ERCC qPCR Titration QA

Nate Olson

2016-12-21

Objective

Quantify the ERCC plasmid spike-in abundance with qPCR to validate two sample titrations.

Overview

1. qPCR Assay Validation
    1. No Template Controls - H20 and pre/ post
    2. Standard Curves - Precision and efficiency

2. Titration Validation
    1. Post
    2. Pre
3. Conclusions
    1. Potential Issues with titrations
    2. Assay limitations

qPCR Assay Validation

Negative Controls

Two types of negative controls were run.

1. No template Controls (NTC) - no DNA added to reactions only water, used to detect potential reagent and cross sample contamination. 6 - NTCs per qPCR plate
2. Negative ERCC Controls - Seperate ERCC plasmids were spiked into the unmixed pre and post treatment samples. The unmixed pre and post-treatment samples were used as a negative ercc control for the post and pre-treatment ERCC assays respectively. 3 pre and post negative ercc controls per plate

```
ercc_ntc <- qpcrERCC %>%
      mutate(sample_type = if_else(grepl("NTC", sampleID),"NTC",sample_type)) %>%
      filter(sample_type == "NTC") %>%
      mutate(undetermined = is.na(Ct),
             sampleID = if_else(grepl("Pre",sampleID),"Pre",sampleID),
             sampleID = if_else(grepl("Post",sampleID),"Post",sampleID))
```

```
n_undetermined <- sum(is.na(ercc_ntc$Ct))
n_ntc <- length(ercc_ntc$Ct)
```

Total negative controls: 60
Number of negative controls with undetermined Ct values: 52

Breakdown of negative control reactions with Ct values. Three pre and post negative controls in addition to 6 no template controls (H20 - blanks) per qPCR plate.

```
ercc_ntc %>% filter(undetermined == F) %>%
      group_by(ercc, sampleID) %>%
      summarise(count = n()) %>% spread(sampleID, count,fill = 0)  %>%
      knitr::kable()
```

ercc	NTC/H2O	Post	Pre
12	0	0	1
34	1	2	0
57	0	1	0
92	0	3	0

- The three negative control reactions for the ERCC-34 assay (two unmixed post treatment and one blank) with Ct values indicates a low level of cross contamination or background noise.

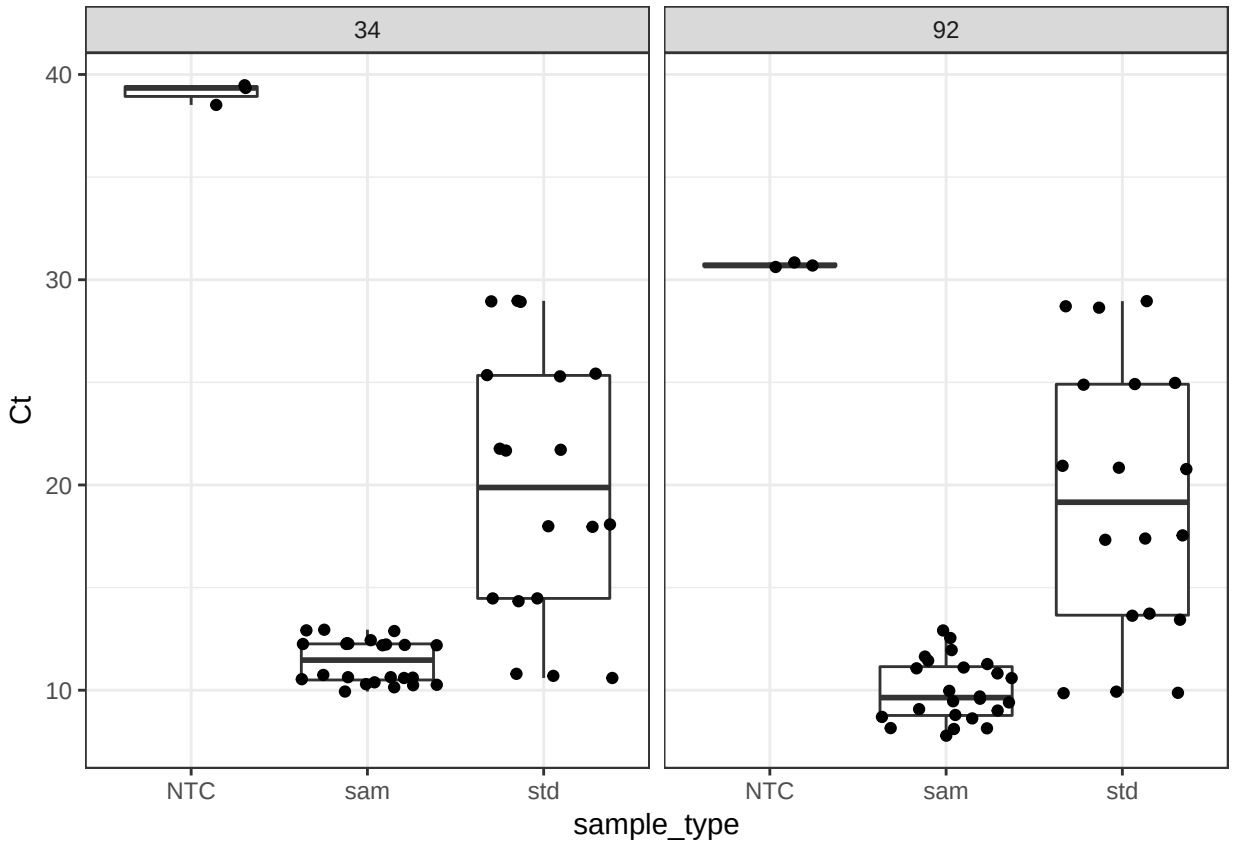
- The three unmixed post treatment samples with Ct values near 30 indicates contaminantion or lack of assay specificity.

Even with the potential low level cross contaminantion the negative control Ct values are much lower than the sample and standard curve Ct values.

```
qpcrERCC %>% filter(ercc %in% c(34, 92)) %>%
      mutate(sample_type = if_else(grepl("NTC", sampleID),"NTC",sample_type)) %>%
      ggplot(aes(x = sample_type, y = Ct)) +
      geom_boxplot() + geom_jitter() +
      theme_bw() + facet_wrap(~ercc)
```

```
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



Conclusion

Potenial low level contamination was observed for ERCC assays 34 and 92. Due to differences in sample and standard curve Ct values compared to the negative controls the potential contamination is unlikely to have negatively impacted the sample or standard curve Ct values.

Standard Curves

Limitation of efficiency assessment is that the standard curve is only plasmid DNA, no stool DNA as background. Stool DNA may contain PCR inhibitors or DNA that may interfere with the qPCR assay.

```
ercc_std <- qpcrERCC %>% filter(sample_type == "std", !grepl("NTC",sampleID)) %>%
      mutate(sampleID = gsub("\\(.*","",sampleID),
             Ct = as.numeric(Ct),
             quat = as.numeric(quant),
             log_quant = log10(quant))
```

Fitting standard curve data to a linear model to assess assay precision (R^2) and efficiency. Efficiency, is a measure of the assay amplification efficiency, whether the amount of template DNA doubles every PCR cycle.

```
fit_mod <- ercc_std %>% mutate(ercc = as.numeric(ercc)) %>%
      group_by(ercc) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~log_quant, data = . )))
```

```
fit_list <- fit_mod$fit %>% set_names(fit_mod$ercc)

fit_coefs <-fit_list %>% map_df(coefficients) %>%
```

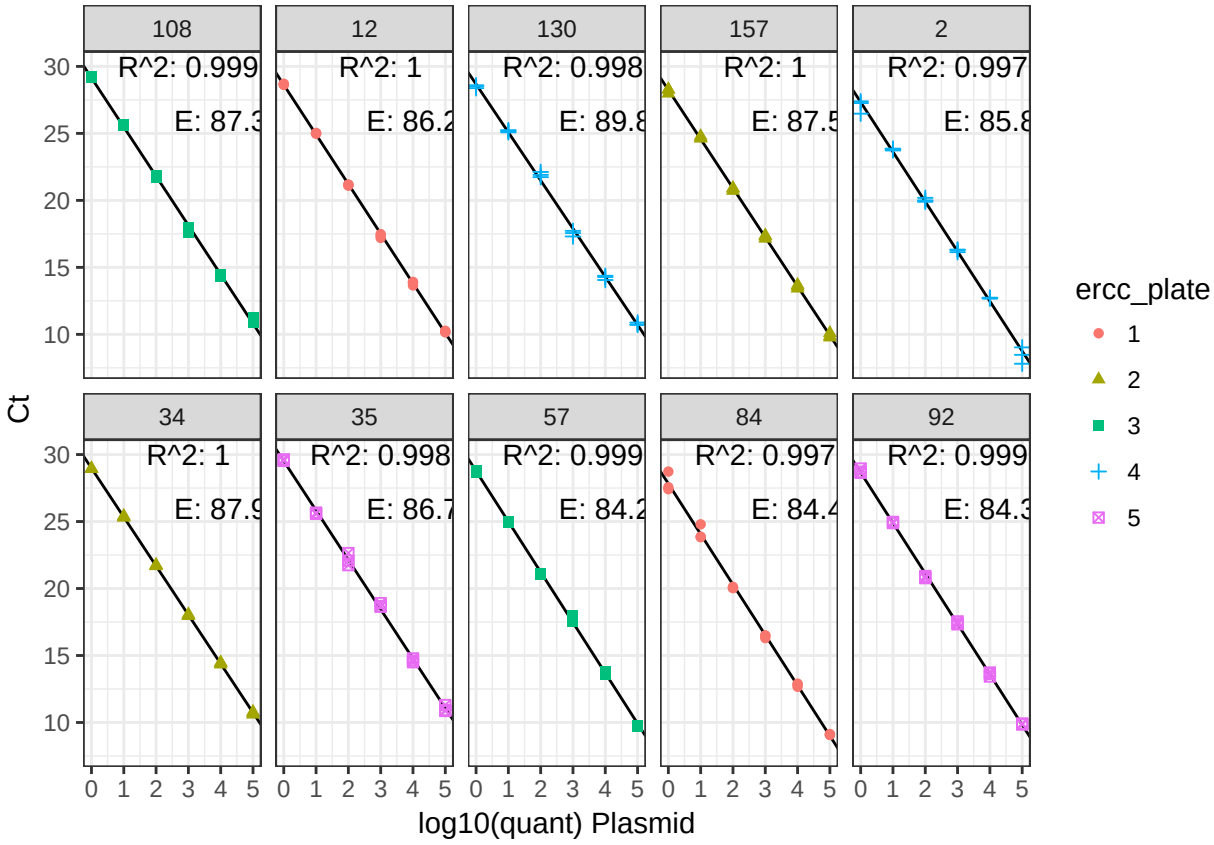
```
      add_column(coefs = c("intercept","slope")) %>%
      gather("ercc","stat",-coefs) %>% spread(coefs, stat)
```

```
std_fit <- fit_list %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(fit_coefs) %>%
      mutate(amplification_factor = 10^(-1/slope),
             efficiency = (amplification_factor - 1) * 100)
```

```
## Joining, by = "ercc"
```

The qPCR assay standard curves had a high level of precision with R^2 values close to 1 for all standard curves. The amplification efficiency was outside of the ideal range (0.9 - 1.1), with still within the acceptable range. Ideal and acceptable ranges based on rule of thumb community accepted guidelines.

```
ggplot(std_fit) +
      geom_abline(aes(intercept = intercept, slope = slope)) +
      geom_text(aes(x = 3, y = 30, label = paste("R^2:", signif(adj.r.squared,3)))) +
      geom_text(aes(x = 4, y = 26, label = paste("E:", signif(efficiency,3)))) +
      geom_point(data = ercc_std, aes(x = log_quant, y = Ct, color = ercc_plate, shape = ercc_plate)) +
      facet_wrap(~ercc, ncol = 5) +
      theme_bw() +
      labs(x = "log10(quant) Plasmid", y = "Ct")
```



Titration Validation

```
post_assays <- c(108,12, 157, 2, 35)
ercc_sam <- qpcrERCC %>% filter(sample_type == "sam") %>%
      mutate(Ct = as.numeric(Ct),
             quant = as.numeric(quant),
             ercc = as.numeric(ercc),
             titration = gsub("._M","",sampleID),
             titration = gsub(".*\\(Pre\\)","20", titration),
             titration = gsub(".*\\(Post\\)","0", titration),
             titration = as.numeric(titration),
             pre_prop = (1 - (2^-titration)),
             assay_type = if_else(ercc %in% post_assays, "Post","Pre"))
```

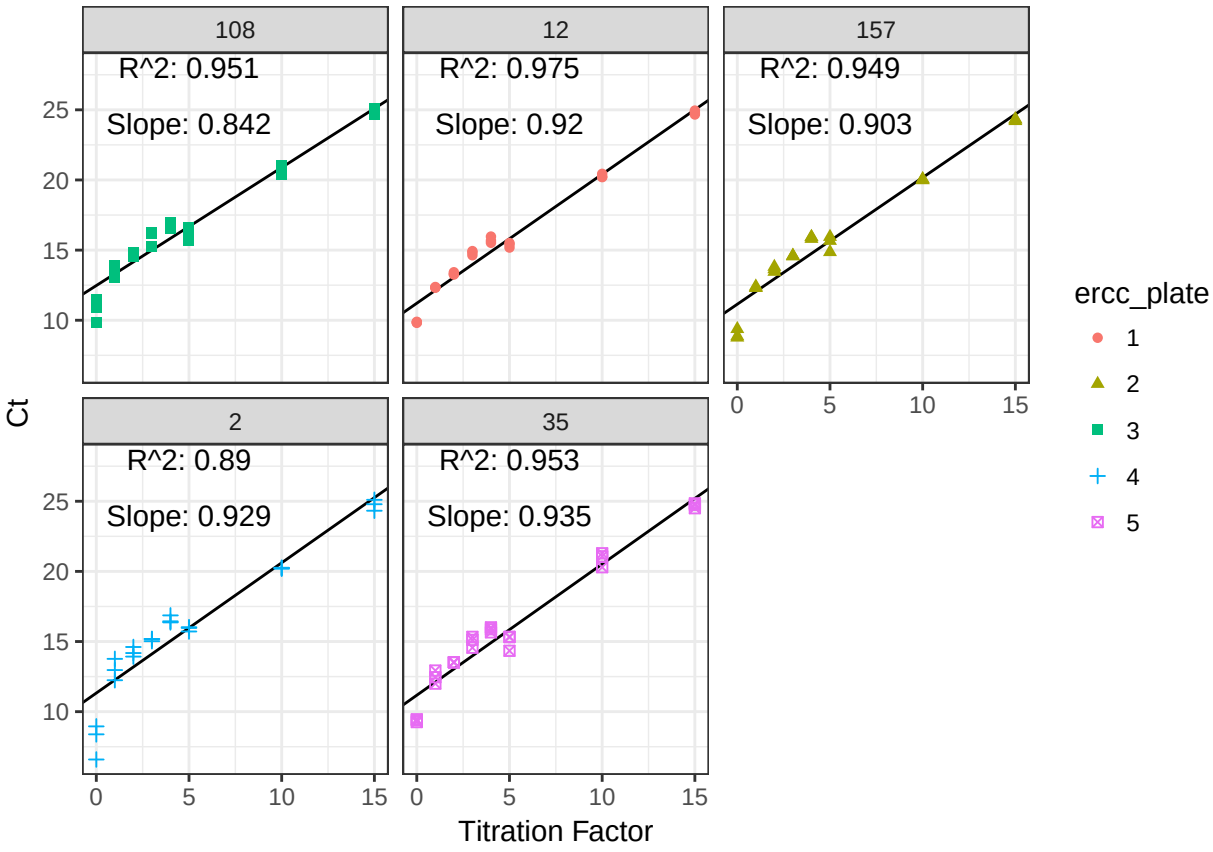
Post-treatment Assays

```
post_fit_mod <- ercc_sam %>% filter(assay_type == "Post") %>%
      group_by(ercc,assay_type) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~titration, data = . )))

post_fit_list <- post_fit_mod$fit %>% set_names(post_fit_mod$ercc)

# Extract fit parameters and calculate efficiency
post_fit_coefs <- post_fit_list %>% map_df(coefficients) %>%
      add_column(coefs = c("intercept","slope")) %>%
      gather("ercc","stat",-coefs) %>% spread(coefs, stat)

post_fit <- post_fit_list %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(post_fit_coefs) %>%
      mutate(amplification_factor = 10^(-1/slope),
             efficiency = (amplification_factor - 1) * 100)
```

```
## Joining, by = "ercc"
```

The post treatment qPCR assays (12, 157, 108, 2, and 35) had good R^2 and slope values. The expected slope is 1, for a doubling every cycle. The 1-4 titration factor samples had Ct values consistently above the regression line.

```
post_fit %>% ggplot() +
      geom_abline(aes(intercept = intercept, slope = slope)) +
      geom_text(aes(x = 5, y = 28, label = paste("R^2:", signif(adj.r.squared,3)))) +
      geom_text(aes(x = 5, y = 24, label = paste("Slope:", signif(slope,3)))) +
      geom_point(data = ercc_sam %>% filter(assay_type == "Post"),
                 aes(x = titration, y = Ct, color = ercc_plate, shape = ercc_plate)) +
      facet_wrap(~ercc) +
      theme_bw() + labs(x = "Titration Factor", y = "Ct")
```



Different slopes for titations 1-4 and titrations 0, 5, 10, and 15.

```
post_fit_mod14 <- ercc_sam %>% filter(assay_type == "Post", titration %in% 1:4) %>%
      group_by(ercc,assay_type) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~titration, data = . )))

post_fit_list14 <- post_fit_mod14$fit %>% set_names(post_fit_mod14$ercc)

# Extract fit parameters and calculate efficiency
post_fit_coefs14 <- post_fit_list14 %>% map_df(coefficients) %>%
      add_column(coefs = c("intercept","slope")) %>%
      gather("ercc","stat",-coefs) %>% spread(coefs, stat)

post_fit14 <- post_fit_list14 %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(post_fit_coefs14) %>% mutate(fit = "1:4")
```

```
## Joining, by = "ercc"
```

```
post_fit_mod05 <- ercc_sam %>% filter(assay_type == "Post", titration %!in% 1:4) %>%
      group_by(ercc,assay_type) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~titration, data = . )))

post_fit_list05 <- post_fit_mod05$fit %>% set_names(post_fit_mod05$ercc)

# Extract fit parameters and calculate efficiency
post_fit_coefs05 <- post_fit_list05 %>% map_df(coefficients) %>%
      add_column(coefs = c("intercept","slope")) %>%
```

```
      gather("ercc","stat",-coefs) %>% spread(coefs, stat)

post_fit05 <- post_fit_list05 %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(post_fit_coefs05) %>%
      mutate(fit = "0,5,10,15")
```

```
## Joining, by = "ercc"
```

```
bind_rows(post_fit14,post_fit05) %>% select(-intercept) %>% arrange(ercc) %>% knitr::kable()
```

ercc	adj.r.squared	slope	fit
108	0.9245472	1.0728149	1:4
108	0.9876631	0.9348699	0,5,10,15
12	0.9824368	1.1606172	1:4
12	0.9973704	0.9955799	0,5,10,15
157	0.9915743	1.1569193	1:4
157	0.9856698	1.0075021	0,5,10,15
2	0.9142569	1.1583663	1:4
2	0.9659081	1.0923516	0,5,10,15
35	0.9471568	1.1640626	1:4
35	0.9876966	1.0377605	0,5,10,15

Pre-treatment Assays

Still need to figure out the expected slope for pre-treatment ERCC spike-ins. Should be 1 Ct difference between the unmixed post and titration factor 1 and 0.5 Ct between titration factor 1 and 2. For the other titration factors the expected difference is to small to detect using qPCR (< 0.5 Ct).

```
pre_fit_mod <- ercc_sam %>% filter(assay_type == "Pre") %>%
      group_by(ercc,assay_type) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~titration, data = . )))

pre_fit_list <- pre_fit_mod$fit %>% set_names(pre_fit_mod$ercc)

# Extract fit parameters and calculate efficiency
pre_fit_coefs <- pre_fit_list %>% map_df(coefficients) %>%
      add_column(coefs = c("intercept","slope")) %>%
      gather("ercc","stat",-coefs) %>% spread(coefs, stat)

pre_fit <- pre_fit_list %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(pre_fit_coefs) %>%
      mutate(amplification_factor = 10^(-1/slope),
             efficiency = (amplification_factor - 1) * 100)
```

```
## Joining, by = "ercc"
```

```
pre_fit %>% ggplot() +
      geom_abline(aes(intercept = intercept, slope = slope)) +
     geom_text(aes(x = 10, y = 16,
                   label = paste("R^2:", signif(adj.r.squared,3)))) +
      geom_text(aes(x = 10, y = 14,
```

```
                label = paste("Slope:", signif(slope,3)))) +
    geom_point(data = ercc_sam %>% filter(assay_type == "Pre"),
               aes(x = titration, y = Ct, color = ercc_plate, shape = ercc_plate)) +
    facet_wrap(~ercc) +
    theme_bw() + labs(x = "Titration Factor", y = "Ct")
```