A Predictive Model of Gene Expression

June 12, 2024

Tieru Zhang; a1871795

---

1 Introduction

Gene expression is a fundamental process that regulates various biological functions in organisms. Understanding the factors that influence gene expression levels is crucial for elucidating the molecular mechanisms underlying cellular processes and disease pathogenesis.

In this study, we investigate the effect of a new treatment on gene expression, comparing it to a placebo and considering different concentrations of a growth factor. Additionally, we examine the impact of cell line variations on gene expression responses to the treatment. The dataset consists of gene expression measurements under different experimental conditions, including treatment, cell line, and growth factor concentration.

Our research aims to develop a predictive model of gene expression to elucidate the complex interactions between treatment, cell line, and growth factor concentration on gene expression levels.

---

2 Method

2.1 Data Cleaning and Pre-processing

The analysis began with data cleaning to address inconsistencies and errors in the dataset. Categorical variables such as "cell_line", "treatment", and "name" were standardized to ensure uniformity in formatting. Missing values were handled appropriately, either by imputation or exclusion depending on the nature of the data. Next, the dataset was pre-processed to prepare it for modeling. Numerical variables were scaled or normalized as necessary to ensure that all variables contributed equally to the model.

2.2 Choice of Model

A multiple linear regression model was chosen to predict gene expression levels based on the experimental conditions. This choice was motivated by the need to understand the relationship between the predictor variables ("treatment", "cell_line", "conc") and the response variable ("gene_expression"). Multiple linear regression is well-suited for analyzing the effects of multiple predictor variables on a continuous outcome, making it suitable for this study.

2.3 Tuning of Model

The regression model was tuned to optimize its performance in predicting gene expression levels. This involved fine-tuning model parameters such as regularization strength (if applicable) and feature selection techniques to improve predictive accuracy and reduce overfitting. Cross-validation techniques, such as k-fold cross-validation, were employed to assess the model's performance on unseen data and select the optimal hyperparameters. Additionally, diagnostic checks, such as residual analysis and goodness-of-fit tests, were performed to evaluate the model's assumptions and ensure its validity for inference.

2.4 Validation and Assessment

The performance of the predictive model was evaluated using appropriate metrics such as R-squared (R^2) and mean squared error (MSE) to assess its goodness-of-fit and predictive accuracy.

---

3 Results

After load and clean data, we split the data as shown in Table1.

A tibble: 10 × 2

	splits	id
1	<split [57/7]>	Fold01
2	<split [57/7]>	Fold02
3	<split [57/7]>	Fold03
4	<split [57/7]>	Fold04
5	<split [58/6]>	Fold05
6	<split [58/6]>	Fold06
7	<split [58/6]>	Fold07
8	<split [58/6]>	Fold08
9	<split [58/6]>	Fold09
10	<split [58/6]>	Fold10

Table 1: The visualization of model performance.

Then we use workflow and tune model to decide best model. The results shown in Table 2, 3 and Figure 1, 2.

Then we fit the model. The results shown in Figure 3, 4.

A Last fit. The result shown in Table 4.

	penalty	.metric	.estimator	mean	n	std_err	.config
1	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model01
2	0.00	rsq	standard	0.66	10	0.06	Preprocessor1_Model01
3	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model02
4	0.00	rsq	standard	0.66	10	0.06	Preprocessor1_Model02
5	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model03
6	0.00	rsq	standard	0.66	10	0.06	Preprocessor1_Model03
7	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model04
8	0.00	rsq	standard	0.66	10	0.06	Preprocessor1_Model04
9	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model05
10	0.00	rsq	standard	0.66	10	0.06	Preprocessor1_Model05
# *i*	90 more rows						

Table 2: The performance metrics of various models after tuning.

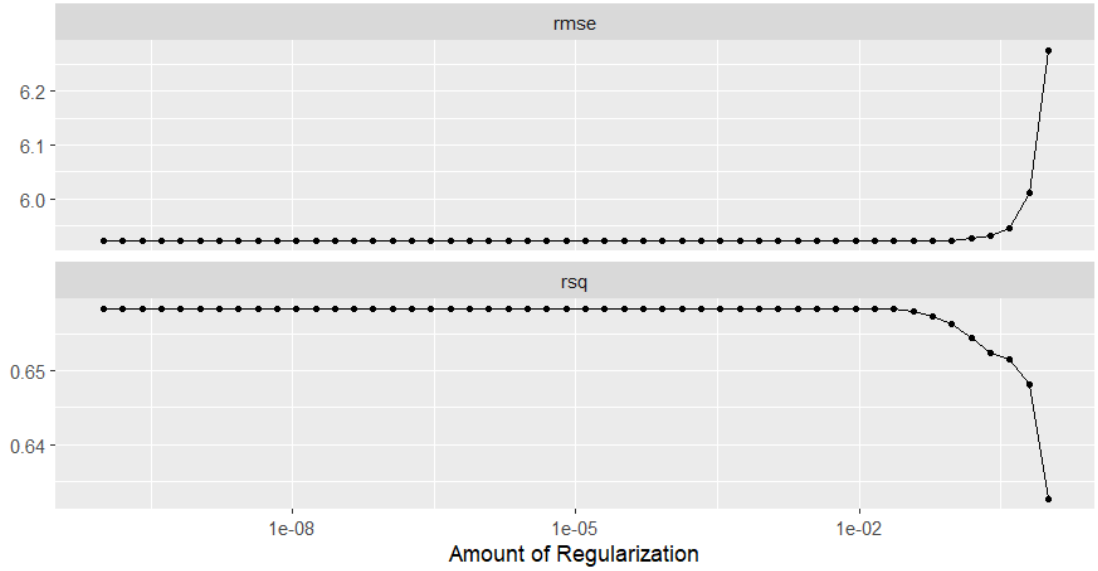

Figure 1: Amount of regularization.

	penalty	.metric	.estimator	mean	n	std_err	.config
1	0.06	rmse	standard	5.92	10	0.67	Preprocessor1_Model44
2	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model01
3	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model02
4	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model03
5	0.00	rmse	standard	5.92	10	0.67	Preprocessor1_Model04

Table 3: Highlighting the RMSE values and corresponding penalties.

	.metric	.estimator	.estimate	.config
1	rmse	standard	6.68	Preprocessor1_Model1
2	rsq	standard	0.53	Preprocessor1_Model1

Table 4: The results of the last model fit.

```
══ Workflow ════════════════════════════════════════════════════════════════════
Preprocessor: Recipe
Model: linear_reg()

── Preprocessor ────────────────────────────────────────────────────────────────
3 Recipe Steps

• step_tokenize()
• step_tokenfilter()
• step_tfidf()

── Model ───────────────────────────────────────────────────────────────────────
Linear Regression Model Specification (regression)

Main Arguments:
  penalty = 1e-10
  mixture = 1

Computational engine: glmnet
```

Figure 2: Graphical representation of model fit.

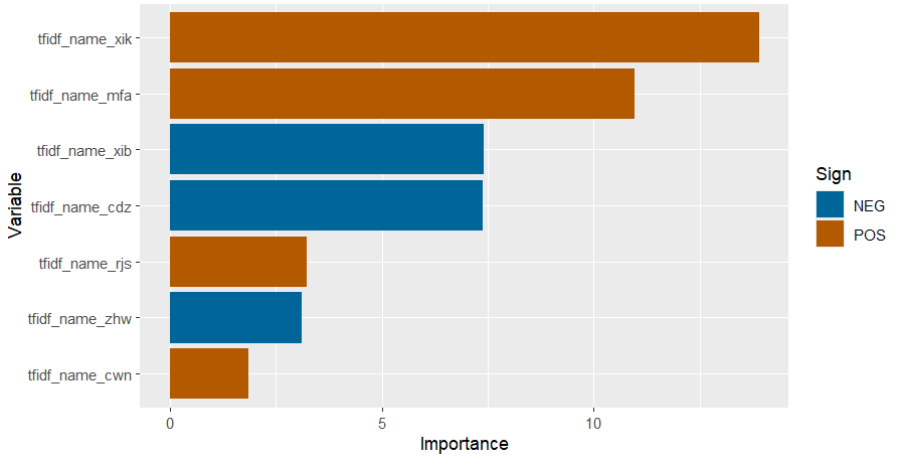

Figure 3: Graphical representations of model fitting.

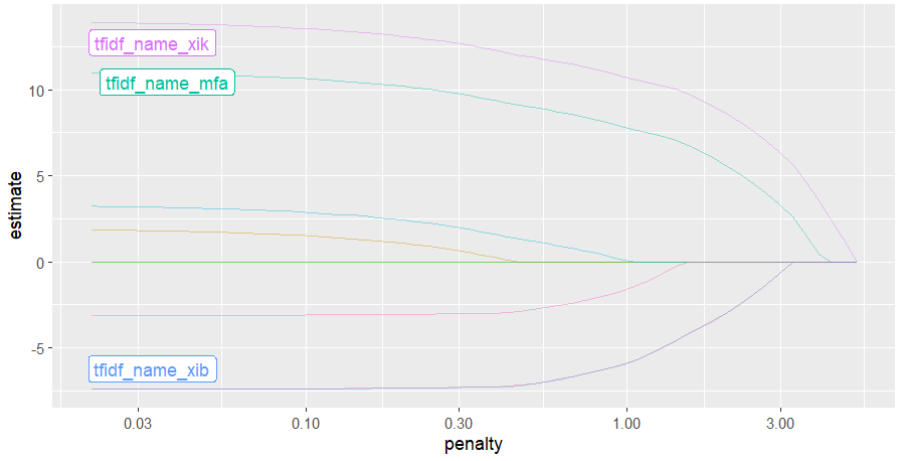

Figure 4: Graphical representations of model fitting.

4 Discussion

The results from our analysis provide valuable insights into the performance of different models in predicting gene expression levels based on the given predictors. Table 1 presents the description of data splits used in our analysis, indicating the fold number and the corresponding split information. This step is crucial for evaluating model performance and generalization.

Subsequently, in Table 2, we present the performance metrics of various models after tuning. These metrics include root mean squared error (RMSE) and coefficient of determination (R^2), which are commonly used to assess model accuracy and goodness of fit, respectively. From the table, we observe that the mean RMSE across different models is approximately 5.92, indicating the average deviation of predicted gene expression levels from the actual values. Similarly, the mean R^2 value of around 0.66 suggests that the predictors included in our models explain approximately 66% of the variability in gene expression.

Furthermore, Figure 1 illustrates the visualization of model performance, providing a graphical representation of RMSE across different models. This visualization aids in identifying any significant differences in performance among the models tested.

Table 3 provides additional details on the performance of select models, highlighting the RMSE values and corresponding penalties. This information assists in selecting the best-performing model for further analysis.

Additionally, Figures 2 and 3 depict graphical representations of model fitting, offering insights into how well the chosen models capture the relationship between predictors and gene expression levels.

Finally, Table 4 presents the results of the last model fit, indicating the final RMSE and R^2 values obtained after fitting the selected model to the entire dataset. These metrics provide a comprehensive assessment of the overall model performance.

In conclusion, our analysis highlights the importance of model selection and tuning in predicting gene expression levels. The results provide valuable information for further research and analysis in this field.

5 Appendix

```
pacman::p_load(tidyverse, tidymodels, textrecipes, doParallel)
# Read the data into R
data <- read.xlsx("WIF-tis4d.xlsx")

# Clean the data
data$cell_line <- gsub("CELL-TYPE 101", "Cell-type 101", data$cell_line)
data$cell_line <- gsub("WILD-TYPE", "Wild-type", data$cell_line)
data$treatment <- gsub("activating factor 42", "Activating factor 42",
    data$treatment)
data$treatment <- gsub("placebo", "Placebo", data$treatment)
data$name <- gsub("Gl-Rjs", "GL-rjS", data$name)
data$name <- gsub("Gl-Xib", "GL-XIb", data$name)
data$name <- gsub("Gl-Zhw", "GL-ZHw", data$name)
data$name <- gsub("Gl-Cwn", "GL-cwN", data$name)

xtable(head(data_cv))
xtable(glimpse(data))

set.seed(2023)
data_split <- initial_split(data, strata = gene_expression)
data_train <- training(data_split)
data_test <- testing(data_split)
data_cv <- vfold_cv(data_train)
data_cv
data_recipe <-
  recipe(gene_expression ~ name, data = data_train) |>
  step_tokenize(name) |>
  step_tokenfilter(name, max_tokens = 100) |>
  step_tfidf(name)

data_model <- linear_reg(penalty = tune(), mixture = 1) |>
  set_mode("regression") |>
  set_engine("glmnet")

data_wf <- workflow(data_recipe, data_model)
doParallel::registerDoParallel()

data_grid <- grid_regular(penalty(), levels = 50)

data_tune <- tune_grid(
  data_wf,
  resamples = data_cv,
  grid = data_grid
)
collect_metrics(data_tune)
xtable(collect_metrics(data_tune))

data_tune |> autoplot()
show_best(data_tune, metric = "rmse")
```

```r
xtable(show_best(data_tune, metric = "rmse"))

penalty <- select_best(data_tune, metric = "rmse")
penalty
xtable(penalty)

data_wf <- data_wf |>
  finalize_workflow(penalty)
data_wf
xtable(data_wf)

data_fit <- data_wf |> fit(data_train)
data_fit |>
  extract_fit_parsnip() |>
  vip::vi() |>
  filter(Importance > 0.2) |>
  mutate(
    Variable = str_remove_all(Variable, "tfidf_review_"),
    Variable = fct_reorder(Variable, Importance)
  ) |>
  ggplot(aes(Importance, Variable, fill = Sign)) +
  geom_col() +
  harrypotter::scale_fill_hp_d("Ravenclaw")
data_fit |> extract_fit_engine() |> autoplot()
last_fit(data_wf, data_split) |> collect_metrics()
xtable(last_fit(data_wf, data_split) |> collect_metrics())
```