Neural_Gene_Expression_Analysis_of_Icelandic_Stickleback_fish_(Optic_

2023-05-07

1. Load packages

```
library(tidyverse)
library(readr)
library(grid)
library(gridExtra)
```

2. Data transformation

```
# Read CSV into R
OT_StickleGene <- read_csv("/stor/work/Bio321G_RY_Spring2023/Exercises/OT_StickleGene.csv")
head(OT_StickleGene, 10)
```

2.a. Import data sets

```
## # A tibble: 10 x 6,901
##    sample_id population sex   turb_combined ENSGACG00000000009
##    <chr>     <chr>      <chr> <chr>                      <dbl>
##  1 ICE094OT2 Blauta     F     high glacial                4.51
##  2 ICE096OT2 Blauta     F     high glacial                4.64
##  3 ICE097OT2 Blauta     M     high glacial                5.06
##  4 ICE098OT1 Blauta     F     high glacial                5.13
##  5 ICE106OT1 Blauta     M     high glacial                4.66
##  6 ICE107OT1 Blauta     F     high glacial                4.70
##  7 ICE108OT1 Blauta     M     high glacial                5.05
##  8 ICE109OT2 Blauta     M     high glacial                4.88
##  9 ICE110OT1 Blauta     M     high glacial                4.91
## 10 ICE115OT1 Pristi     F     high glacial                5.07
## # i 6,896 more variables: ENSGACG00000000013 <dbl>, ENSGACG00000000014 <dbl>,
## #   ENSGACG00000000016 <dbl>, ENSGACG00000000024 <dbl>,
## #   ENSGACG00000000025 <dbl>, ENSGACG00000000027 <dbl>,
## #   ENSGACG00000000037 <dbl>, ENSGACG00000000038 <dbl>,
## #   ENSGACG00000000043 <dbl>, ENSGACG00000000048 <dbl>,
## #   ENSGACG00000000057 <dbl>, ENSGACG00000000061 <dbl>,
## #   ENSGACG00000000065 <dbl>, ENSGACG00000000067 <dbl>, ...
```

2.b. Add origin of each species One piece of data that is not included in the data frame is the hypothesized source population: North America or Europe.

1. North America populations include: Pristi, Galta, LittlaLon, Lon, Hops, Thanga
2. Europe populations include: Frosta and Blauta

```
OT_StickleGene <- OT_StickleGene %>%  # Pipe the OT data set
  mutate(origin = ifelse(population %in% c("Frosta", "Blauta"),  # If TRUE (i.e. population is in the v
                         "Europe",
                         "N_America"),   # If FALSE, origin is North America
         .after = population)  # add the origin column after the population column
```

2.c. Specify the turbidity Regarding the water turbidity, those that live in spring water can be further separate into two categories: High (elevation) Spring or Low (elevation) Spring

1. High Spring: Frosta, Galta
2. Low Spring: Hops, LittlaLon

```
OT_StickleGene$turb_combined[OT_StickleGene$population %in% c("Frosta", "Galta")] <- "high spring"
OT_StickleGene$turb_combined[OT_StickleGene$population %in% c("Hops", "LittlaLon")] <- "low spring"
```

2.d. Add the elevation Another piece of data that is not included in the data frame is the elevation (i.e., height) at which the fish live: High, Mid, Low. This can be classified in accordance to its habitat, specifically the water turbitdity

1. High elevation: High Glacial, High Spring
2. Mid elevation: Low Spring
3. Low elevation: Marine

```
OT_StickleGene <- OT_StickleGene %>%
  mutate(elevation = ifelse(OT_StickleGene$turb_combined %in% c("high glacial", "high spring"),
                            "high",
                            (ifelse(OT_StickleGene$turb_combined == "low spring",   # Nested ifelse()
                                    "mid", "low"))))
```

3. Mean summarized gene expression by population, sex, origin, turbidity, and elevation

Make a new data frame where gene expression is mean summarized by population, sex, turbidity, and elevation.

```
OT_StickleGene_summarize <- OT_StickleGene %>%
  group_by(population, sex, origin, turb_combined, elevation) %>%   # Use group_by() to group the data
  summarise_if(is.numeric, ~mean(.,na.rm = TRUE))    # Use summarise_if() to mean summarized each gene

# (.) is used as a placeholder, which represents the argument/object pass from the left hand side of th
```

```
head(OT_StickleGene_summarize, 20)   # Only return 16 rows because it only has 16 rows
```

```
## # A tibble: 16 x 6,902
## # Groups:   population, sex, origin, turb_combined [16]
##    population sex   origin    turb_combined elevation ENSGACG00000000009
##    <chr>      <chr> <chr>     <chr>         <chr>                  <dbl>
##  1 Blauta     F     Europe    high glacial  high                    4.74
##  2 Blauta     M     Europe    high glacial  high                    4.91
##  3 Frosta     F     Europe    high spring   high                    5.06
##  4 Frosta     M     Europe    high spring   high                    5.12
##  5 Galta      F     N_America high spring   high                    4.67
##  6 Galta      M     N_America high spring   high                    4.88
##  7 Hops       F     N_America low spring    mid                     4.80
##  8 Hops       M     N_America low spring    mid                     4.99
##  9 LittlaLon  F     N_America low spring    mid                     5.15
## 10 LittlaLon  M     N_America low spring    mid                     4.69
## 11 Lon        F     N_America marine        low                     5.23
## 12 Lon        M     N_America marine        low                     4.91
## 13 Pristi     F     N_America high glacial  high                    5.34
## 14 Pristi     M     N_America high glacial  high                    4.88
## 15 Thanga     F     N_America marine        low                     5.22
## 16 Thanga     M     N_America marine        low                     4.99
## # i 6,896 more variables: ENSGACG00000000013 <dbl>, ENSGACG00000000014 <dbl>,
## #   ENSGACG00000000016 <dbl>, ENSGACG00000000024 <dbl>,
## #   ENSGACG00000000025 <dbl>, ENSGACG00000000027 <dbl>,
## #   ENSGACG00000000037 <dbl>, ENSGACG00000000038 <dbl>,
## #   ENSGACG00000000043 <dbl>, ENSGACG00000000048 <dbl>,
## #   ENSGACG00000000057 <dbl>, ENSGACG00000000061 <dbl>,
## #   ENSGACG00000000065 <dbl>, ENSGACG00000000067 <dbl>, ...
```

As you can see, the data frame is now reduced to 16 rows (i.e., 16 observations) because we've group the data set by 8 species, each species have 2 sex.

4. Make a metadata data frame

It is often useful to have a metadata data frame to aid analysis. Let's make a data frame that does not contain any gene expression columns (i.e., only `sample_id`, `population`, `sex`, `origin`, `turb_combined`, and `elevation`)

```
OT_StickleGene_metadata <- OT_StickleGene %>%
  dplyr::select(sample_id, population, sex, origin, turb_combined, elevation)

head(OT_StickleGene_metadata, 10)
```

```
## # A tibble: 10 x 6
##    sample_id population sex   origin turb_combined elevation
##    <chr>     <chr>      <chr> <chr>  <chr>         <chr>
##  1 ICE094OT2 Blauta     F     Europe high glacial  high
##  2 ICE096OT2 Blauta     F     Europe high glacial  high
##  3 ICE097OT2 Blauta     M     Europe high glacial  high
##  4 ICE098OT1 Blauta     F     Europe high glacial  high
##  5 ICE106OT1 Blauta     M     Europe high glacial  high
##  6 ICE107OT1 Blauta     F     Europe    high glacial  high
##  7 ICE108OT1 Blauta     M     Europe    high glacial  high
##  8 ICE109OT2 Blauta     M     Europe    high glacial  high
##  9 ICE110OT1 Blauta     M     Europe    high glacial  high
## 10 ICE115OT1 Pristi     F     N_America high glacial  high
```

5. Filter high expression genes

For this task, we can approach it in two different ways

1. Calculate the mean expression across all species and sex (i.e., 16 combinations) and select the same number as the second method (which is 18)
2. As introduced by Dr. Rebecca L. Young, we can choose an arbitrary high value (in this case, 12) and select the gene expression columns which have the maximum value is greater than 12

```
OT_high_mean_expression_genes <- colMeans(OT_StickleGene_summarize[,7:ncol(OT_StickleGene_summarize)])
OT_high_mean_expression_genes <- tail(sort(OT_high_mean_expression_genes), 18)
OT_high_mean_expression_genes <- as.data.frame(OT_high_mean_expression_genes)
OT_high_mean_expression_genes <- rownames(OT_high_mean_expression_genes)  # Return a list of genes
OT_high_mean_expression_genes <- OT_StickleGene %>%
  dplyr::select(c("sample_id", "population", "sex", "turb_combined", "elevation", OT_high_mean_expressi


# Join with OT_StickleGene_metadata
OT_high_mean_expression_genes <- right_join(OT_StickleGene_metadata, OT_high_mean_expression_genes)

head(OT_high_mean_expression_genes, 10)
```

5.a. Mean expression across species and sex

```
## # A tibble: 10 x 24
##    sample_id population sex   origin  turb_combined elevation ENSGACG00000010148
##    <chr>     <chr>      <chr> <chr>   <chr>         <chr>                  <dbl>
##  1 ICE094OT2 Blauta     F     Europe  high glacial  high                    9.56
##  2 ICE096OT2 Blauta     F     Europe  high glacial  high                   10.7
##  3 ICE097OT2 Blauta     M     Europe  high glacial  high                   10.3
##  4 ICE098OT1 Blauta     F     Europe  high glacial  high                   10.3
##  5 ICE106OT1 Blauta     M     Europe  high glacial  high                   10.2
##  6 ICE107OT1 Blauta     F     Europe  high glacial  high                   10.2
##  7 ICE108OT1 Blauta     M     Europe  high glacial  high                   11.2
##  8 ICE109OT2 Blauta     M     Europe  high glacial  high                    9.29
##  9 ICE110OT1 Blauta     M     Europe  high glacial  high                    9.26
## 10 ICE115OT1 Pristi     F     N_Amer~ high glacial  high                   10.1
## # i 17 more variables: ENSGACG00000005112 <dbl>, ENSGACG00000020925 <dbl>,
## #   ENSGACG00000013530 <dbl>, ENSGACG00000020947 <dbl>,
## #   ENSGACG00000004758 <dbl>, ENSGACG00000012607 <dbl>,
## #   ENSGACG00000013716 <dbl>, ENSGACG00000009520 <dbl>,
## #   ENSGACG00000012080 <dbl>, ENSGACG00000020954 <dbl>,
## #   ENSGACG00000015622 <dbl>, ENSGACG00000013415 <dbl>,
## #   ENSGACG00000005864 <dbl>, ENSGACG00000020941 <dbl>, ...
```

```
OT_high_max12_expression_genes <- OT_StickleGene %>%
  column_to_rownames("sample_id") %>%
  select_if(is.numeric) %>%
  select_if(~ max(., na.rm = TRUE) > 12) %>%   # Tilde operator (~)
  rownames_to_column("sample_id")  # Add the sample_id column back. This basically "undo" the first com


OT_high_max12_expression_genes <- right_join(OT_StickleGene_metadata, OT_high_max12_expression_genes)

head(OT_high_max12_expression_genes, 10)
```

5.b. Gene expression columns which have the maximum value is greater than 12

```
## # A tibble: 10 x 24
##    sample_id population sex   origin  turb_combined elevation ENSGACG00000003467
##    <chr>     <chr>      <chr> <chr>   <chr>         <chr>                  <dbl>
##  1 ICE094OT2 Blauta     F     Europe  high glacial  high                    7.81
##  2 ICE096OT2 Blauta     F     Europe  high glacial  high                    5.50
##  3 ICE097OT2 Blauta     M     Europe  high glacial  high                   10.5
##  4 ICE098OT1 Blauta     F     Europe  high glacial  high                    4.52
##  5 ICE106OT1 Blauta     M     Europe  high glacial  high                    6.44
##  6 ICE107OT1 Blauta     F     Europe  high glacial  high                    3.69
##  7 ICE108OT1 Blauta     M     Europe  high glacial  high                    4.90
##  8 ICE109OT2 Blauta     M     Europe  high glacial  high                    5.61
##  9 ICE110OT1 Blauta     M     Europe  high glacial  high                    5.40
## 10 ICE115OT1 Pristi     F     N_Amer~ high glacial  high                    7.83
## # i 17 more variables: ENSGACG00000006034 <dbl>, ENSGACG00000006710 <dbl>,
## #   ENSGACG00000006713 <dbl>, ENSGACG00000013533 <dbl>,
## #   ENSGACG00000014492 <dbl>, ENSGACG00000015622 <dbl>,
## #   ENSGACG00000020365 <dbl>, ENSGACG00000020371 <dbl>,
## #   ENSGACG00000020925 <dbl>, ENSGACG00000020929 <dbl>,
## #   ENSGACG00000020935 <dbl>, ENSGACG00000020938 <dbl>,
## #   ENSGACG00000020941 <dbl>, ENSGACG00000020942 <dbl>, ...
```

Among the 18 genes, there are 8 common genes (almost half) between the two methods - the genes that ended with 15622, 20925, 20935, 20938, 20941, 20942, 20947, 20954.

6. Data transformation for visualization

This process can be carry out in two steps

1. Mean-summarize the gene expression
2. Transform the data set from wide to long format

```
# Mean-summarize the gene expression
OT_high_mean_expression_genes <- OT_high_mean_expression_genes %>%
  group_by(population, sex, origin, turb_combined, elevation) %>%
  summarise_if(is.numeric, ~ mean(., na.rm = TRUE))

# Transform the data from wide to long format
OT_high_mean_expression_genes <- OT_high_mean_expression_genes %>%
  pivot_longer(cols = starts_with("ENS"),
               names_to = "gene_id",       # Name the now-flipped column "gene_id"
               values_to = "expression")   # Name the column of corresponding value of each gene to "exp

head(OT_high_mean_expression_genes, 10)
```

```
## # A tibble: 10 x 7
## # Groups:   population, sex, origin, turb_combined [1]
##    population sex   origin turb_combined elevation gene_id            expression
##    <chr>      <chr> <chr>  <chr>         <chr>     <chr>                   <dbl>
##  1 Blauta     F     Europe high glacial  high      ENSGACG00000010148      10.2
##  2 Blauta     F     Europe high glacial  high      ENSGACG00000005112      10.2
##  3 Blauta     F     Europe high glacial  high      ENSGACG00000020925      10.2
##  4 Blauta     F     Europe high glacial  high      ENSGACG00000013530       9.98
##  5 Blauta     F     Europe high glacial  high      ENSGACG00000020947      10.4
##  6 Blauta     F     Europe high glacial  high      ENSGACG00000004758      10.4
##  7 Blauta     F     Europe high glacial  high      ENSGACG00000012607      10.6
##  8 Blauta     F     Europe high glacial  high      ENSGACG00000013716      10.6
##  9 Blauta     F     Europe high glacial  high      ENSGACG00000009520      10.6
## 10 Blauta     F     Europe high glacial  high      ENSGACG00000012080      11.0
```

```
# Rinse and repeat for the second method
OT_high_max12_expression_genes <- OT_high_max12_expression_genes %>%
  group_by(population, sex, origin, turb_combined, elevation) %>%
  summarise_if(is.numeric, ~ mean(., na.rm = TRUE))

OT_high_max12_expression_genes <- OT_high_max12_expression_genes %>%
  pivot_longer(cols = starts_with("ENS"),
               names_to = "gene_id",
               values_to = "expression")

head(OT_high_max12_expression_genes, 10)
```

```
## # A tibble: 10 x 7
## # Groups:   population, sex, origin, turb_combined [1]
##    population sex   origin turb_combined elevation gene_id            expression
##    <chr>      <chr> <chr>  <chr>         <chr>     <chr>                   <dbl>
##  1 Blauta     F     Europe high glacial  high      ENSGACG00000003467       5.38
##  2 Blauta     F     Europe high glacial  high      ENSGACG00000006034       5.83
##  3 Blauta     F     Europe high glacial  high      ENSGACG00000006710       8.49
##  4 Blauta     F     Europe high glacial  high      ENSGACG00000006713       6.65
##  5 Blauta     F     Europe high glacial  high      ENSGACG00000013533       6.35
##  6 Blauta     F     Europe high glacial  high      ENSGACG00000014492       9.30
##  7 Blauta     F     Europe high glacial  high      ENSGACG00000015622      11.0
##  8 Blauta     F     Europe high glacial  high      ENSGACG00000020365       7.16
##  9 Blauta     F     Europe high glacial  high      ENSGACG00000020371       9.81
## 10 Blauta     F     Europe high glacial  high      ENSGACG00000020925      10.2
```

7. Data visualization

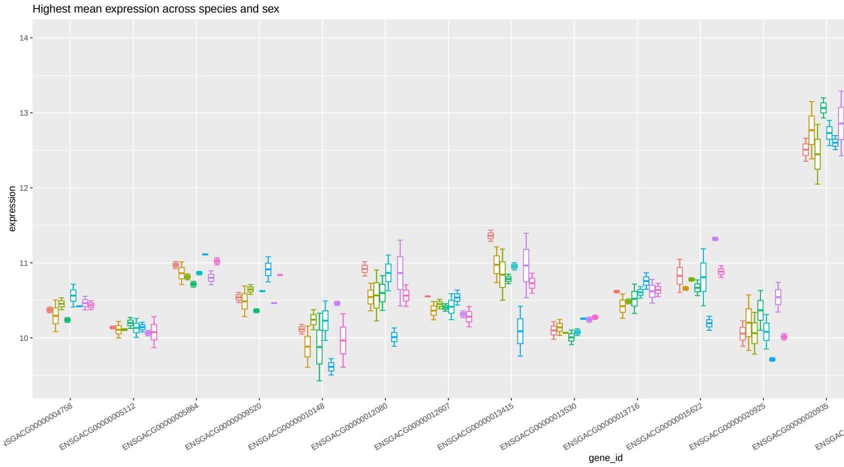
Using our metadata, we can determine which factor help differentiate the most in terms of the Stickleback fish sensory requirement

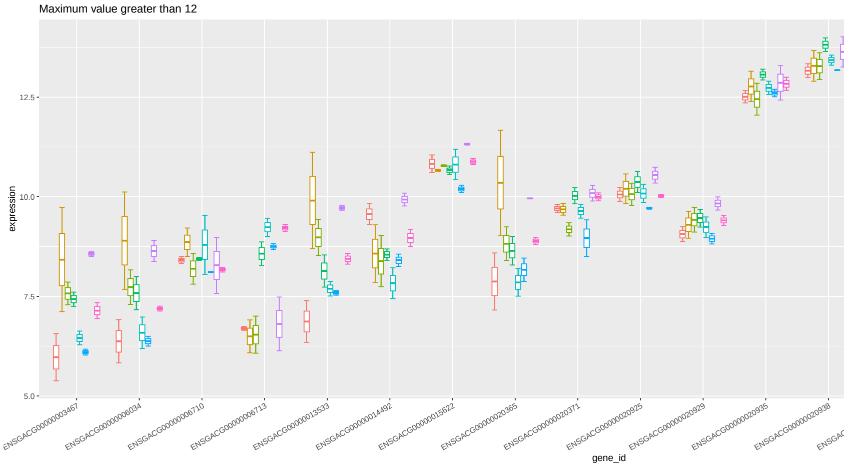
1. By `population`
2. By `sex`
3. By `origin`
4. By `elevation`
5. By `turb_combined`

```r
boxplot_mean_population <- OT_high_mean_expression_genes %>% ggplot(aes(x = gene_id,
                                                                        y = expression,
                                                                        col = population)) +
  stat_boxplot(geom = "errorbar") +   # Add wiskers to the box plot
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Highest mean expression across species and sex")


boxplot_max12_population <- OT_high_max12_expression_genes %>% ggplot(aes(x = gene_id,
                                                                          y = expression,
                                                                          col = population)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Maximum value greater than 12")


# Set fig.height = 15, fig.width = 20
grid.arrange(boxplot_mean_population, boxplot_max12_population, ncol = 1)
```





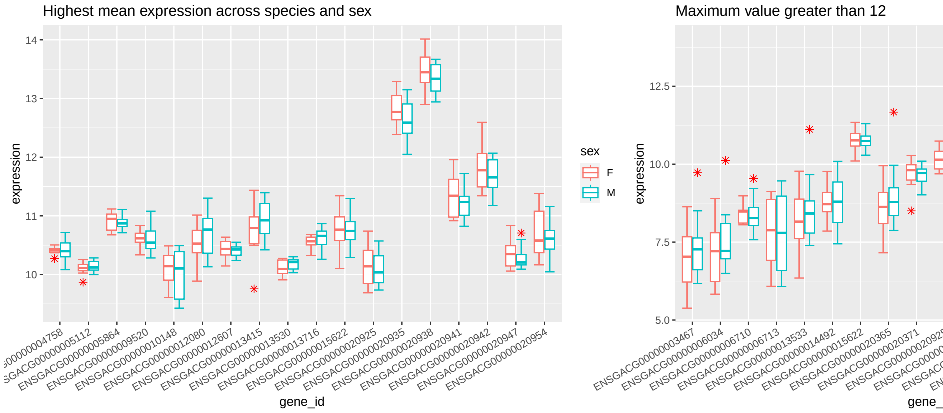
7.a. Gene expression analysis by population

We can see there's some clear separation, particularly coming from the second method (in comparison to the first method) in the left-half of the boxplot.

```
boxplot_mean_sex <- OT_high_mean_expression_genes %>% ggplot(aes(x = gene_id,
                                                                 y = expression,
                                                                 col = sex)) +
  stat_boxplot(geom = "errorbar") +    # Add wiskers to the box plot
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Highest mean expression across species and sex")


boxplot_max12_sex <- OT_high_max12_expression_genes %>% ggplot(aes(x = gene_id,
                                                                   y = expression,
                                                                   col = sex)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Maximum value greater than 12")


# Set fig.height = 5, fig.width = 15
grid.arrange(boxplot_mean_sex, boxplot_max12_sex, ncol = 2)
```



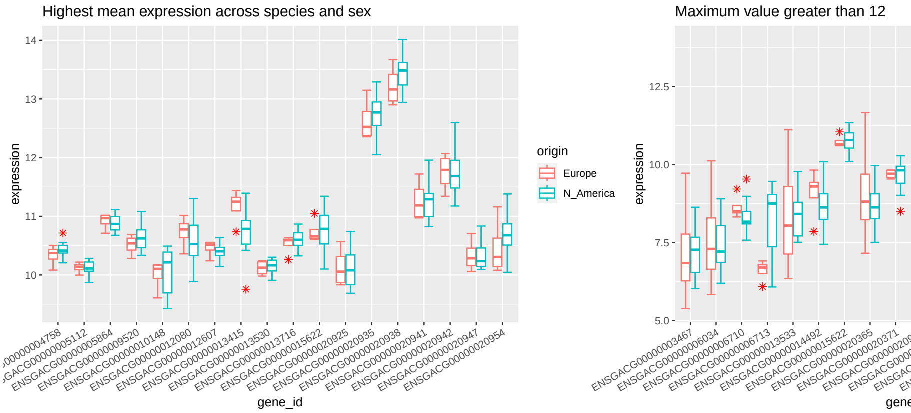
7.b. Gene expression analysis by sex

We can see that the distribution of the genes differentiated on sex basis doesn't provide a clear separation as the gene expression level is similar

```
boxplot_mean_origin <- OT_high_mean_expression_genes %>% ggplot(aes(x = gene_id,
                                                                    y = expression,
                                                                    col = origin)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Highest mean expression across species and sex")


boxplot_max12_origin <- OT_high_max12_expression_genes %>% ggplot(aes(x = gene_id,
                                                                      y = expression,
                                                                      col = origin)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Maximum value greater than 12")
```

```r
grid.arrange(boxplot_mean_origin, boxplot_max12_origin, ncol = 2)
```



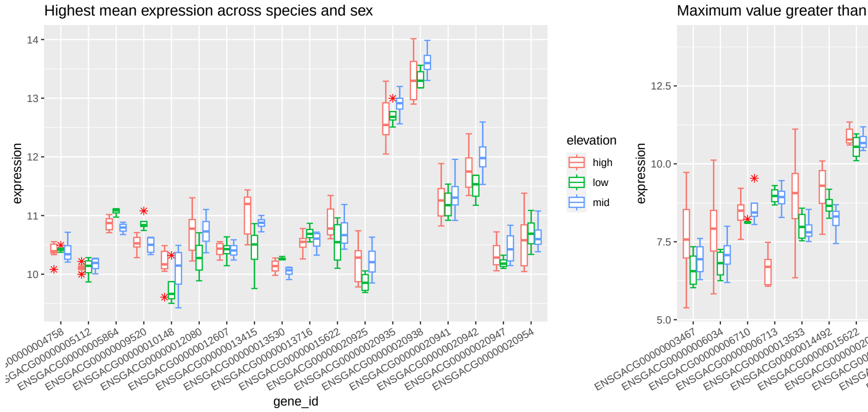
7.c. Gene expression analysis by origin

Here, we can see a better distinction as there's a more noticeable difference in the gene expression level in some genes such as ENSGACG00000012080, ENSGACG00000013415, and ENSGACG00000006713

```r
boxplot_mean_elevation <- OT_high_mean_expression_genes %>% ggplot(aes(x = gene_id,
                                                                        y = expression,
                                                                        col = elevation)) +
  stat_boxplot(geom = "errorbar") +    # Add wiskers to the box plot
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Highest mean expression across species and sex")


boxplot_max12_elevation <- OT_high_max12_expression_genes %>% ggplot(aes(x = gene_id,
                                                                          y = expression,
                                                                          col = elevation)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Maximum value greater than 12")


grid.arrange(boxplot_mean_elevation, boxplot_max12_elevation, ncol = 2)
```



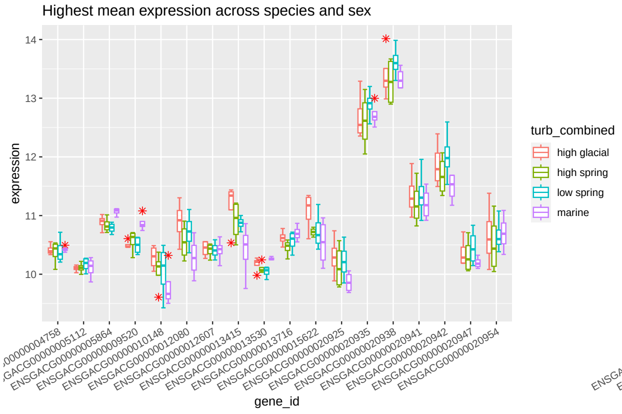
7.d. Gene expression analysis by elevation

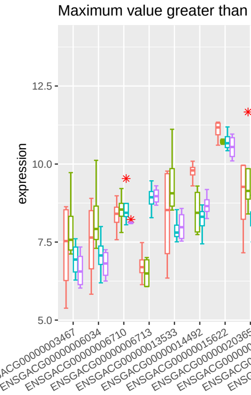
We can see there's some clear separation, again particularly coming from the second method (in comparison to the first method). Overall, we can see that the predominant pattern is that the high elevation has the highest expression level, follow by the mid and low.

```
boxplot_mean_turbidity <- OT_high_mean_expression_genes %>% ggplot(aes(x = gene_id,
                                                                       y = expression,
                                                                       col = turb_combined)) +
  stat_boxplot(geom = "errorbar") +    # Add wiskers to the box plot
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Highest mean expression across species and sex")


boxplot_max12_turbidity <- OT_high_max12_expression_genes %>% ggplot(aes(x = gene_id,
                                                                         y = expression,
                                                                         col = turb_combined)) +
  stat_boxplot(geom = "errorbar") +
  geom_boxplot(outlier.colour = "red",
               outlier.shape = 8,
               outlier.size = 2) +
  theme(axis.text.x = element_text(angle = 30, hjust=1)) +
  labs(title = "Maximum value greater than 12")


grid.arrange(boxplot_mean_turbidity, boxplot_max12_turbidity, ncol = 2)
```





7.e. Gene expression analysis by turbidity

The general pattern can be deduced that high glacial and high spring turbidity has higher expression level than that of low spring and marine.

Given the visualization and analysis, we can see that the Icelandic Stickleback fish's Optic Tectum gene expression level can be best differentiated by the two factors: elevation and water turbidity - fish that live at higher elevation in glacial lake or spring will likely have higher gene expression for certain genes in the Optic Tectum in comparison with those living at lower elevation in spring or marine environment. This can be explain by the murky water at higher elevation in glacial lake and high spring, which requires better sensory for the Stickleback to navigate the environment.