Gene expression analysis report

AUTHOR
Ethan Johnson

PUBLISHED
May 26, 2023

Introduction

The dataset consists of the results of an experiment that looked at the effect of a new treatment on gene expression. It contains trials for two types of cell lines (which the new treatment and a placebo were tested. For each cell line, different concentrations of a growth factor were tested (0, 2, 4, 6, 8, 10 micrograms/mL). The key variables of interest were:

- gene expression: a number determining the gene expression,
- concentration: a number between 0 and 10 (micrograms/mL) that reflects the concentration of the growth factor,
- treatment: the type of treatment used (treatment or placebo),
- cell line: the type of cell line the treatment was tested on, and
- gene line: the type of gene line used in the trial.

The key research question was to use the data to predict the effect of the treatment on the effect of growth factor on gene expression.

Methods

The data was analysed using the packages `tidyverse` (Wickham et al. 2019), `lme4` (Bates et al. 2015), `ggrepel` (Slowikowski 2023), `patchwork` (Pedersen 2022), `showtext` (Qiu and See file AUTHORS for details. 2023), `lmerTest` (Kuznetsova, Brockhoff, and Christensen 2017), `MuMIn` (Bartoń 2023), and `gt` (iannone et al. 2023) in RStudio[1].

The dataset was preprocessed by manually combining each Excel worksheet together into the same page. This worksheet contained a column for concentration and gene expression. A column containing each cell line and gene line associated with each data entry was then added. The cell lines, treatment, and gene line variables were then converted to factors. All entries with a value of gene expression less than 0 were converted to NA.

A mixed effects model was fitted to the data with gene expression as a numeric outcome, the fixed effects were all of the interactions between treatment, concentration and cell lines, and the random effect was the gene line which was fitted as the intercept.

The model was tuned by removing all insignificant variables at the 5% level from the full model. The significance of the random gene line intercept was assessed using the `ranova()` function. The significance of the fixed effects were assessed using the `anova()` function by comparing a model for each level of interaction to the full interaction model (i.e. all first, second and third order interactions).

Results

Figure 1 shows the experimental gene expression results for each concentration of growth factor. The results for both cell line types and treatments are shown, with each trend labelled with the gene line used in the trial. There exists some clustering of the results between the different treatment used. As such, we determine that we want interactions in the final model.

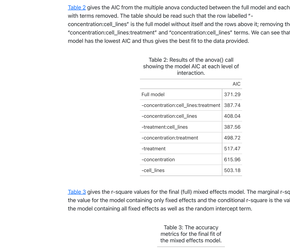

Figure 1: Scatterplot of concentration of growth factor vs gene expression for each cell line and treatment.

Table 1 gives the results of the anova conducted between a model with and without the random effect intercept term (the gene line used). We see that the model with the random intercept term has a lower AIC than the model without. Hence, it follows that we want to keep this random effect in the final model.

Table 1: Results of the ranovaU call showing the significance of the random effect.

	Number of Parameters	Log-likelihood	AIC	LRT	Degrees of Freedom	Pr(>Chisq)
Random intercept	10.00	−174.56	369.13	NA	NA	NA
No Random Intercept	9.00	−214.15	446.29	79.17	1.00	0.00

Table 2 gives the AIC from the multiple anova conducted between the full model and each model with terms removed. The table should be read such that the row labelled "-concentration:cell_lines" is the full model without itself and the rows above it; removing the "concentration:cell_lines:treatment" and "concentration:cell_lines" terms. We can see that the full model has the lowest AIC and thus gives the best fit to the data provided.

Table 2: Results of the anova() call showing the model AIC at each level of interaction.

	AIC
Full model	371.29
-concentration:cell_lines:treatment	387.74
-concentration:cell_lines	408.04
-treatment:cell_lines	387.56
-concentration:treatment	498.72
-treatment	517.47
-concentration	615.96
-cell_lines	503.18

Table 3 gives the r-square values for the final (full) mixed effects model. The marginal r-square is the value for the model containing only fixed effects and the conditional r-square is the value for the model containing all fixed effects as well as the random intercept term.

Table 3: The accuracy metrics for the final fit of the mixed effects model.

Marginal rsq.	Conditional rsq.
0.89	0.98

Table 4 gives the coefficients for the final mixed effects model fitted.

Table 4: The coefficients for the final mixed effects model.

	coeffs
(Intercept)	9.92
treatmentplacebo	−4.92
concentration	3.05
cell_linewild-type	−0.36
treatmentplacebo:concentration	−1.41
treatmentplacebo:cell_linewild-type	0.08
concentration:cell_linewild-type	−0.12
treatmentplacebo:concentration:cell_linewild-type	−0.97

Discussion

We see from the final model that the key variables associated with gene expression are concentration, treatment, cell lines (as well as all their interactions) and gene line. In terms of the research question, the effect of treatment on gene expression is strong. This conclusion can be drawn because all of the treatments interactions with other variables are deemed to be significant and included in the final model. Having said this, the variables of gene line, cell line, and concentration were also significant enough to be kept in the final model and so each of these effects on gene expression are also strong.

The final model is quite strong, we see a marginal and conditional r-square of 0.89 and 0.98 respectively from Table 3. This indicates that in the full model, with random effects, the model explains 98% of the variability in gene expression.

Appendix

Code for analysis

```
pacman::p_load(tidyverse, lme4, ggrepel, patchwork, showtext, lmerTest, MuMIn, gt

font_add(
  family = "times",
  regular = here::here(
    "figs","Times New Roman.ttf"
  )
)

cellData <- read_csv(here::here("data","cellData.csv"))
cellData$cell_lines <- as.factor(cellData$cell_lines)
cellData$treatment <- as.factor(cellData$treatment)
cellData$GL <- as.factor(cellData$GL)


cellData <- cellData %>%
  mutate(gene_expression = replace(gene_expression, which(gene_expression < 0), N
M1.Full <- lmer(data = cellData, gene_expression ~ treatment+concentration+cell_l
# random effects
randomEffects <- ranova(M1.Full)

# fixed effects
# 3-way interactions
M1.A <- update(M1.Full, . ~ . -concentration:cell_lines:treatment)

df <- anova(M1.Full, M1.A)["AIC"][2,1]
df <- rbind(df, anova(M1.Full, M1.A)["AIC"][1,1])

# 2-way interactions
M2.Full <- M1.A

M2.A <- update(M2.Full, . ~ . -concentration:cell_lines)
M2.B <- update(M2.Full, . ~ . -treatment:cell_lines)
M2.C <- update(M2.Full, . ~ . -concentration:treatment)

df <- rbind(df, anova(M1.Full, M2.A)["AIC"][1,1])
df <- rbind(df, anova(M1.Full, M2.B)["AIC"][1,1])
df <- rbind(df, anova(M1.Full, M2.C)["AIC"][1,1])

# 1-way interactions
M3.Full <- update(M2.Full, . ~ . -concentration:cell_lines-treatment:cell_lines-c

M3.A <- update(M3.Full, . ~ . -treatment)
M3.B <- update(M3.Full, . ~ . -concentration)
M3.C <- update(M3.Full, . ~ . -cell_lines)

df <- rbind(df, anova(M1.Full, M3.A)["AIC"][1,1])
df <- rbind(df, anova(M1.Full, M3.B)["AIC"][1,1])
df <- rbind(df, anova(M1.Full, M3.C)["AIC"][1,1])
# colours

col_path <- c("#81A7CD", "#D2BF9D")

# plot
showtext_auto()
plot1 <- cellData %>%
  filter(cell_lines == "cell-type 101") %>%
  ggplot(aes(concentration, gene_expression, color = treatment, label = GL)) +
  geom_point(aes(fill = treatment), shape=21, colour = "black", size = 3) +
  geom_label_repel(aes(fill = treatment),
                   color = "black",
                   min.segment.length = 0,
                   data = . %>% subset(., concentration==10),
                   max.overlaps = Inf,
                   nudge_x = 1,
                   show.legend = FALSE,
                   family = "times"
                   ) +

  theme_bw() +
  scale_x_continuous(minor_breaks = seq(0, 12.5, 0.5), breaks = seq(0, 10, by = 1
  scale_fill_manual(values = col_path, labels = c("Activating factor 42", "Placeb
  labs(
    x = expression(paste("Concentration of Growth Factor (", mu, "g/ml)")),
    y = "Gene Expression"
  ) +
  ggtitle("Cell-type 101")+
  guides(fill = guide_legend(title = "Treatment")) +
  theme(text = element_text(family = "times"))

plot2<-cellData %>%
  filter(cell_lines == "wild-type") %>%
  ggplot(aes(concentration, gene_expression, color = treatment, label = GL)) +
  geom_point(aes(fill = treatment), shape=21, colour = "black", size = 3) +
  geom_label_repel(aes(fill = treatment),
                   color = "black",
                   min.segment.length = 0,
                   data = . %>% subset(., concentration==10),
                   max.overlaps = Inf,
                   nudge_x = 1,
                   show.legend = FALSE,
                   family = "times"

  ) +
  theme_bw() +
  scale_x_continuous(minor_breaks = seq(0, 12.5, 0.5), breaks = seq(0, 10, by = 1
  scale_fill_manual(values = col_path, labels = c("Activating factor 42", "Placeb
  labs(
    x = expression(paste("Concentration of Growth Factor (", mu, "g/ml)")),
    y = "Gene Expression"
  ) +
  ggtitle("Wild-type") +
  guides(fill = guide_legend(title = "Treatment")) +
  theme(text = element_text(family = "times"))



patchwork <- plot2 + plot1 + plot_annotation(tag_levels = 'A') +
  plot_layout(guides = "collect") &
  theme(legend.position = "bottom")
patchwork

rownames(randomEffects) <- c("Random Intercept", "No Random Intercept")

randomEffects %>% gt(rownames_to_stub = TRUE) %>%
  cols_label(
    npar = md("*nMumber of Parameters*"),
    logLik = md("*eLog-Likelihood*"),
    AIC = md("*eAIC*"),
    LRT = md("*eLRT*"),
    Df = md("*eDegrees of Freedom*"),
    "Pr(>Chisq)" = md("*ePr(>Chisq)*")
  ) %>%
  fmt_number(decimals = 2)
rownames(df) <- c("Full model", "-concentration:cell_lines:treatment", "-concentr
                  "-treatment:cell_lines", "-concentration:treatment", "-treatmen
                  "-concentration", "-cell_lines")

colnames(df) <- c("AIC")


df <- as.data.frame(df)

df %>% gt(rownames_to_stub = TRUE) %>% fmt_number(decimals = 2)
as.data.frame(r.squaredGLMM(M1.Full)) %>% gt() %>%
  cols_label(R2m = "Marginal rsq", R2c = "Conditional rsq") %>% fmt_number(decima
coeffs <- summary(M1.Full)$coefficients[,1]
coeffs <- as.data.frame(coeffs)

coeffs %>% gt(rownames_to_stub = TRUE) %>% fmt_number(decimals = 2)
```

References

Bartoń, Kamil. 2023. *MuMIn: Multi-Model Inference*. https://CRAN.R-project.org/package=MuMIn.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. https://doi.org/10.18637/jss.v067.i01.

Iannone, Richard, Joe Cheng, Barret Schloerke, Ellis Hughes, Alexandra Lauer, and Jooheung Seo. 2023. *Gt: Easily Create Presentation-Ready Display Tables*. https://CRAN.R-project.org/package=gt.

Kuznetsova, Alexandra, Per B. Brockhoff, and Rune H. B. Christensen. 2017. "lmerTest Package: Tests in Linear Mixed Effects Models." *Journal of Statistical Software* 82 (13): 1–26. https://doi.org/10.18637/jss.v082.i13.

Pedersen, Thomas Lin. 2022. *Patchwork: The Composer of Plots*. https://CRAN.R-project.org/package=patchwork.

Qiu, Yixuan, and authors/contributors of the bundled software. See file AUTHORS for details. 2023. *Showtext: Using Fonts More Easily in r Graphs*. https://CRAN.R-project.org/package=showtext.

Slowikowski, Kamil. 2023. *Ggrepel: Automatically Position Non-Overlapping Text Labels with 'Ggplot2'*. https://CRAN.R-project.org/package=ggrepel.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Footnotes

1. https://www.rstudio.com/↩