Single cell expression analyses

Jennifer Tran

2024-08-19

Load necessary packages for these graphs:

```
require('pacman')

p_load(dplyr, data.table, ggplot2, tidyr, RColorBrewer, stringr, ggridges, colourpicker, ggforce)
```

Flow data ridgeplots for plasmid expression (Fig 4)

Load all data tables (.csv for each sample) and sample names

```
##     SampleID IPTG_uM
##  1:    eco01      EV
##  2:    eco02       0
##  3:    eco03     7.8
##  4:    eco04    15.6
##  5:    eco05    31.3
##  6:    eco06    62.5
##  7:    eco07     125
##  8:    eco08     250
##  9:    eco09     500
## 10:    eco10    1000
## 11:    aba01      EV
## 12:    aba02       0
## 13:    aba03     7.8
## 14:    aba04    15.6
## 15:    aba05    31.3
## 16:    aba06    62.5
## 17:    aba07     125
## 18:    aba08     250
## 19:    aba09     500
## 20:    aba10    1000
```

Place all loaded dataframes into a list

```
#A. baumannii list
list_of_aba <- list(aba01 = aba01, aba02 = aba02, aba03 = aba03, aba04 = aba04,
                    aba05 = aba05, aba06 = aba06, aba07 = aba07, aba08 = aba08,
                    aba09 = aba09, aba10 = aba10)

combined_aba <- rbindlist(list_of_aba, idcol = "SampleID")
```

```
#E. coli list
list_of_eco <- list(eco01 = eco01, eco02 = eco02, eco03 = eco03, eco04 = eco04,
                    eco05 = eco05, eco06 = eco06, eco07 = eco07, eco08 = eco08,
                    eco09 = eco09, eco10 = eco10)

combined_eco <- rbindlist(list_of_eco, idcol = "SampleID")
```

```
## [1] "A. baumannii list"
```

```
## Rows: 607,869
## Columns: 8
## $ SampleID  <chr> "aba01", "aba01", "aba01", "aba01", "aba01", "aba01", "aba01~
## $ ‘FSC-A‘   <int> 545, 594, 611, 509, 608, 611, 668, 560, 598, 649, 611, 552, ~
## $ ‘FSC-H‘   <int> 476, 498, 501, 415, 506, 473, 581, 458, 506, 511, 477, 410, ~
## $ ‘SSC-A‘   <int> 508, 592, 613, 508, 567, 570, 626, 543, 457, 646, 519, 514, ~
## $ ‘SSC-H‘   <int> 567, 605, 603, 536, 578, 559, 620, 585, 546, 612, 536, 551, ~
## $ autofluor <int> 214, 288, 240, 206, 221, 234, 209, 281, 201, 231, 201, 210, ~
## $ GFP       <int> 40, 221, 268, 208, 213, 254, 215, 250, 63, 299, 298, 165, 23~
## $ Time      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

```
## [1] "E. coli list"
```

```
## Rows: 815,202
## Columns: 8
## $ SampleID  <chr> "eco01", "eco01", "eco01", "eco01", "eco01", "eco01", "eco01~
## $ ‘FSC-A‘   <int> 459, 655, 658, 513, 582, 638, 504, 441, 538, 608, 517, 581, ~
## $ ‘FSC-H‘   <int> 536, 572, 615, 562, 526, 569, 523, 509, 520, 588, 510, 546, ~
## $ ‘SSC-A‘   <int> 604, 707, 692, 635, 653, 576, 627, 642, 619, 672, 601, 681, ~
## $ ‘SSC-H‘   <int> 609, 674, 675, 646, 630, 607, 617, 632, 611, 663, 581, 656, ~
## $ autofluor <int> 243, 252, 257, 226, 238, 269, 227, 232, 242, 223, 226, 254, ~
## $ GFP       <int> 430, 530, 418, 423, 423, 384, 385, 490, 468, 446, 436, 431, ~
## $ Time      <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

Normalize fluorescence data (GFP readings) to empty vector control
This centers no fluorescence readings around 0; relative fluorescence

```
##for A. baumannii
# Filter the negative control for A. baumannii
neg_control_ab <- combined_aba %>% filter(SampleID == "aba01")

# Calculate the mean and standard deviation for the negative control
mean_ab <- mean(neg_control_ab$GFP, na.rm = TRUE)
sd_ab <- sd(neg_control_ab$GFP, na.rm = TRUE)

# Normalize GFP values by subtracting the mean of the negative control
combined_aba <- combined_aba %>%
  mutate(norm_GFP = GFP - mean_ab)

# Add IDs from the sample map
combined_aba <- combined_aba %>%
  left_join(sample_map, by = "SampleID")
```

```r
# Create a factor for IPTG concentration levels
combined_aba$IPTG_uM_factor <- factor(
  combined_aba$IPTG_uM,
  levels = unique(combined_aba$IPTG_uM[order(combined_aba$SampleID)])
)

# Add IDs from the sample map and calculate statistics for each sample
combined_aba_stats <- combined_aba %>%
  group_by(SampleID) %>%
  summarise(
    mean = mean(GFP, na.rm = TRUE),
    sd = sd(GFP, na.rm = TRUE),
    N = n(),
    .groups = 'drop'
  ) %>%
  mutate(
    sd_adj = sqrt(sd_ab^2 + sd^2),
    mean_adj = mean - mean_ab,
    N_final = N
  ) %>%
  left_join(sample_map, by = "SampleID")

print(combined_aba_stats)
```

```
## # A tibble: 10 x 8
##    SampleID  mean    sd     N sd_adj mean_adj N_final IPTG_uM
##    <chr>    <dbl> <dbl> <int>  <dbl>    <dbl>   <int> <chr>
##  1 aba01     240.  74.6 56044   105.      0     56044 EV
##  2 aba02     265.  81.9 59754   111.     24.5   59754 0
##  3 aba03     325. 105.  56622   129.     84.7   56622 7.8
##  4 aba04     408. 115.  60162   137.    168.    60162 15.6
##  5 aba05     503. 123.  58565   144.    263.    58565 31.3
##  6 aba06     590. 123.  59589   144.    350.    59589 62.5
##  7 aba07     647. 120.  62186   142.    407.    62186 125
##  8 aba08     689. 116.  65013   138.    449.    65013 250
##  9 aba09     692. 118.  61259   140.    452.    61259 500
## 10 aba10     703. 112.  68675   135.    463.    68675 1000
```

```r
##for E. coli##
# Filter the negative control for E. coli
neg_control_eco <- combined_eco %>% filter(SampleID == "eco01")

# Calculate the mean and standard deviation for the negative control
mean_eco <- mean(neg_control_eco$GFP, na.rm = TRUE)
sd_eco <- sd(neg_control_eco$GFP, na.rm = TRUE)
neg_n_eco <- nrow(neg_control_eco)

# Normalize GFP values by subtracting the mean of the negative control
combined_eco <- combined_eco %>%
  mutate(norm_GFP = GFP - mean_eco)

# Add IDs from the sample map
combined_eco <- combined_eco %>%
  left_join(sample_map, by = "SampleID")

# Create a factor for IPTG concentration levels
combined_eco$IPTG_uM_factor <- factor(
  combined_eco$IPTG_uM,
  levels = unique(combined_eco$IPTG_uM[order(combined_eco$SampleID)])
)

# Add IDs from the sample map and calculate statistics for each sample
combined_eco_stats <- combined_eco %>%
  group_by(SampleID) %>%
  summarise(
    mean = mean(GFP, na.rm = TRUE),
    sd = sd(GFP, na.rm = TRUE),
    N = n(),
    .groups = 'drop'
  ) %>%
  mutate(
    sd_adj = sqrt(sd_eco^2 + sd^2),
    mean_adj = mean - mean_eco,
    N_final = N
  ) %>%
    left_join(sample_map, by = "SampleID")

print(combined_eco_stats)
```

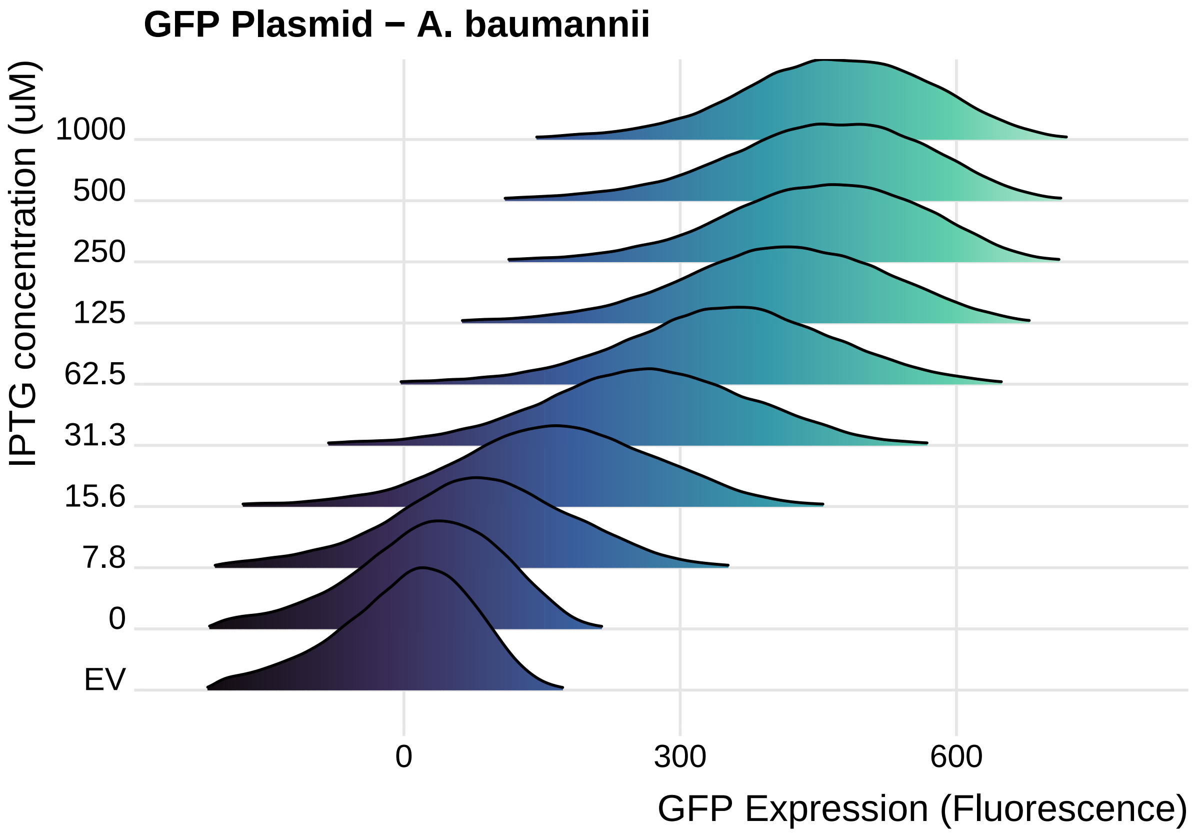
```
## # A tibble: 10 x 8
##    SampleID  mean    sd     N sd_adj mean_adj N_final IPTG_uM
##    <chr>    <dbl> <dbl> <int>  <dbl>    <dbl>   <int> <chr>
##  1 eco01     364.  78.8 71113  111.       0     71113 EV
##  2 eco02     463.  69.6 81428  105.      99.2   81428 0
##  3 eco03     498.  65.1 81185  102.     134.    81185 7.8
##  4 eco04     525.  59.1 82785   98.5    161.    82785 15.6
##  5 eco05     569.  55.5 81978   96.4    205.    81978 31.3
##  6 eco06     619.  56.5 83112   96.9    255.    83112 62.5
##  7 eco07     662.  58.6 83752   98.2    298.    83752 125
##  8 eco08     695.  59.2 84954   98.5    331.    84954 250
##  9 eco09     707.  66.2 83821  103.     343.    83821 500
## 10 eco10     719.  62.6 81074  101.     355.    81074 1000
```

Plot stacked density plots for each IPTG concentration

#####*A. baumannii* ATCC 17978

```
ggplot(combined_aba, aes(x = norm_GFP, y = IPTG_uM_factor,
                         height = after_stat(density))) +
  geom_density_ridges_gradient(
    scale = 2, # Could adjust for scale
    aes(fill = after_stat(x)),
    gradient_lwd = 0.0,
    rel_min_height = 0.02
  ) +
  scale_fill_viridis_c(name = "norm_GFP", option = "G") +
  labs(x = "GFP Expression (Fluorescence)", y = "IPTG concentration (uM)",
       title = "GFP Plasmid - A. baumannii") +
  theme_ridges() + theme(legend.position = "none")
```

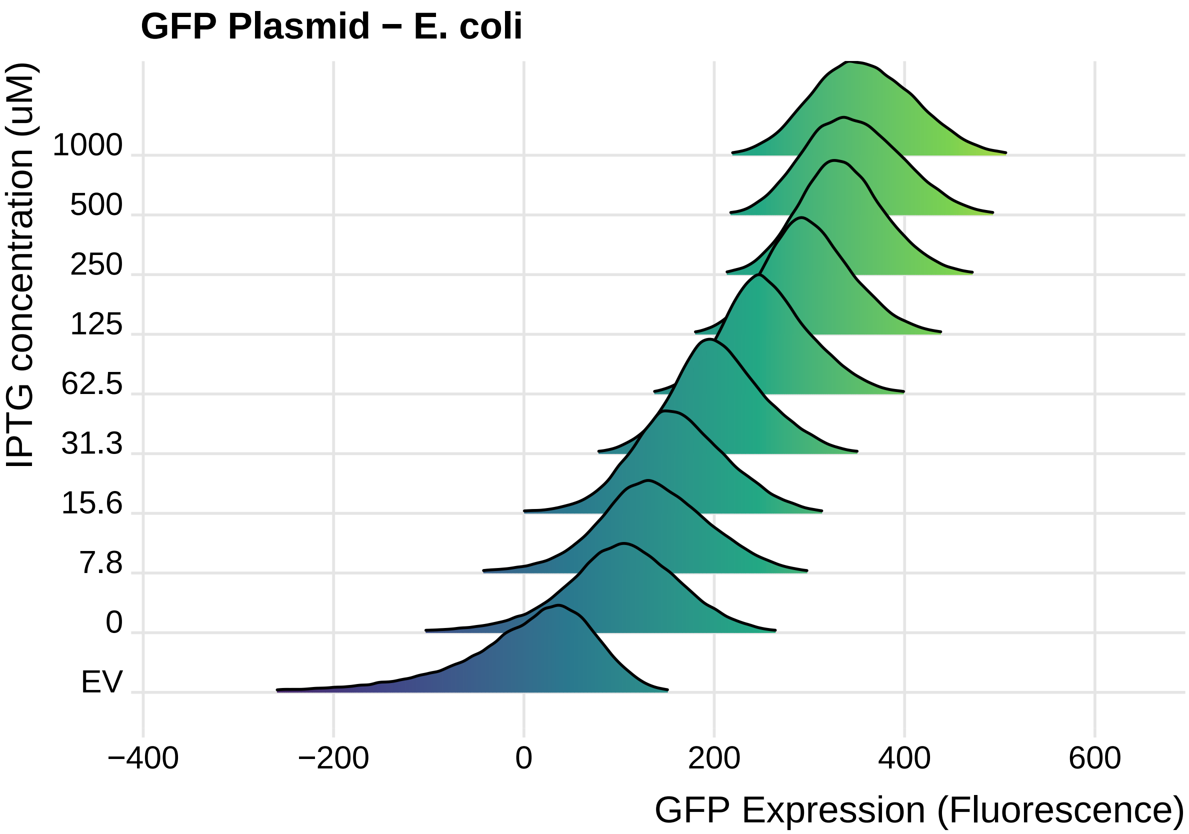
```
## Picking joint bandwidth of 10.4
```



#####for *E. coli* BW25113

```
ggplot(combined_eco, aes(x = norm_GFP, y = IPTG_uM_factor, height = after_stat(density))) +
  geom_density_ridges_gradient(
    scale = 2,
    aes(fill = after_stat(x)),
    gradient_lwd = 0.0,
    rel_min_height = 0.02
  ) +
  scale_fill_viridis_c(name = "norm_GFP", option = "D") +
  labs(x = "GFP Expression (Fluorescence)", y = "IPTG concentration (uM)",
       title = "GFP Plasmid - E. coli") +
  theme_ridges() + theme(legend.position = "none")
```

```
## Picking joint bandwidth of 5.01
```



```
# Function to perform pairwise unpaired t-tests with Bonferroni adjustment
perform_pairwise_t_tests <- function(data) {
  IPTG_conc <- unique(data$IPTG_uM)
  results <- list()
  num_comparisons <- length(IPTG_conc) * (length(IPTG_conc) - 1) / 2  # Total number of comparisons

  for (i in 1:(length(IPTG_conc) - 1)) {
    for (j in (i + 1):length(IPTG_conc)) {
      IPTG_conc1 <- IPTG_conc[i]
      IPTG_conc2 <- IPTG_conc[j]

      data1 <- data %>% filter(IPTG_uM == IPTG_conc1)
      data2 <- data %>% filter(IPTG_uM == IPTG_conc2)

      # Ensure that the number of rows in each group being compared is the same
      if(nrow(data1) == nrow(data2)) {
        # Extract values
        mean1 <- data1$mean_adj
        mean2 <- data2$mean_adj
        sd1 <- data1$sd_adj
        sd2 <- data2$sd_adj
        n1 <- data1$N_final
        n2 <- data2$N_final

        # Calculate standard error of the difference
        se_diff <- sqrt(sd1^2 / n1 + sd2^2 / n2)
        if(all(se_diff > 0)) { # Ensure that no division by zero occurs
          # Calculate Welch's t-statistic
          t_statistic <- (mean1 - mean2) / se_diff

          # Calculate degrees of freedom for Welch's t-test
          num <- (sd1^2 / n1 + sd2^2 / n2)^2
          denom <- ((sd1^2 / n1)^2 / (n1 - 1)) + ((sd2^2 / n2)^2 / (n2 - 1))
          df <- num / denom

          p_value <- 2 * pt(-abs(t_statistic), df)
          bonferroni_adj <- min(p_value * num_comparisons, 1)  # Bonferroni adjustment

          # Store results
          results[[paste(IPTG_conc1, IPTG_conc2, sep = "_vs_")]] <- data.frame(
            IPTG_Conc1 = IPTG_conc1,
            IPTG_Conc2 = IPTG_conc2,
            t_statistic = t_statistic,
            SEM = se_diff,
            degrees_of_freedom = df,
            p_value = p_value,
            Bonferroni_adj = bonferroni_adj
          )
        }
      }
    }
  }

  do.call(rbind, results)
}



# Perform pairwise comparisons within each strain
ab_comparisons <- perform_pairwise_t_tests(combined_aba_stats)
eco_comparisons <- perform_pairwise_t_tests(combined_eco_stats)
```

Welch's t-test results

```
## [1] "A. baumannii stats"
```

```
##               IPTG_Conc1 IPTG_Conc2 t_statistic       SEM degrees_of_freedom
## EV_vs_0              EV          0  -38.535962 0.6354261           115768.7
## EV_vs_7.8            EV        7.8 -120.919732 0.7002993           108923.4
## EV_vs_15.6           EV       15.6 -234.158872 0.7159574           112149.1
## EV_vs_31.3           EV       31.3 -354.480401 0.7428984           107418.5
## EV_vs_62.5           EV       62.5 -473.480580 0.7393979           109171.1
## EV_vs_125            EV        125 -564.340179 0.7219755           114248.8
## EV_vs_250            EV        250 -640.541715 0.7008506           119394.0
## EV_vs_500            EV        500 -628.579929 0.7187622           113328.4
## EV_vs_1000           EV       1000 -680.486600 0.6807012           124496.4
## 0_vs_7.8              0        7.8  -85.368633 0.7050979           111842.8
## 0_vs_15.6             0       15.6 -198.654920 0.7206518           115025.6
## 0_vs_31.3             0       31.3 -319.572688 0.7474235           109985.6
## 0_vs_62.5             0       62.5 -437.672224 0.7439444           111803.0
## 0_vs_125              0        125 -527.025449 0.7266310           117130.0
## 0_vs_250              0        250 -601.487996 0.7056455           122579.0
## 0_vs_500              0        500 -590.669157 0.7234384           116212.2
## 0_vs_1000             0       1000 -639.873991 0.6856370           128004.8
## 7.8_vs_15.6         7.8       15.6 -106.580093 0.7784547           116777.4
## 7.8_vs_31.3         7.8       31.3 -222.410529 0.8033024           114481.1
## 7.8_vs_62.5         7.8       62.5 -331.735719 0.8000663           115757.9
## 7.8_vs_125          7.8        125 -411.686968 0.7839932           118804.7
## 7.8_vs_250          7.8        250 -476.395220 0.7645837           121076.0
## 7.8_vs_500          7.8        500 -470.042240 0.7810351           117877.5
## 7.8_vs_1000         7.8       1000 -507.303218 0.7461574           122698.0
## 15.6_vs_31.3       15.6       31.3 -117.131517 0.8169887           118106.9
## 15.6_vs_62.5       15.6       62.5 -224.184319 0.8138071           119370.7
## 15.6_vs_125        15.6        125 -300.487177 0.7980108           122344.9
## 15.6_vs_250        15.6        250 -361.096356 0.7789507           124489.7
## 15.6_vs_500        15.6        500 -357.376400 0.7951048           121418.1
## 15.6_vs_1000       15.6       1000 -388.449213 0.7608724           125945.7
## 31.3_vs_62.5       31.3       62.5 -103.566096 0.8376067           118121.5
## 31.3_vs_125        31.3        125 -175.243274 0.8222678           120067.3
## 31.3_vs_250        31.3        250 -230.884603 0.8037830           120985.0
## 31.3_vs_500        31.3        500 -229.979968 0.8194478           119151.1
## 31.3_vs_1000       31.3       1000 -254.192139 0.7862759           121190.5
## 62.5_vs_125        62.5        125  -70.014386 0.8191066           121346.5
## 62.5_vs_250        62.5        250 -123.457141 0.8005489           122520.4
## 62.5_vs_500        62.5        500 -124.601194 0.8162757           120427.0
## 62.5_vs_1000       62.5       1000 -144.472436 0.7829694           122975.8
## 125_vs_250          125        250  -52.880794 0.7844857           126557.5
## 125_vs_500          125        500  -55.413016 0.8005282           123443.0
## 125_vs_1000         125       1000  -72.753410 0.7665380           128042.9
## 250_vs_500          250        500   -3.679266 0.7815294           125629.7
## 250_vs_1000         250       1000  -19.130179 0.7466748           132890.9
## 500_vs_1000         500       1000  -14.942223 0.7635122           127121.5
##                     p_value Bonferroni_adj
## EV_vs_0      7.905050e-323  3.557273e-321
## EV_vs_7.8     0.000000e+00   0.000000e+00
## EV_vs_15.6    0.000000e+00   0.000000e+00
## EV_vs_31.3    0.000000e+00   0.000000e+00
## EV_vs_62.5    0.000000e+00   0.000000e+00
## EV_vs_125     0.000000e+00   0.000000e+00
## EV_vs_250     0.000000e+00   0.000000e+00
## EV_vs_500     0.000000e+00   0.000000e+00
## EV_vs_1000    0.000000e+00   0.000000e+00
## 0_vs_7.8      0.000000e+00   0.000000e+00
## 0_vs_15.6     0.000000e+00   0.000000e+00
## 0_vs_31.3     0.000000e+00   0.000000e+00
## 0_vs_62.5     0.000000e+00   0.000000e+00
## 0_vs_125      0.000000e+00   0.000000e+00
## 0_vs_250      0.000000e+00   0.000000e+00
## 0_vs_500      0.000000e+00   0.000000e+00
## 0_vs_1000     0.000000e+00   0.000000e+00
## 7.8_vs_15.6   0.000000e+00   0.000000e+00
## 7.8_vs_31.3   0.000000e+00   0.000000e+00
## 7.8_vs_62.5   0.000000e+00   0.000000e+00
## 7.8_vs_125    0.000000e+00   0.000000e+00
## 7.8_vs_250    0.000000e+00   0.000000e+00
## 7.8_vs_500    0.000000e+00   0.000000e+00
## 7.8_vs_1000   0.000000e+00   0.000000e+00
## 15.6_vs_31.3  0.000000e+00   0.000000e+00
## 15.6_vs_62.5  0.000000e+00   0.000000e+00
## 15.6_vs_125   0.000000e+00   0.000000e+00
## 15.6_vs_250   0.000000e+00   0.000000e+00
## 15.6_vs_500   0.000000e+00   0.000000e+00
## 15.6_vs_1000  0.000000e+00   0.000000e+00
## 31.3_vs_62.5  0.000000e+00   0.000000e+00
## 31.3_vs_125   0.000000e+00   0.000000e+00
## 31.3_vs_250   0.000000e+00   0.000000e+00
## 31.3_vs_500   0.000000e+00   0.000000e+00
## 31.3_vs_1000  0.000000e+00   0.000000e+00
## 62.5_vs_125   0.000000e+00   0.000000e+00
## 62.5_vs_250   0.000000e+00   0.000000e+00
## 62.5_vs_500   0.000000e+00   0.000000e+00
## 62.5_vs_1000  0.000000e+00   0.000000e+00
## 125_vs_250    0.000000e+00   0.000000e+00
## 125_vs_500    0.000000e+00   0.000000e+00
## 125_vs_1000   0.000000e+00   0.000000e+00
## 250_vs_500    2.340042e-04   1.053019e-02
## 250_vs_1000   1.823224e-81   8.204507e-80
## 500_vs_1000   1.932143e-50   8.694643e-49
```

```
## [1] "E. coli stats"
```

```
##              IPTG_Conc1 IPTG_Conc2 t_statistic       SEM degrees_of_freedom
## EV_vs_0              EV          0  -178.20311 0.5569436           147057.8
## EV_vs_7.8            EV        7.8  -243.85654 0.5505618           145396.3
## EV_vs_15.6           EV       15.6  -297.93268 0.5400754           143221.9
## EV_vs_31.3           EV       31.3  -382.31839 0.5364250           141633.5
## EV_vs_62.5           EV       62.5  -475.32972 0.5362329           142086.2
## EV_vs_125            EV        125  -553.09809 0.5380611           143008.1
## EV_vs_250            EV        250  -616.43196 0.5374000           143322.6
## EV_vs_500            EV        500  -625.54052 0.5484089           146284.9
## EV_vs_1000           EV       1000  -649.47506 0.5471762           144445.9
## 0_vs_7.8              0        7.8   -68.09160 0.5141459           162507.4
## 0_vs_15.6             0       15.6  -122.60279 0.5029007           163129.2
## 0_vs_31.3             0       31.3  -212.10549 0.4989784           161988.7
## 0_vs_62.5             0       62.5  -312.04314 0.4987718           162868.4
## 0_vs_125              0        125  -396.11918 0.5007368           163652.7
## 0_vs_250              0        250  -464.01838 0.5000264           164505.1
## 0_vs_500              0        500  -476.32658 0.5118398           164826.7
## 0_vs_1000             0       1000  -501.70182 0.5105188           162250.7
## 7.8_vs_15.6         7.8       15.6   -53.74494 0.4958238           163450.2
## 7.8_vs_31.3         7.8       31.3  -144.00276 0.4918450           162401.6
## 7.8_vs_62.5         7.8       62.5  -245.36332 0.4916355           163347.5
## 7.8_vs_125          7.8        125  -330.90133 0.4936289           164097.9
## 7.8_vs_250          7.8        250  -399.69395 0.4929082           165036.9
## 7.8_vs_500          7.8        500  -413.54480 0.5048882           164898.8
## 7.8_vs_1000         7.8       1000  -439.12156 0.5035490           162225.3
## 15.6_vs_31.3       15.6       31.3   -92.02471 0.4800778           164736.7
## 15.6_vs_62.5       15.6       62.5  -195.85021 0.4798631           165829.1
## 15.6_vs_125        15.6        125  -283.65419 0.4819052           166496.7
## 15.6_vs_250        15.6        250  -354.06504 0.4811670           167629.0
## 15.6_vs_500        15.6        500  -369.14069 0.4934321           166443.7
## 15.6_vs_1000       15.6       1000  -395.21707 0.4920617           163563.9
## 31.3_vs_62.5       31.3       62.5  -104.68141 0.4757508           165078.0
## 31.3_vs_125        31.3        125  -193.62363 0.4778106           165726.5
## 31.3_vs_250        31.3        250  -264.50301 0.4770660           166900.8
## 31.3_vs_500        31.3        500  -281.89067 0.4894339           165488.8
## 31.3_vs_1000       31.3       1000  -307.94278 0.4880523           162567.9
## 62.5_vs_125        62.5        125   -89.43385 0.4775948           166858.2
## 62.5_vs_250        62.5        250  -160.18272 0.4768499           168059.0
## 62.5_vs_500        62.5        500  -180.21336 0.4892233           166499.1
## 62.5_vs_1000       62.5       1000  -205.98902 0.4878411           163550.5
## 125_vs_250          125        250   -70.30619 0.4789049           168686.4
## 125_vs_500          125        500   -92.52641 0.4912265           167214.4
## 125_vs_1000         125       1000  -117.94788 0.4898500           164281.2
## 250_vs_500          250        500   -24.01915 0.4905023           168237.9
## 250_vs_1000         250       1000   -49.28566 0.4891237           165268.4
## 500_vs_1000         500       1000   -24.59195 0.5011941           164872.0
##                    p_value Bonferroni_adj
## EV_vs_0       0.000000e+00   0.000000e+00
## EV_vs_7.8     0.000000e+00   0.000000e+00
## EV_vs_15.6    0.000000e+00   0.000000e+00
## EV_vs_31.3    0.000000e+00   0.000000e+00
## EV_vs_62.5    0.000000e+00   0.000000e+00
## EV_vs_125     0.000000e+00   0.000000e+00
## EV_vs_250     0.000000e+00   0.000000e+00
## EV_vs_500     0.000000e+00   0.000000e+00
## EV_vs_1000    0.000000e+00   0.000000e+00
## 0_vs_7.8      0.000000e+00   0.000000e+00
## 0_vs_15.6     0.000000e+00   0.000000e+00
## 0_vs_31.3     0.000000e+00   0.000000e+00
## 0_vs_62.5     0.000000e+00   0.000000e+00
## 0_vs_125      0.000000e+00   0.000000e+00
## 0_vs_250      0.000000e+00   0.000000e+00
## 0_vs_500      0.000000e+00   0.000000e+00
## 0_vs_1000     0.000000e+00   0.000000e+00
## 7.8_vs_15.6   0.000000e+00   0.000000e+00
## 7.8_vs_31.3   0.000000e+00   0.000000e+00
## 7.8_vs_62.5   0.000000e+00   0.000000e+00
## 7.8_vs_125    0.000000e+00   0.000000e+00
## 7.8_vs_250    0.000000e+00   0.000000e+00
## 7.8_vs_500    0.000000e+00   0.000000e+00
## 7.8_vs_1000   0.000000e+00   0.000000e+00
## 15.6_vs_31.3  0.000000e+00   0.000000e+00
## 15.6_vs_62.5  0.000000e+00   0.000000e+00
## 15.6_vs_125   0.000000e+00   0.000000e+00
## 15.6_vs_250   0.000000e+00   0.000000e+00
## 15.6_vs_500   0.000000e+00   0.000000e+00
## 15.6_vs_1000  0.000000e+00   0.000000e+00
## 31.3_vs_62.5  0.000000e+00   0.000000e+00
## 31.3_vs_125   0.000000e+00   0.000000e+00
## 31.3_vs_250   0.000000e+00   0.000000e+00
## 31.3_vs_500   0.000000e+00   0.000000e+00
## 31.3_vs_1000  0.000000e+00   0.000000e+00
## 62.5_vs_125   0.000000e+00   0.000000e+00
## 62.5_vs_250   0.000000e+00   0.000000e+00
## 62.5_vs_500   0.000000e+00   0.000000e+00
## 62.5_vs_1000  0.000000e+00   0.000000e+00
## 125_vs_250    0.000000e+00   0.000000e+00
## 125_vs_500    0.000000e+00   0.000000e+00
## 125_vs_1000   0.000000e+00   0.000000e+00
## 250_vs_500   2.878744e-127  1.295435e-125
## 250_vs_1000   0.000000e+00   0.000000e+00
## 500_vs_1000  2.683516e-133  1.207582e-131
```

Microscopy image analysis for plasmid expression (Fig S3)

Load all data tables (.csv for each sample) and sample names

```
## Rows: 437
## Columns: 8
## $ Aba20 <dbl> 33.617, 596.428, 543.023, 138.870, 51.583, 36.907, 284.417, 614.~
## $ Aba23 <dbl> 690.347, 1369.399, 730.903, 7.999, 1278.827, 1823.991, 1575.411,~
## $ Aba26 <dbl> 5525.054, 2936.634, 8705.279, 6302.199, 4828.207, 9343.641, 7474~
## $ Aba29 <dbl> 59979.475, 25093.045, 162.864, 21168.394, 43167.129, 20462.820, ~
## $ Eco5  <dbl> 5.401, 14.128, 96.037, 140.865, 173.175, 219.949, 65.093, 13.584~
## $ Eco8  <dbl> 11489.474, 8266.120, 8854.634, 6941.750, 7557.569, 4782.041, 625~
## $ Eco11 <dbl> 12326.367, 4109.619, 13567.807, 6964.584, 1941.373, 15152.469, 1~
## $ Eco14 <dbl> 65351.943, 63789.949, 43536.042, 65437.946, 64335.826, 59819.540~
```

```
##    Sample_name Strain IPTG_conc
## 1:       Aba20    Aba        EV
## 2:       Aba23    Aba         0
## 3:       Aba26    Aba      62.5
## 4:       Aba29    Aba      1000
## 5:        Eco5    Eco        EV
## 6:        Eco8    Eco         0
## 7:       Eco11    Eco      62.5
## 8:       Eco14    Eco      1000
```

Convert data to long format and combine with map

```
# Convert to long format
mic_long <- mic_data %>%
  pivot_longer(cols = everything(), names_to = "Sample_name", values_to = "fluor_int")

# Join with the second dataframe
merged_mic <- mic_long %>%
  left_join(mic_sample_map, by = "Sample_name")

# View the resulting dataframe
head(merged_mic)
```

```
## # A tibble: 6 x 4
##   Sample_name fluor_int Strain IPTG_conc
##   <chr>           <dbl> <chr>  <chr>
## 1 Aba20           33.6  Aba    EV
## 2 Aba23          690.   Aba    0
## 3 Aba26         5525.   Aba    62.5
## 4 Aba29        59979.   Aba    1000
## 5 Eco5             5.40 Eco    EV
## 6 Eco8         11489.   Eco    0
```

Plot the data as Sina plots.

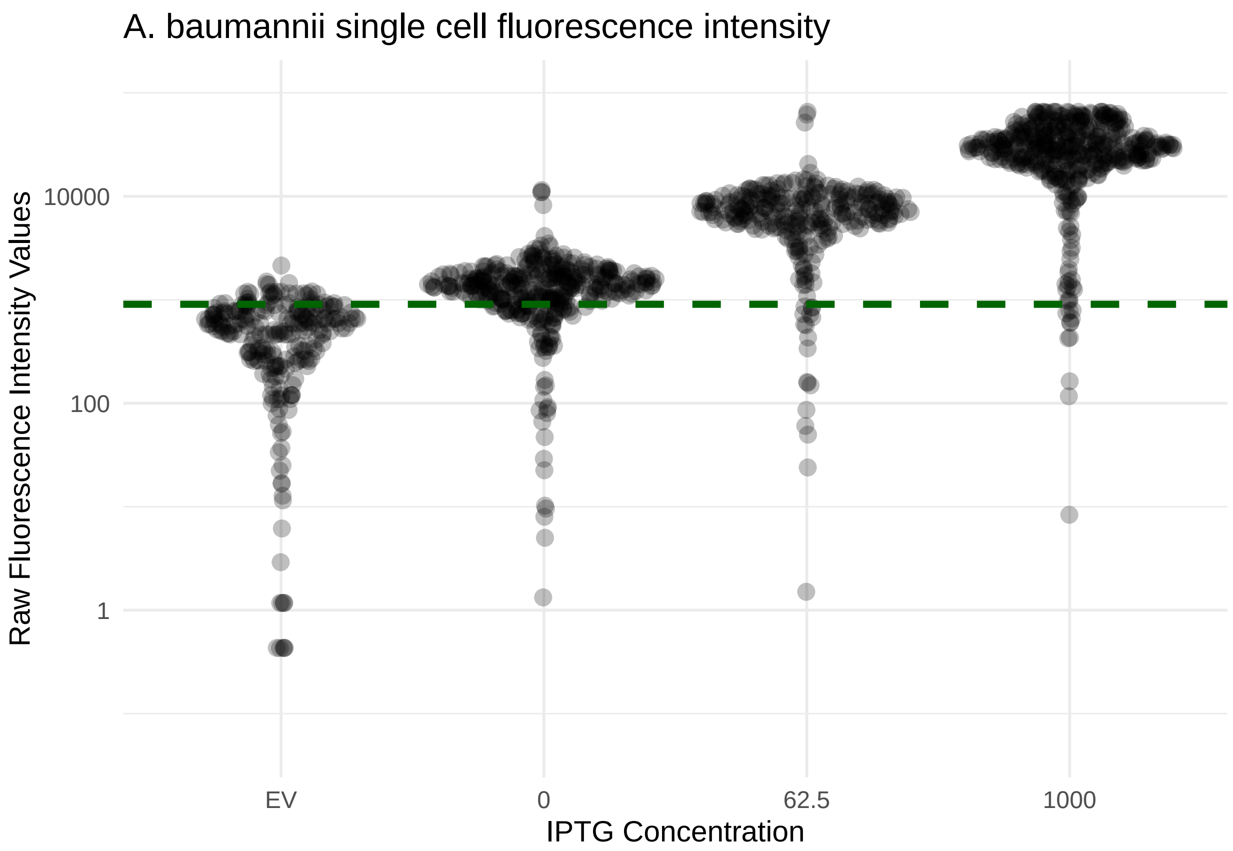
First for A. baumannii.

```
merged_mic <- merged_mic %>%
  mutate(IPTG_conc = factor(IPTG_conc, levels = c("EV", "0", "62.5", "1000")))

aba_data <- merged_mic %>% filter(Strain == "Aba")

# Call fluorescence threshold (mean of EV + 1 SD)
mean_threshold <- aba_data %>% filter(IPTG_conc == "EV") %>%
  summarise(mean_threshold = mean(fluor_int, na.rm = TRUE) +
              sd(fluor_int, na.rm = TRUE)) %>%
  pull(mean_threshold)

# Create the Sina plots
ggplot(aba_data, aes(x = IPTG_conc, y = fluor_int, fill = IPTG_conc)) +
  geom_sina(alpha = 0.25, size = 2.5) +
  geom_hline(yintercept = mean_threshold, linetype = "dashed",
             color = "darkgreen", linewidth=1.25) +
  scale_y_log10(limits = c(0.05, 100000)) +
  theme_minimal() +
  labs(title = "A. baumannii single cell fluorescence intensity",
       x = "IPTG Concentration",
       y = "Raw Fluorescence Intensity Values") +
  theme(legend.position = "none")
```



Calculate percent of cells that are above the mean+SD threshold above background

```
IPTG_percent <- aba_data %>%
  filter(!is.na(fluor_int)) %>%  # Exclude rows where fluor_int is NA
  group_by(IPTG_conc) %>%
  summarise(
    total = n(),
    above_threshold = sum(fluor_int > mean_threshold, na.rm = TRUE),
    percent_above_threshold = above_threshold / total * 100
  )

print(IPTG_percent)
```

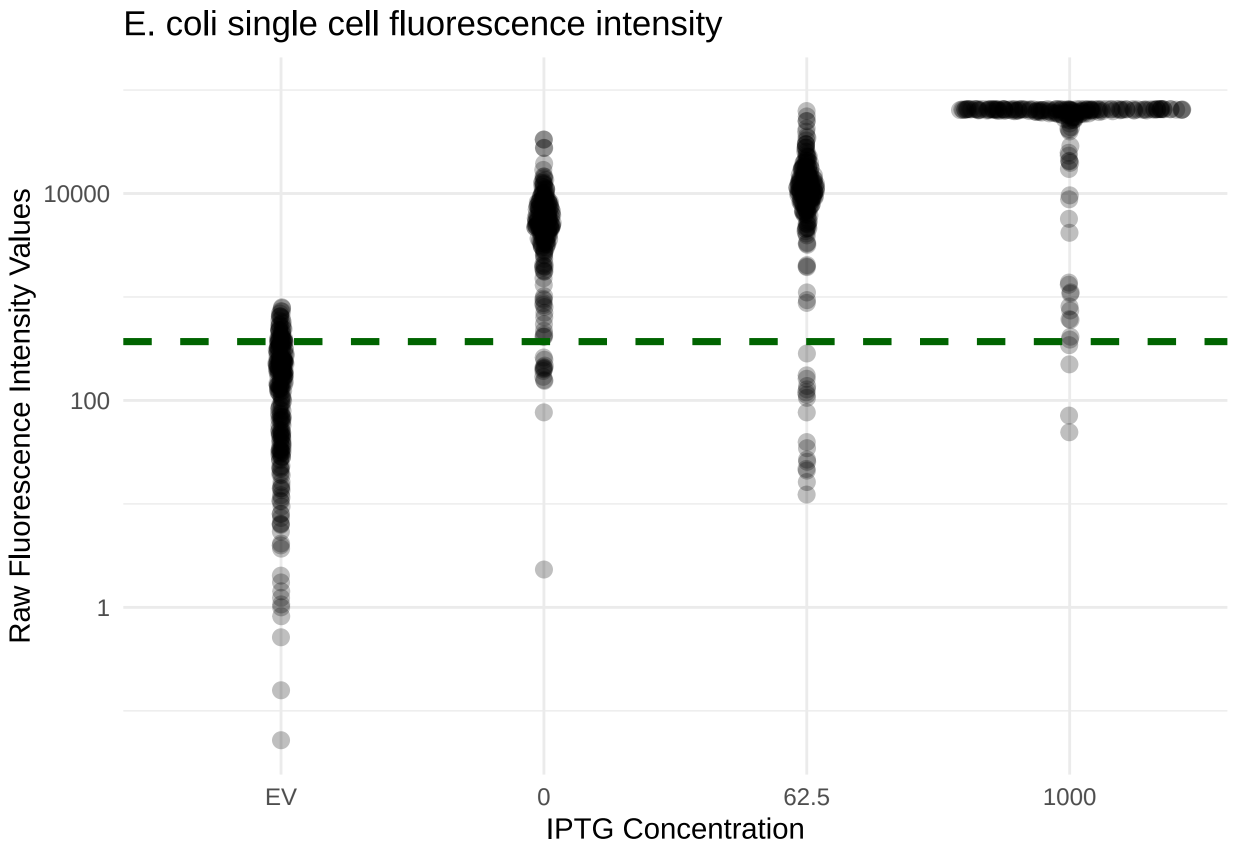
```
## # A tibble: 4 x 4
##   IPTG_conc total above_threshold percent_above_threshold
##   <fct>     <int>           <int>                   <dbl>
## 1 EV          242              34                    14.0
## 2 0           375             292                    77.9
## 3 62.5        289             271                    93.8
## 4 1000        437             426                    97.5
```

Then for E. coli.

```
eco_data <- merged_mic %>% filter(Strain == "Eco")
```

```
# Call fluorescence threshold (mean of EV + 1 SD)
mean_threshold <- eco_data %>% filter(IPTG_conc == "EV") %>%
  summarise(mean_threshold = mean(fluor_int, na.rm = TRUE) +
              sd(fluor_int, na.rm = TRUE)) %>%
  pull(mean_threshold)

# Create the Sina plots
ggplot(eco_data, aes(x = IPTG_conc, y = fluor_int, fill = IPTG_conc)) +
  geom_sina(alpha = 0.25, size = 2.5) +
  geom_hline(yintercept = mean_threshold, linetype = "dashed",
             color = "darkgreen", linewidth=1.25) +
  scale_y_log10(limits = c(0.05, 100000)) +
  theme_minimal() +
  labs(title = "E. coli single cell fluorescence intensity",
       x = "IPTG Concentration",
       y = "Raw Fluorescence Intensity Values") +
  theme(legend.position = "none")
```



Calculate percent of cells that are above the mean+SD threshold above background

```
IPTG_percent <- eco_data %>%
  filter(!is.na(fluor_int)) %>%  # Exclude rows where fluor_int is NA
  group_by(IPTG_conc) %>%
  summarise(
    total = n(),
    above_threshold = sum(fluor_int > mean_threshold, na.rm = TRUE),
    percent_above_threshold = above_threshold / total * 100
  )

print(IPTG_percent)
```

```
## # A tibble: 4 x 4
##   IPTG_conc total above_threshold percent_above_threshold
##   <fct>     <int>           <int>                   <dbl>
## 1 EV          265              41                    15.5
## 2 0           250             237                    94.8
## 3 62.5        231             214                    92.6
## 4 1000        210             206                    98.1
```

And run statistics to determine differences (Welch's t-tests)

```
pairwise_results <- pairwise.t.test(aba_data$fluor_int, aba_data$IPTG_conc,
                                    p.adjust.method = "bonferroni",  # Bonferroni correction
                                    na.action = na.omit)  # Omit NAs
print("A. baumannii stats")
```

```
## [1] "A. baumannii stats"
```

```
print(pairwise_results)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  aba_data$fluor_int and aba_data$IPTG_conc
##
##      EV      0       62.5
## 0    1       -       -
## 62.5 5.2e-16 3.5e-15 -
## 1000 < 2e-16 < 2e-16 < 2e-16
##
## P value adjustment method: bonferroni
```

```
pairwise_results <- pairwise.t.test(eco_data$fluor_int, eco_data$IPTG_conc,
                                    p.adjust.method = "bonferroni",  # Bonferroni correction
                                    na.action = na.omit)  # Omit NAs
print("E. coli stats")
```

```
## [1] "E. coli stats"
```

```
print(pairwise_results)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  eco_data$fluor_int and eco_data$IPTG_conc
##
```

```
##      EV      0       62.5
## 0    1.0e-09 -       -
## 62.5 < 2e-16 5.6e-11 -
## 1000 < 2e-16 < 2e-16 < 2e-16
##
## P value adjustment method: bonferroni
```