qpcr_linear_model

Nate Olson

2016-10-05

Summary

- Objective - compare linear modeling methods for qPCR bacterial abundance data
- Method - bacterial abundance quantified using zymo kit
- Results
  - R^2 standard curve 0.99
  - Large fraction of samples (especially unmixed) with Ct values outside standard curve.

Objective

The proportion of pre and post exposure samples in individual titrations is dependent on the ratios at which the two samples were mixed. This assumes that the pre and post samples have equivalent proportions of bacterial to non-bacterial DNA. To validate this assumption the concentration of bacterial DNA was assayed using qPCR. Additionally, the concentation of bacterial DNA in the titrations was assayed.

Methods

- zymo qPCR assay - https://www.zymoresearch.com/dna/dna-analysis/femto-bacterial-dna-quantification-kit
- 45 Samples - all mixed and unmixed
- diluted samples - need to find out how they were diluted
- triplicates per sample - 135 reactions
  - three qPCR plates, one replicate of each sample ran on each plate
- 7 concentration standard curve - for the assay
  - issue with fourth standard

Munging qPCR Data

```
bac_con <- read_excel(path = "../data/MixStudy_Nate_20160919.xls",
                      sheet = "QDNA_20160919",
                      skip = 11, na = "Undetermined", col_names = FALSE)
colnames(bac_con) <- c("well","sample_name",
                       "plate1_Ct","plate1_quant",
                       "plate2_Ct","plate2_quant",
                       "plate3_Ct","plate3_quant")
bac_con <- bac_con %>% gather("id","value", -well, -sample_name) %>%
      separate(id, c("plate","var"), sep = "_") %>%
      spread(var,value)
```

```
bac_std <- read_excel(path = "../data/MixStudy_Nate_20160919.xls",
                      sheet = "QDNA_20160919",skip = 3, col_names = FALSE) %>%
      select(-X12, -X13, -X5,-X7,-X9) %>% filter(X2 %in% paste0("Std",1:7))
colnames(bac_std) <- c("well","sample_name","conc","plate1","plate2","plate3")
bac_std <- bac_std %>% gather("plate","Ct",-well, -sample_name, -conc) %>%
      mutate(conc = as.numeric(conc), Ct = as.numeric(Ct)) %>% filter(!is.na(Ct))
```

```
## Warning in eval(substitute(expr), envir, enclos): NAs introduced by
## coercion
```

```
## NAs introduced when converting samples with undetermined and omit Ct values.
```

Analysis of Standard Curve

Fitted model using the three plates as replicates and not fitting individual models for each plate. Using a random effect model is likely a better approach.

Fitting the standard curve using linear models

```
bac_std <- mutate(bac_std, log_conc = log10(conc))
std_fit1 <- lm(log_conc~Ct, data = bac_std)
# extracting values for plots
std_slope <- std_fit1$coefficients[2]
std_intercept <- std_fit1$coefficients[1]
fit1_df <- tibble(plate = paste0("plate",c(1:3)),
                  slope = std_fit1$coefficients[2],
                  intercept = std_fit1$coefficients[1],
                  mod = "log_conc~Ct")
bac_fit <- bac_con %>% select(sample_name,plate, Ct) %>% add_predictions(std_fit1)
```

```
fit1_df %>% knitr::kable()
```

plate	slope	intercept	mod
plate1	-0.2131823	4.365381	log_conc~Ct
plate2	-0.2131823	4.365381	log_conc~Ct
plate3	-0.2131823	4.365381	log_conc~Ct

```
std_fit2 <- lm(log_conc~plate/Ct, data = bac_std)
bac_fit <- bac_con %>% select(sample_name,plate, Ct) %>% add_predictions(std_fit2)
std_coef <- std_fit2 %>% coefficients()
coef_df <- frame_data(
      ~plate, ~intercept, ~slope,
      "plate1",std_coef[1], std_coef[4],
      "plate2",std_coef[1] + std_coef[2], std_coef[5],
      "plate3",std_coef[1] + std_coef[3], std_coef[6]
)
fit2_df <- coef_df %>% mutate(mod = "log_conc~plate/Ct")
```

Question Why do the "log_conc~plate/Ct" and "log_conc~plate*Ct" fit the data the same? Only appearant difference is that the t-test for the Ct:plate interactions for the "log_conc~plate/Ct" model are calculated independently and making them all significant.

```r
fit2_df %>% knitr::kable()
```

plate	intercept	slope	mod
plate1	4.417411	-0.2139791	log_conc~plate/Ct
plate2	4.828613	-0.2314146	log_conc~plate/Ct
plate3	4.070008	-0.2042542	log_conc~plate/Ct

```r
std_fit3 <- lm(log_conc~plate*Ct, data = bac_std)
std_coef <- std_fit3 %>% coefficients()
## not sure if I need to add Ct
fit3_df <- frame_data(
      ~plate, ~intercept, ~slope,
      "plate1", std_coef[1], std_coef[4],
      "plate2",std_coef[1] + std_coef[2], std_coef[4] + std_coef[5],
      "plate3", std_coef[1] + std_coef[3], std_coef[4] + std_coef[6]
      ) %>%
      mutate(mod = "log_conc~plate*Ct")
```

```r
fit3_df %>% knitr::kable()
```

plate	intercept	slope	mod
plate1	4.417411	-0.2139791	log_conc~plate*Ct
plate2	4.828613	-0.2314146	log_conc~plate*Ct
plate3	4.070008	-0.2042542	log_conc~plate*Ct

Question Why do all of the plates have the same slope and intercept for random effects model?

```r
std_fit4 <- lmer(log_conc~ Ct + (1|plate), data = bac_std)
fit4_df <- tibble(plate = paste0("plate",c(1:3)),
                  slope = coef(std_fit4)$plate[1,2],
                  intercept = coef(std_fit4)$plate[1,1],
                  mod = "log_conc~ Ct + (1|plate)")
```

```r
fit4_df %>% knitr::kable()
```

plate	slope	intercept	mod
plate1	-0.2131823	4.365381	log_conc~ Ct + (1|plate)
plate2	-0.2131823	4.365381	log_conc~ Ct + (1|plate)
plate3	-0.2131823	4.365381	log_conc~ Ct + (1|plate)

Model Fit Summary

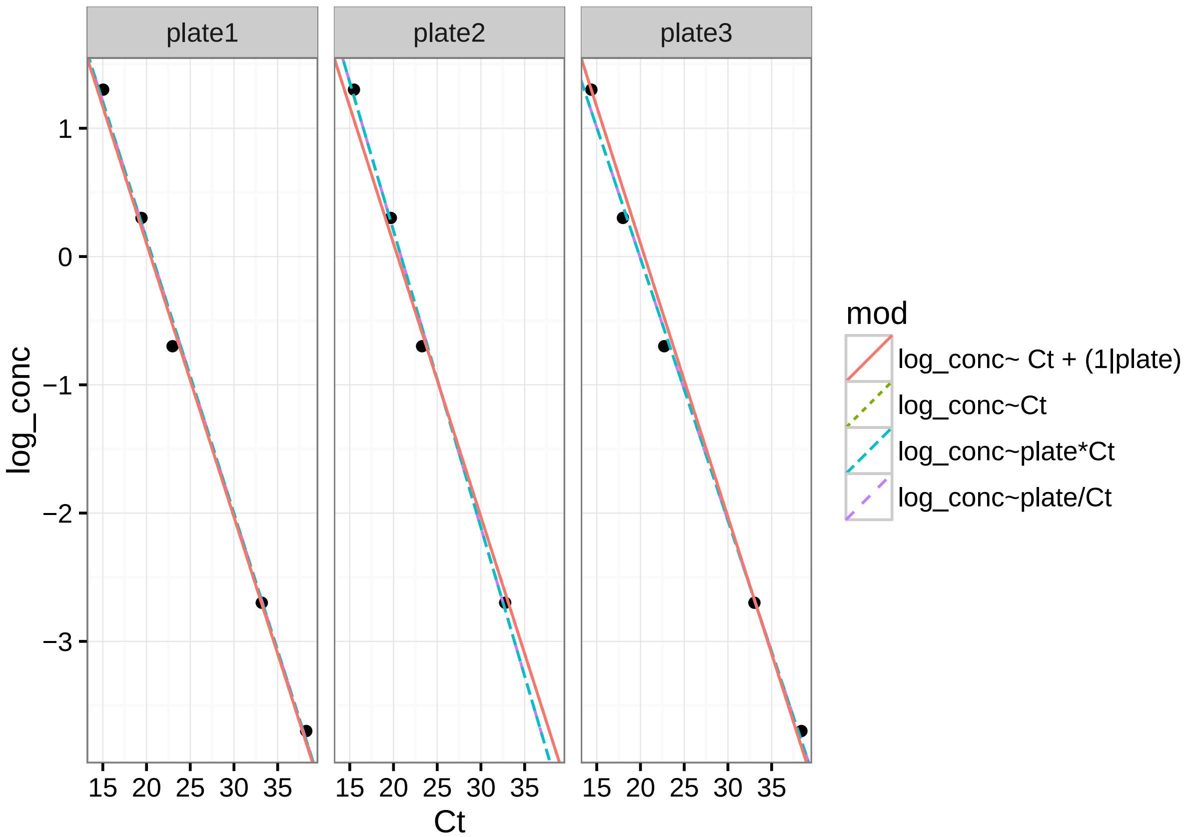
```r
fit_summary_df <-list(std_fit1, std_fit2, std_fit3, std_fit4) %>%
      map_df(glance, .id = "fit")
fit_summary_df$mod <- c("log_conc~Ct","log_conc~plate/Ct",
                        "log_conc~plate*Ct","log_conc~ Ct + (1|plate)")
```

```r
fit_summary_df %>% knitr::kable()
```

fit	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
1	0.9939065	0.9933987	0.1500805	1957.313	0	2	7.7660805	-9.532161	-7.614989	0.2702900
2	0.9967288	0.9946843	0.1346755	487.521	0	6	12.1205896	-10.241179	-5.767778	0.1450999
3	0.9967288	0.9946843	0.1346755	487.521	0	6	12.1205896	-10.241179	-5.767778	0.1450999
4	NA	NA	0.1500805	NA	NA	NA	0.9735191	6.052962	8.609191	-15.5321610

Comparing Regression Models

```
fit_coefs <- bind_rows(fit1_df, fit2_df, fit3_df,fit4_df)
ggplot(fit_coefs) + geom_point(data = bac_std, aes(y = log_conc, x = Ct)) +
      geom_abline(aes(slope = slope, intercept = intercept, color = mod, linetype = mod)) +
      facet_grid(.~plate) + theme_bw()
```



Model Summaries

log_conc~Ct

```
summary(std_fit1)
```

```
##
## Call:
## lm(formula = log_conc ~ Ct, data = bac_std)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
```

```
## -0.228775 -0.096812  0.006189  0.116039  0.235924
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.365381   0.125876   34.68 2.11e-13 ***
## Ct          -0.213182   0.004819  -44.24 1.16e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1501 on 12 degrees of freedom
## Multiple R-squared:  0.9939, Adjusted R-squared:  0.9934
## F-statistic:  1957 on 1 and 12 DF,  p-value: 1.158e-14
```

log_conc~plate/Ct

```r
summary(std_fit2)
```

```
##
## Call:
## lm(formula = log_conc ~ plate/Ct, data = bac_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19978 -0.07599  0.03538  0.06859  0.16901
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.417411   0.189273  23.339 1.21e-08 ***
## plateplate2     0.411203   0.313488   1.312    0.226
## plateplate3    -0.347403   0.260241  -1.335    0.219
## plateplate1:Ct -0.213979   0.006960 -30.743 1.36e-09 ***
## plateplate2:Ct -0.231415   0.010552 -21.930 1.97e-08 ***
## plateplate3:Ct -0.204254   0.006644 -30.741 1.36e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1347 on 8 degrees of freedom
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9947
## F-statistic: 487.5 on 5 and 8 DF,  p-value: 1.029e-09
```

log_conc~plate*Ct

```r
summary(std_fit3)
```

```
##
## Call:
## lm(formula = log_conc ~ plate * Ct, data = bac_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19978 -0.07599  0.03538  0.06859  0.16901
##
```

```
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.417411   0.189273  23.339 1.21e-08 ***
## plateplate2     0.411203   0.313488   1.312    0.226
## plateplate3    -0.347403   0.260241  -1.335    0.219
## Ct             -0.213979   0.006960 -30.743 1.36e-09 ***
## plateplate2:Ct -0.017436   0.012641  -1.379    0.205
## plateplate3:Ct  0.009725   0.009623   1.011    0.342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1347 on 8 degrees of freedom
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9947
## F-statistic: 487.5 on 5 and 8 DF,  p-value: 1.029e-09
```

log_conc~ Ct + (1|plate)

```r
summary(std_fit4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_conc ~ Ct + (1 | plate)
##    Data: bac_std
##
## REML criterion at convergence: -1.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.52435 -0.64507  0.04124  0.77318  1.57198
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  plate    (Intercept) 2.520e-17 5.020e-09
##  Residual             2.252e-02 1.501e-01
## Number of obs: 14, groups:  plate, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  4.365381   0.125876   34.68
## Ct          -0.213182   0.004819  -44.24
##
## Correlation of Fixed Effects:
##    (Intr)
## Ct -0.948
```