

ERCC qPCR Titration QA

Nate Olson

2016-12-17

Objective

Use qPCR to quantify the ERCC plasmid spike-in abundance to validate titrations.

No Template Controls

qPCR Assay Efficiency

Limitation of efficiency assessment is that the standard curve is only plasmid DNA, no stool DNA extract as background.

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

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

Extract fit parameters and calculate efficiency

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

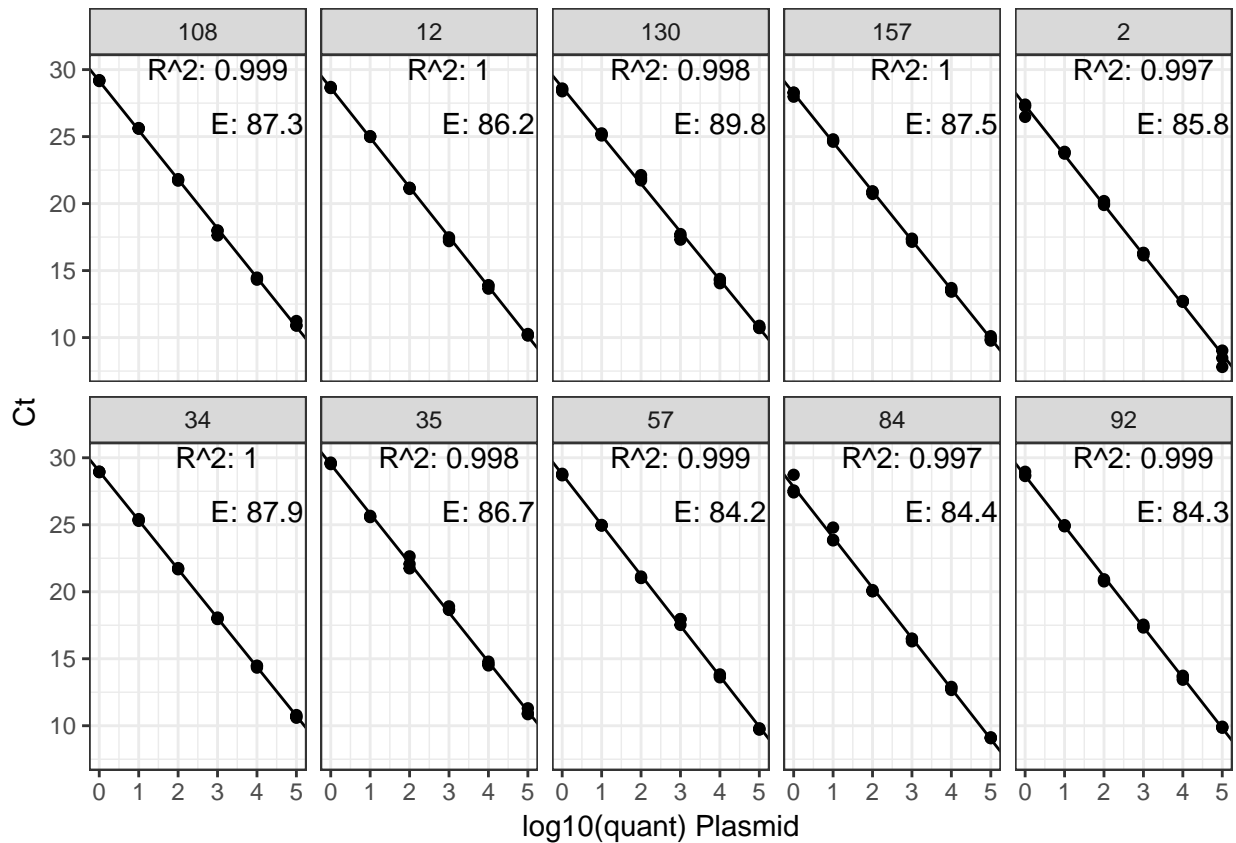
```
fit_coefs <- fit_list %>% map_df(coefficients) %>%
  add_column(coefs = c("intercept", "slope")) %>%
  gather("ercc", "stat", -coefs) %>% spread(coefs, stat)
```

```
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"
```

High R^2 values. Efficiency for all assays lower than expected (between 0.9 and 1.1).

```
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)) +
  facet_wrap(~ercc, ncol = 5) +
  theme_bw() +
  labs(x = "log10(quant) Plasmid", y = "Ct")
```



Titration Validation

```
ercc_sam <- qpcrERCC %>% filter(sample_type == "sam") %>%
  mutate(Ct = as.numeric(Ct),
         quant = as.numeric(quant),
         titration = gsub("._M", "", sampleID),
         titration = gsub(".*\\(Pre\\)", "20", titration),
         titration = gsub(".*\\(Post\\)", "0", titration),
         titration = as.numeric(titration))
```

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

Extract fit parameters and calculate efficiency

```
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)
```

```
sam_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 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 standard curve efficiency was less than 1, indicating that the lower than expected slope values are likely due to low qPCR assay efficiency than pipetting errors while generating the mixtures. Not sure about the 1-4 titration factor samples having Ct values consistently above the regression line.

TODO

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 too small to detect using qPCR (< 0.5 Ct).

```
ggplot(sam_fit) +
  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, aes(x = titration, y = Ct)) + facet_wrap(~ercc) +
  theme_bw() + labs(x = "Titration Factor", y = "Ct")
```

