README

Douglas Dally - a1767927

6/2/23

Data and Cleaning

The data was provided in an excel spreadsheet with 8 tabs. Each tab contained:

- The name of the cell: "GL-xxx".
- The cell line: Wild type or type-101.
- The type of treatment applied to the cell type: placebo or Activating Factor 42 (AF42).
- A dataset with gene expression values for 11 different treatment concentrations. Gene expression was measured from a concentration of 0 μg/L to 10 μg/L, with observations recorded for every 1 μg/L increase.

Data was initially cleaned manually in excel. The "concentration" and "gene_expression" values from each of the eight experiment blocks were combined in a tidy format with new columns "cell_line" and "block" added by hand. -99 entry for gene expression was also converted by hand to `NA` in excel.

Within R, we remove the observation with a missing value, then a new column is added to the dataset called `cell_type`, which contains the name of type of cell that each block (excel tab) represented.

The column `treatment` was created to replace the column `group`, and consisted of two levels: placebo and AF42. We factorised the variables `cell_type`, `cell_line` and `treatment`, and removed the `block` and `group` columns to obtain the final dataset.

```
data
```

```
# A tibble: 87 x 5
   gene_exp concentration cell_line cell_type treatment
      <dbl>         <dbl> <chr>     <chr>     <chr>
```

```
 1     5.51             0 wild      GL-CsE    placebo
 2     6.41             1 wild      GL-CsE    placebo
 3     5.71             2 wild      GL-CsE    placebo
 4     7.94             3 wild      GL-CsE    placebo
 5     6.87             4 wild      GL-CsE    placebo
 6     7.29             5 wild      GL-CsE    placebo
 7    10.0              6 wild      GL-CsE    placebo
 8     8.85             7 wild      GL-CsE    placebo
 9     8.91             8 wild      GL-CsE    placebo
10     9.68             9 wild      GL-CsE    placebo
# i 77 more rows
```

The code which produced the final cleaned dataset is found in "2023-03-10_gene-data-cleaning.R".

Task 1 - EDA

The following figures and tables were produced in "2023-03-10_gene-exp_EDA.R", and can be found in the "figs" and "tabs" folders respectively. All figures were saved as .png files, and all tables were saved as .docx files.

Firstly we looked at the distribution of gene expression over the whole dataset.

Figures 1 and 2 show that the distribution of gene expression is slightly right skewed, with some potential outliers existing above a gene expression value of 40. In future I would have asked the collaborator about these points.

We then produced the following scatter plots to visualise gene expression against concentration:

From these scatterplots, we see that there is a linear relationship between gene expression and treatment (growth factor) concentration for each cell type. Could potentially use linear mixed effects to model this data, as seen by varying slopes and intercepts for each cell type.

Also produced the following table of summary statistics for the data, grouped by cell line.

```r
wild_summary <- read.csv(here::here(
  "data", "2023-06-01_wild-summary.csv"
))

cell101_summary <- read.csv(here::here(
  "data", "2023-06-01_cell101-summary.csv"
))
```

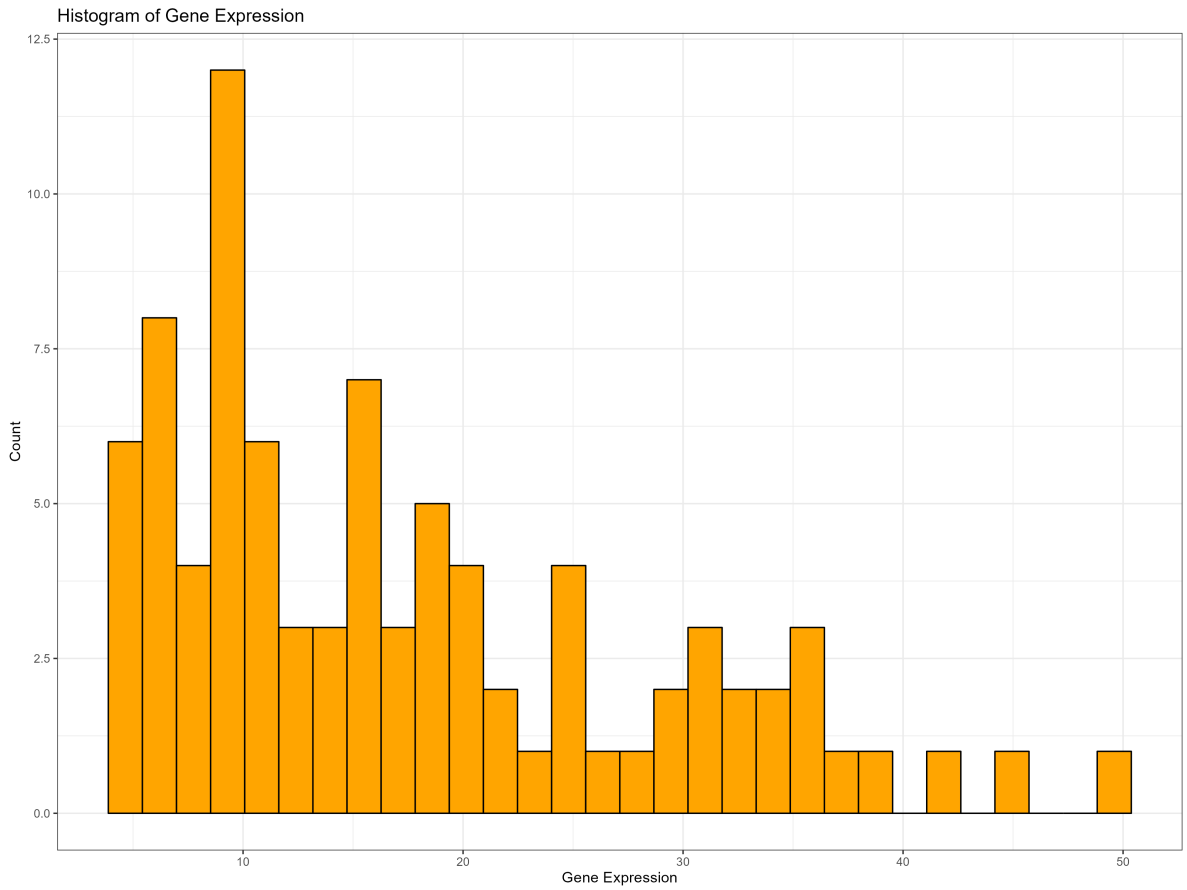


Figure 1: Histogram of Gene Expression for all cell types over both cell lines.

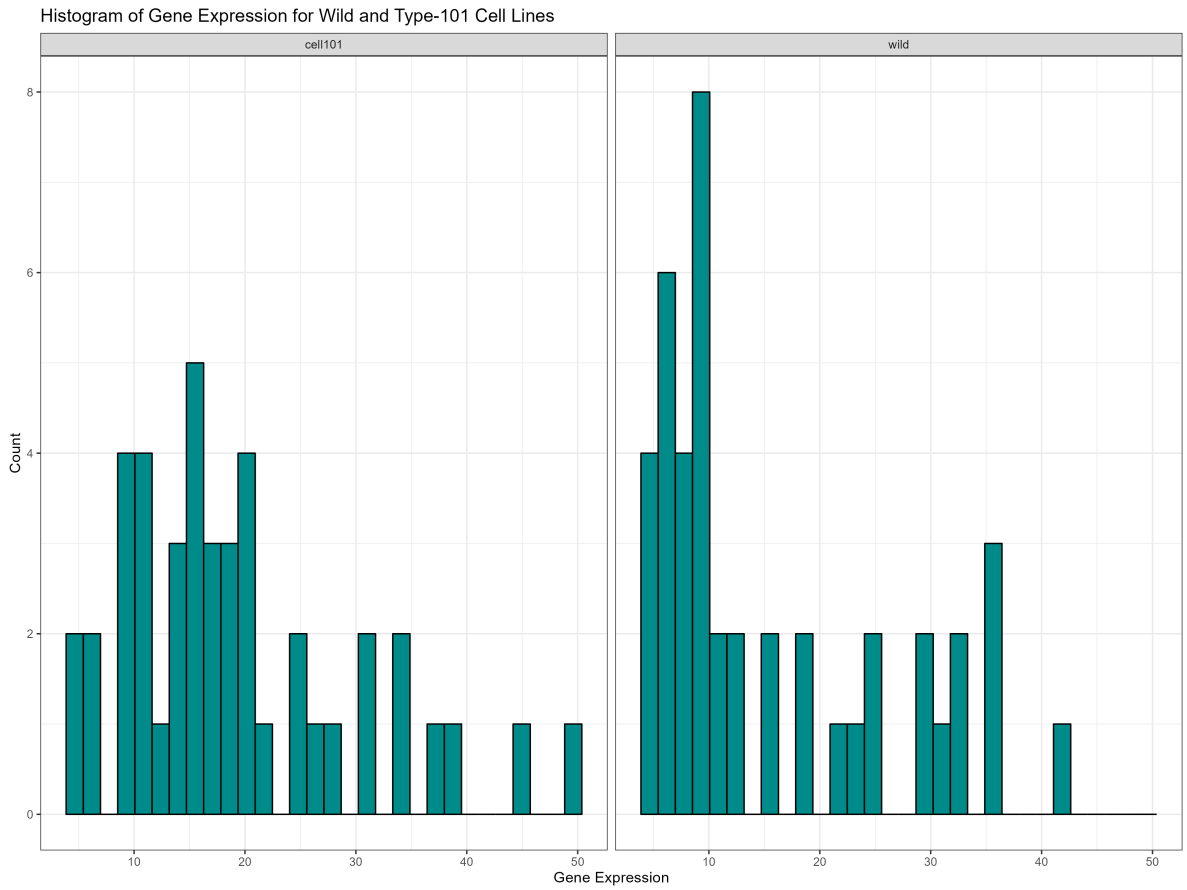


Figure 2: Histogram of Gene Expression for all cell types, seperated by cell line.

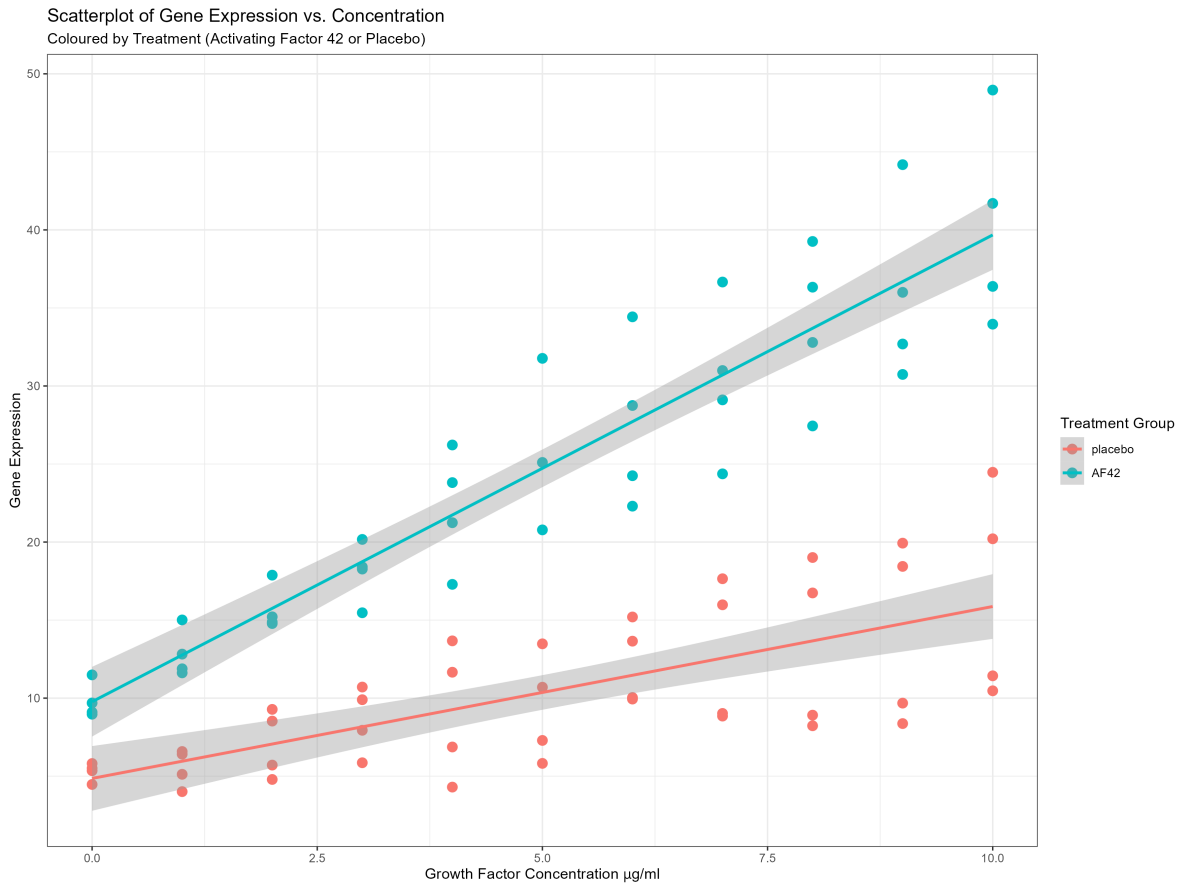


Figure 3: Scatterplot of gene expression vs concentration for all cell types and cell lines.

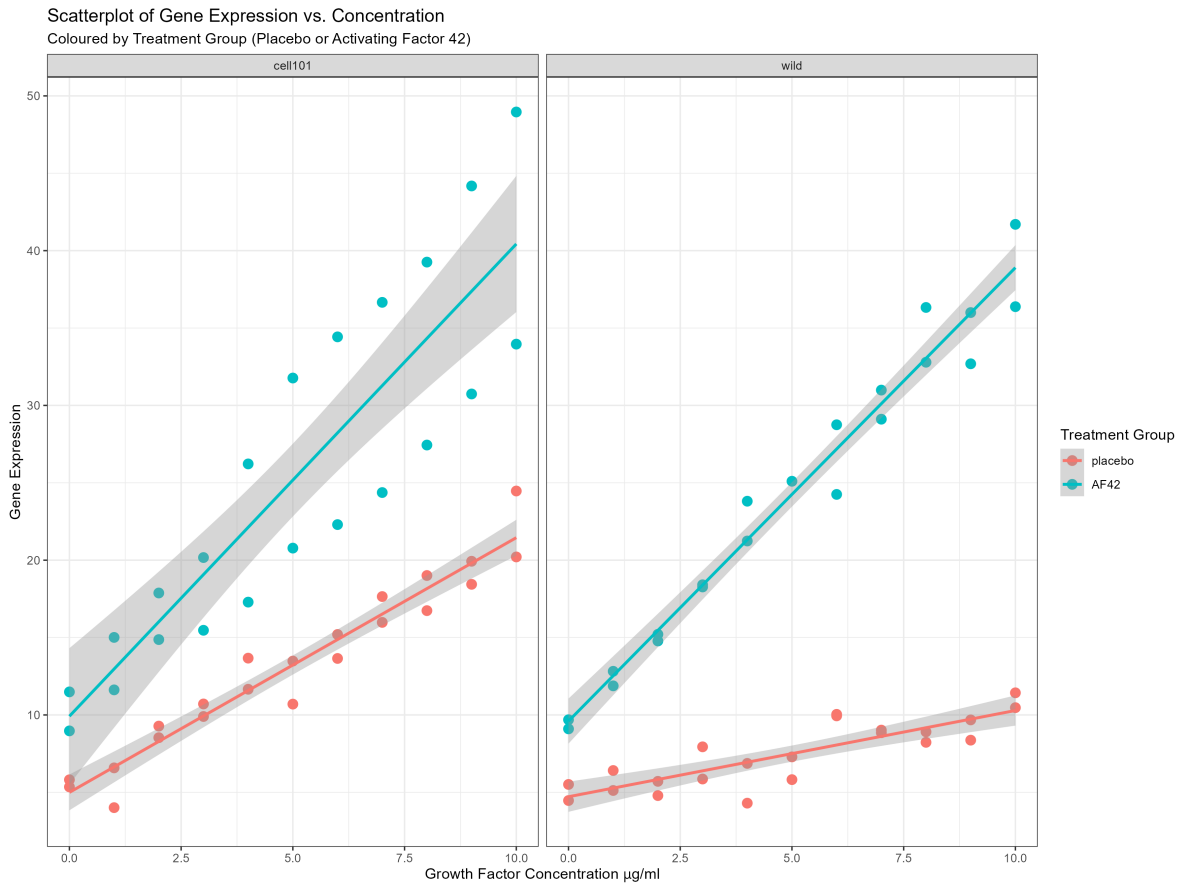


Figure 4: Scatterplot of gene expression vs concentration for all cell types and cell lines.

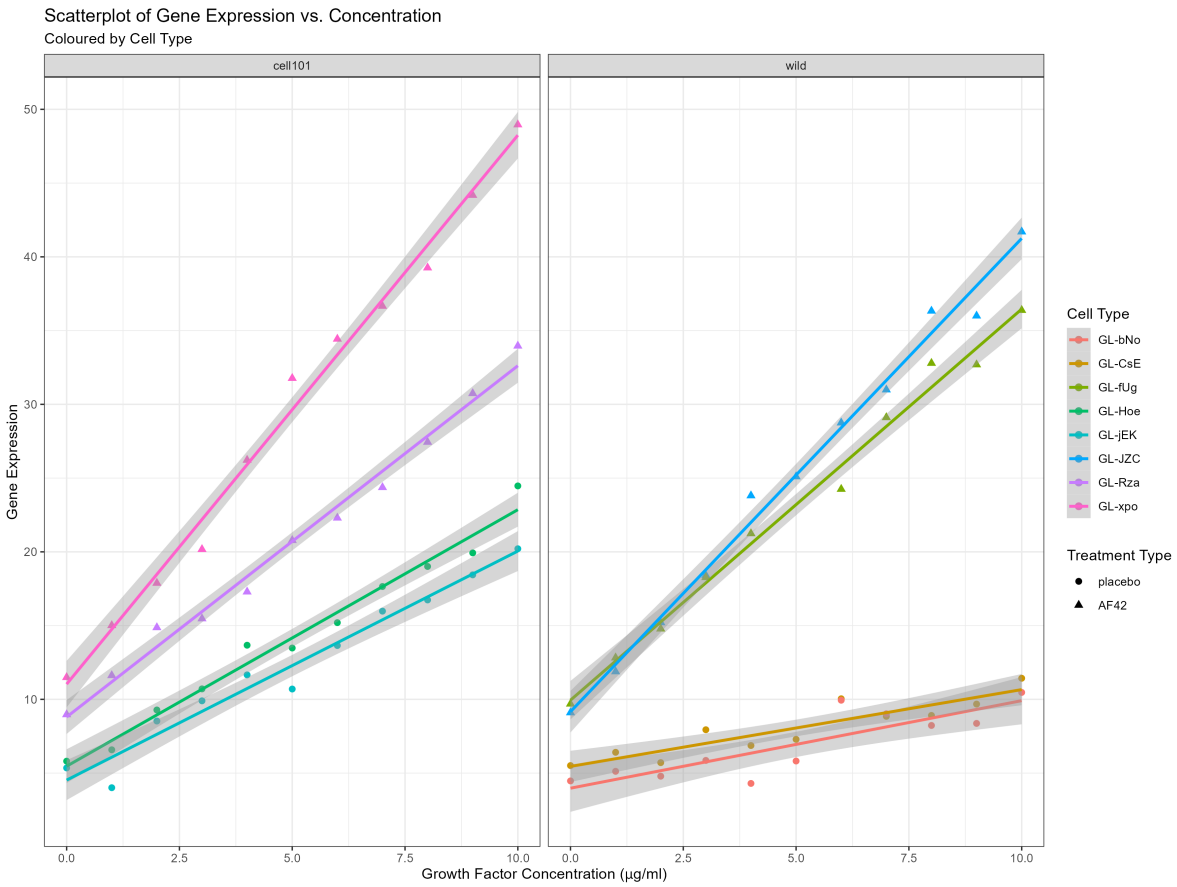


Figure 5: Scatterplot of gene expression vs concentration for all cell types and cell lines.

```
wild_summary |> gt() |>
  tab_header(
    title = "Summary Statistics",
    subtitle = "Wild Type Cell Line"
  ) |>
  tab_caption(caption = "Table 1: Summary statistics for wild-type cell line.")
```

Summary Statistics
Wild Type Cell Line

treatment	min	max	median	mean	std_dev	iqr
placebo	4.3	11.43	7.615	7.501364	2.135443	17.480
AF42	9.1	41.70	24.250	24.252381	9.862842	3.255

```
cell101_summary |> gt() |>
  tab_header(
    title = "Summary Statistics",
    subtitle = "Type-101 Cell Line"
  ) |>
  tab_caption(caption = "Table 2: Summary statistics for type-101 cell line.")
```

Summary Statistics
Type-101 Cell Line

treatment	min	max	median	mean	std_dev	iqr
placebo	4.01	24.47	13.565	13.22545	5.497236	17.4875
AF42	8.97	48.96	23.335	25.17455	11.141347	7.9875

Task 2 - Scatterplot as .tif

The collaborator required a particular figure as a .tif file for a conference. The figure required was a scatterplot of gene expression vs. concentration for the type-101 cell line, as shown in the image below. Note that the original figure was sent as a pdf. for simplicity in loading the image into quarto it was converted to .png file online.

The conference requires all text in the figure to be in Times New Roman and the final figure a tiff file (9in x 6in) with a resolution of 500. The code to reproduce this figure with the correct font and format is found in "2023-04-04_gene-exp-figure.R" file in the R folder. The process used was as follows:

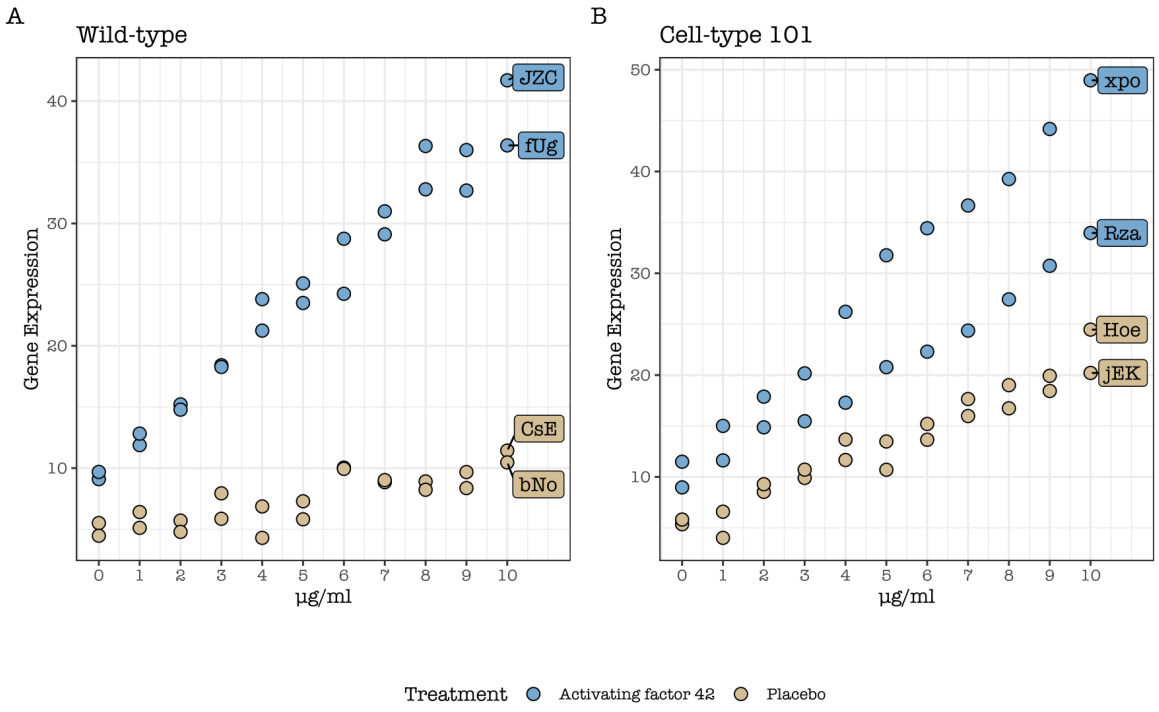
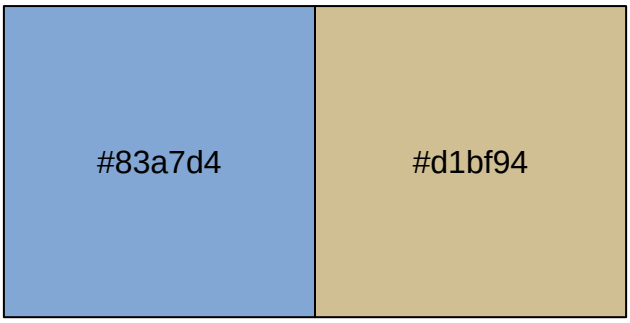


Figure 6: Required figure for conference with incorrect font.

- Load clean data from "2023-06-01_cleaned-data-final.csv".
- Create a new column in dataset called `exp_label` which contains the the prefixes of the gene type name (e.g Rza, jEK, fUg, etc.) for the observations with `concentration = 10`. These will be used as labels in the final plot.
- Filter data by cell line. We will produce a seperate figure for both wild and type-101 cell lines and join them later.
- For each cell line, Use `ggplot` to produce a basic scatterplot of gene expression vs. concentration by treatment type. We want a black bordered, coloured circular data points, and so `shape=21` and `size=3` inside `geom_point` was used.
- Add required gridlines and x-axis to the plot using `scale_x_continuous`.
- Add required colour scheme and legend names using `scale_fill_manual`. Colours were obtained by using Google Eye Dropper on the original figure sent by the collaborator so as to get the exact same results. These colours were found to be blue: "#83a7d4", and yellow: "#d1bf94".

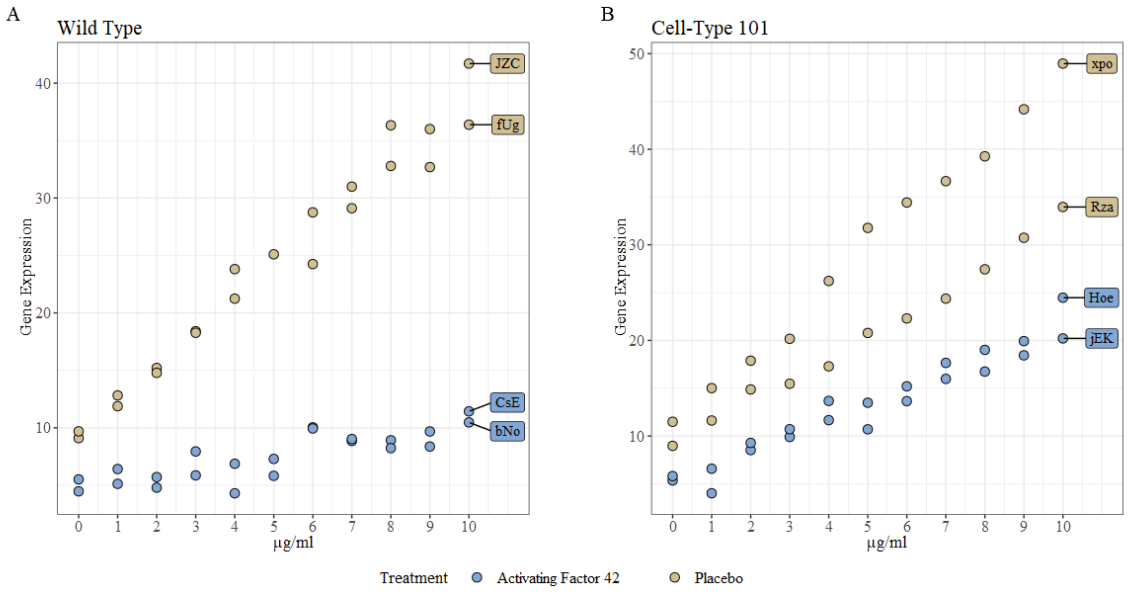
```
scales::show_col(c("#83a7d4", "#d1bf94"))
```



- Use `ggrepel` to produce the correct label and font for each cell type. The parameters nudge_x and size in this function were obtained via trial and error.

- Finish each plot by adding a tag, titles and themes.

Once each figure was produced, `ggarrange` was used to combine both figures under a single legend placed below the plots. The final result was the following.



This figure was sent to the collaborator as a .tif file with the required formatting. This was a large file and so was compressed online before being sent in an email. The compressing was easy as most of the figure is white.

Task 3 - Sample Size calculation

The collaborators wish to fit a linear regression of gene expression with the predictors concentration, cell age, treatment (two levels), cell type (two levels), and media (two levels). They needed to know the required sample size to achieve:

- A power of 90%.
- An R-squared value of $R^2 = 0.1$ between the predictors and response variable.

At a significance level of 5%.

To calculate the required sample size, we use the `pwr.f2.test()` function from the `pwr` package. This function takes the number of independent predictors, u, in the proposed model (5 predictors), the required power (90%) and effect size f, and returns the error degrees of freedom, v resulting from a one way ANOVA F-test.

Here, we define the effect size f as

$$f = \frac{R^2}{1 - R^2}.$$

The sample size is calculated using

$$n = u + v + 1$$

Using the code from "2023-05-01_sample-size-calc.R" in the R folder, it was found that the required sample size to achieve the desired power and R^2 value at the 5% significance level was $n = 154$.

Task 4 - Fitting Predictive Model & IMRaD

The collaborators decided that they wanted to fit a predictive model of gene expression. The results of this analysis were to be described in an IMRaD report. It was decided that a linear fixed effects model would be used to predict gene expression with potential fixed effects:

- Concentration (integer value ranging from 0 to 10).
- Cell Line (factor with 2 levels: wild and cell101)
- Treatment (factor with 2 levels: placebo and AF42)

The random effect in the model is cell type, which will be our random intercept term. The full linear mixed effects model containing all predictors and their interactions was fitted using the `lmer` function from the `lmerTest` package. Code for this section can be found in "2023-05-20_IMRaD-code.R" in the R folder.

Computing the ANOVA table of the full model, we find that there are three predictors with insignificant p-values, namely `treatment`, `cell_line` and `treatment:cell_line`.

```
# Full fixed effect model
M_full <- lmer(gene_exp ~ concentration*treatment*cell_line + (1|cell_type), data = data)
print(anova(M_full))
```

```
Type III Analysis of Variance Table with Satterthwaite's method
                                 Sum Sq Mean Sq NumDF  DenDF   F value
concentration                    3684.1  3684.1     1 75.002 1412.7592
treatment                          10.7    10.7     1  4.441    4.0858
cell_line                           0.0     0.0     1  4.441    0.0176
concentration:treatment           785.2   785.2     1 75.002  301.1072
concentration:cell_line            80.6    80.6     1 75.002   30.8960
treatment:cell_line                 0.0     0.0     1  4.441    0.0003
concentration:treatment:cell_line  51.5    51.5     1 75.002   19.7389
```

```
                                    Pr(>F)
concentration                     < 2.2e-16 ***
treatment                           0.1063
cell_line                           0.9002
concentration:treatment           < 2.2e-16 ***
concentration:cell_line           3.976e-07 ***
treatment:cell_line                 0.9872
concentration:treatment:cell_line 3.015e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Backwards selection was used to reduce model size until all predictors were significant. At each step in the process, a new model was fitted using the `update` function and an ANOVA table calculated. Insignificant terms were identified by a p-value greater than 0.05.

- Step 1: Remove 3-way interaction term `concentration:treatment:cell_line` by principle of marginality.
- Step 2: Remove the insignificant predictors `treatment`, `cell_line` and `treatment:cell_line`.
- Step 3: Remove 2-way interaction terms `concentration:treatment` and `concentration:cell_line` by principle of marginality.
- Step 4: Remove insignificant predictor `cell_line`.

This process resulted in 5 total models. For each model, the `r.squaredGLMM` function from the `MuMIn` package was used to calculate both:

- The marginal R-squared R^2_M: R^2 is only calculated from the fixed effects of the model.
- The conditional R-squared R^2_C: R^2 is calculated using both fixed and random effects of the model.

The `anova` function was then used to compare the 5 models by AIC. These AIC values were combined with the R^2_M and R^2_C values to produce a summary table of these 5 models. The number of fixed effect features present in each model was also stored in a column `n_pred`.

```
model_summary <- read.csv(here::here(
  "data", "2023-06-01_models-summary.csv"
))

model_summary |>
  gt() |>
  tab_header(
    title = "Summary of Backward Selection from Full Model") |>
```

```
tab_caption(caption = "Table 3: Results from Backwards Selection.")
```

Summary of Backward Selection from Full Model

model	n_fixed_effects	AIC	R2m	R2c
step 5	2	503	0.785	0.882
step 4	3	502	0.802	0.883
step 3	5	388	0.885	0.973
step 2	6	388	0.886	0.974
full	7	371	0.891	0.979

As seen in Table 3, it was found that the full model performed the best in terms of both AIC (lowest with 371) and R^2, and so this model was included in the IMRaD report as the final predictive model. The full model has:

- A marginal R-squared of $R^2_M = 0.891$, meaning that 89.1% of the variance in the data is captured by the fixed effects in the model.
- A conditional R-squared of $R^2_C = 0.979$, meaning that 97.9% of the variance in the data is captured by the full mixed effects model, incorporating both random and fixed effects.

The final model was visually tested by predicting the gene expression for wild-type GL-bNo cells with placebo treatment. This was done by using the `predict` function on the corresponding subset of the data with the recorded gene expression values removed. These recorded gene expressions were stored separately as the actual results.

The predicted values for gene expression were then plotted on top of the actual values to visualise the accuracy of final model, as seen below.

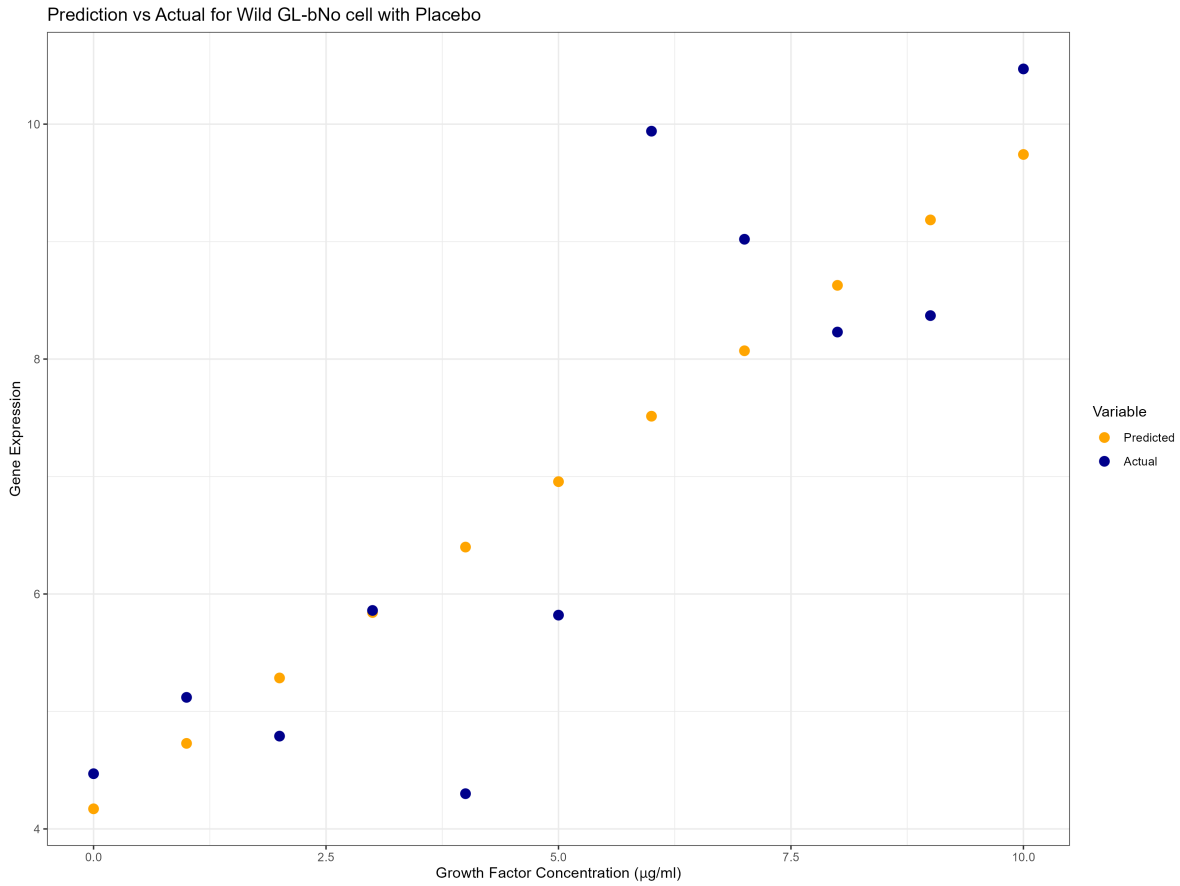


Figure 7: Predicted and Actual values for gene expression vs concentration, calculated from the full mixed effects model.