reQTL Poisson Mixed Model Example

Genevieve Roberts

2024-06-14

Load Packages

```r
library(lme4)
library(lmerTest)
library(emmeans)
library(data.table)
library(here)
library(dplyr)
library(stringr)
library(tidyr)
library(tibble)
library(purrr)
library(ggplot2)
library(broom)
library(broom.mixed)
```

Why use an interaction regression model?

An interaction model allows us to look at the differences in the slopes between genotypes.

Simulate Data for a reQTL

```r
#Simulate some data with an interaction effect
library(tibble)
library(dplyr)

simulate_expression_data <- function(n_donors = 80,
                                     p_minor_allele = 0.3,
                                     interaction_effect_magnitude = 0.5,
                                     snp_id="SNP1") {
  # Function to simulate genotype frequencies according to Hardy-Weinberg equilibrium
  simulate_genotype <- function(n, p_minor_allele) {
    p_major_allele <- 1 - p_minor_allele
    sample(0:2, size = n, replace = TRUE, prob = c(p_major_allele^2,
                                                   2 * p_major_allele * p_minor_allele,
                                                   p_minor_allele^2))
  }

  # Simulate the dataframe
  df <- tibble(
    donor = rep(1:n_donors, each = 2),
    IFNg_treatment = rep(c(TRUE, FALSE), n_donors),
    genotype = rep(simulate_genotype(n_donors, p_minor_allele), each = 2)
  )

  # Add the interaction effect and simulate expression values
  df <- df %>%
    mutate(
      interaction_effect = 1 + IFNg_treatment * genotype * interaction_effect_magnitude,
      expression_value = rpois(n = n(), lambda = interaction_effect * 20) # Scale lambda
    ) %>%
    mutate(snp_id = snp_id) %>%
    select(snp_id, donor, IFNg_treatment, genotype, expression_value)

  return(df)
}

set.seed(123)
df <- simulate_expression_data(n_donors = 80,
                               p_minor_allele = 0.3,
                               interaction_effect_magnitude = 0.5)

# View the first few rows of the simulated dataframe
pander(head(df))
```

snp_id	donor	IFNg_treatment	genotype	expression_value
SNP1	1	TRUE	0	16
SNP1	1	FALSE	0	23
SNP1	2	TRUE	1	29

snp_id	donor	IFNg_treatment	genotype	expression_value
SNP1	2	FALSE	1	25
SNP1	3	TRUE	0	15
SNP1	3	FALSE	0	18

Fit the Negative Binomial Mixed Effects Regression Model & Interperet Interaction Term

```r
# Fit negative binomial regression with random intercept (the outcome looks like count data)
poisson_model <- glmer.nb(expression_value ~ IFNg_treatment*genotype + (1 | donor),
                          data = df)

#clean-up the model output and pull the interaction term
tidy_poisson <- tidy(poisson_model)
tidy_poisson %>% pander()
```

Table 2: Table continues below

effect	group	term	estimate	std.error	statistic	p.value
fixed	NA	(Intercept)	3.025	0.03281	92.17	0
fixed	NA	IFNg_treatmentTRUE	-0.0439	0.04623	-0.9495	0.3424
fixed	NA	genotype	-0.001929	0.04034	-0.04781	0.9619
fixed	NA	IFNg_treatmentTRUE:genotype	0.4023	0.05277	7.625	2.45e-14
ran_pars	donor	sd__(Intercept)	1.952e-06	NA	NA	NA

The estimate (AKA beta) for the interaction term IFNg_treatmentTRUE:genotype is 0.40, with a p-value of 2.5e-14:

- The estimate of 0.40 means that for each one-unit increase in genotype, the log of the expected expression_value count increases by 0.40 when the treatment is TRUE. Exponentiating this, exp(0.40)=1.49, suggests that the expected count of expression_value is approximately 49% higher for each additional minor allele when the treatment is applied compared to when it is not.
- The p-value of 2.5e-14 indicates that this interaction effect is statistically significant at the 5% level, meaning there is strong evidence that the effect of genotype on expression_value is indeed influenced by whether or not the cells were treated with IFNg. Thus, this SNP is a significant reQTL.

Simulate both a real reQTL and no eQTL

```
# Parameters for the SNPs
snp_params <- tibble(
  n_donors = 80,
  p_minor_allele = 0.3,
  interaction_effect_magnitude = c(0.8, 0.25, 0),  # Large interaction effect vs no interaction effect
  snp_id = c("Strong reQTL SNP", "Weak reQTL SNP", "Regular SNP")
)

# Simulate data for each SNP and combine into one dataframe
combined_df <- pmap_df(snp_params, ~ simulate_expression_data(..1, ..2, ..3, ..4))

# View the first few rows of the combined dataframe
pander(head(combined_df))
```

snp_id	donor	IFNg_treatment	genotype	expression_value
Strong reQTL SNP	1	TRUE	1	34
Strong reQTL SNP	1	FALSE	1	16
Strong reQTL SNP	2	TRUE	0	20
Strong reQTL SNP	2	FALSE	0	21
Strong reQTL SNP	3	TRUE	1	36
Strong reQTL SNP	3	FALSE	1	12

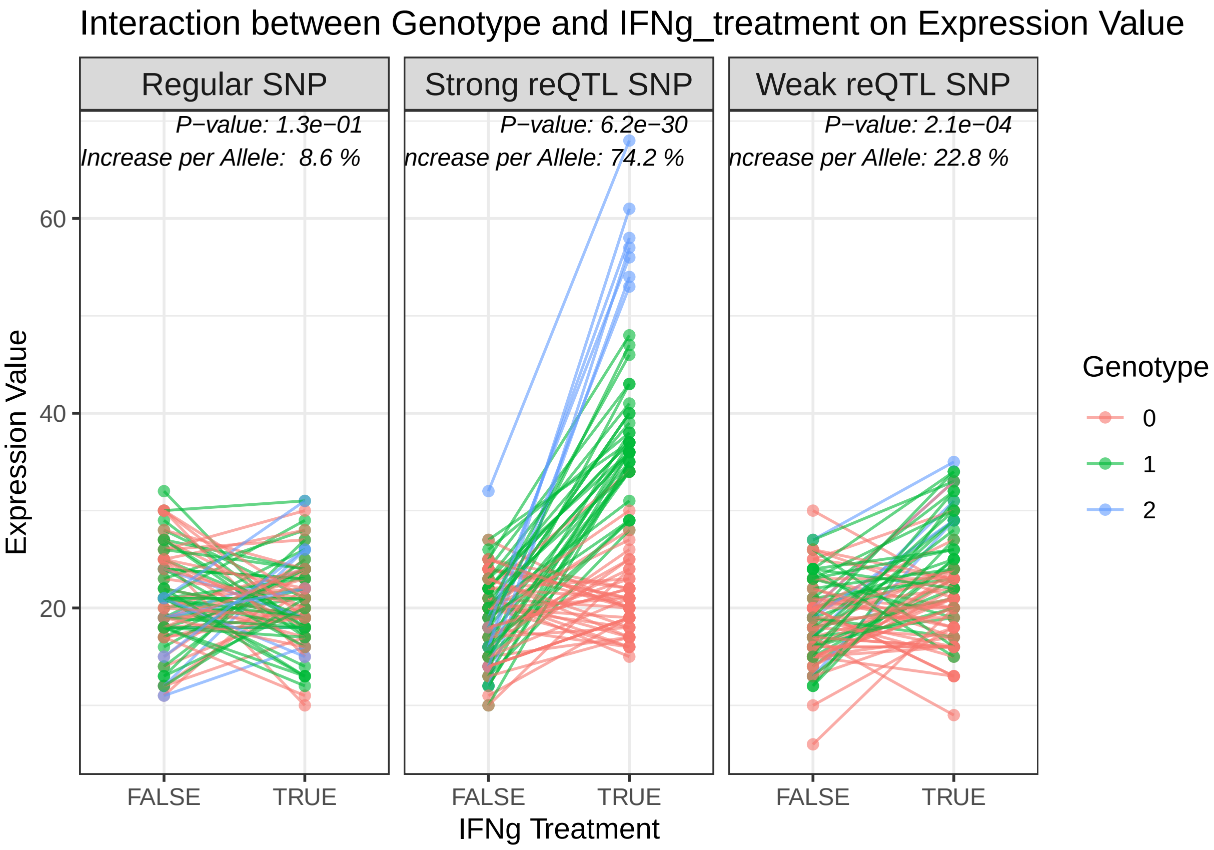
Fit the Negative Binomial Mixed Effects Regression Model for Each SNP

```r
# Function to fit Poisson regression model and extract the interaction term for each SNP
fit_poisson_and_extract <- function(data) {
  snp_ids <- unique(data$snp_id)

  # Define a function to fit the model and extract the interaction term for a given SNP
  fit_and_extract <- function(snp_id) {
    # Filter data for the current SNP
    df_snp <- data %>% filter(snp_id == {{ snp_id }})

    # Fit the Poisson regression model
    poisson_model <- glmer.nb(expression_value ~ IFNg_treatment * genotype + (1 | donor),
                           data = df_snp)

    # Clean up and pull the interaction term
    tidy_poisson <- tidy(poisson_model)
    interaction_term <- tidy_poisson %>%
      filter(str_detect(term, ":")) %>%
      select(term, estimate, p.value)

    # Add SNP identifier to the results
    interaction_term <- interaction_term %>%
      mutate(snp_id = snp_id) %>%
      select(snp_id, everything())

    return(interaction_term)
  }

  # Apply the function to each SNP and combine the results into one data frame
  results <- map_df(snp_ids, fit_and_extract)

  return(results)
}

sum_stats <- fit_poisson_and_extract(combined_df)
pander(sum_stats)
```

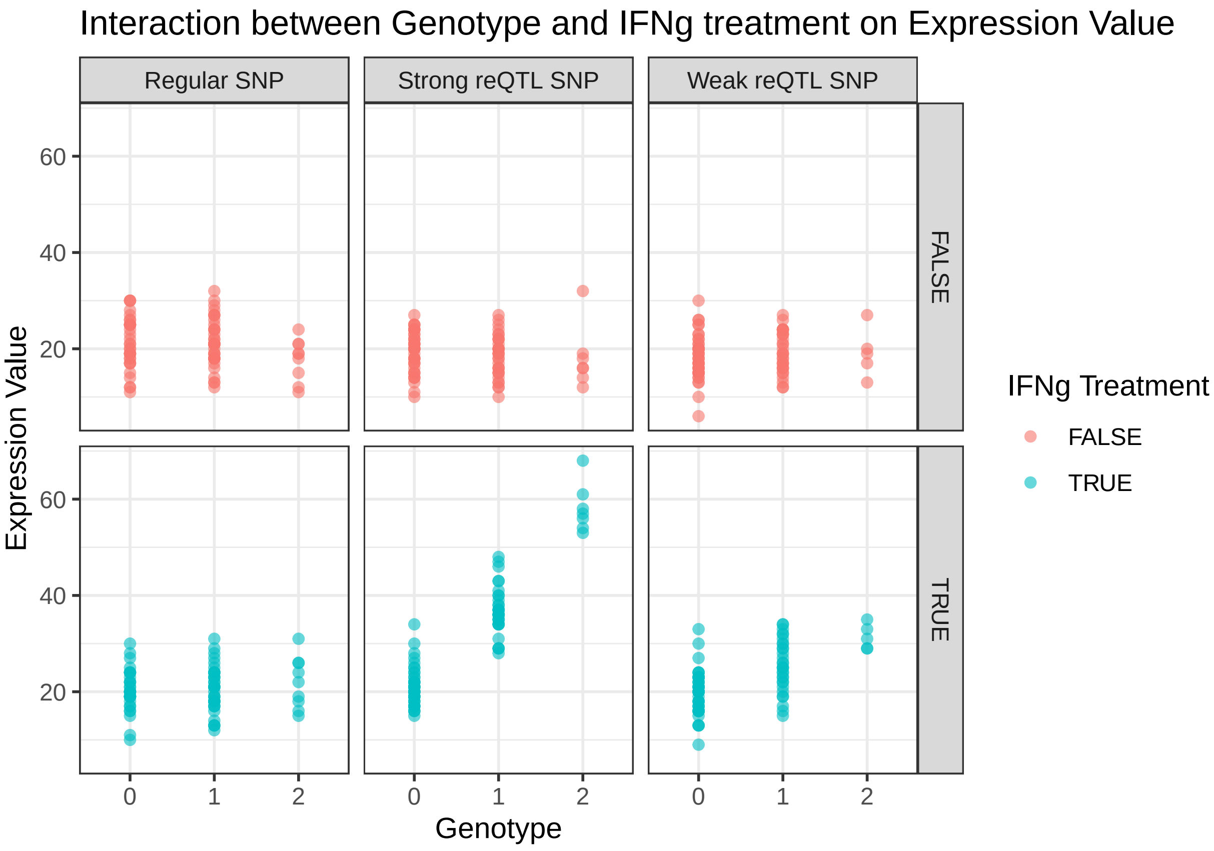
snp_id	term	estimate	p.value
Strong reQTL SNP	IFNg_treatmentTRUE:genotype	0.5552	6.206e-30
Weak reQTL SNP	IFNg_treatmentTRUE:genotype	0.2051	0.0002056
Regular SNP	IFNg_treatmentTRUE:genotype	0.08218	0.1332

Plot the Data with P-values to Demonstrate

```r
# Merge the SNP table with your combined_df
combined_df_annotated <- combined_df %>%
  left_join(sum_stats, by = "snp_id") %>%
  mutate(`Percentage Increase` = (exp(estimate) - 1) * 100)

# Create the plot
ggplot(combined_df_annotated, aes(x = factor(IFNg_treatment),
                                  y = expression_value, color = factor(genotype))) +
  geom_point(alpha = 0.6) +
  geom_line(aes(group = donor, color = factor(genotype)), alpha = 0.6) +
  labs(
    title = "Interaction between Genotype and IFNg_treatment on Expression Value",
    x = "IFNg Treatment",
    y = "Expression Value",
    color = "Genotype"
  ) +
  facet_grid(".~snp_id") +
  theme_bw() +
  geom_text(data = combined_df_annotated %>% distinct(snp_id, p.value, `Percentage Increase`),
            aes(x = Inf, y = Inf, label = paste("P-value:",
                                                format(p.value, digits = 2),
                                                "\n Increase per Allele:",
                                                format(`Percentage Increase`, digits = 2),
                                                "%")),
            hjust = 1.1, vjust = 1.1, size = 3, color = "black", fontface = "italic") +
  theme(strip.text = element_text(size = 12)) # Adjust facet label text size if needed
```


Interaction between Genotype and IFNg_treatment on Expression Value
Regular SNP
Strong reQTL SNP
Weak reQTL SNP
P−value: 1.3e−01
Increase per Allele: 8.6 %
P−value: 6.2e−30
ncrease per Allele: 74.2 %
P−value: 2.1e−04
ncrease per Allele: 22.8 %
Expression Value
60
40
20
FALSE
TRUE
FALSE
TRUE
FALSE
TRUE
IFNg Treatment
Genotype
0
1
2

```r
# Create the plot
ggplot(combined_df_annotated, aes(x = factor(genotype),
                                  y = expression_value, color = factor(IFNg_treatment))) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Interaction between Genotype and IFNg treatment on Expression Value",
    x = "Genotype",
    y = "Expression Value",
    color = "IFNg Treatment"
  ) +
  facet_grid("factor(IFNg_treatment)~snp_id") +
  theme_bw()
```

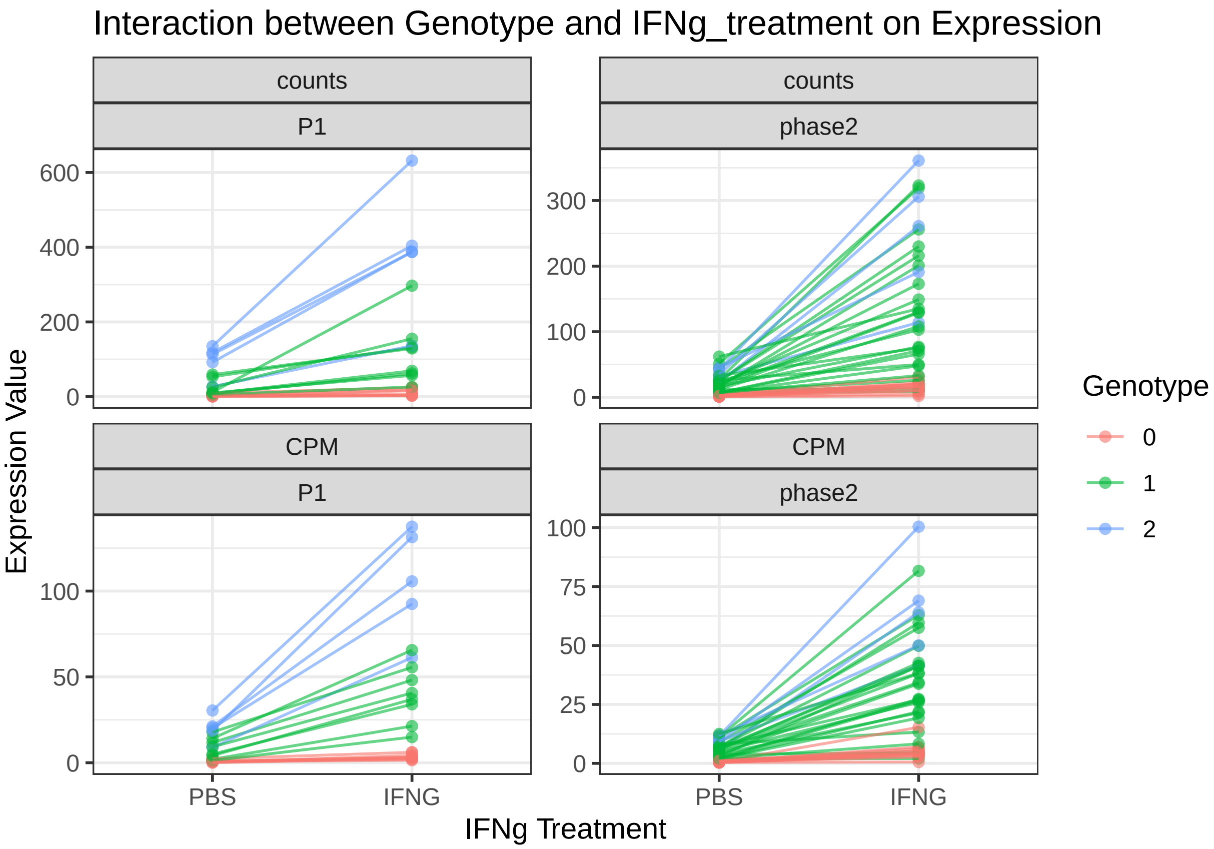


Explore the real data

```r
#load the real data
real_dat <- read.csv(here::here("notebooks/long_form_reQTL_data.csv")) %>%
  arrange(donor)

#make sure the reference group is set to "PBS"
real_dat$condition <- relevel(factor(real_dat$condition), ref = "PBS")
real_dat$genotype.nt <- relevel(factor(real_dat$genotype.nt), ref = "TT")
```

Plot the real data

```r
# Create the plot

# Reshape to long format for faceting
real_dat_plot <- real_dat %>%
  pivot_longer(cols = c(counts, CPM), names_to = "scale_type", values_to = "expression_value")

# Create the plot
ggplot(real_dat_plot, aes(x = factor(condition),
                          y = expression_value,
                          color = factor(genotype.num))) +
  geom_point(alpha = 0.6) +
  geom_line(aes(group = donor), alpha = 0.6) +
  labs(
    title = "Interaction between Genotype and IFNg_treatment on Expression",
    x = "IFNg Treatment",
    y = "Expression Value",
    color = "Genotype"
  ) +
  facet_wrap(scale_type ~ phase, scales = "free_y") +
  theme_bw()
```



Fit a linear mixed effects model

```
# Fit the Poisson regression model
real_linear_model <- lmer(counts ~ condition * genotype.num  +
                           #(1 + condition | phase) + #random slope for phase
                           (1 | phase/donor), #random intercept for donor, nested within phase
                          data = real_dat)

# Clean up and pull the interaction term
real_tidy_linear <- tidy(real_linear_model)
real_interaction_term <- real_tidy_linear %>%
  filter(str_detect(term, ":")) %>%
  select(term, estimate, p.value)

real_tidy_linear
```

```
## # A tibble: 7 x 8
##   effect   group       term         estimate std.error statistic    df   p.value
##   <chr>    <chr>       <chr>           <dbl>     <dbl>     <dbl> <dbl>     <dbl>
## 1 fixed    <NA>        (Intercept)    -0.190      15.5   -0.0123  4.37  9.91e- 1
## 2 fixed    <NA>        conditionI~     1.21       15.5    0.0778 59.3   9.38e- 1
## 3 fixed    <NA>        genotype.n~    29.4        12.3    2.39  104.    1.88e- 2
## 4 fixed    <NA>        conditionI~   115.         14.4    8.00   59.3   5.37e-11
## 5 ran_pars donor:phase sd__(Inter~    36.3        NA     NA        NA    NA
## 6 ran_pars phase       sd__(Inter~    10.9        NA     NA        NA    NA
## 7 ran_pars Residual    sd__Observ~    54.2        NA     NA        NA    NA
```

Fit the Negative Binomial Mixed Effects Regression Model on the Real Data

```
# Fit the Poisson regression model
real_poisson_model <- glmer.nb(counts ~ condition * genotype.num +
                                 (1 | phase) +
                                 (1 | phase:donor),
                               data = real_dat)

# Clean up and pull the interaction term
real_tidy_poisson <- tidy(real_poisson_model)
real_interaction_term <- real_tidy_poisson %>%
  filter(str_detect(term, ":")) %>%
  select(term, estimate, p.value)

real_tidy_poisson
```

```
## # A tibble: 6 x 7
##   effect   group       term                 estimate std.error statistic   p.value
##   <chr>    <chr>       <chr>                   <dbl>     <dbl>     <dbl>     <dbl>
## 1 fixed    <NA>        (Intercept)          9.91e-1     0.191     5.18   2.25e- 7
## 2 fixed    <NA>        conditionIFNG        1.66e+0     0.161    10.3    6.96e-25
## 3 fixed    <NA>        genotype.num         1.68e+0     0.164    10.2    1.23e-24
## 4 fixed    <NA>        conditionIFNG:gen~   1.62e-2     0.134     0.121  9.04e- 1
## 5 ran_pars phase:donor sd__(Intercept)      6.43e-1    NA        NA     NA
## 6 ran_pars phase       sd__(Intercept)      1.41e-6    NA        NA     NA
```

```
# Fit a reduced model
real_poisson_model_red <- glmer.nb(counts ~ condition + genotype.nt + (1 | donor),
                           data = real_dat)

# Fit the Poisson regression model
real_poisson_model <- glmer.nb(counts ~ condition * genotype.nt + (1 | donor),
                           data = real_dat)

# Do a likelihood ratio test
anova(real_poisson_model_red, real_poisson_model, test = "LRT")
```

```
## Data: real_dat
## Models:
## real_poisson_model_red: counts ~ condition + genotype.nt + (1 | donor)
## real_poisson_model: counts ~ condition * genotype.nt + (1 | donor)
##                        npar    AIC    BIC  logLik -2*log(L)  Chisq Df
## real_poisson_model_red    6 1043.4 1060.2 -515.69    1031.4
## real_poisson_model        8 1047.1 1069.6 -515.57    1031.1 0.2461  2
##                        Pr(>Chisq)
## real_poisson_model_red
## real_poisson_model         0.8842
```