
 group\_by(PG, IPEC.type) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

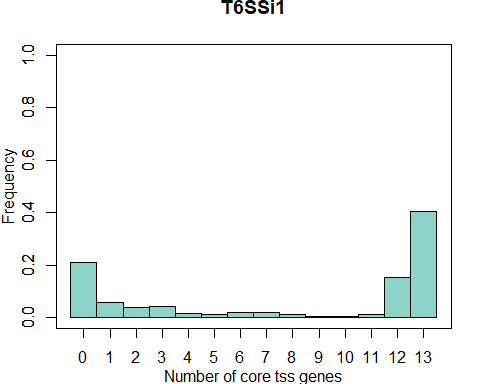
ggplot(proportions\_PG\_IPEC.type, aes(x = PG, y = Percentage, fill = IPEC.type)) +   
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "IPEC type") +  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_IPEC), "Paired"),  
 labels = custom\_labels\_IPEC) + # Use custom labels  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Supplementary histograms - example for T6SSi1 DB2

hist\_plot\_NST6SSi1 <- function() {  
 hist(EB\_T6\_20241024[,"NST6SSi1"],   
 breaks = seq(-0.5, 13.5, 1), # Bins corresponding to integer values  
 main = "T6SSi1",   
 xlab = "Number of core tss genes",   
 ylab = "Frequency",   
 col = "#8DD3C7", # Set3 color from ggplot  
 freq = FALSE, # Use probability densities instead of frequencies  
 xaxt = 'n', # Suppress the default x-axis  
 ylim = c(0, 1))  
   
 # Customize the x-axis and y-axis  
 axis(1, at = 0:13, labels = 0:13, col.axis = "black", col = "black") # X-axis only  
 axis(2, col.axis = "black", col = "black") # Y-axis only  
   
 # Add box outline for axis lines  
 box(col = "black")  
}

as.ggplot(hist\_plot\_NST6SSi1)



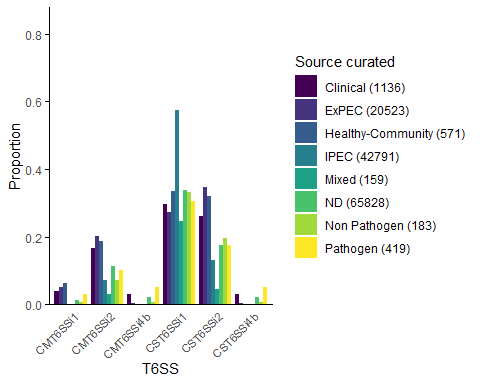
### Figure S5 T6SS (Completeness) in Clinical groups for both DB

Clinical\_CT6\_curated\_dbs <- EB\_T6\_20241024 %>%  
 select(Source.curated, CMT6SSi1, CMT6SSi2, CMT6SSi4b, CST6SSi1, CST6SSi2, CST6SSi4b)  
  
counts\_Source\_curated\_dbs <- Clinical\_CT6\_curated\_dbs %>% #Calculate counts for each category in Source.curated  
 group\_by(Source.curated) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
print(counts\_Source\_curated\_dbs) #Display the counts

## # A tibble: 8 × 2  
## Source.curated Count  
## <fct> <int>  
## 1 Clinical 1136  
## 2 ExPEC 20523  
## 3 Healthy-Community 571  
## 4 IPEC 42791  
## 5 Mixed 159  
## 6 ND 65828  
## 7 Non Pathogen 183  
## 8 Pathogen 419

data\_long\_Clinical\_T6C\_dbs <- Clinical\_CT6\_curated\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -Source.curated, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_Clinical\_T6C\_dbs <- data\_long\_Clinical\_T6C\_dbs %>% #Calculate proportions  
 group\_by(Source.curated, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels <- c( #Create a named vector for custom labels  
 "Clinical" = "Clinical (1136)",  
 "ExPEC" = "ExPEC (20523)",  
 "Healthy-Community" = "Healthy-Community (571)",  
 "IPEC" = "IPEC (42791)",  
 "Mixed" = "Mixed (159)",  
 "ND" = "ND (65828)",  
 "Non Pathogen" = "Non Pathogen (183)",  
 "Pathogen" = "Pathogen (419)"  
)

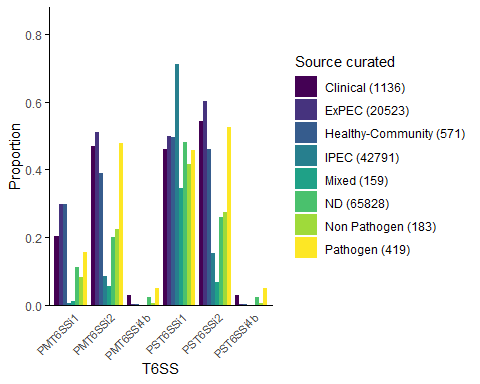
ggplot(proportions\_Clinical\_T6C\_dbs, aes(x = T6SS, y = Proportion, fill = Source.curated)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "Source curated") + # Add a label for fill legend  
 scale\_fill\_manual(values = viridis::viridis(length(custom\_labels)),   
 labels = custom\_labels) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S5b T6SS in Clinical groups for both DB (Presence)

Clinical\_PT6\_curated\_dbs <- EB\_T6\_20241024 %>%  
 select(Source.curated, PMT6SSi1, PMT6SSi2, PMT6SSi4b, PST6SSi1, PST6SSi2, PST6SSi4b)  
  
data\_long\_Clinical\_T6P\_dbs <- Clinical\_PT6\_curated\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -Source.curated, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_Clinical\_T6P\_dbs <- data\_long\_Clinical\_T6P\_dbs %>% #Calculate proportions  
 group\_by(Source.curated, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')

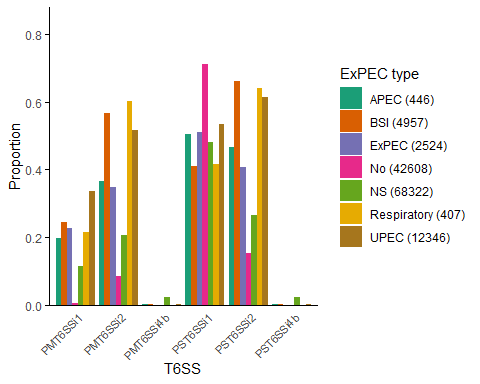
ggplot(proportions\_Clinical\_T6P\_dbs, aes(x = T6SS, y = Proportion, fill = Source.curated)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "Source curated") + # Add a label for fill legend  
 scale\_fill\_manual(values = viridis::viridis(length(custom\_labels)),   
 labels = custom\_labels) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S6a T6SS (Presence) in ExPEC type for both DB

ExPECtype\_PT6\_dbs <- EB\_T6\_20241024 %>%  
 select(ExPEC.type, PMT6SSi1, PMT6SSi2, PMT6SSi4b, PST6SSi1, PST6SSi2, PST6SSi4b)  
  
data\_long\_ExPECtype\_T6P\_dbs <- ExPECtype\_PT6\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -ExPEC.type, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_ExPECtype\_T6P\_dbs <- data\_long\_ExPECtype\_T6P\_dbs %>% #Calculate proportions  
 group\_by(ExPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')  
  
custom\_labels\_ExPEC <- c( #Create a named vector for custom labels  
 "APEC" = "APEC (446)",  
 "BSI" = "BSI (4957)",  
 "ExPEC" = "ExPEC (2524)",  
 "No" = "No (42608)",  
 "NS" = "NS (68322)",  
 "ND" = "ND (65828)",  
 "Respiratory" = "Respiratory (407)",  
 "UPEC" = "UPEC (12346)"  
)

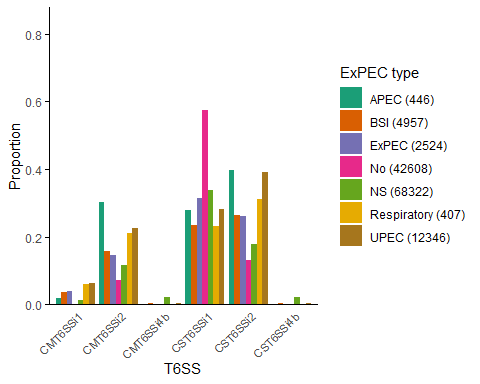
ggplot(proportions\_ExPECtype\_T6P\_dbs, aes(x = T6SS, y = Proportion, fill = ExPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "ExPEC type") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_ExPEC), "Dark2"),  
 labels = custom\_labels\_ExPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S6b T6SS (Complete) in ExPEC type

ExPECtype\_CT6\_dbs <- EB\_T6\_20241024 %>%  
 select(ExPEC.type, CMT6SSi1, CMT6SSi2, CMT6SSi4b, CST6SSi1, CST6SSi2, CST6SSi4b)  
  
data\_long\_ExPECtype\_T6C\_dbs <- ExPECtype\_CT6\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -ExPEC.type, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_ExPECtype\_T6C\_dbs <- data\_long\_ExPECtype\_T6C\_dbs %>% #Calculate proportions  
 group\_by(ExPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')

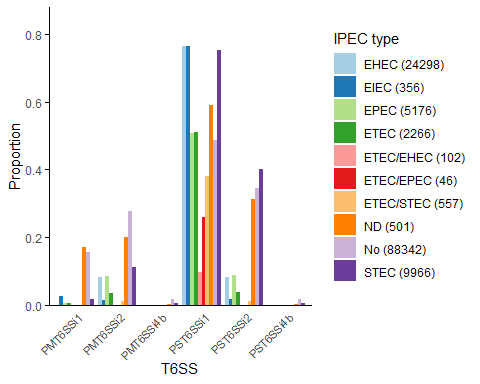
ggplot(proportions\_ExPECtype\_T6C\_dbs, aes(x = T6SS, y = Proportion, fill = ExPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "ExPEC type") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_ExPEC), "Dark2"),  
 labels = custom\_labels\_ExPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S7a T6SS (Presence) proportion in IPEC type for both DBs

IPECtype\_PT6\_dbs <- EB\_T6\_20241024 %>%  
 select(IPEC.type, PMT6SSi1, PMT6SSi2, PMT6SSi4b, PST6SSi1, PST6SSi2, PST6SSi4b)  
  
data\_long\_IPECtype\_T6P\_dbs <- IPECtype\_PT6\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -IPEC.type, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_IPECtype\_T6P\_dbs <- data\_long\_IPECtype\_T6P\_dbs %>% #Calculate proportions  
 group\_by(IPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')

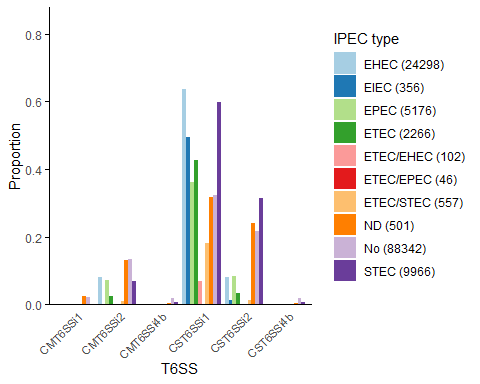
ggplot(proportions\_IPECtype\_T6P\_dbs, aes(x = T6SS, y = Proportion, fill = IPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "IPEC type") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_IPEC), "Paired"),  
 labels = custom\_labels\_IPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S7b T6SS proportion (Complete) in IPEC type

IPECtype\_CT6\_dbs <- EB\_T6\_20241024 %>%  
 select(IPEC.type, CMT6SSi1, CMT6SSi2, CMT6SSi4b, CST6SSi1, CST6SSi2, CST6SSi4b)  
  
data\_long\_IPECtype\_T6C\_dbs <- IPECtype\_CT6\_dbs %>% #Reshape the data to long format  
 pivot\_longer(cols = -IPEC.type, names\_to = "T6SS", values\_to = "Value")  
  
proportions\_IPECtype\_T6C\_dbs <- data\_long\_IPECtype\_T6C\_dbs %>% #Calculate proportions  
 group\_by(IPEC.type, T6SS) %>%  
 summarize(Proportion = mean(Value == 1), .groups = 'drop')

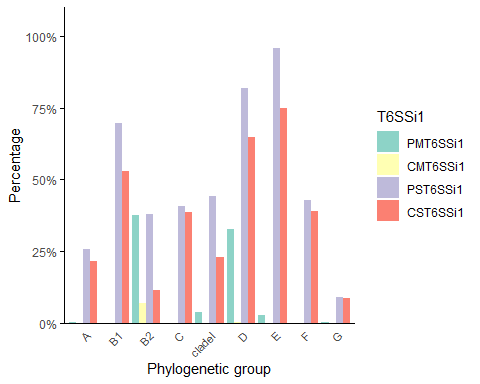
ggplot(proportions\_IPECtype\_T6C\_dbs, aes(x = T6SS, y = Proportion, fill = IPEC.type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "T6SS", y = "Proportion", fill = "IPEC type") + # Add a label for fill legend  
 scale\_fill\_manual(values = brewer.pal(length(custom\_labels\_IPEC), "Paired"),  
 labels = custom\_labels\_IPEC) + # Use custom labels  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits without percentage  
 coord\_cartesian(ylim = c(0, 0.8)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S3a T6SSi1 for both DB

summary\_data\_i1\_dbs <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PMT6SSi1 = mean(PMT6SSi1 == 1) \* 100,  
 CMT6SSi1 = mean(CMT6SSi1 == 1) \* 100,  
 PST6SSi1 = mean(PST6SSi1 == 1) \* 100,  
 CST6SSi1 = mean(CST6SSi1 == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PMT6SSi1", "CMT6SSi1", "PST6SSi1", "CST6SSi1"))) # Reorder levels

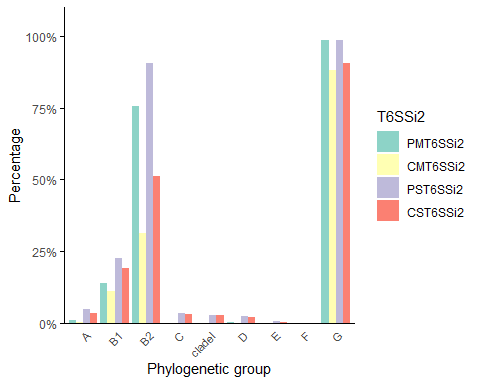
ggplot(summary\_data\_i1\_dbs, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi1") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S3b T6SSi2 in both DBs

summary\_data\_i2\_dbs <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PMT6SSi2 = mean(PMT6SSi2 == 1) \* 100,  
 CMT6SSi2 = mean(CMT6SSi2 == 1) \* 100,  
 PST6SSi2 = mean(PST6SSi2 == 1) \* 100,  
 CST6SSi2 = mean(CST6SSi2 == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PMT6SSi2", "CMT6SSi2", "PST6SSi2", "CST6SSi2"))) # Reorder levels

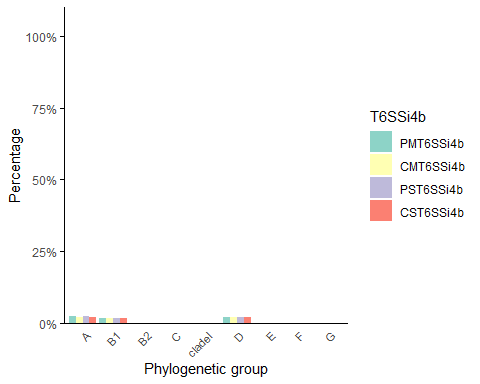
ggplot(summary\_data\_i2\_dbs, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi2") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S3c T6SSi4b in both DBs

summary\_data\_i4b\_dbs <- EB\_T6\_20241024 %>%  
 group\_by(PG) %>%  
 summarize(  
 PMT6SSi4b = mean(PMT6SSi4b == 1) \* 100,  
 CMT6SSi4b = mean(CMT6SSi4b == 1) \* 100,  
 PST6SSi4b = mean(PST6SSi4b == 1) \* 100,  
 CST6SSi4b = mean(CST6SSi4b == 1) \* 100  
 ) %>%  
 pivot\_longer(cols = -PG, names\_to = "T6SS", values\_to = "Percentage") %>%  
 mutate(T6SS = factor(T6SS, levels = c("PMT6SSi4b", "CMT6SSi4b", "PST6SSi4b", "CST6SSi4b"))) # Reorder levels

ggplot(summary\_data\_i4b\_dbs, aes(x = PG, y = Percentage, fill = T6SS)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(x = "Phylogenetic group", y = "Percentage", fill = "T6SSi4b") +  
 scale\_fill\_brewer(palette = "Set3") + # Use the Set3 color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) +  
 theme\_minimal() +  
 coord\_cartesian(ylim = c(0, 100)) + # Set y-axis limits  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )

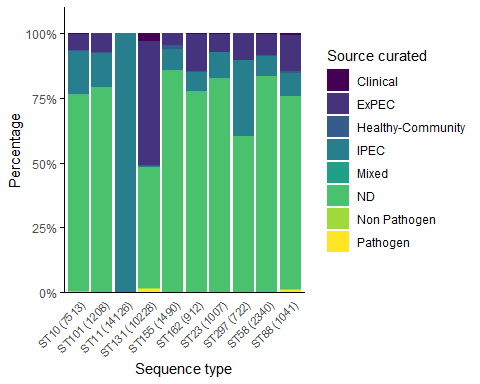


### Figure S9 STs with ExPEC/IPEC overlap

ST\_filtered <- subset(EB\_T6\_20241024, ST %in% c("ST10", "ST23", "ST101", "ST88", "ST162", "ST297", "ST155", "ST58", "ST131", "ST11"))  
  
counts\_clinical\_ST <- ST\_filtered %>%  
 group\_by(ST) %>%  
 summarize(Count = n(), .groups = 'drop')  
  
ST\_counted\_clinical <- ST\_filtered %>%  
 left\_join(counts\_clinical\_ST, by = "ST") %>%  
 # Format the Source.Niche column to include the count in brackets  
 mutate(ST = paste0(ST, " (", Count, ")")) %>%  
 # Remove the Count column if it's no longer needed  
 select(-Count)  
  
proportions\_ST\_clinical <- ST\_counted\_clinical %>%  
 group\_by(ST, Source.curated) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

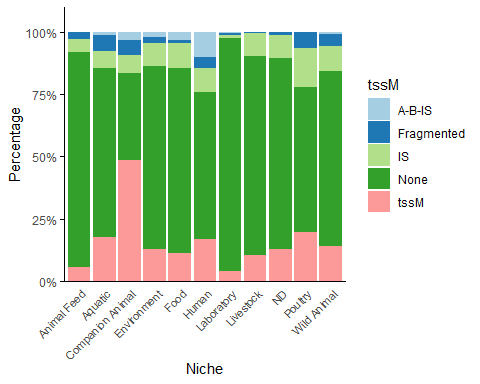
## `summarise()` has grouped output by 'ST'. You can override using the `.groups`  
## argument.

ggplot(proportions\_ST\_clinical, aes(x = ST, y = Percentage, fill = Source.curated)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Sequence type",  
 y = "Percentage",  
 fill = "Source curated") +  
 scale\_fill\_viridis\_d() + # Use the viridis color palette  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



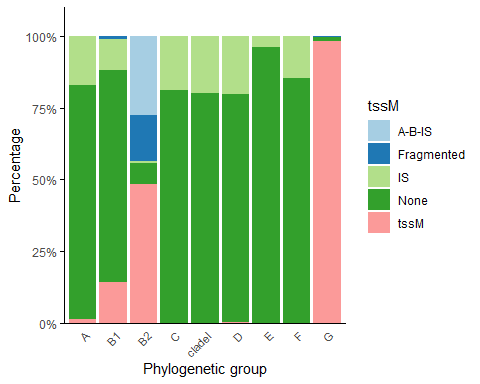
### Figure S10a whole collection and tssM for Niche

ggplot(proportions\_Niche\_tssM, aes(x = Source.Niche, y = Percentage, fill = tssM\_frag)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "tssM") +  
 scale\_fill\_brewer(palette = "Paired") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



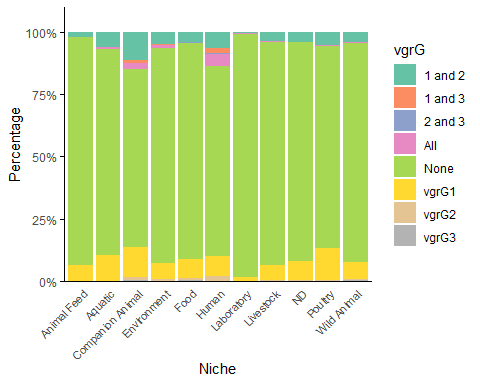
### Figure S10b whole collection and tssM for PG

ggplot(proportions\_PG\_tssM, aes(x = PG, y = Percentage, fill = tssM\_frag)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "tssM") +  
 scale\_fill\_brewer(palette = "Paired") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Figure S10c whole collection and vgrG for Niche

ggplot(proportions\_Niche\_vgrG, aes(x = Source.Niche, y = Percentage, fill = vgrG\_KO178)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Niche",  
 y = "Percentage",  
 fill = "vgrG") +  
 scale\_fill\_brewer(palette = "Set2") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )

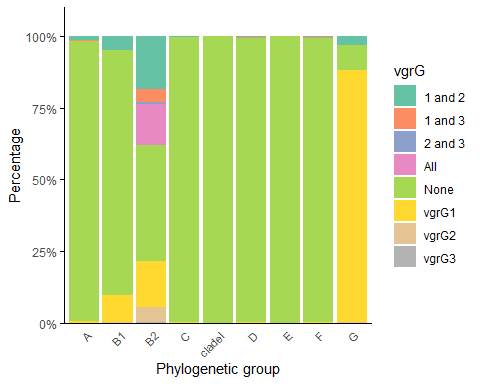


### Figure S10d whole collection and vgrG for PG

proportions\_PG\_vgrG <- EB\_T6\_20241024 %>%  
 group\_by(PG, vgrG\_KO178) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = (Count / sum(Count)) \* 100)

## `summarise()` has grouped output by 'PG'. You can override using the `.groups`  
## argument.

ggplot(proportions\_PG\_vgrG, aes(x = PG, y = Percentage, fill = vgrG\_KO178)) +  
 geom\_bar(stat = "identity", position = "stack") + # Use stack for cumulative bars  
 labs(x = "Phylogenetic group",  
 y = "Percentage",  
 fill = "vgrG") +  
 scale\_fill\_brewer(palette = "Set2") +  
 scale\_y\_continuous(labels = scales::percent\_format(scale = 1), expand = expansion(mult = c(0, 0.1))) + # Adjust y-axis limits  
 theme\_minimal() +  
 theme(  
 panel.background = element\_blank(),  
 panel.grid = element\_blank(),  
 axis.line = element\_line(color = "black"),  
 axis.ticks.y = element\_line(color = "black"),  
 axis.text.x = element\_text(size = 8, angle = 45, hjust = 1)  
 )



### Supplementary Figure 11A - correlations Cluster1

cluster1\_indices <- which(groups == 1)  
  
cor\_matrix\_cluster1 <- cor\_matrix\_T6\_CP\_no\_tss[cluster1\_indices, cluster1\_indices]

pdf("Cluster1.pdf", width = 8, height = 8)  
corrplot(cor\_matrix\_cluster1, method = "color", tl.cex = 0.3, tl.col = "black", col = col\_palette)  
dev.off()

## png   
## 2

### Supplementary Figure 11B - correlations Cluster1

cluster2\_indices <- which(groups == 2)  
  
cor\_matrix\_cluster2 <- cor\_matrix\_T6\_CP\_no\_tss[cluster2\_indices, cluster2\_indices]

pdf("Cluster2.pdf", width = 8, height = 8)  
corrplot(cor\_matrix\_cluster2, method = "color", tl.cex = 0.3, tl.col = "black", col = col\_palette)  
dev.off()

## png   
## 2

## Tables and Supplementary Tables

Supplementary Table 1

write.table(EB\_T6\_20241024, file = "Supplementary\_Table1.txt", sep = "\t", quote = FALSE, row.names = F)

### Table 1, Table 2, Supplementary Table 2 and 3

Intermediate table to calculate prevalences for given categories

selected\_columns <- c("Species", "Source.Niche", "Source.curated", "ExPEC.type",   
 "IPEC.type", "ExPEC\_STs", "Gender", "ST", "PG",   
 "CM\_bin", "PM\_bin",   
 "CT6SSi2", "CT6SSi1", "CT6SSi4b", "Cmin1", "Pmin1",   
 "PT6SSi2", "PT6SSi1", "PT6SSi4b",   
 "CMT6SSi2", "CMT6SSi1", "CMT6SSi4b",   
 "PMT6SSi2", "PMT6SSi1", "PMT6SSi4b",   
 "CST6SSi2", "CST6SSi1", "CST6SSi4b",   
 "PST6SSi2", "PST6SSi1", "PST6SSi4b")  
  
  
EB\_T6\_20241024\_prevalence <- EB\_T6\_20241024[, selected\_columns]  
  
write.table(EB\_T6\_20241024\_prevalence, file = "EB\_T6\_20241024\_prevalence.txt", sep = "\t", quote = FALSE, row.names = F)

Prevalence table updated to contain suffix \_bin for each T6SS binary column

Calculating prevalence for give metadata group (example with PG)

EB\_T6\_20241024\_prevelences <- read.csv("~/T6-20241024/Intermediate tables/EB\_T6\_20241024\_prevelences.csv", sep=";", stringsAsFactors=TRUE)  
  
grouped\_PG <- EB\_T6\_20241024\_prevelences %>%   
 group\_by(PG)  
  
prevalence\_data\_PG <- grouped\_PG %>%   
 summarise\_at(vars(ends\_with("\_bin")), ~ mean(.))

### Supplementary Table 4 (ExPEC) and 6 (IPEC) - counting STs within specific group; example for ExPEC (STable 4)

ExPEC <- subset(EB\_T6\_20241024, Source.curated == "ExPEC")  
  
category\_counts\_ExPEC\_ST <- ExPEC %>% count(ST)  
  
write.table(category\_counts\_ExPEC\_ST, file = "ExPEC\_ST.txt", sep = "\t", quote = FALSE, row.names = F)

### Supplementary Table 5 (ExPEC types) and 7 (IPEC types) - counting STs within specific ExPEC/IPEC type - example for APEC

APEC <- subset(EB\_T6\_20241024, ExPEC.type == "APEC")  
  
category\_counts\_APEC\_ST <- APEC %>% count(ST)  
  
write.table(category\_counts\_APEC\_ST, file = "APEC\_ST.txt", sep = "\t", quote = FALSE, row.names = F)

### Supplementary tables 10: Statistics - Wilcoxon rank-sum test adjusted by Benjamini-Hochberg - example for presence of T6SSi1 and Clinical groups (Source.curated)

df1 <- EB\_T6\_20241024  
group\_names <- unique(EB\_T6\_20241024$Source.curated)  
p\_values\_curated\_PT6i1 <- c() # Initialize the vector to store p-values for CT6i4b  
comparisons <- c()   
for (i in 1:(length(group\_names) - 1)) {  
 for (j in (i + 1):length(group\_names)) {  
 group\_i <- df1[df1$Source.curated == group\_names[i], "PT6SSi1"]  
 group\_j <- df1[df1$Source.curated == group\_names[j], "PT6SSi1"]  
   
 # Perform Wilcoxon rank-sum test  
 test\_result <- wilcox.test(group\_i, group\_j)  
   
 # Store the p-value and comparison name in the correct vectors  
 p\_values\_curated\_PT6i1 <- c(p\_values\_curated\_PT6i1, test\_result$p.value)  
 comparisons <- c(comparisons, paste(group\_names[i], "vs", group\_names[j]))  
 }  
 }  
adjusted\_p\_values\_curated\_PT6i1 <- p.adjust(p\_values\_curated\_PT6i1, method = "BH")  
cat("Length of comparisons:", length(comparisons), "\n")

## Length of comparisons: 28

cat("Length of p\_values\_curated\_PT6i1:", length(p\_values\_curated\_PT6i1), "\n")

## Length of p\_values\_curated\_PT6i1: 28

cat("Length of adjusted\_p\_values\_curated\_PT6i1:", length(adjusted\_p\_values\_curated\_PT6i1), "\n")

## Length of adjusted\_p\_values\_curated\_PT6i1: 28

results\_df\_curated\_PT6i1 <- data.frame(  
 Comparison = comparisons,  
 P\_value = p\_values\_curated\_PT6i1,  
 Adjusted\_P\_value = adjusted\_p\_values\_curated\_PT6i1)  
write.csv(results\_df\_curated\_PT6i1, "pairwise\_comparisons\_curated\_PT6i1.csv", row.names = FALSE)