Mixing Titrations and Titration Validation

Nate Olson

2017-01-19

Mixing Titrations

Table with volumes used to dilute samples

```
biosampleInfo %>% kable()
```

biosample_id	lab_id	treatment	timepoint	stool_weight	processing_date	conc_ngul	vol_ul	total_ug	samp
E01JH0004	3	Pre	-1	0.19	2012-03-29	241	20	4.82	5.1
E01JH0011	8	Pre	-1	0.26	2012-03-29	547	20	10.94	2.2
E01JH0016	12	Pre	-1	0.20	2012-03-30	266	20	5.32	4.6
E01JH0017	13	Pre	-1	0.16	2012-03-30	399	20	7.98	3.1
E01JH0038	28	Pre	-1	0.34	2012-03-30	324	20	6.48	3.8
E01JH0004	138	Post	4	0.57	2012-04-16	203	20	4.06	6.1
E01JH0011	115	Post	2	NA	NA	NA	NA	NA	
E01JH0016	999	Post	2	NA	NA	NA	NA	NA	
E01JH0017	177	Post	5	0.50	2012-04-19	199	20	3.98	6.2
E01JH0038	105	Post	2	1.01	2012-04-10	543	20	10.86	2.3

Table with volumes used to make titrations

```
tstPrep %>% kable()
```

Titration	ERCC_ul	Prep_DNA	Prep_vol	Final_vol
(-1)_Pre unmixed	2	100ul_12.5ng/ul	102	21
(1)_Post unmixed	2	100ul_12.5ng/ul	102	92
M1-Mixed	NA	(-1)*10ul+(1)*10ul	20	10
M2-Mixed	NA	(-1)*10ul+M1*10ul	20	10
M3-Mixed	NA	(-1)*10ul+M2*10ul	20	10
M4-Mixed	NA	(-1)*10ul+M3*10ul	20	10
M5-Mixed	NA	(-1)*10ul+M4*10ul	20	19.5
M10-Mixed	NA	(-1)*15.5ul+M5*0.5ul	16	15.5
M15-Mixed	NA	(-1)*15.5ul+M10*0.5ul	16	16

Tube Rack Image

Image of tubes layout prior to 16S PCR.

ERCC table - plasmid and spike-ins

```
erccMeta %>% kable()
```

ercc_id	biosample_id	treatment	length	GC	assay_id	amplicon_length
ERCC-00002	E01JH0017	Post	1061	0.53	Ac03459872_a1	69
ERCC-00012	E01JH0004	Post	994	0.52	Ac03459877_a1	77
ERCC-00034	E01JH0011	Pre	1019	0.50	Ac03459987_a1	58
ERCC-00035	E01JH0038	Post	1130	0.52	Ac03459892_a1	65
ERCC-00057	E01JH0016	Pre	1021	0.51	Ac03460000_a1	78
ERCC-00084	E01JH0004	Pre	994	0.52	Ac03459922_a1	63
ERCC-00092	E01JH0038	Pre	1124	0.51	Ac03459925_a1	87
ERCC-00108	E01JH0016	Post	1022	0.50	Ac03460028_a1	74
ERCC-00130	E01JH0017	Pre	1059	0.47	Ac03460039_a1	72
ERCC-00157	E01JH0011	Post	1019	0.51	Ac03459958_a1	71

Titration Validation

qPCR Assay Evaluation

Fitting the standard curve to a linear model, `Ct ~ log10(concentration)`.
The expected slope for the standard curve is -3.33 indicating a perfect doubling every PCR cycle, for a amplification factor (AF) of 2 and efficiency (E) of 1.

$$AF = 10^{-1/slope}$$

$$E = 10^{-1/slope} - 1$$

qPCR Bacterial Abundance

Standard Curves

TODO Amplification Curves for standards

Linear Model

The model was fit using the full standard curve and only points in the standard curve with concentrations greater than 0.02 ng/ul. Fitting the regression to all concentrations in the standard curve resulted in a lower amplification efficiency and R^2 for all three standard curves.

```
fit_mod <- qpcrBacStd %>%
      filter(!is.na(Ct)) %>%
      mutate(log_conc = log10(conc), date != "2016-09-19", conc > 0.002) %>%
      ## excluding standard curve outlier
      filter(std != "zymo" | date != "2016-12-09" | conc != 0.00002 | plate != "plate3") %>%
      group_by(date, std) %>% nest() %>%
      mutate(fit = map(data, .f=~lm(Ct~log_conc ,data = .)))

fit_list <- fit_mod$fit %>% set_names(paste(fit_mod$date, fit_mod$std))

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

std_fit <- fit_list %>% map_df(broom::glance, .id = "std") %>%
```

```
      select(std, adj.r.squared) %>% left_join(fit_coefs) %>%
      separate(std, c("date","std","mod"), sep = " ") %>%
      mutate(amplification_factor = 10^(-1/slope),
             efficiency = (amplification_factor - 1) * 100)
```

```
## Joining, by = "std"
```

```
## Warning: Too few values at 3 locations: 1, 2, 3
```

The efficiency and precision (R^2) were higher for the in-house standard curve. Fitting the regression model for the experiment run on 12/09 without the low concentration plate 3 outlier resulted in a lower efficiency (76% versus 80%) but higher R^2 (0.997 versus 0.986).
Fitting the regression to only the standard with concentrations within the observed range of the samples (20 ng/ul, 2 ng/ul, and 0.2 ng/ul) compared to all standard curve samples resulted in a higher amplification efficiency but slightly lower R^2 values for all three standard curves. The higher efficiency

```
std_fit %>% select(std, date, mod, efficiency, adj.r.squared) %>%
      arrange(date, std) %>% knitr::kable()
```

std	date	mod	efficiency	adj.r.squared
zymo	2016-09-19	NA	63.86615	0.9933987
shan	2016-12-09	NA	86.86251	0.9992791
zymo	2016-12-09	NA	75.63477	0.9973140

```
qpcrBacStd %>% mutate(log_conc = log10(conc)) %>% ggplot(aes(y = Ct, x = log_conc)) +
      geom_vline(aes(xintercept = log10(0.2)), color = "grey60") +
      geom_abline(data = std_fit, aes(intercept = intercept, slope = slope)) +
      geom_point(aes(color = plate, shape = plate)) +
      facet_grid(std~date) + theme_bw() +
      theme(legend.position = "bottom")
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

Sample Concentrations

Using in-house with standard concentrations 20 ng/ul, 2ng/ul, and 0.2 ng/ul to predict sample concentrations. The unmixed sample concentrations were diluted to 10 ng/ul prior to making the titrations therefore all samples are expected to have concentrations less than 10 ng/ul.

Why do the unmixed post-treatment samples consistently have predicted concentrations greater than 10 ng/ul. Potential reasons:

1. The DNA in the post treatment samples have a higher 16S copy/ genome sequence ratio than *E. coli*, not likely as *E. coli* has 6 copies per genome.

2. Low Ct values resulting in noisy quantification. Clean amplification curves makes this unlikely.
3. Something else ...

```
mod <- qpcrBacStd %>%
      filter(std == "shan", conc >= 0.2) %>% mutate(log_conc = log10(conc)) %>%
      {lm(log_conc~Ct, data = .)}

bac_abu <- qpcrBacAbu %>% filter(!is.na(Ct), std == "shan") %>% add_predictions(mod) %>%
      mutate(quant = 10^pred) %>% group_by(sample_name) %>%
      mutate(quant_min = min(quant), quant_max = max(quant))
```

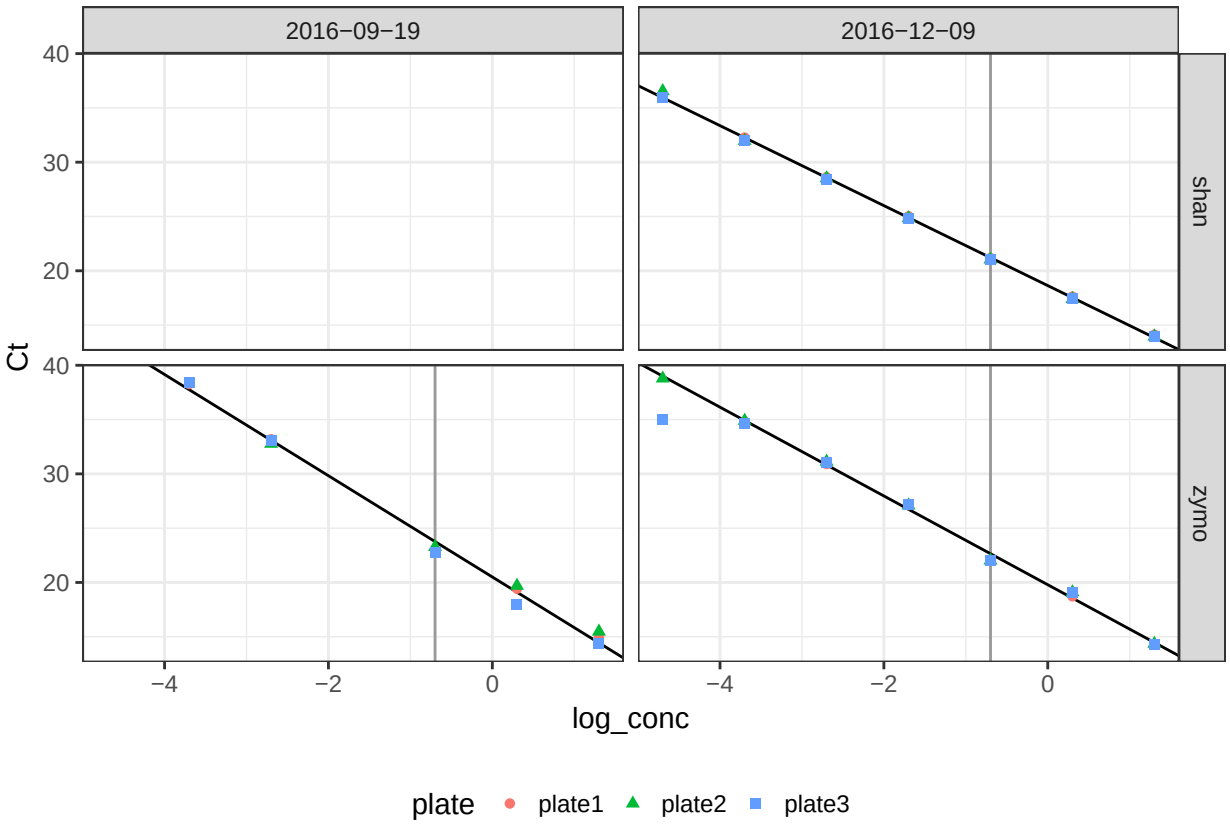


Figure 1: qPCR bacterial abundance standard curves. Using two different standards and performed on two different days. Two models were fit to the standard curves, one with all data point and a second with only the 20 ng/ul, 2 ng/ul, and 0.2 ng/ul standards. Grey vertical line indicates concentration cutoff for subset model.

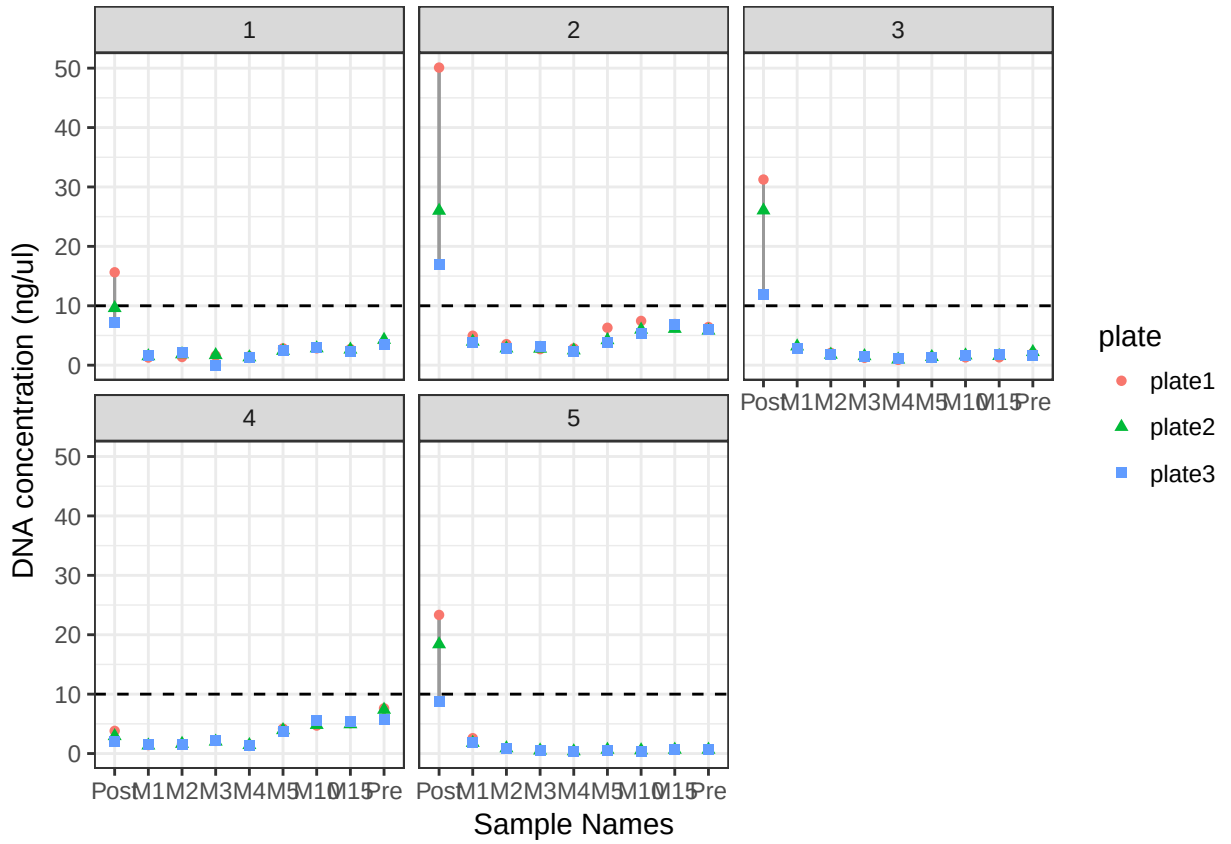


Figure 2: Predicted mixture study sample concentrations. Dashed line indicates the expected max concentration of 10 ng/ul.

```
bac_abu %>% filter(sample_name != "NTC") %>% ungroup() %>%
      mutate(sample_name = gsub(" ","_", sample_name)) %>%
      separate(sample_name, c("bio_rep","titration"), sep = "_") %>%
      mutate(titration = fct_relevel(titration, c("Post",paste0("M",c(1,2,3,4,5,10,15)),"Pre"))) %>%
      ggplot() +
            geom_hline(aes(yintercept = 10), linetype = 2) +
            geom_linerange(aes(x = titration,ymin = quant_min, ymax = quant_max), color = "grey60") +
            geom_point(aes(y = quant, x = titration, color = plate, shape = plate)) +
            facet_wrap(~bio_rep) +
            theme_bw() + labs(x = "Sample Names", y = "DNA concentration (ng/ul)")
```

Bacterial DNA proportions

Need to estimate the proportion of DNA in the unmixed sample that is bacterial to correct for differences in bacterial DNA proportions. As the pre and post treatment samples were diluted to 10 ng/ul prior to generating the titration series the unmixed samples should have less than 10 ng/ul and the proportion of bacterial DNA in the samples is obtained by dividing the estimated concentration by 10. Due to the estimated post treatment sample concentrations greater than 10, we will set these samples at 10 ng/ul.

```
bac_abu %>% filter(sam_type == "unmixed") %>% ungroup() %>%
      mutate(sample_name = gsub(" ","_", sample_name), quant = if_else(quant > 10, 10, quant)) %>%
      separate(sample_name, c("bio_rep","titration"), sep = "_")  %>%
```

```
      group_by(bio_rep, titration) %>% summarise(bac_prop = median(quant)/10) %>%
      spread(titration, bac_prop) %>% knitr::kable()
```

bio_rep	Post	Pre
1	0.9619490	0.3533205
2	1.0000000	0.6063395
3	1.0000000	0.2033600
4	0.2944102	0.7376526
5	1.0000000	0.0682492

qPCR ERCC

Standard Curves

Amplification Curves

TODO Show differences in baseline for amplification 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)
```

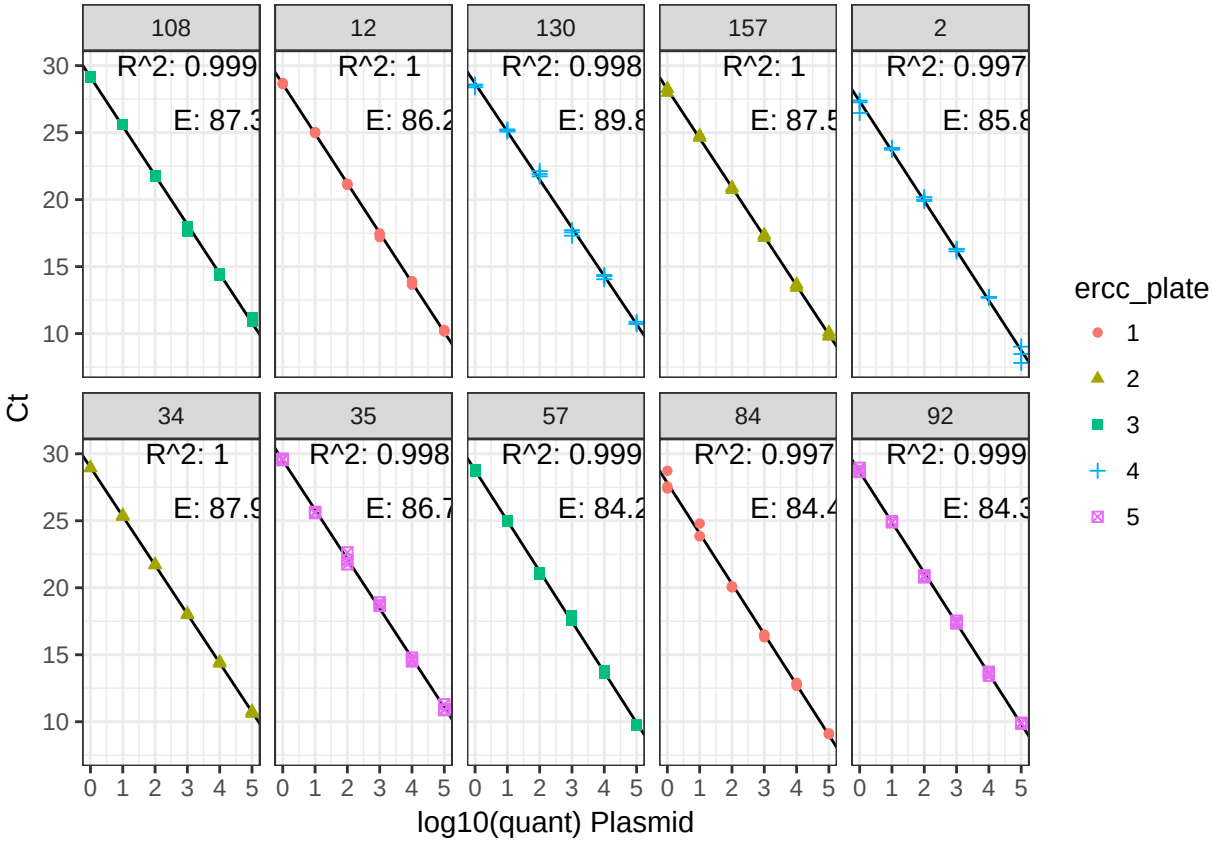
```
std_fit <- fit_list %>% map_df(broom::glance, .id = "ercc") %>%
      select(ercc, adj.r.squared) %>%
      left_join(fit_coefs) %>%
      mutate(amplification_factor = 10^(-1/slope),
             efficiency = (amplification_factor - 1) * 100)
```

```
## 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")
```



Sample Cts

```
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"))
```

Pre-treatment

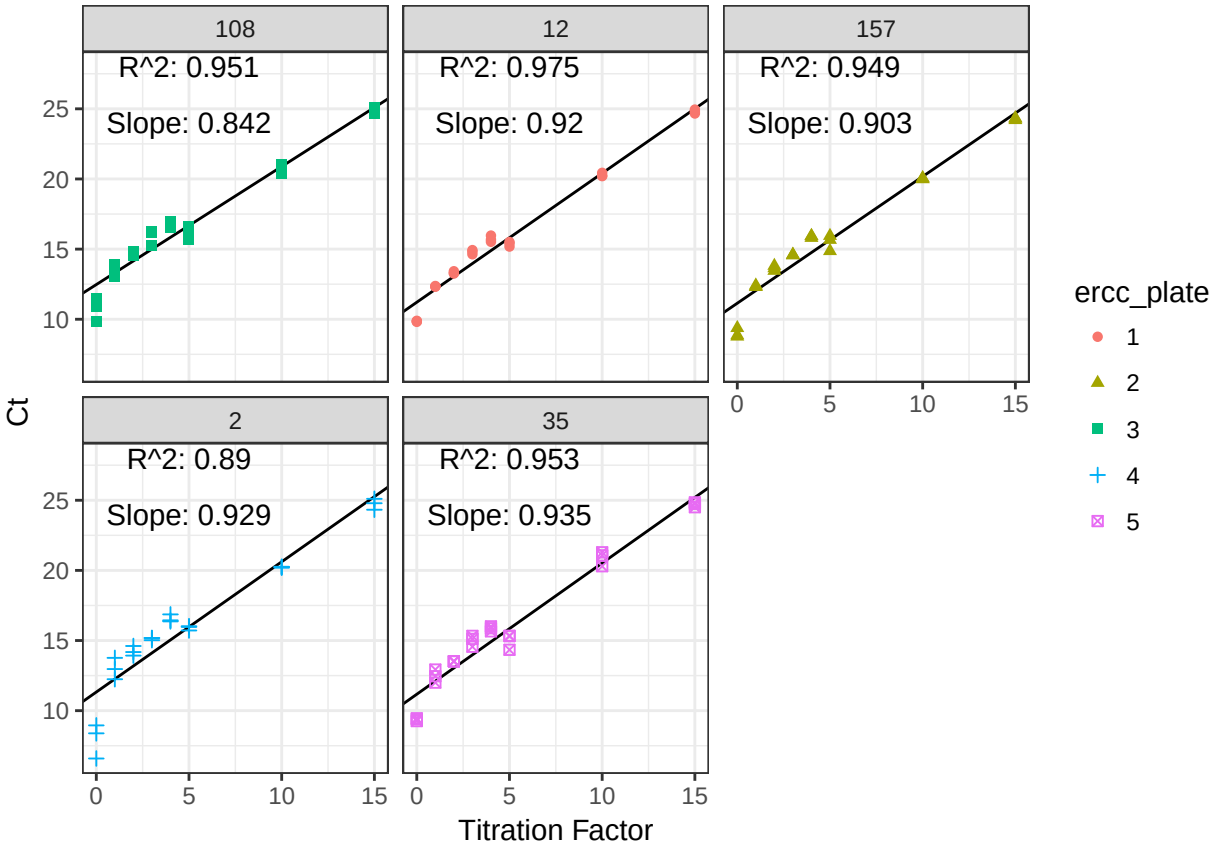
```
post_fit_mod <- ercc_sam %>% filter(assay_type == "Post") %>%
      group_by(ercc,assay_type) %>% nest() %>%
      mutate(fit = map(data, ~lm(Ct~titration, data = . )))
```

```
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

Post-treatment

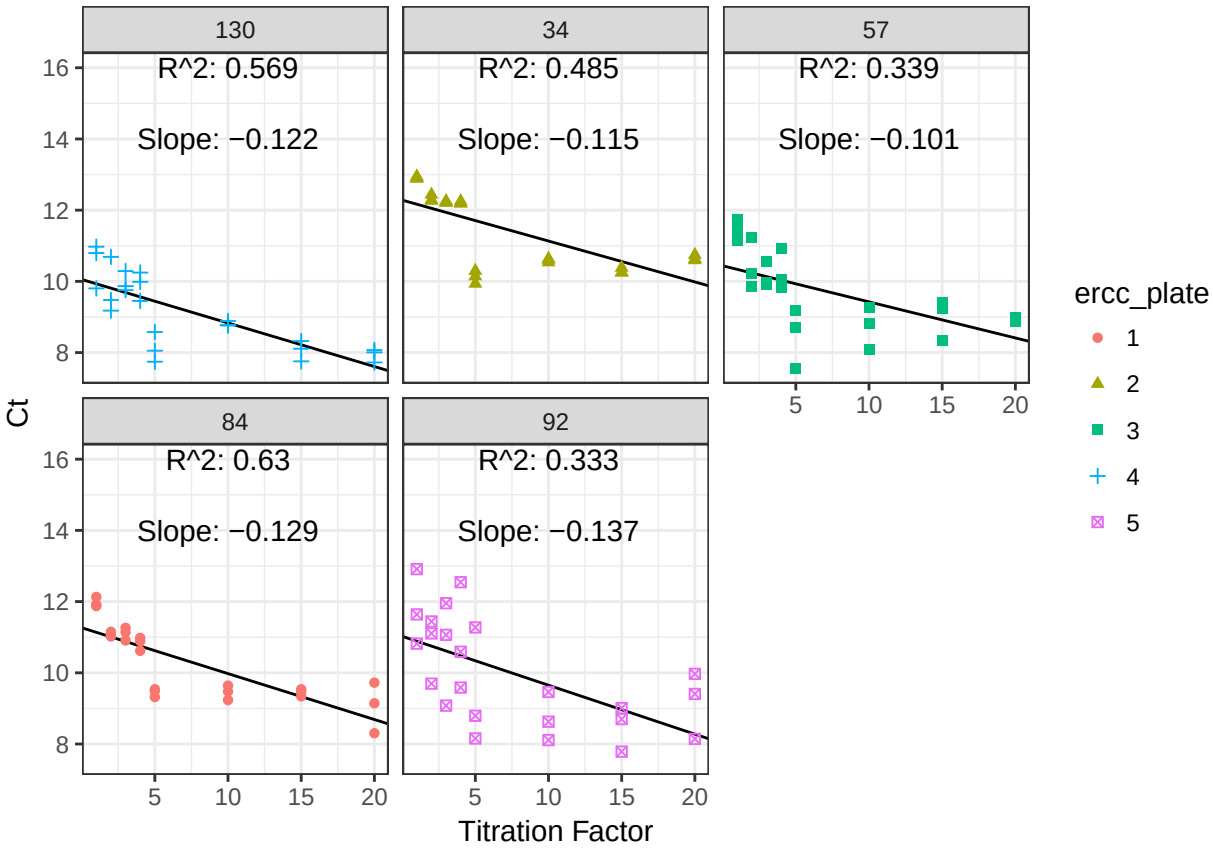
Still need to figure out the expected slope for pre-treatment ERCC spike-ins. Should be 1 Ct difference between the unmixed post and titration factor 1 and 0.5 Ct between titration factor 1 and 2. For the other

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")
```


130
R^2: 0.569
Slope: −0.122
34
R^2: 0.485
Slope: −0.115
57
R^2: 0.339
Slope: −0.101
84
R^2: 0.63
Slope: −0.129
92
R^2: 0.333
Slope: −0.137
Ct
Titration Factor
ercc_plate
1
2
3
4
5