Bacterial Abundance qPCR

Nate Olson

2016-09-27

Summary

- Objective - quantify bacterial abundance in starting DNA
- Method - bacterial abundance quantified using zymo kit
- Results
  - clean no template controls
  - R^2 standard curve 0.99
  - Large fraction of samples (especially unmixed) with Ct values outside standard curve.
- Conclusion - need to rerun samples, specifically unmixed samples, diluting so that Ct values are within standard curve.

Questions

- How much template was added to each reaction?

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.
```

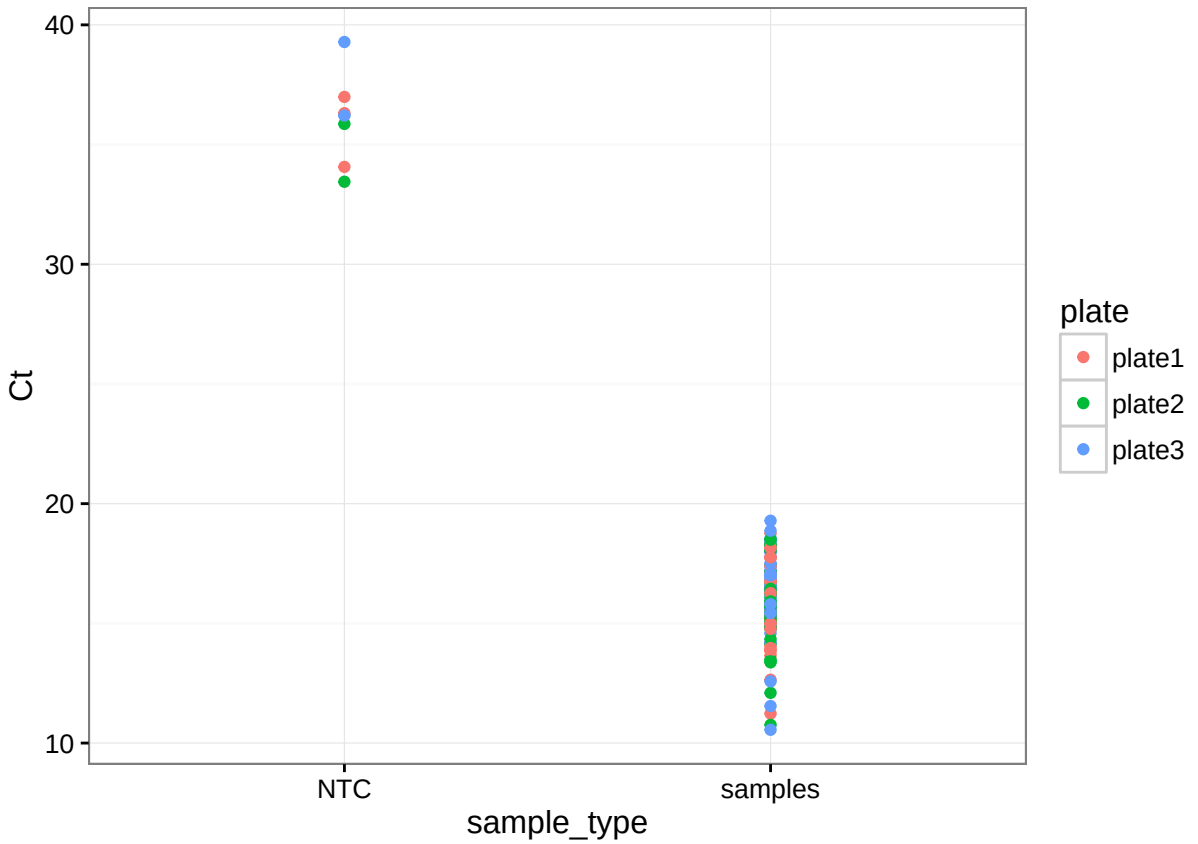
Results

NTC Check

For NTC (no template control) samples the Ct values were Comparison of Ct values between NTC and samples. NTC Ct values > 30, and sample Ct values < 20, excluding NTC from the rest of the analysis.

```
bac_con %>% mutate(sample_type = if_else(sample_name == "NTC", "NTC","samples")) %>%
ggplot() + geom_point(aes(x = sample_type, y = Ct, color = plate)) + theme_bw()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
## Warning about missing value due to one NTC reaction with a Ct value of NA

## plot shows okay to exclude NTC
bac_con <- bac_con %>% filter(sample_name != "NTC")
```

Analysis of Standard Curve

Fitting the standard curve using a linear model, Ct~log10(concentration).

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.

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

```
summary(std_fit)
```

```
##
## 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
```

```
std_fit2 <- lm(log_conc~plate/Ct, data = bac_std)
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
```

```
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")
```

```
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")
```

```
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
```

```
fit3_df
```

```
## # A tibble: 3 × 4
##    plate intercept      slope              mod
##    <chr>     <dbl>      <dbl>            <chr>
## 1 plate1  4.417411 -0.2139791 log_conc~plate*Ct
## 2 plate2  4.828613 -0.2314146 log_conc~plate*Ct
## 3 plate3  4.070008 -0.2042542 log_conc~plate*Ct
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
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)")
```

```
coef(std_fit4)
```

```
## $plate
##        (Intercept)         Ct
## plate1    4.365381 -0.2131823
## plate2    4.365381 -0.2131823
## plate3    4.365381 -0.2131823
##
## attr(,"class")
## [1] "coef.mer"
```

```
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
```

Comparing Regression Models

Model Matrix

```
std_fit$model
```

```
##    log_conc     Ct
## 1   1.30103 15.042
## 2   0.30103 19.425
## 3  -0.69897 22.977
## 4  -2.69897 33.186
## 5  -3.69897 38.270
## 6   1.30103 15.481
## 7   0.30103 19.697
## 8  -0.69897 23.272
```

```
## 9  -2.69897 32.773
## 10  1.30103 14.384
## 11  0.30103 17.992
## 12 -0.69897 22.717
## 13 -2.69897 33.033
## 14 -3.69897 38.407
```

```
std_fit2$model
```

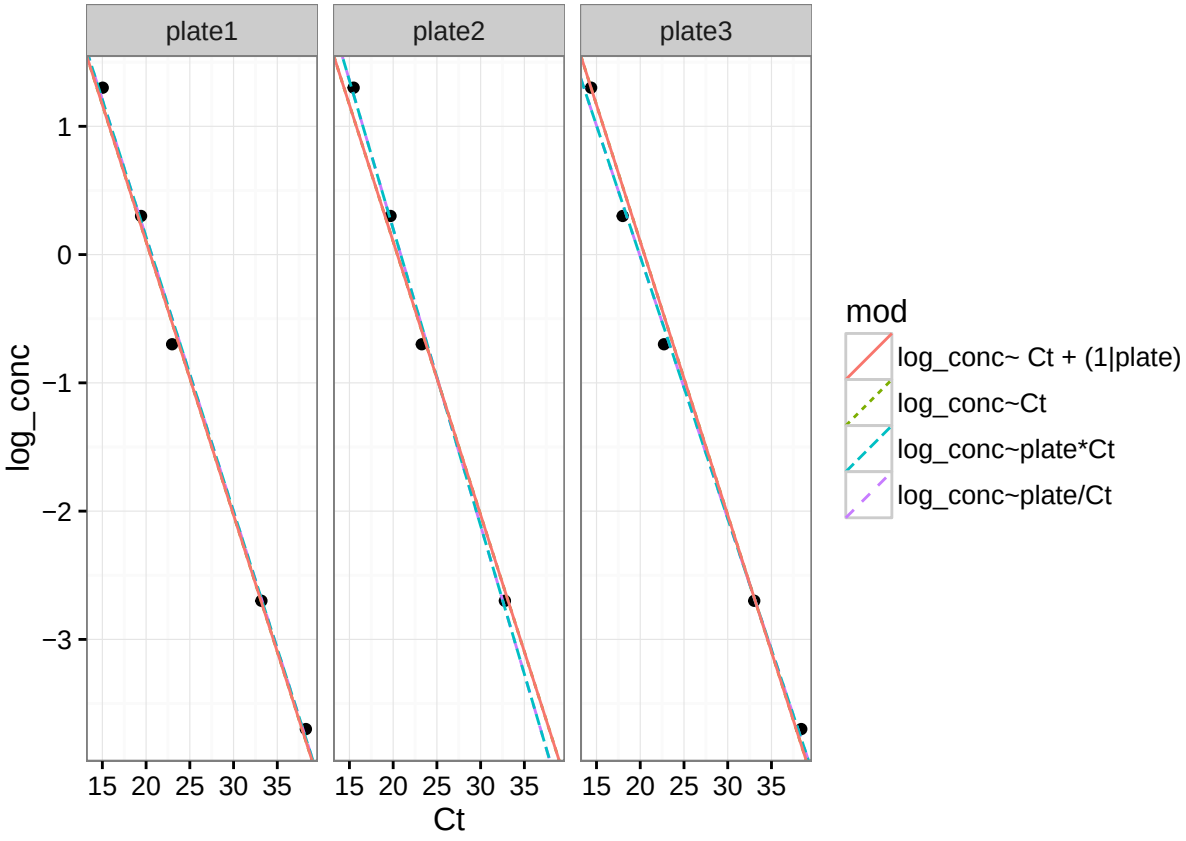
```
##    log_conc  plate     Ct
## 1   1.30103 plate1 15.042
## 2   0.30103 plate1 19.425
## 3  -0.69897 plate1 22.977
## 4  -2.69897 plate1 33.186
## 5  -3.69897 plate1 38.270
## 6   1.30103 plate2 15.481
## 7   0.30103 plate2 19.697
## 8  -0.69897 plate2 23.272
## 9  -2.69897 plate2 32.773
## 10  1.30103 plate3 14.384
## 11  0.30103 plate3 17.992
## 12 -0.69897 plate3 22.717
## 13 -2.69897 plate3 33.033
## 14 -3.69897 plate3 38.407
```

```
std_fit3$qr
```

```
## $qr
##    (Intercept) plateplate2 plateplate3           Ct plateplate2:Ct
## 1   -3.7416574  -1.0690450 -1.33630621 -92.647713076  -2.438037e+01
## 2    0.2672612   1.6903085 -0.84515425  -4.627304060   3.854875e+01
## 3    0.2672612   0.1333828  1.58113883  -0.748511122  -3.082758e-15
## 4    0.2672612   0.1333828  0.22016954 -30.791337969  -5.289791e+00
## 5    0.2672612   0.1333828  0.22016954   0.495148219   1.161455e+01
## 6    0.2672612  -0.4582252  0.03419583  -0.217992572   5.560054e-01
## 7    0.2672612  -0.4582252  0.03419583  -0.081070948   2.398987e-01
## 8    0.2672612  -0.4582252  0.03419583   0.035033133  -2.814719e-02
## 9    0.2672612  -0.4582252  0.03419583   0.343593937  -7.405120e-01
## 10   0.2672612   0.1333828 -0.41228599  -0.266174682  -9.114631e-02
## 11   0.2672612   0.1333828 -0.41228599  -0.148998871  -5.102175e-02
## 12   0.2672612   0.1333828 -0.41228599   0.004453376   1.524971e-03
## 13   0.2672612   0.1333828 -0.41228599   0.339482663   1.162492e-01
## 14   0.2672612   0.1333828 -0.41228599   0.514012266   1.760134e-01
##    plateplate3:Ct
## 1    -33.81736672
## 2    -21.38798066
## 3     40.01324792
## 4    -13.34239940
## 5     -6.07673005
## 6    -13.99585865
## 7      0.04184597
## 8     -0.02320216
## 9     -0.19607552
## 10    -0.58345600
## 11    -0.41237173
```

```
## 12    -0.18832151
## 13     0.30084294
## 14     0.55566746
## attr(,"assign")
## [1] 0 1 1 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$plate
## [1] "contr.treatment"
##
##
## $qraux
## [1] 1.267261 1.133383 1.220170 1.330037 1.169554 1.118557
##
## $pivot
## [1] 1 2 3 4 5 6
##
## $tol
## [1] 1e-07
##
## $rank
## [1] 6
##
## attr(,"class")
## [1] "qr"
```

```
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()
```



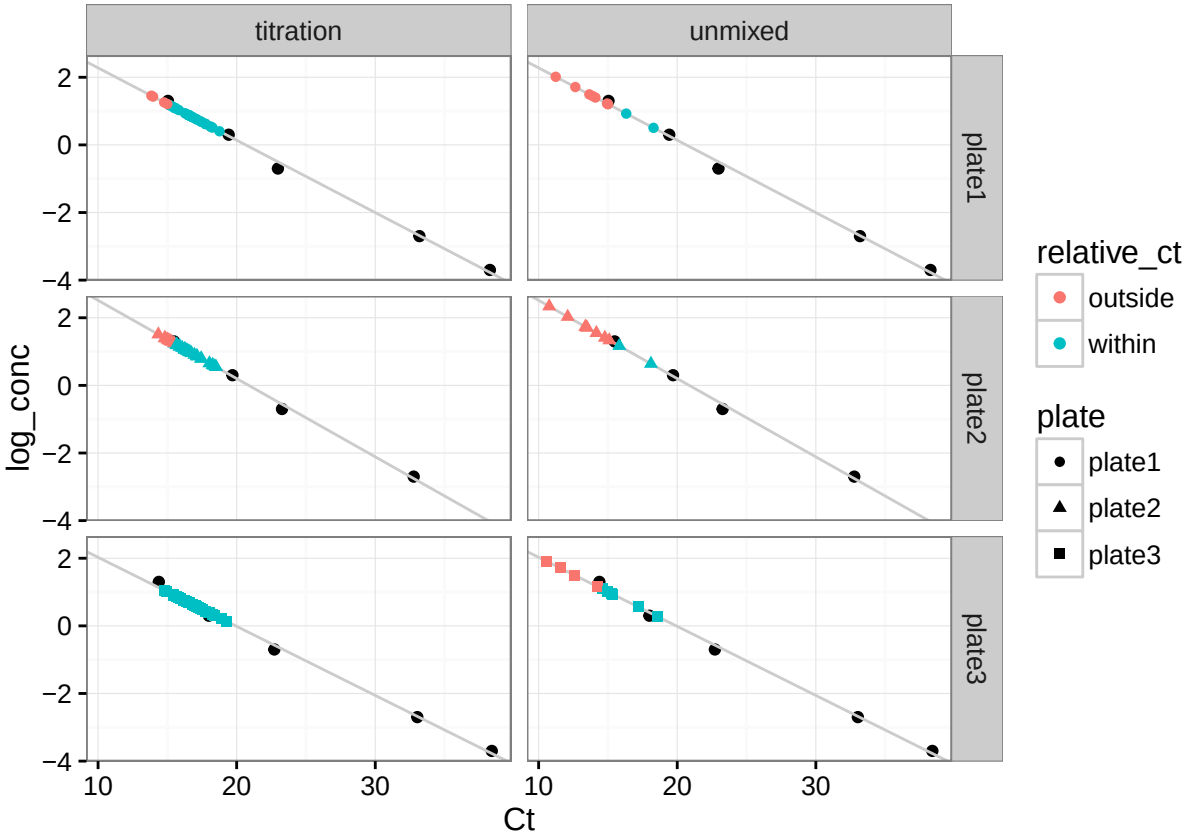
Samples outside of standard curve range.

Samples have Ct values lower then the standard curve samples and fall outside of the standard curve. This is not ideal. Only 2 of the 10 unmixed pre and post treatment samples have Ct values within the standard curve for all three plates, and none of the remaining 8 have Ct within the standard curve for 2 of the three plates. The linearity of the standard curve outside of the standard concentration range is unknown.

```
min_std <- bac_std %>% filter(conc == 20) %>% select(plate, Ct) %>% rename(std_ct = Ct)
bac_fit <- left_join(bac_fit, min_std) %>%
      mutate(relative_ct = if_else(Ct < std_ct, "outside","within")) %>%
      mutate(sam_type = if_else(grepl("\\(",sample_name),"unmixed","titration"))
```

```
## Joining, by = "plate"
```

```
# ggplot(bac_std) + geom_point(aes(y = log_conc, x = Ct)) +
#       geom_abline(aes(slope = std_slope, intercept = std_intercept), color = "grey80") +
#       geom_point(data = bac_fit, aes(y = pred, x = Ct, shape = plate, color = relative_ct)) +
#       facet_grid(plate~sam_type) +
#       theme_bw()
ggplot(bac_std) + geom_point(aes(y = log_conc, x = Ct)) +
      geom_abline(data = coef_df, aes(slope = slope, intercept = intercept), color = "grey80") +
      geom_point(data = bac_fit, aes(y = pred, x = Ct, shape = plate, color = relative_ct)) +
      facet_grid(plate~sam_type) +
      theme_bw()
```



Difference in Ct value between unmixed samples and highest concentration (lowest Ct) standard. Samples with negative values are outside of the standard curve.

```
bac_fit %>% filter(sam_type == "unmixed") %>%
            mutate(sample_name = str_replace(sample_name, ".*_", ""),
             sample_name = str_replace(sample_name, '\\(',"_"),
             sample_name = str_replace(sample_name, '\\)',"")) %>%
      mutate(ct_diff = std_ct - Ct) %>% select(sample_name, plate, ct_diff) %>%
      spread(plate,ct_diff) %>% kable()
```

sample_name	plate1	plate2	plate3
1 Post	1.381	1.987	-0.572
1 Pre	0.029	0.384	-0.954
2 Post	3.810	4.720	3.829
2 Pre	0.056	2.111	-0.203
3 Post	2.395	3.384	2.836
3 Pre	-1.278	-0.336	-2.801
4 Post	0.107	0.705	-0.886
4 Pre	0.946	1.314	0.164
5 Post	1.153	2.067	1.806
5 Pre	-3.246	-2.623	-4.171

Variability of samples outside of the standard curve

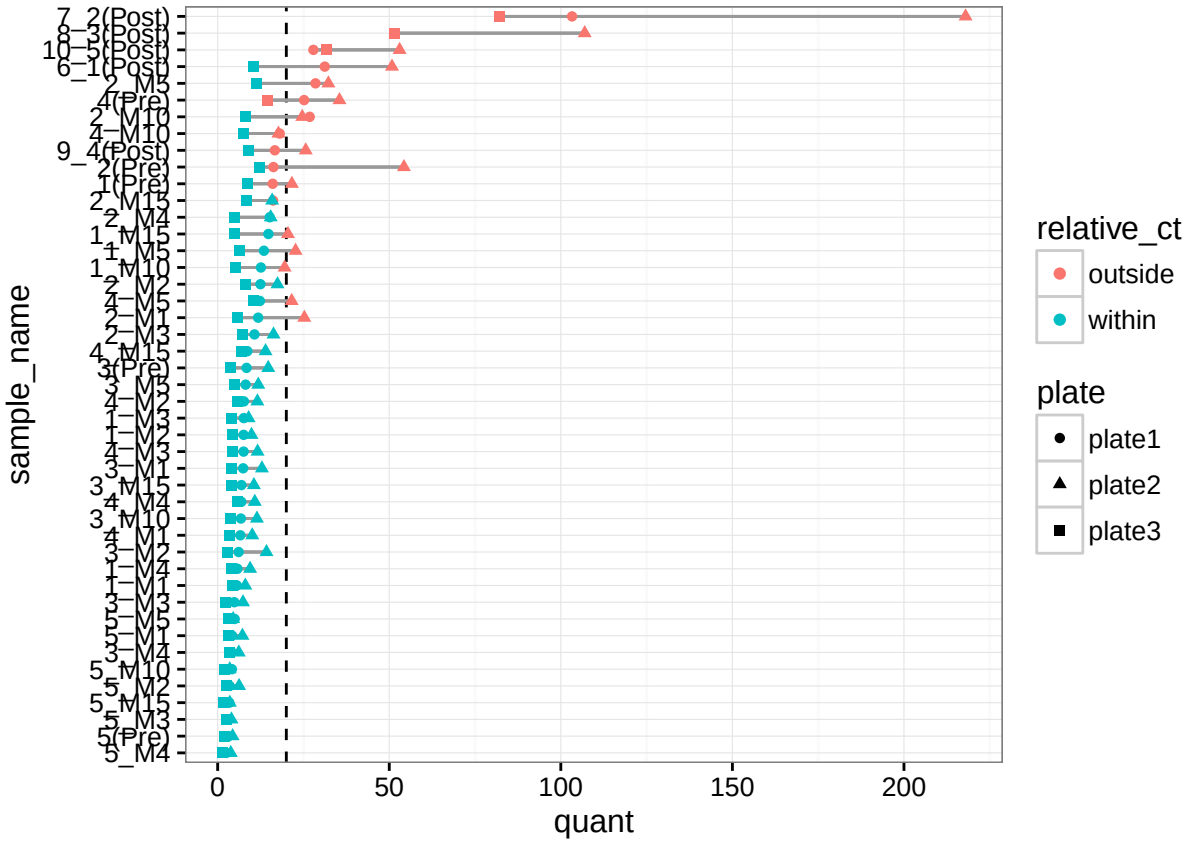
The variability in concentration for the unmixed samples, especially post treatment biological replicates 2 (7_2(Post)) and 3 (8_3(Post)), is likely due to the sample concentration falling outside the range of the

standard curve.

Dashed line is at 20ng/ul, to indicate the highest concentation standard.

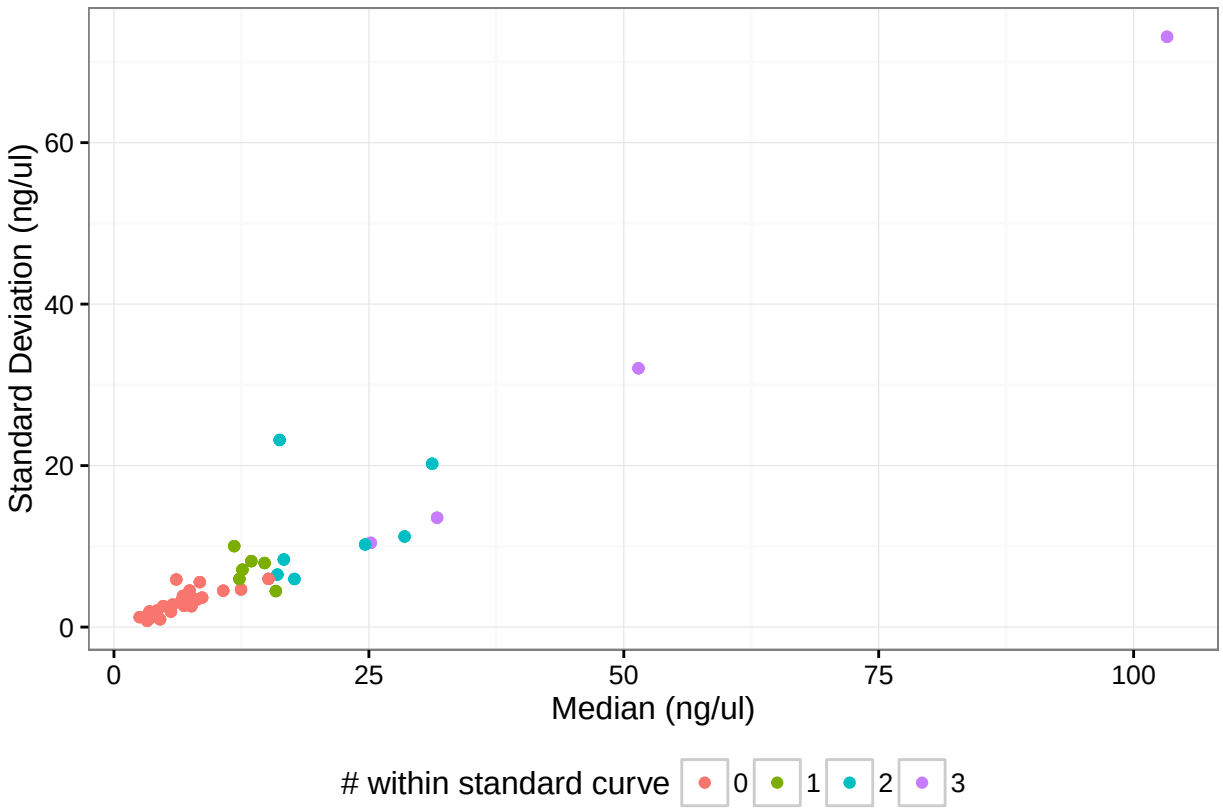
```
bac_fit <- bac_fit %>%
      mutate(quant = 10^pred) %>%
      group_by(sample_name) %>%
      mutate(quant_min = min(quant), quant_max = max(quant)) %>%
      ungroup() %>%
      mutate(sample_name = fct_reorder(sample_name, quant, fun = "median"))

ggplot(bac_fit) +
      geom_hline(aes(yintercept = 20), linetype = 2) +
      geom_linerange(aes(x = sample_name,ymin = quant_min, ymax = quant_max), color = "grey60") +
      geom_point(aes(y = quant, x = sample_name, color = relative_ct, shape = plate)) +
      theme_bw() + coord_flip()
```



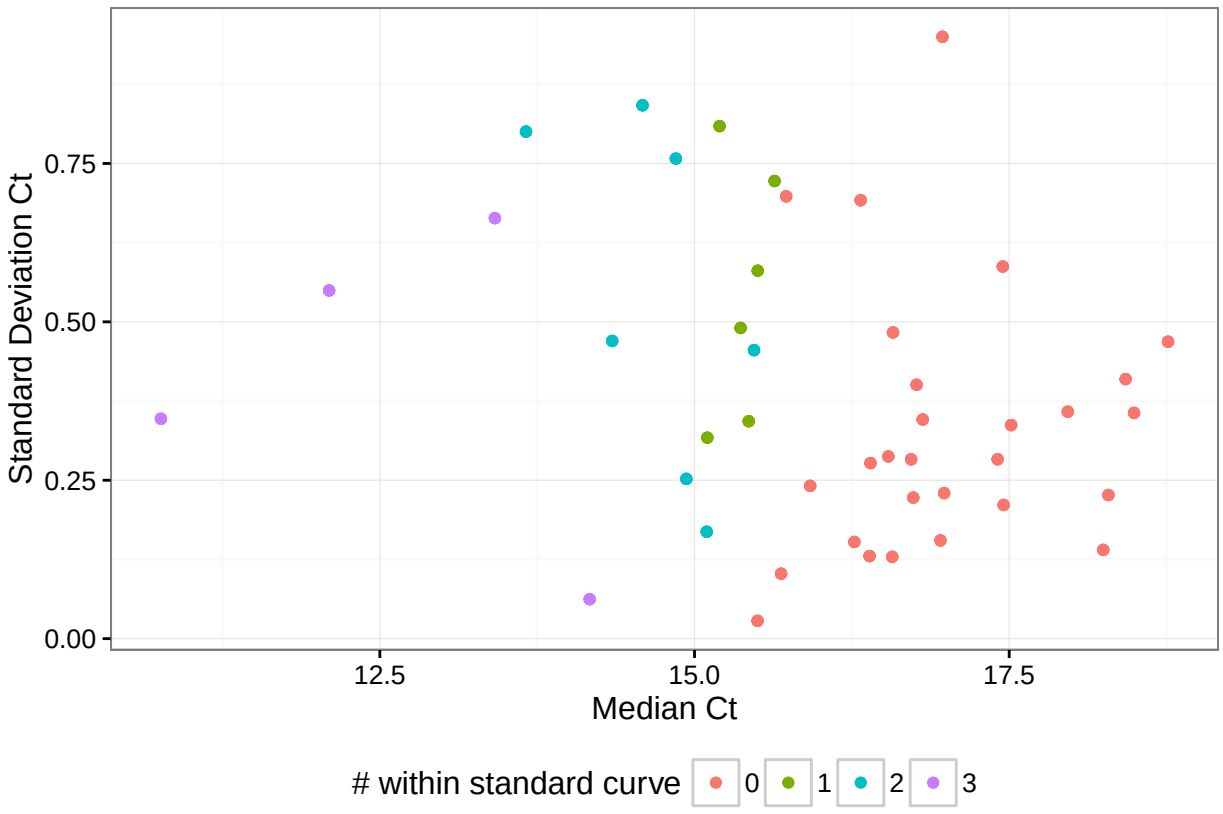
Relationship between standard deviation and median concentration values.

```
bac_var <- bac_fit %>% group_by(sample_name) %>%
      mutate(quant_med = median(quant), quant_sd = sd(quant),
            relative_ct_binary = if_else(relative_ct == "within", 0,1),
            n_within = sum(relative_ct_binary) %>% as.factor())
ggplot(bac_var) +
      geom_point(aes(x = quant_med, y = quant_sd, color = n_within)) +
      theme_bw() +
      labs(x = "Median (ng/ul)", y = "Standard Deviation (ng/ul)", color = "# within standard curve") +
      theme(legend.position = "bottom")
```



Relationship between median and standard deviation Ct values.

```
bac_var <- bac_fit %>% group_by(sample_name) %>%
      mutate(Ct_med = median(Ct), Ct_sd = sd(Ct),
             relative_ct_binary = if_else(relative_ct == "within", 0,1),
             n_within = sum(relative_ct_binary) %>% as.factor())
ggplot(bac_var) +
      geom_point(aes(x = Ct_med, y = Ct_sd, color = n_within)) +
      theme_bw() +
      labs(x = "Median Ct", y = "Standard Deviation Ct", color = "# within standard curve") +
      theme(legend.position = "bottom")
```



Tritration Concentration Deviation from Expected Value

```
bac_unmixed <- bac_fit %>% filter(sam_type == "unmixed") %>%
            mutate(sample_name = str_replace(sample_name, ".*_", ""),
                   sample_name = str_replace(sample_name, '\\(',"_"),
                   sample_name = str_replace(sample_name, '\\)',"")) %>%
            separate(sample_name,into = c("bio_rep","titration"),remove = FALSE) %>%
            mutate(titration_factor = if_else(titration == "Pre", 20, 0)) %>%
            select(bio_rep, plate,pred, titration) %>%
            spread(titration,pred)
```

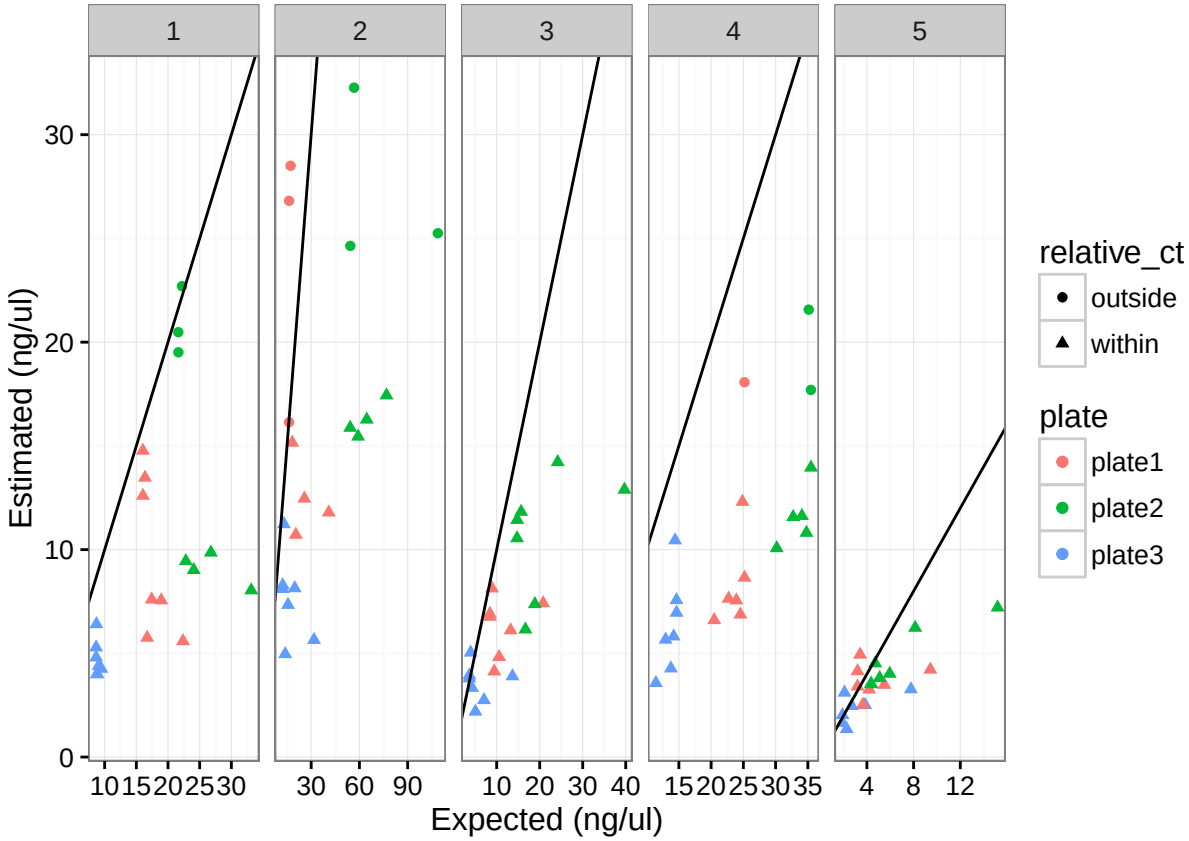
```
bac_mixed <- bac_fit %>% filter(sam_type == "titration") %>%
      separate(sample_name, into = c("bio_rep","titration"), remove = FALSE) %>%
      mutate(titration_factor = str_replace(titration, "M","") %>% as.numeric())
```

```
bac_fit2 <- left_join(bac_mixed, bac_unmixed) %>% select(-sample_name, -Ct, -std_ct, -sam_type) %>%
      mutate(exp_pred = Post*(2^-titration_factor) + Pre*(1-2^-titration_factor))
```

```
## Joining, by = c("bio_rep", "plate")
```

Expected Concentration - calculation based on concentration measurements for pre and post unmixed samples
Estimated Concentration - calculated using standard curve

```
bac_fit2 %>% ggplot(aes(x = 10^exp_pred, y = 10^pred, color = plate, shape = relative_ct)) +
      geom_point() +
      geom_abline(aes(intercept = 0, slope = 1)) +
      facet_grid(~bio_rep, scales = "free") +
      labs(x = "Expected (ng/ul)", y = "Estimated (ng/ul)") + theme_bw()
```



Conclusions

The samples should be diluted and run again.

Caveats

- To reduce experimental design complexity, sample name confounded with well except for negative controls (No Template Control, NTC).