Neural_Gene_Expression_Analysis_of_Icelandic_Stickleback_fish_(Torus_

2023-05-07

1. Load packages

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

2. Data transformation

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

2.a. Import data sets

```
## # A tibble: 10 x 5,923
##    sample_id population sex   turb_combined ENSGACG00000000009
##    <chr>     <chr>      <chr> <chr>                      <dbl>
##  1 ICE096TS1 Blauta     F     high glacial                5.34
##  2 ICE098TS1 Blauta     F     high glacial                5.73
##  3 ICE106TS1 Blauta     M     high glacial                5.41
##  4 ICE107TS1 Blauta     F     high glacial                5.32
##  5 ICE108TS1 Blauta     M     high glacial                4.31
##  6 ICE110TS1 Blauta     M     high glacial                5.22
##  7 ICE115TS1 Pristi     F     high glacial                5.26
##  8 ICE119TS1 Pristi     F     high glacial                5.31
##  9 ICE120TS1 Pristi     M     high glacial                5.11
## 10 ICE124TS1 Pristi     F     high glacial                4.98
## # i 5,918 more variables: ENSGACG00000000016 <dbl>, ENSGACG00000000024 <dbl>,
## #   ENSGACG00000000025 <dbl>, ENSGACG00000000031 <dbl>,
## #   ENSGACG00000000037 <dbl>, ENSGACG00000000038 <dbl>,
## #   ENSGACG00000000043 <dbl>, ENSGACG00000000048 <dbl>,
## #   ENSGACG00000000057 <dbl>, ENSGACG00000000061 <dbl>,
## #   ENSGACG00000000065 <dbl>, ENSGACG00000000067 <dbl>,
## #   ENSGACG00000000074 <dbl>, ENSGACG00000000075 <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

```
TS_StickleGene <- TS_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

```
TS_StickleGene$turb_combined[TS_StickleGene$population %in% c("Frosta", "Galta")] <- "high spring"
TS_StickleGene$turb_combined[TS_StickleGene$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

```
TS_StickleGene <- TS_StickleGene %>%
  mutate(elevation = ifelse(TS_StickleGene$turb_combined %in% c("high glacial", "high spring"),
                            "high",
                            (ifelse(TS_StickleGene$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.

```
TS_StickleGene_summarize <- TS_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(TS_StickleGene_summarize, 20)   # Only return 16 rows because it only has 16 rows
```

```
## # A tibble: 16 x 5,924
## # 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                     5.46
##  2 Blauta     M     Europe    high glacial  high                     4.98
##  3 Frosta     F     Europe    high spring   high                     5.02
##  4 Frosta     M     Europe    high spring   high                     5.07
##  5 Galta      F     N_America high spring   high                     5.65
##  6 Galta      M     N_America high spring   high                     4.05
##  7 Hops       F     N_America low spring    mid                      4.99
##  8 Hops       M     N_America low spring    mid                      5.24
##  9 LittlaLon  F     N_America low spring    mid                      5.29
## 10 LittlaLon  M     N_America low spring    mid                      5.07
## 11 Lon        F     N_America marine        low                      5.54
## 12 Lon        M     N_America marine        low                      5.03
## 13 Pristi     F     N_America high glacial  high                     5.18
## 14 Pristi     M     N_America high glacial  high                     5.14
## 15 Thanga     F     N_America marine        low                      5.10
## 16 Thanga     M     N_America marine        low                      5.16
## # i 5,918 more variables: ENSGACG00000000016 <dbl>, ENSGACG00000000024 <dbl>,
## #   ENSGACG00000000025 <dbl>, ENSGACG00000000031 <dbl>,
## #   ENSGACG00000000037 <dbl>, ENSGACG00000000038 <dbl>,
## #   ENSGACG00000000043 <dbl>, ENSGACG00000000048 <dbl>,
## #   ENSGACG00000000057 <dbl>, ENSGACG00000000061 <dbl>,
## #   ENSGACG00000000065 <dbl>, ENSGACG00000000067 <dbl>,
## #   ENSGACG00000000074 <dbl>, ENSGACG00000000075 <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`)

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

head(TS_StickleGene_metadata, 10)
```

```
## # A tibble: 10 x 6
##    sample_id  population sex   origin    turb_combined elevation
##    <chr>      <chr>      <chr> <chr>     <chr>         <chr>
##  1 ICE096TS1  Blauta     F     Europe    high glacial  high
##  2 ICE098TS1  Blauta     F     Europe    high glacial  high
##  3 ICE106TS1  Blauta     M     Europe    high glacial  high
##  4 ICE107TS1  Blauta     F     Europe    high glacial  high
##  5 ICE108TS1  Blauta     M     Europe    high glacial  high
```

```
##  6 ICE110TS1 Blauta     M     Europe    high glacial  high
##  7 ICE115TS1 Pristi     F     N_America high glacial  high
##  8 ICE119TS1 Pristi     F     N_America high glacial  high
##  9 ICE120TS1 Pristi     M     N_America high glacial  high
## 10 ICE124TS1 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 10)
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

```
TS_high_mean_expression_genes <- colMeans(TS_StickleGene_summarize[,7:ncol(TS_StickleGene_summarize)])
TS_high_mean_expression_genes <- tail(sort(TS_high_mean_expression_genes), 10)
TS_high_mean_expression_genes <- as.data.frame(TS_high_mean_expression_genes)
TS_high_mean_expression_genes <- rownames(TS_high_mean_expression_genes)  # Return a list of genes
TS_high_mean_expression_genes <- TS_StickleGene %>%
  dplyr::select(c("sample_id", "population", "sex", "turb_combined", "elevation", TS_high_mean_expressio


# Join with TS_StickleGene_metadata
TS_high_mean_expression_genes <- right_join(TS_StickleGene_metadata, TS_high_mean_expression_genes)

head(TS_high_mean_expression_genes, 10)
```

5.a. Mean expression across species and sex

```
## # A tibble: 10 x 16
##    sample_id population sex   origin  turb_combined elevation ENSGACG00000015409
##    <chr>     <chr>      <chr> <chr>   <chr>         <chr>                   <dbl>
##  1 ICE096TS1 Blauta     F     Europe  high glacial  high                    11.9
##  2 ICE098TS1 Blauta     F     Europe  high glacial  high                    12.1
##  3 ICE106TS1 Blauta     M     Europe  high glacial  high                    13.0
##  4 ICE107TS1 Blauta     F     Europe  high glacial  high                    11.3
##  5 ICE108TS1 Blauta     M     Europe  high glacial  high                    12.8
##  6 ICE110TS1 Blauta     M     Europe  high glacial  high                    12.6
##  7 ICE115TS1 Pristi     F     N_Amer~ high glacial  high                     9.40
##  8 ICE119TS1 Pristi     F     N_Amer~ high glacial  high                    10.1
##  9 ICE120TS1 Pristi     M     N_Amer~ high glacial  high                     9.80
## 10 ICE124TS1 Pristi     F     N_Amer~ high glacial  high                    10.1
## # i 9 more variables: ENSGACG00000013716 <dbl>, ENSGACG00000017217 <dbl>,
## #   ENSGACG00000009520 <dbl>, ENSGACG00000020371 <dbl>,
## #   ENSGACG00000005864 <dbl>, ENSGACG00000020941 <dbl>,
## #   ENSGACG00000020942 <dbl>, ENSGACG00000020935 <dbl>,
## #   ENSGACG00000020938 <dbl>
```

```
TS_high_max12_expression_genes <- TS_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


TS_high_max12_expression_genes <- right_join(TS_StickleGene_metadata, TS_high_max12_expression_genes)

head(TS_high_max12_expression_genes, 10)
```

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

```
## # A tibble: 10 x 16
##    sample_id population sex   origin  turb_combined elevation ENSGACG00000006710
##    <chr>     <chr>      <chr> <chr>   <chr>         <chr>                   <dbl>
##  1 ICE096TS1 Blauta     F     Europe  high glacial  high                     9.48
##  2 ICE098TS1 Blauta     F     Europe  high glacial  high                     9.29
##  3 ICE106TS1 Blauta     M     Europe  high glacial  high                     8.79
##  4 ICE107TS1 Blauta     F     Europe  high glacial  high                     9.16
##  5 ICE108TS1 Blauta     M     Europe  high glacial  high                     9.11
##  6 ICE110TS1 Blauta     M     Europe  high glacial  high                     8.77
##  7 ICE115TS1 Pristi     F     N_Amer~ high glacial  high                    10.9
##  8 ICE119TS1 Pristi     F     N_Amer~ high glacial  high                     9.80
##  9 ICE120TS1 Pristi     M     N_Amer~ high glacial  high                     7.04
## 10 ICE124TS1 Pristi     F     N_Amer~ high glacial  high                     7.72
## # i 9 more variables: ENSGACG00000015409 <dbl>, ENSGACG00000017217 <dbl>,
## #   ENSGACG00000020365 <dbl>, ENSGACG00000020371 <dbl>,
## #   ENSGACG00000020935 <dbl>, ENSGACG00000020938 <dbl>,
## #   ENSGACG00000020941 <dbl>, ENSGACG00000020942 <dbl>,
## #   ENSGACG00000020954 <dbl>
```

Among the 10 genes, there are 7 common genes(70%) between the two methods - the genes that ended with 15049, 17217, 20371, 20935, 20938, 20941, 20942.

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
TS_high_mean_expression_genes <- TS_high_mean_expression_genes %>%
  group_by(population, sex, origin, turb_combined, elevation) %>%
  summarise_if(is.numeric, ~ mean(., na.rm = TRUE))
```

```r
# Transform the data from wide to long format
TS_high_mean_expression_genes <- TS_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(TS_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      ENSGACG00000015409       11.7
##  2 Blauta     F     Europe high glacial  high      ENSGACG00000013716       10.9
##  3 Blauta     F     Europe high glacial  high      ENSGACG00000017217       12.1
##  4 Blauta     F     Europe high glacial  high      ENSGACG00000009520       11.0
##  5 Blauta     F     Europe high glacial  high      ENSGACG00000020371       11.8
##  6 Blauta     F     Europe high glacial  high      ENSGACG00000005864       11.5
##  7 Blauta     F     Europe high glacial  high      ENSGACG00000020941       11.9
##  8 Blauta     F     Europe high glacial  high      ENSGACG00000020942       12.6
##  9 Blauta     F     Europe high glacial  high      ENSGACG00000020935       13.5
## 10 Blauta     F     Europe high glacial  high      ENSGACG00000020938       14.0
```

```r
# Rinse and repeat for the second method
TS_high_max12_expression_genes <- TS_high_max12_expression_genes %>%
  group_by(population, sex, origin, turb_combined, elevation) %>%
  summarise_if(is.numeric, ~ mean(., na.rm = TRUE))

TS_high_max12_expression_genes <- TS_high_max12_expression_genes %>%
  pivot_longer(cols = starts_with("ENS"),
               names_to = "gene_id",
               values_to = "expression")

head(TS_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      ENSGACG00000006710       9.31
##  2 Blauta     F     Europe high glacial  high      ENSGACG00000015409      11.7
##  3 Blauta     F     Europe high glacial  high      ENSGACG00000017217      12.1
##  4 Blauta     F     Europe high glacial  high      ENSGACG00000020365       5.26
##  5 Blauta     F     Europe high glacial  high      ENSGACG00000020371      11.8
##  6 Blauta     F     Europe high glacial  high      ENSGACG00000020935      13.5
##  7 Blauta     F     Europe high glacial  high      ENSGACG00000020938      14.0
##  8 Blauta     F     Europe high glacial  high      ENSGACG00000020941      11.9
##  9 Blauta     F     Europe high glacial  high      ENSGACG00000020942      12.6
## 10 Blauta     F     Europe high glacial  high      ENSGACG00000020954      11.0
```

7. Data visualization

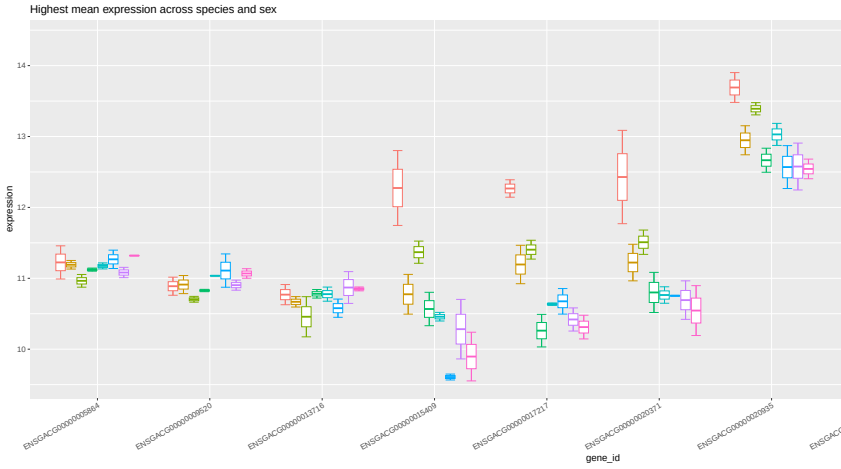
Using our metadata, we can determine which factor help differentiate the most in terms of the Stickleback fish sensory requirement

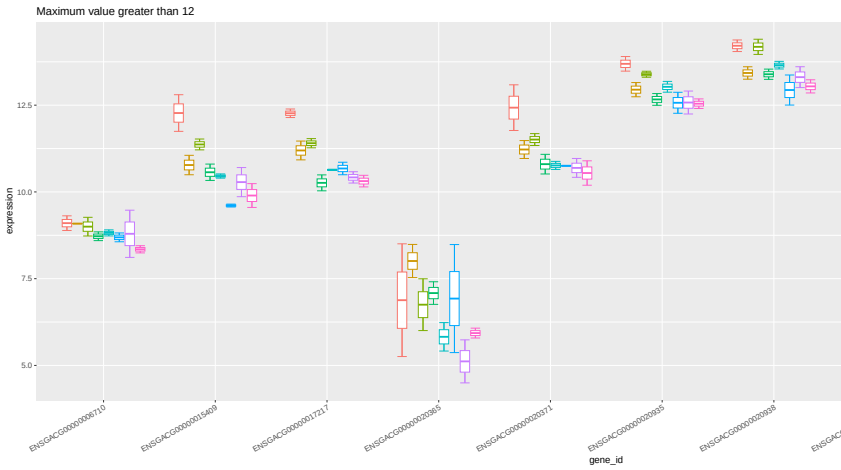
1. By `population`
2. By `sex`
3. By `origin`
4. By `elevation`
5. By `turb_combined`

```
boxplot_mean_population <- TS_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 <- TS_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

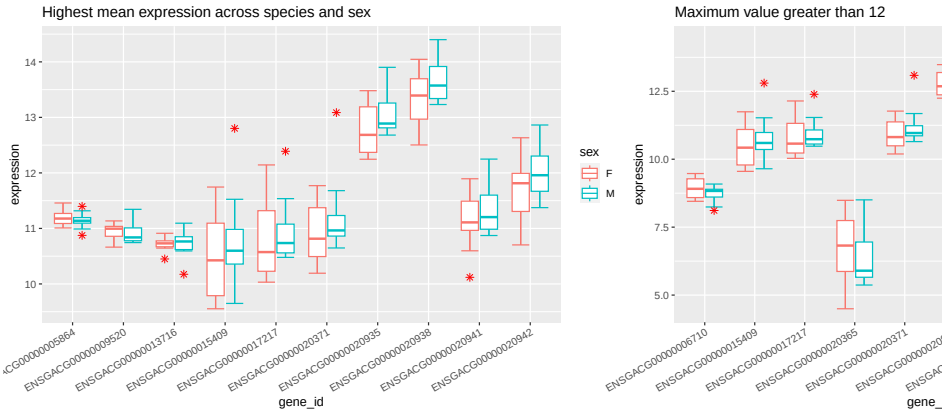
We can see there's some clear separation. More particularly, we can see the species Blauta, Frosta, and Galta generally have a higher gene expression level, whereas the species Lon, Pristi, Thanga have lower gene expression level.

```
boxplot_mean_sex <- TS_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 <- TS_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)
```



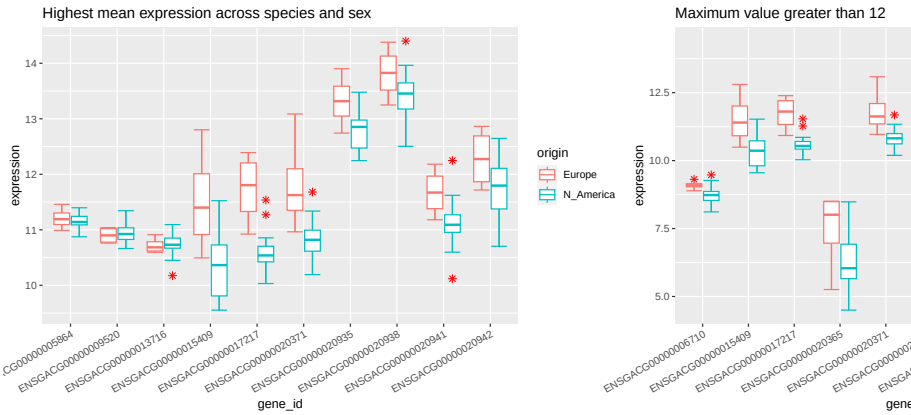
7.b. Gene expression analysis by sex

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 for the most part.

```
boxplot_mean_origin <- TS_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 <- TS_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")
```

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



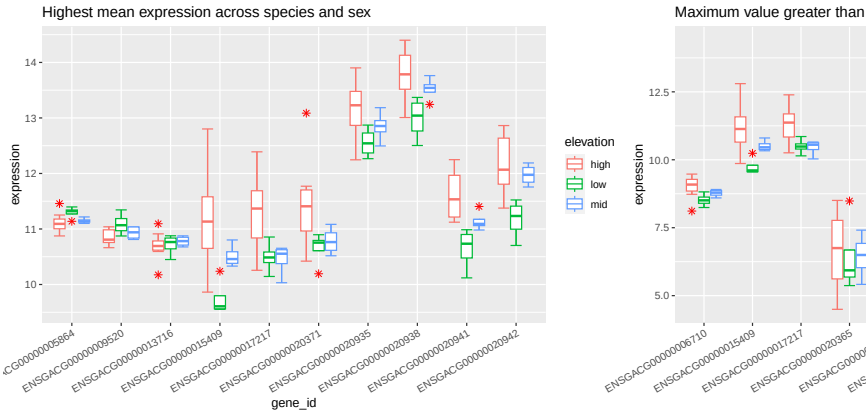
7.c. Gene expression analysis by origin

Here, we can see a better distinction as there's a more noticeable difference in the gene expression level as the European species displayed a generally higher gene expression level than their North America-originated counterpart

```
boxplot_mean_elevation <- TS_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 <- TS_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)
```



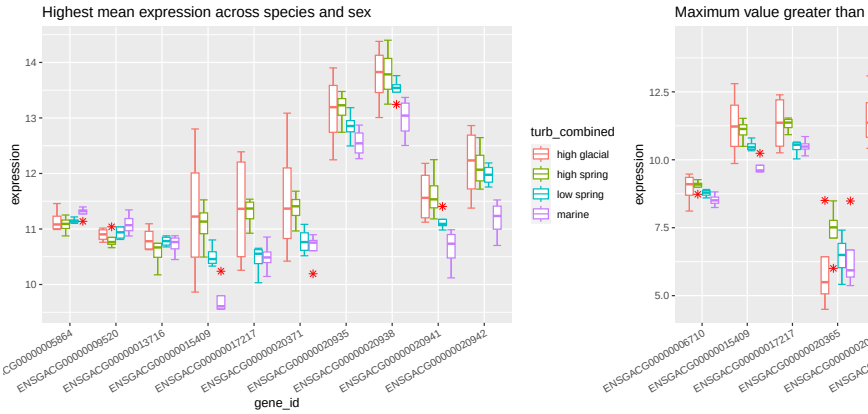
7.d. Gene expression analysis by elevation

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 <- TS_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 <- TS_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, follow by low spring, then marine.

Given the visualization and analysis, we can see that the Icelandic Stickleback fish's Torus Semicircularis 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 auditory sensory for the Stickleback to navigate the environment as it cannot solely relies on its vision.

This can be reflected where species that live in high elevation in glacial lake (Blauta) and high spring(Frosta and Galta) generally have a higher gene expression level, whereas species that live in low elevation and marine water Lon and Thanga have lower gene expression level.