logFC Summary

Nate Olson

2017-09-29

0.1 Objective

Based on the mixture design the logFC between sequential titrations for features only present in post-exposure samples should be 1. Due to differences in the proportion of bacterial DNA the expected logFC of 1 does not hold. However, the logFC between sequential titrations should be constant and increase when comparing non-adjacent titrations.

0.2 Approach

- Characterize observed logFC between pre- and post-exposure samples for E01JH0011. This individual had inferred theta values that agreed best with the mixture design.

0.3 Post-specific features

Most of the post-specific features were not present in none of the titrations and therefore cannot be used to evalute logFC estimates. Interestingly there was no correlation between logCPM between the pre- and post-exposure samples and the number of titration PCR replicates with observed counts. For post-specific features we expect a linear relationship between the logFC estimate and difference in the titration factor for the samples being compared. For features with high abundance in the post-exposure samples (large logCPM) observed in over half of the titration PCR replicates there is inconsistent behavior even for the highest abundance (logCPM > 6.5) and prevalent features (observed in > 15 titration PCRs).

```
pa_summary_anno_df %>%
      filter(biosample_id == "E01JH0011", T00 == 4, T20 == 0) %>%
      ggplot() +
      geom_histogram(aes(x = pa_mixed)) + facet_wrap(~pipe, nrow = 1) +
      theme_bw() +
      labs(x = "Titration PCR Replicates with Non-Zero Counts")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20, post_specific == 1) %>%
      ggplot() +
      geom_point(aes(x = pa_mixed, y = logCPM)) +
      facet_wrap(~pipe) + theme_bw() + labs(x = "Number of Titration PCR Replicates")
```

```
post_feature_logFC <- logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20,
             post_specific == 1, pa_mixed != 0) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC) %>%
      rename(prepost_logCPM = logCPM, prepost_logFC = logFC) %>%
      left_join(logFC_biosam_11)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```

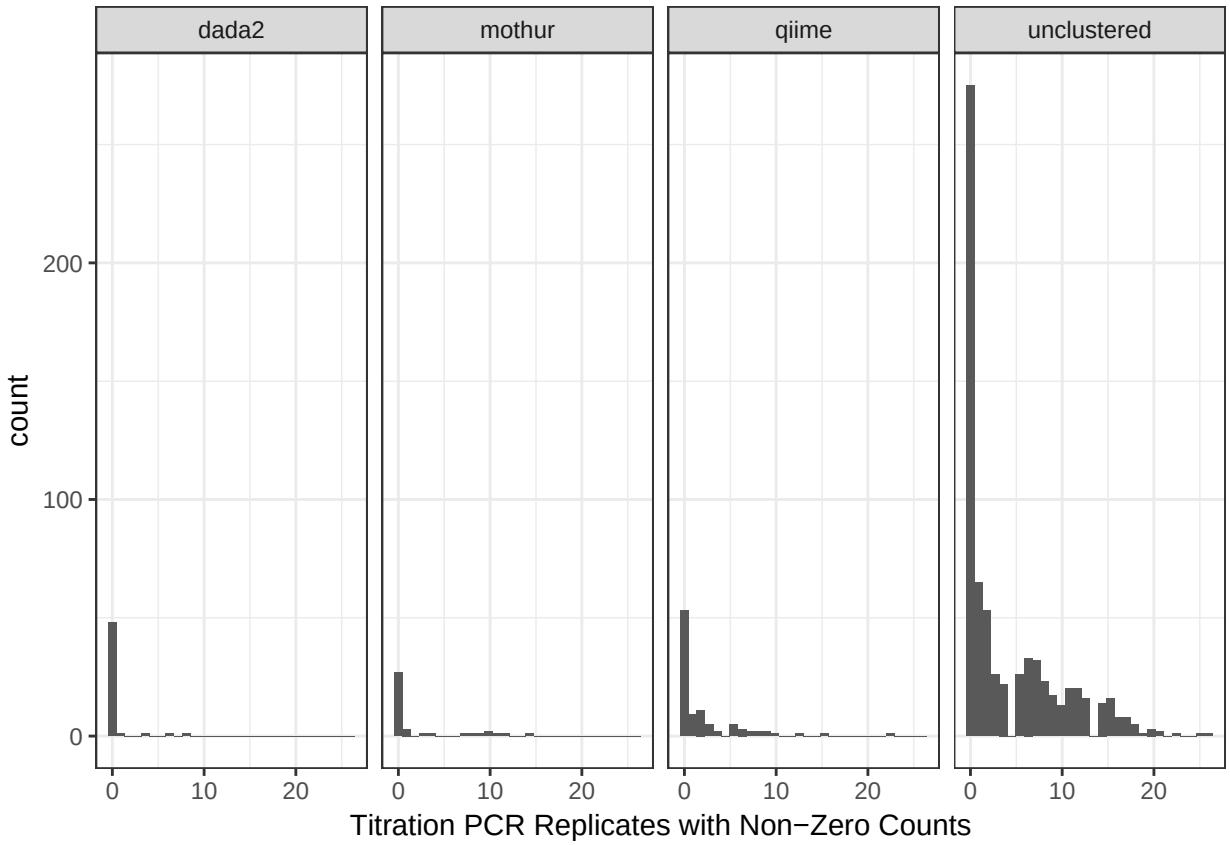


Figure 1: Distribution titration PCR replicates with non-zero counts for post-specific features.

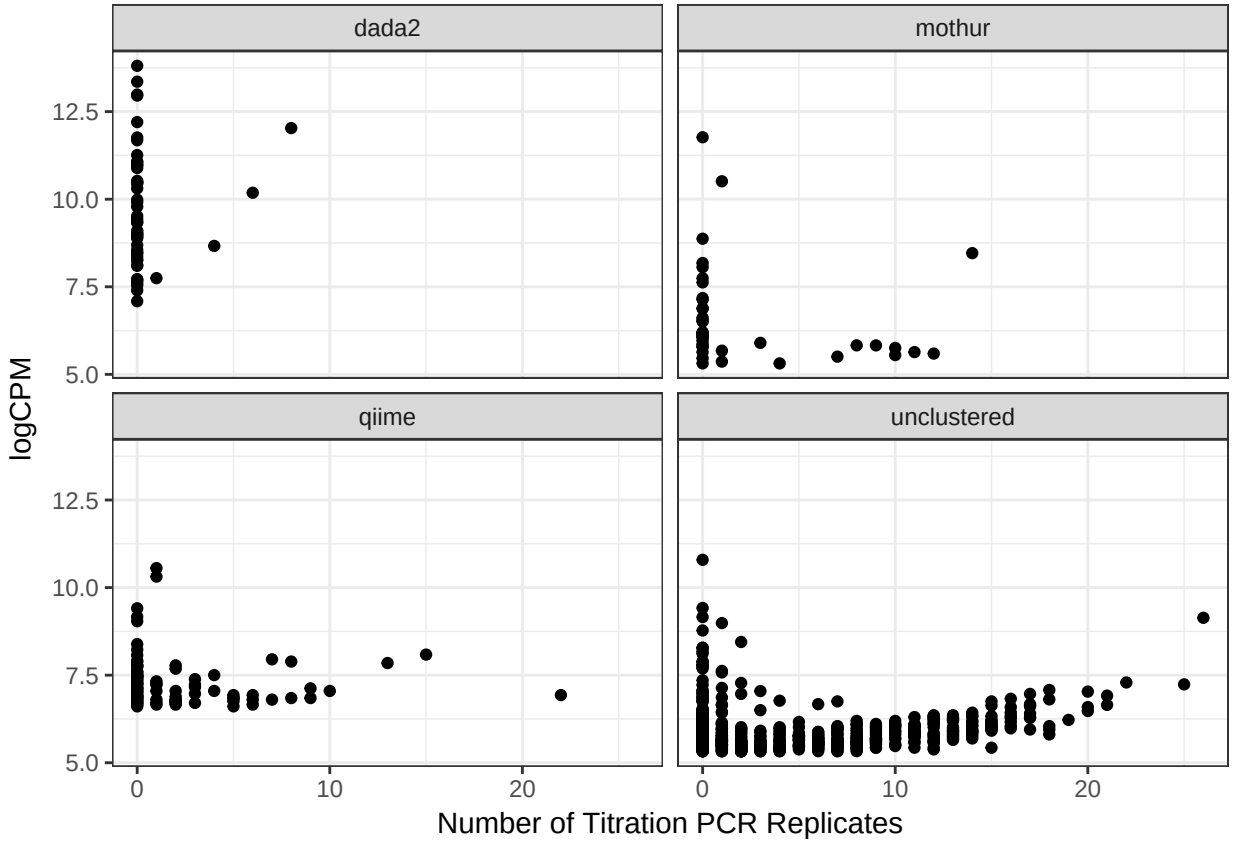
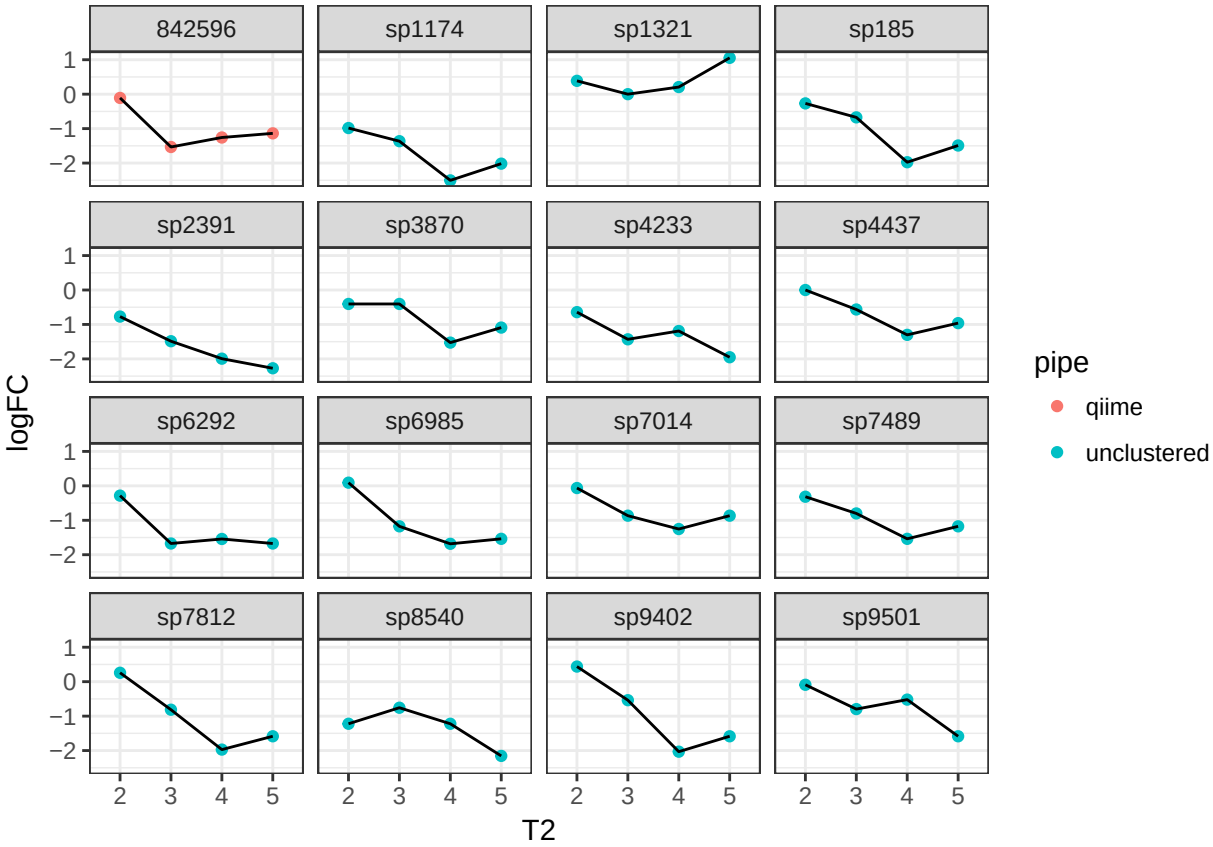


Figure 2: Relationship between the logFC pre- and post-exposure samples of post-specific features to the number of titration PCR replicats the feature was observed in.

```
post_feature_logFC %>% ungroup() %>%
      filter(pa_mixed > 15, prepost_logCPM > 6.58)  %>%
      filter(T1 == 1, T2 %in% 1:5)  %>%
      ggplot() +
      geom_point(aes(x = T2, y = logFC, color = pipe)) +
      geom_line(aes(x = T2, y = logFC, group = feature_id)) +
      facet_wrap(~feature_id) +
      theme_bw()
```



Escherichia/ Shigella Features logFC estimates for features classified as Escherichia with logFC > -4 between unmixed pre- and post-exposure samples exhibited the expected behavior of linearly increasing when comparing non-successive titrations.

```
logFC_biosam_11 %>% filter(T1 == 0, T2 == 20) %>%
      ggplot() +
      geom_point(aes(x = logCPM, y = logFC,
                     fill = factor(ec_feature)),
                 color = "grey80", shape = 21) +
      geom_hline(aes(yintercept = -4), linetype = 2) +
      facet_wrap(~pipe) + theme_bw()
```

No QIIME features were classified as **Escherichia/Shigella**, will include QIIME features classified as Enterobacteriaceae with logFC < -4.

```
logFC_biosam_11 %>% filter(Rank5 == "f__Enterobacteriaceae") %>%
      filter(T1 == 0, T2 == 20) %>%
      ggplot() +
      geom_point(aes(x = logCPM, y = logFC),
```

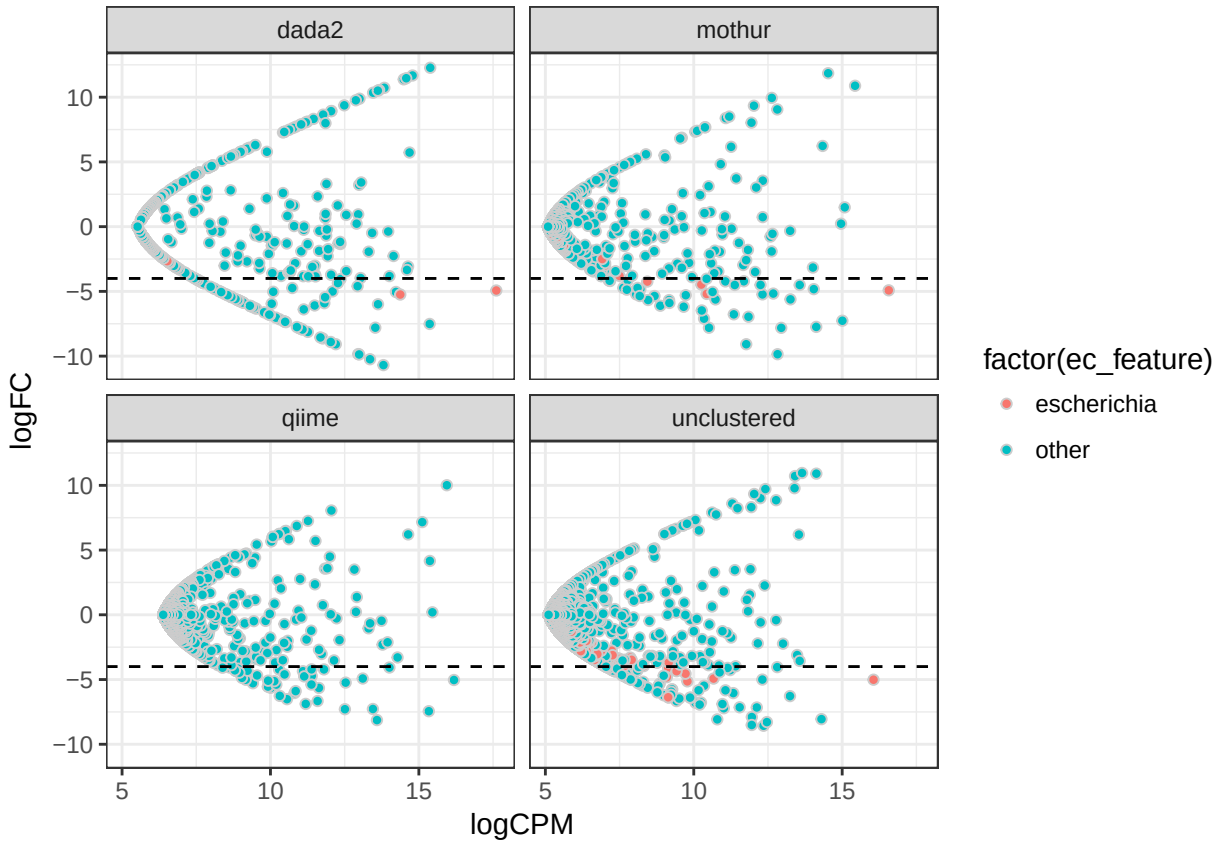
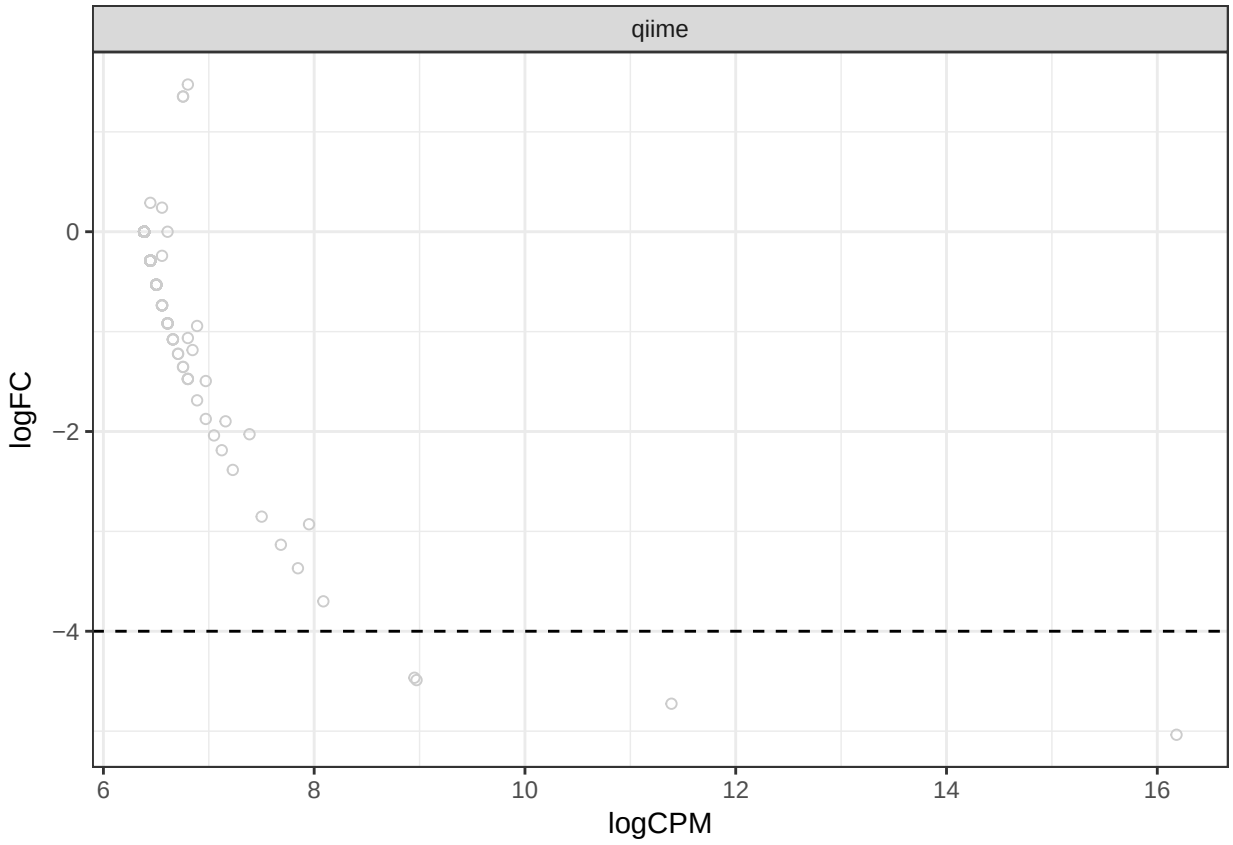


Figure 3: MA plot comparing pre and post unmixed E01JH0011 samples. Teal points indicates features classified as Escherichia.

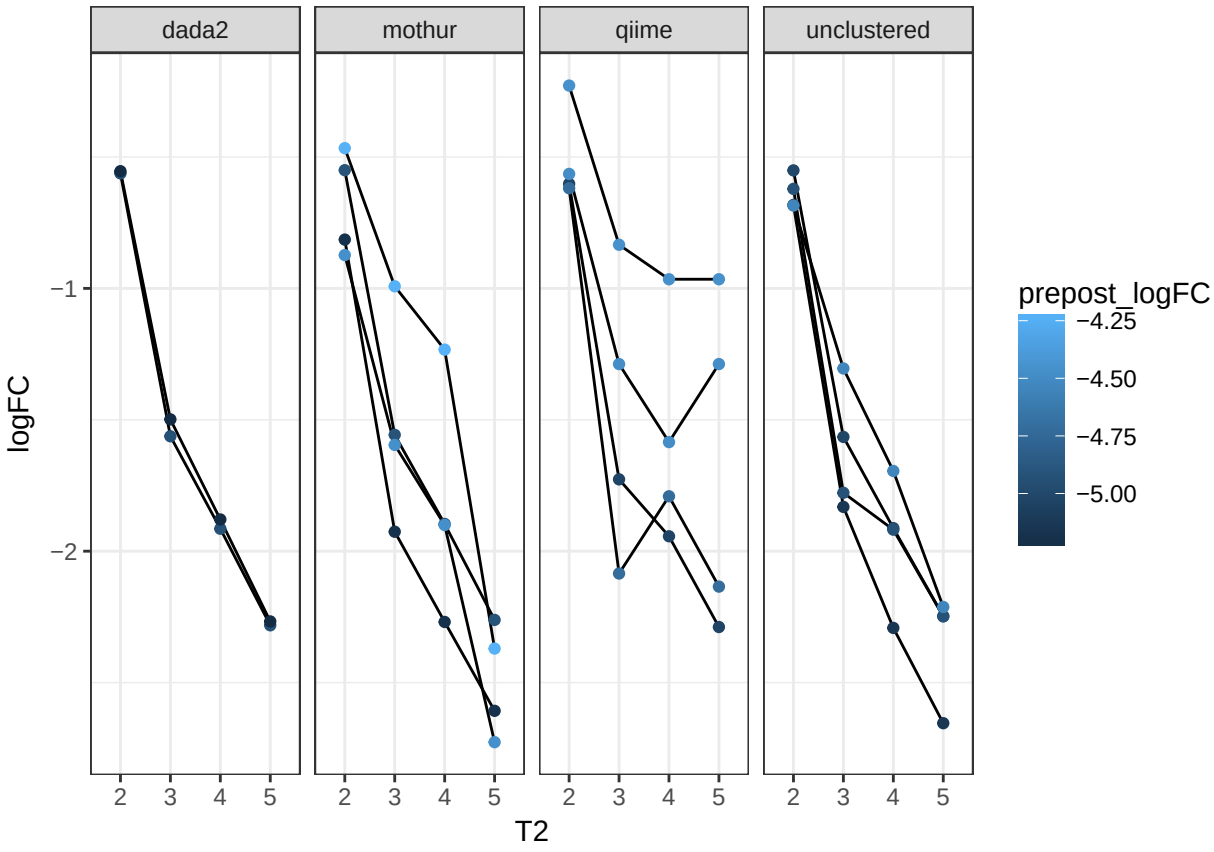
```
                color = "grey80", shape = 21) +
      geom_hline(aes(yintercept = -4), linetype = 2) +
      facet_wrap(~pipe) + theme_bw()
```



Subset of Escherichia features that are well behaved.

```
logFC_biosam_11 %>%
      mutate(ec_feature = if_else(Rank5 == "f__Enterobacteriaceae",
                                  "escherichia",ec_feature)) %>%
      filter(T1 == 0, T2 == 20, pa_mixed != 0,
             ec_feature == "escherichia", logFC < -4) %>%
      group_by(pipe) %>%
      top_n(n = 4, wt = logCPM) %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC) %>%
      rename(prepost_logCPM = logCPM, prepost_logFC = logFC) %>%
      left_join(logFC_biosam_11) %>%
      filter(T1 == 1, T2 %in% 1:5) %>%
      ggplot() +
      geom_path(aes(x = T2, y = logFC, group = feature_id)) +
            geom_point(aes(x = T2, y = logFC, color = prepost_logFC)) +
      theme_bw() + facet_wrap(~pipe, nrow = 1)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```



0.4 Other Post-Dominant Features

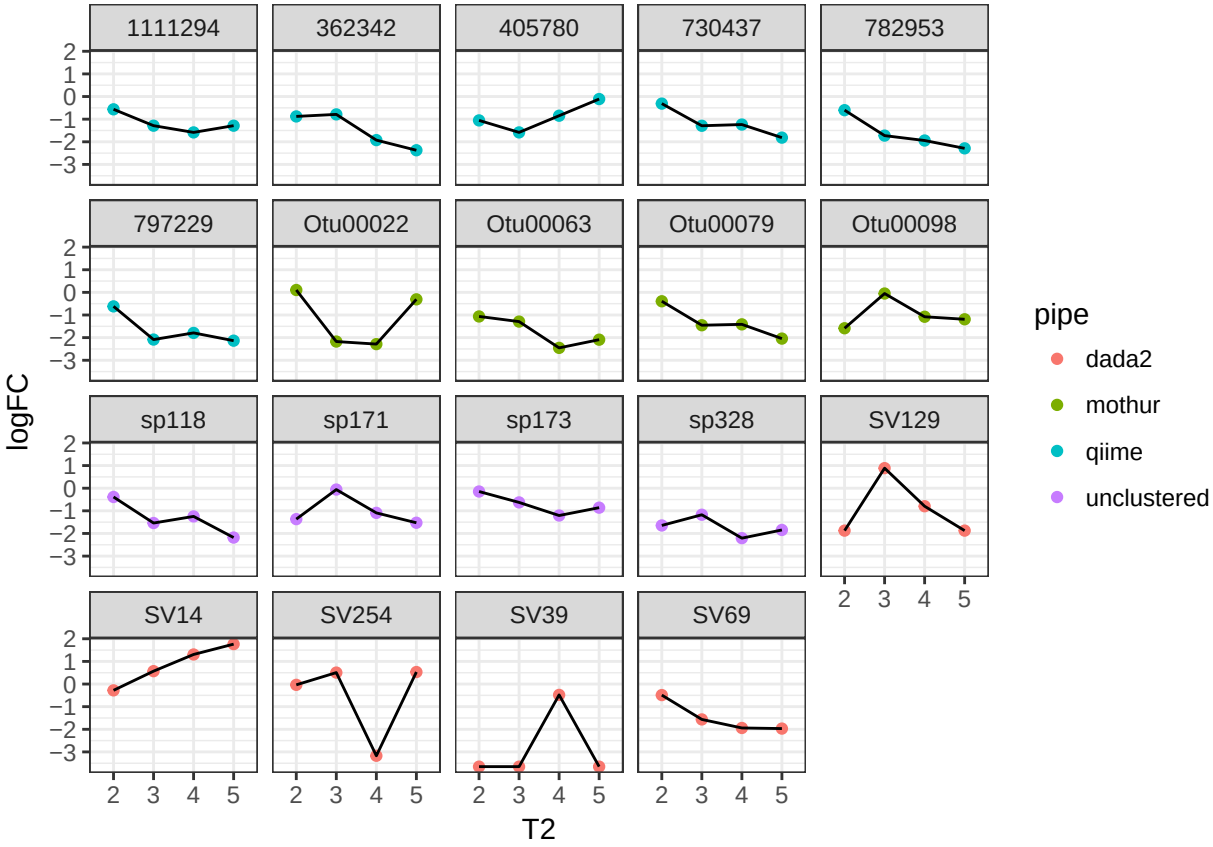
As **Escherichia** was not a post-specific feature but was significantly more abundant in post-exposure samples than pre-exposure samples we looked at the logFC between the first titration and the second - fifth titrations. Overall the logFC values are not consistent with our expectations or with other feature with similar pre- post logFC characteristics. Though feature behavior is not always consistent with our expectations.

```
other_features_logFC <- logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20,
             post_specific != 1,
             pa_mixed != 0,
             ec_feature != "escherichia",
             logFC < -4) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC, pa_mixed) %>%
      mutate(pp_logCPM_bin = cut_number(n = 4, logCPM),
             pa_mixed_bin = cut_number(n = 4, pa_mixed)) %>%
      rename(prepost_logCPM = logCPM, prepost_logFC = logFC) %>%
      left_join(logFC_biosam_11)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id", "pa_mixed")
```

```
other_features_logFC %>%
      ungroup() %>%
      filter(T1 == 1, T2 %in% 1:5)  %>%
      group_by(feature_id) %>%
      mutate(med_logFC = median(logFC),
```

```
            range_logFC = max(abs(logFC)) - min(abs(logFC))) %>%
      filter(med_logFC != 0, range_logFC > 1) %>%
      ggplot() +
      geom_point(aes(x = T2, y = logFC, color = pipe)) +
      geom_line(aes(x = T2, y = logFC, group = feature_id)) +
      theme_bw()  + facet_wrap(~feature_id)
```



0.5 Characterizing Feature Behavior

To determine whether logFC value are inconsistent with our expectations or due to varaiablity in the measurement process we characterized features based on prepost_logFC, prepost_CPM, pa_mixed, linear model for T1 = 1 and T2 = 1:5 - R2 and slope.

```
logFC_model_dat <- logFC_biosam_11 %>%
      ## Only including features with significant pre-post differential abundance estimates
      filter(T1 == 0, T2 == 20,pa_mixed != 0, FDR < 0.05) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC, pa_mixed, post_specific) %>%
      rename(prepost_logCPM = logCPM, prepost_logFC = logFC) %>%
      left_join(logFC_biosam_11) %>%
      filter(T1 == 1, T2 %in% 2:5)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id", "pa_mixed", "post_specific")
```

```
## fitting a linear model to the logFC between the first titration and titrations 2-5.
logFC_model_fit <- logFC_model_dat %>%
      mutate(T2 = as.numeric(as.character(T2))) %>%
```

```r
    group_by(pipe, feature_id, prepost_logFC, prepost_logCPM, pa_mixed) %>%
    mutate(mean_logFC = mean(logFC)) %>%
    ## excluding features with no change between titrations - all logFC 0 most
    ## likely 0 abundance features
    filter(mean_logFC != 0) %>%
    nest() %>%
    mutate(fit = map(data, ~lm(logFC~T2, data = .)),
           fit_glance = map(fit, glance),
           fit_tidy = map(fit, tidy))
```

```
## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in stats::summary.lm(x): essentially perfect fit: summary may be
## unreliable
```

```
## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable

## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable
```

```r
logFC_lm_glance <- logFC_model_fit %>% select(-data, -fit, -fit_tidy) %>% unnest() %>%
      select(-p.value, -statistic)

logFC_lm_df <- logFC_model_fit %>% select(-data, -fit, -fit_glance) %>% unnest() %>%
      left_join(logFC_lm_glance) %>%
      mutate(term = if_else(term == "(Intercept)", "intercept", "slope"))
```

```
## Joining, by = c("pipe", "feature_id", "prepost_logFC", "prepost_logCPM", "pa_mixed")
```

High R2 values tend to increase as slope estimates get further from 0.

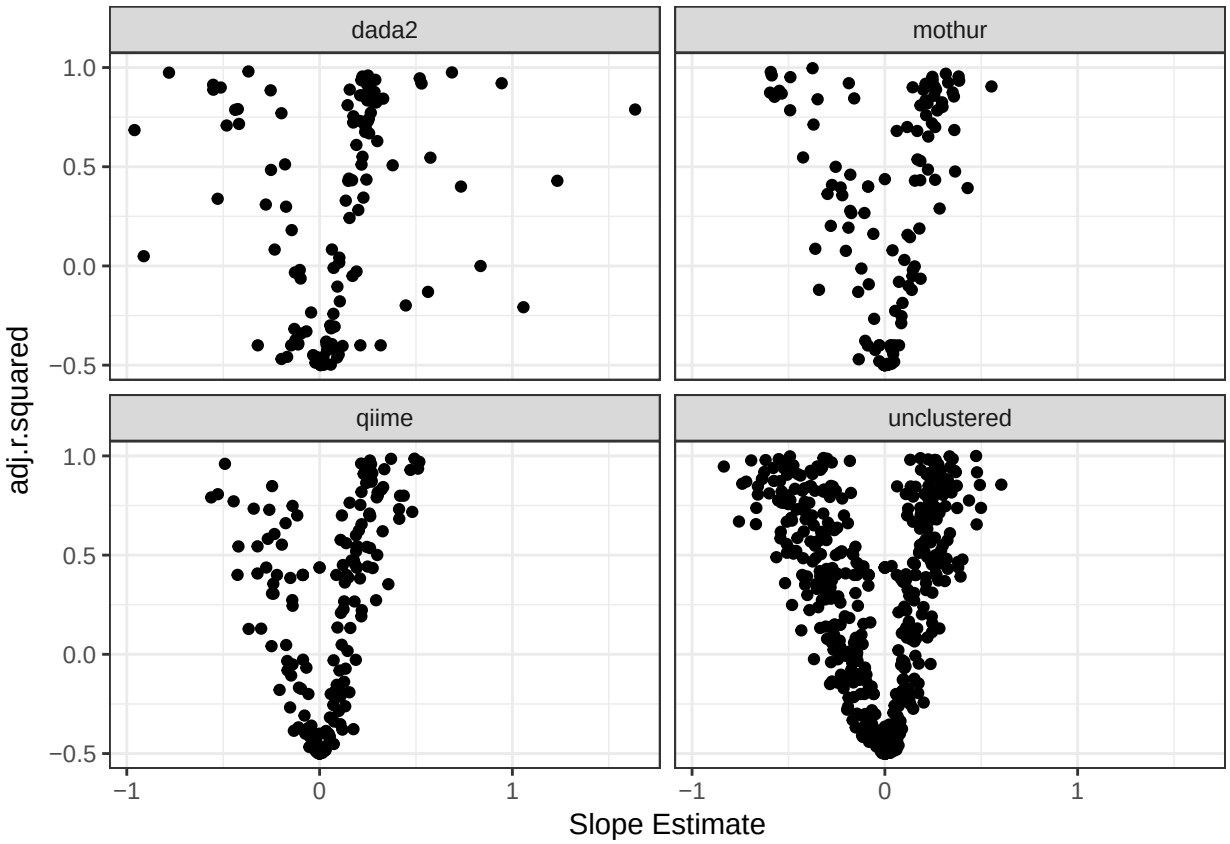


Figure 4: Relationship between R2 and slope estimates.

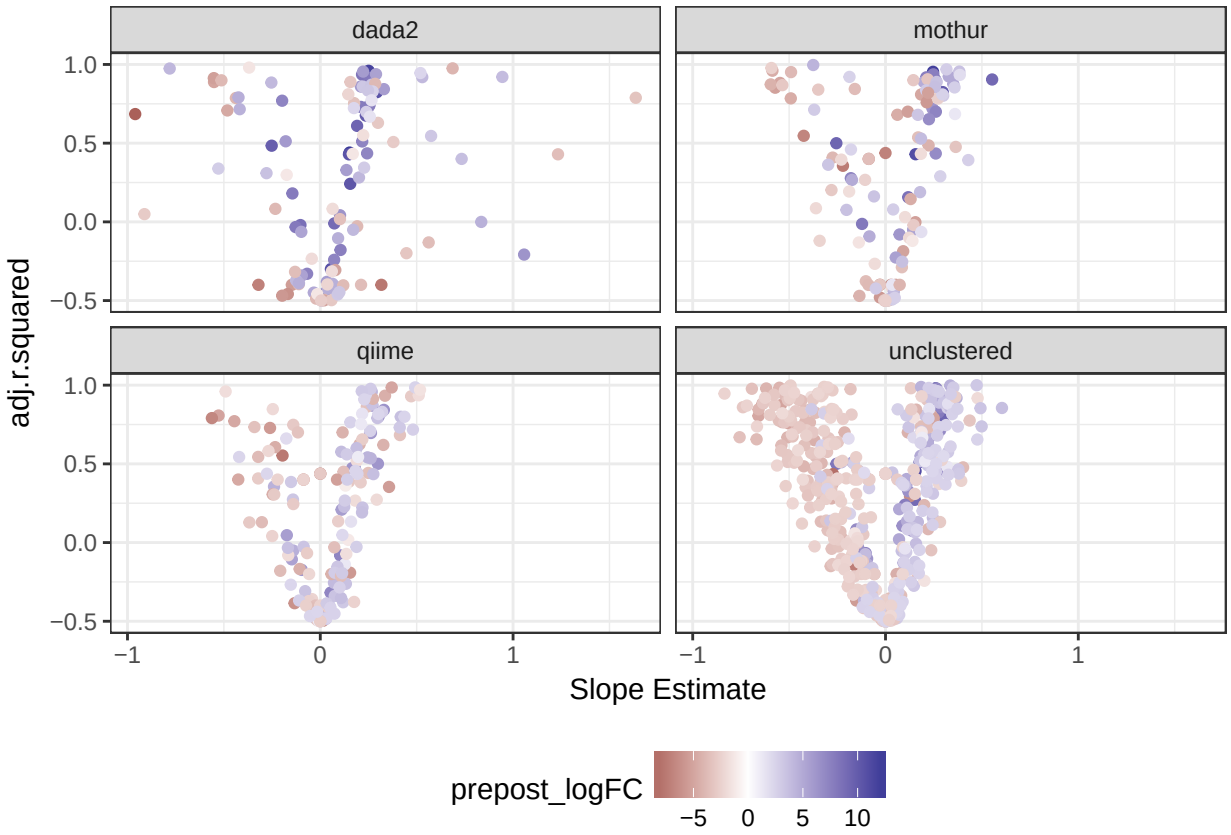
```
logFC_lm_df %>% filter(term == "slope") %>%
      ggplot() + geom_point(aes(x = estimate, y = adj.r.squared)) +
      theme_bw() + theme(legend.position = "bottom") + facet_wrap(~pipe) +
      labs(x = "Slope Estimate")
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

No clear relationship between pre-post logFC or logCPM and slope estimate or R2.

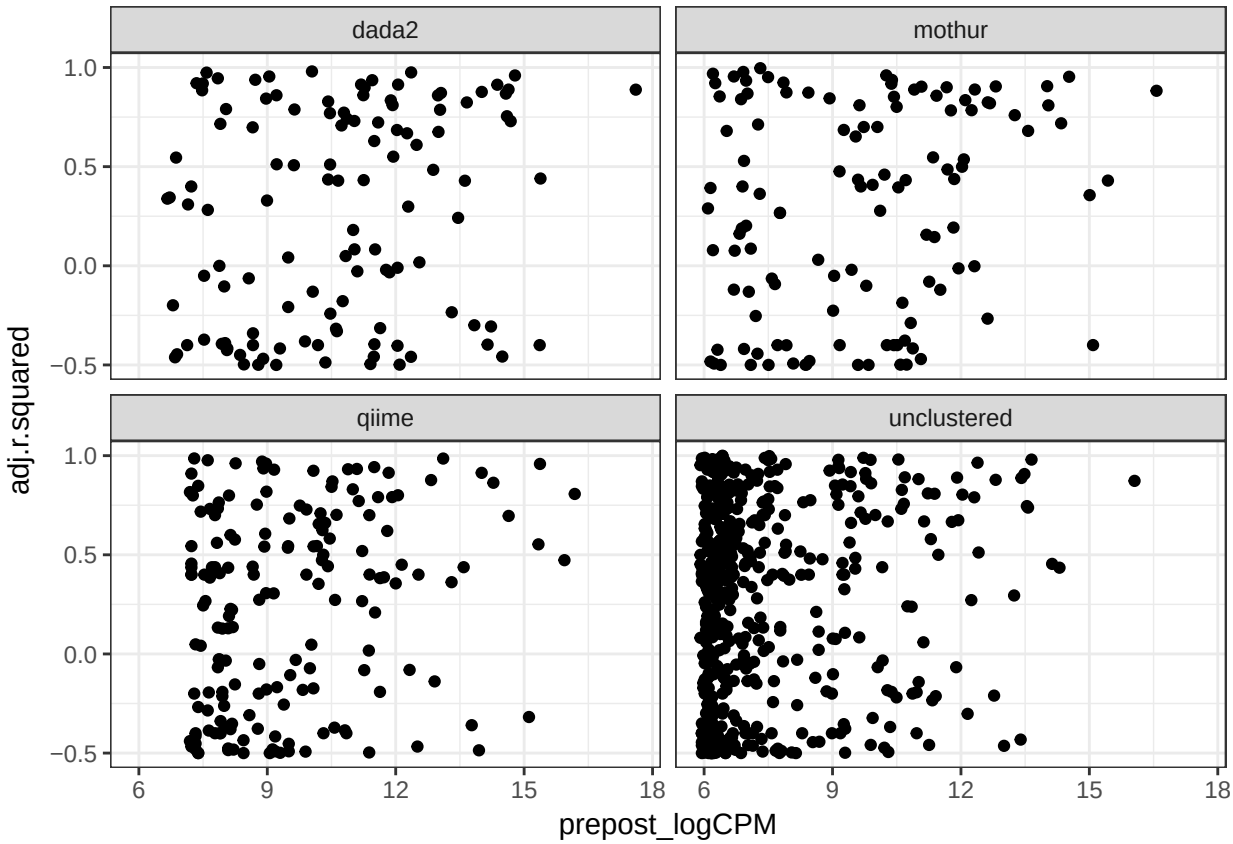
```
logFC_lm_df %>% filter(term == "slope") %>%
      ggplot() + geom_point(aes(x = estimate, y = adj.r.squared, color = prepost_logFC)) +
      theme_bw() + theme(legend.position = "bottom") + facet_wrap(~pipe) +
      scale_color_gradient2() + labs(x = "Slope Estimate")
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
logFC_lm_df %>% filter(term == "slope") %>%
      ggplot() + geom_point(aes(x = prepost_logCPM, y = adj.r.squared)) +
      theme_bw() + theme(legend.position = "bottom") + facet_wrap(~pipe)
```

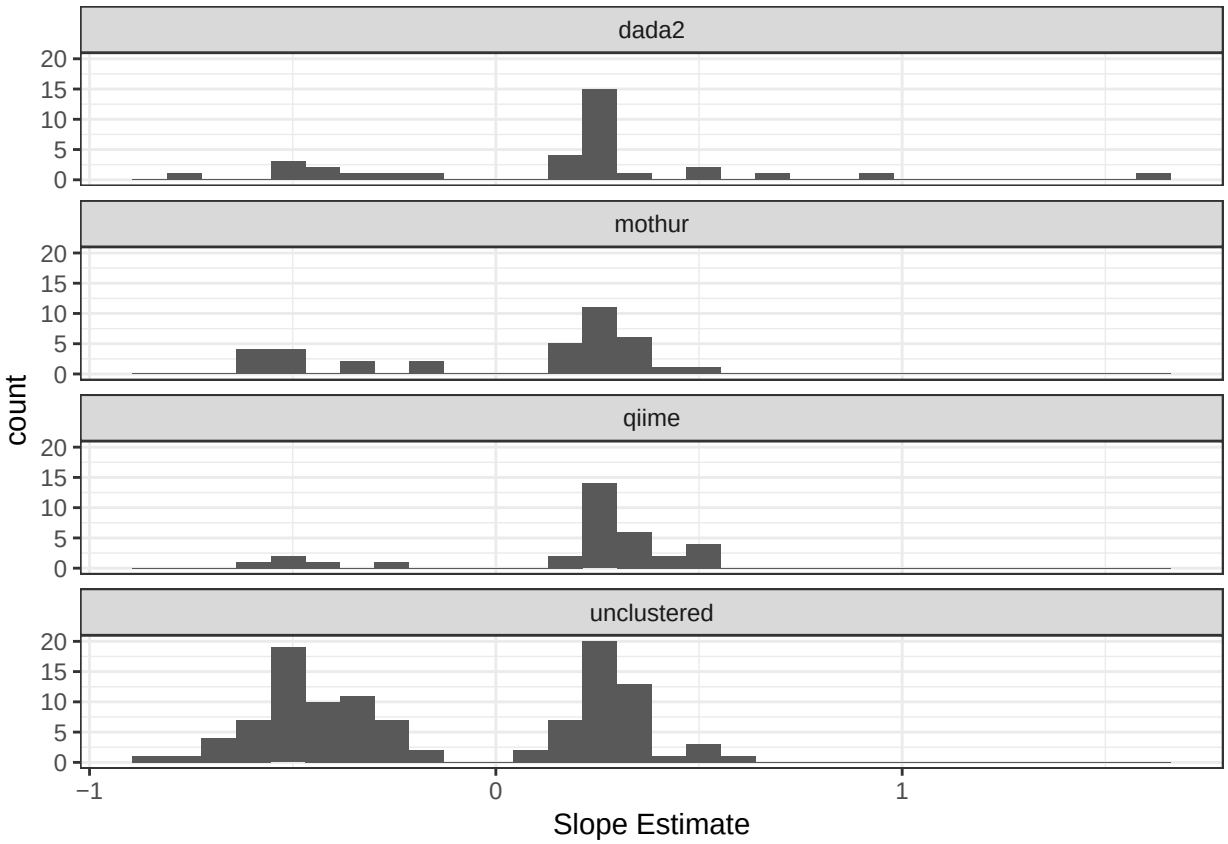
```
## Warning: Removed 5 rows containing missing values (geom_point).
```



For features with high R2 > 0.75, there are peaks in the slope estimates around 0.25 and -0.5. Assuming the pre- and post-exposure samples were mixed according to the experimental design a slope of 1 is expected.

```
logFC_lm_df %>% filter(term == "slope", adj.r.squared > 0.75) %>%
      ggplot() + geom_histogram(aes(x = estimate)) +
      theme_bw() +
      theme(legend.position = "bottom") +
      facet_wrap(~pipe, ncol = 1) +
      labs(x = "Slope Estimate")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



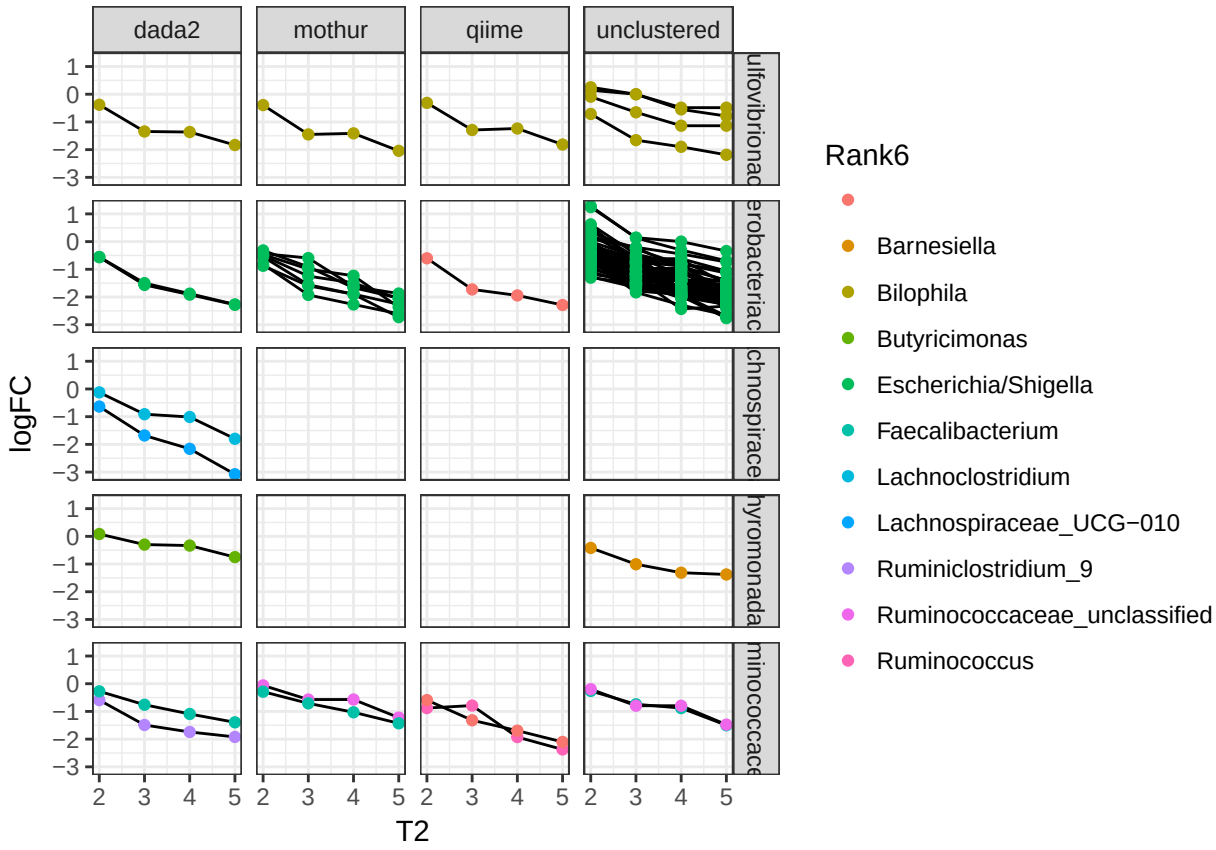

```
lm_feature_subset <- logFC_lm_df %>% filter(term == "slope") %>%
      filter(adj.r.squared > 0.75, estimate < -0.25) %>%
      select(pipe, feature_id) %>%
      left_join(logFC_model_fit) %>%
      select(-fit, -fit_glance, -fit_tidy) %>%
      unnest()
```

```
## Joining, by = c("pipe", "feature_id")
```

logFC estimate performance is independent of pre-post logFC and pre-exposure sample abundance. Though a large logFC and starting relative abundance are necessary to detect changes in relative abundance across titrations.

Escherichia/Shigella, Ruminoccocus, Sulfovibrioaceae are the only taxa that has consistent behavior across pipelines.

```
lm_feature_subset %>%
      mutate(Rank3 = str_replace(Rank3, "c__",""),
             Rank4 = str_replace(Rank4, "o__",""),
             Rank5 = str_replace(Rank5, "f__",""),
             Rank6 = str_replace(Rank6, "g__","")) %>%
      ggplot() +
      geom_line(aes(x = T2, y = logFC, group = feature_id)) +
      geom_point(aes(x = T2, y = logFC, color = Rank6)) +
      facet_grid(Rank5~pipe) +
      theme_bw()
```

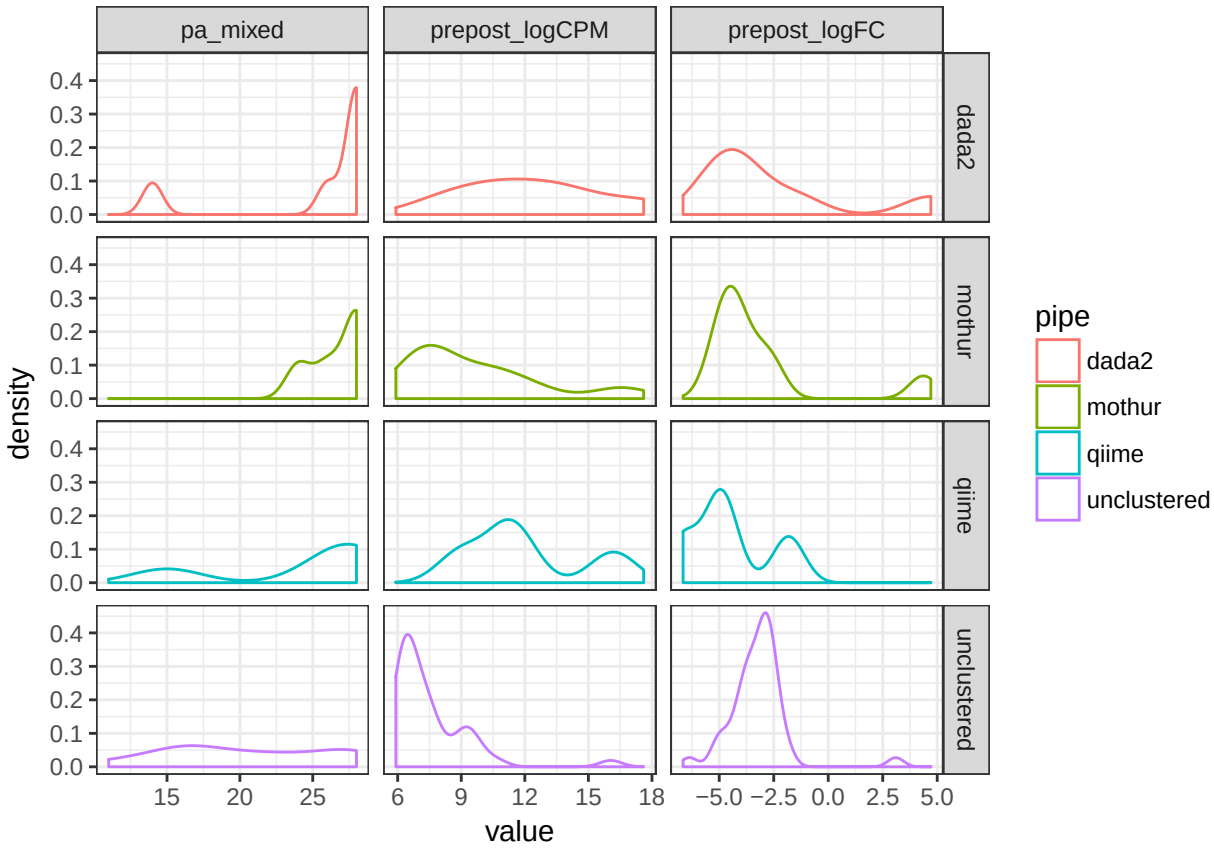



```
lm_feature_subset <- logFC_lm_df %>%
      filter(term == "slope") %>%
      filter(adj.r.squared > 0.75, estimate < -0.35, estimate > -0.65) %>%
      select(pipe, feature_id) %>%
      left_join(logFC_model_fit) %>%
      select(-fit, -fit_glance, -fit_tidy) %>%
      unnest()
```

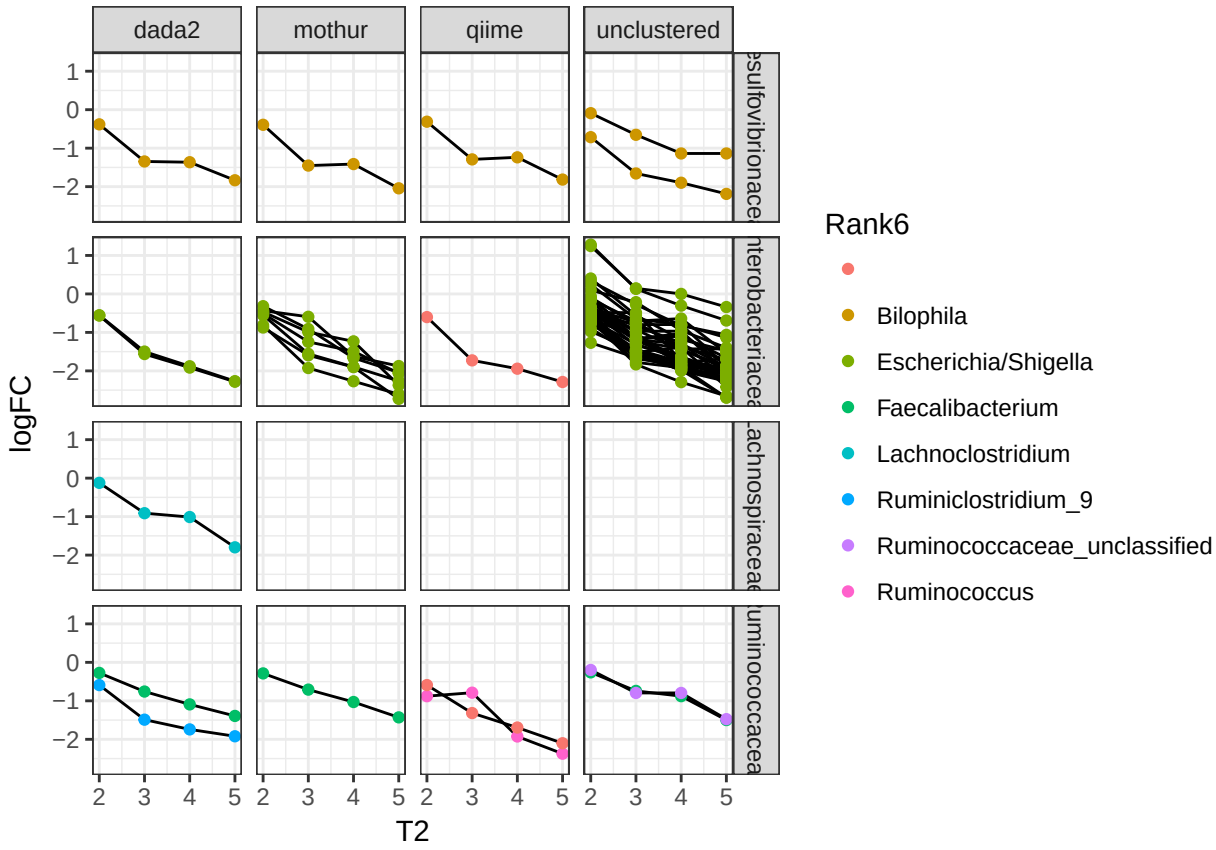
```
## Joining, by = c("pipe", "feature_id")
```

Some features with negative slope estimates but positive pre-post logFC estimates, these are inconsistent with expectations.

```
logFC_lm_df %>%
      filter(term == "slope") %>%
      filter(adj.r.squared > 0.75, estimate < -0.35, estimate > -0.65) %>%
      select(pipe, feature_id, prepost_logFC, prepost_logCPM,
             pa_mixed) %>%
      gather("key","value", -pipe, -feature_id) %>%
      ggplot() + geom_density(aes(x = value, color = pipe)) +
      facet_grid(pipe~key, scales = "free_x") + theme_bw()
```




```
lm_feature_subset %>%
      mutate(Rank3 = str_replace(Rank3, "c__",""),
             Rank4 = str_replace(Rank4, "o__",""),
             Rank5 = str_replace(Rank5, "f__",""),
             Rank6 = str_replace(Rank6, "g__","")) %>%
      ggplot() +
      geom_line(aes(x = T2, y = logFC, group = feature_id)) +
      geom_point(aes(x = T2, y = logFC, color = Rank6)) +
      facet_grid(Rank5~pipe) +
      theme_bw()
```

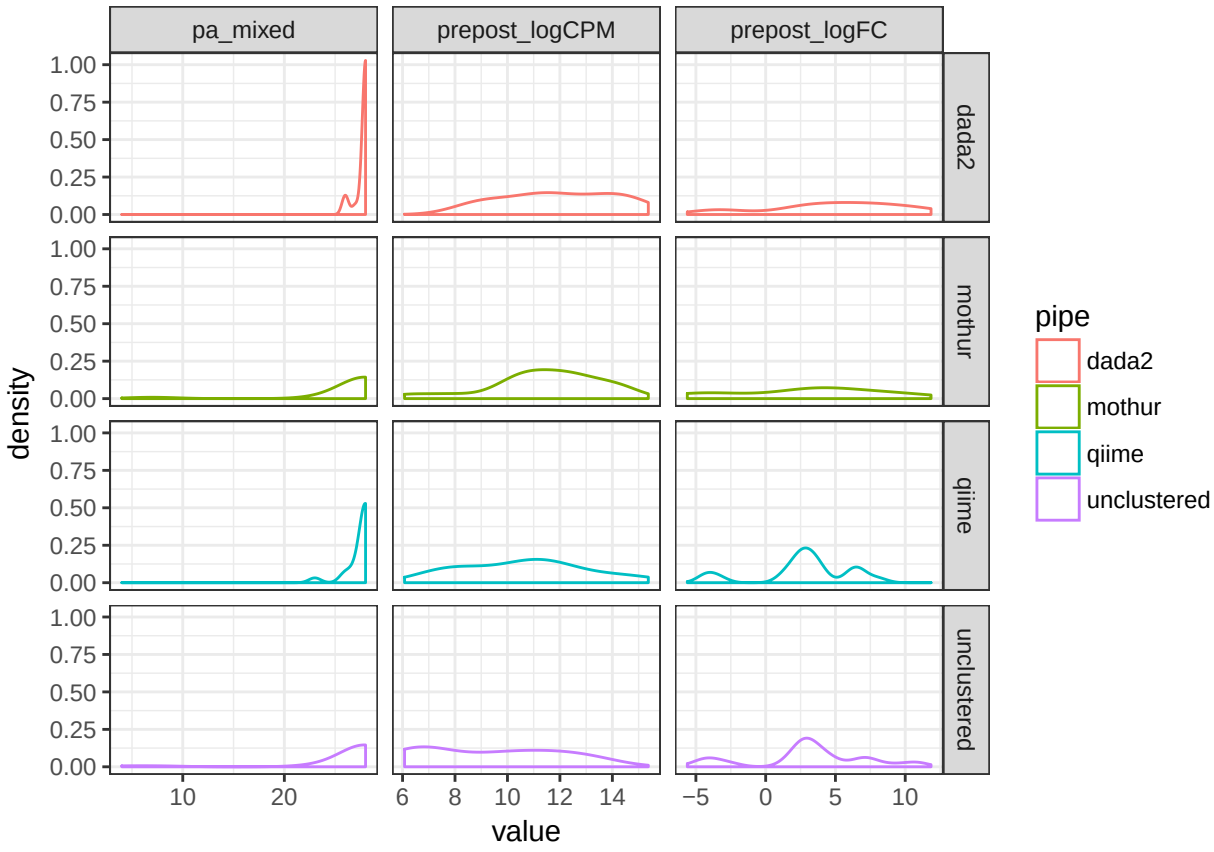


Features with slope estimates around the peak in distribution of 0.25

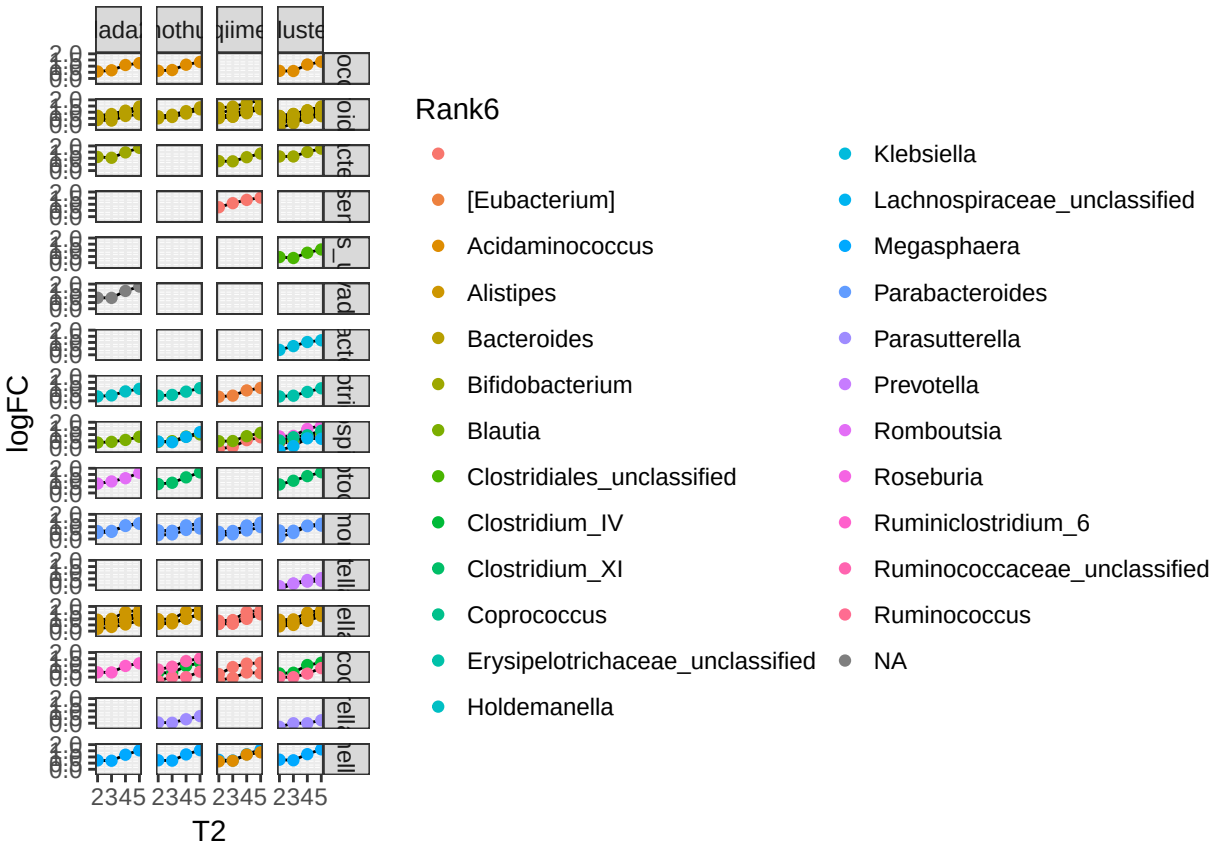
```
lm_feature_subset <- logFC_lm_df %>%
      filter(term == "slope") %>%
      filter(adj.r.squared > 0.75, estimate > 0.15, estimate < 0.35) %>%
      select(pipe, feature_id) %>%
      left_join(logFC_model_fit) %>%
      select(-fit, -fit_glance, -fit_tidy) %>%
      unnest()
```

```
## Joining, by = c("pipe", "feature_id")
```

```
logFC_lm_df %>%
      filter(term == "slope") %>%
      filter(adj.r.squared > 0.75, estimate > 0.15, estimate < 0.35) %>%
      select(pipe, feature_id, prepost_logFC, prepost_logCPM,
             pa_mixed) %>%
      gather("key","value", -pipe, -feature_id) %>%
      ggplot() + geom_density(aes(x = value, color = pipe)) +
      facet_grid(pipe~key, scales = "free_x") + theme_bw()
```




```
lm_feature_subset %>%
      mutate(Rank3 = str_replace(Rank3, "c__",""),
             Rank4 = str_replace(Rank4, "o__",""),
             Rank5 = str_replace(Rank5, "f__",""),
             Rank6 = str_replace(Rank6, "g__","")) %>%
      ggplot() +
      geom_line(aes(x = T2, y = logFC, group = feature_id)) +
      geom_point(aes(x = T2, y = logFC, color = Rank6)) +
      facet_grid(Rank5~pipe) +
      theme_bw()
```



0.6 Regression Tree

Interested in the relationship between logFC, logCPM, and number of titration PCR replicates with observed counts to the linear model fit - R2 and slope estimate.

```
logFC_lm_anno <- logFC_lm_df %>%
      filter(term == "slope")
```

Looking at features with negative slope estimates first

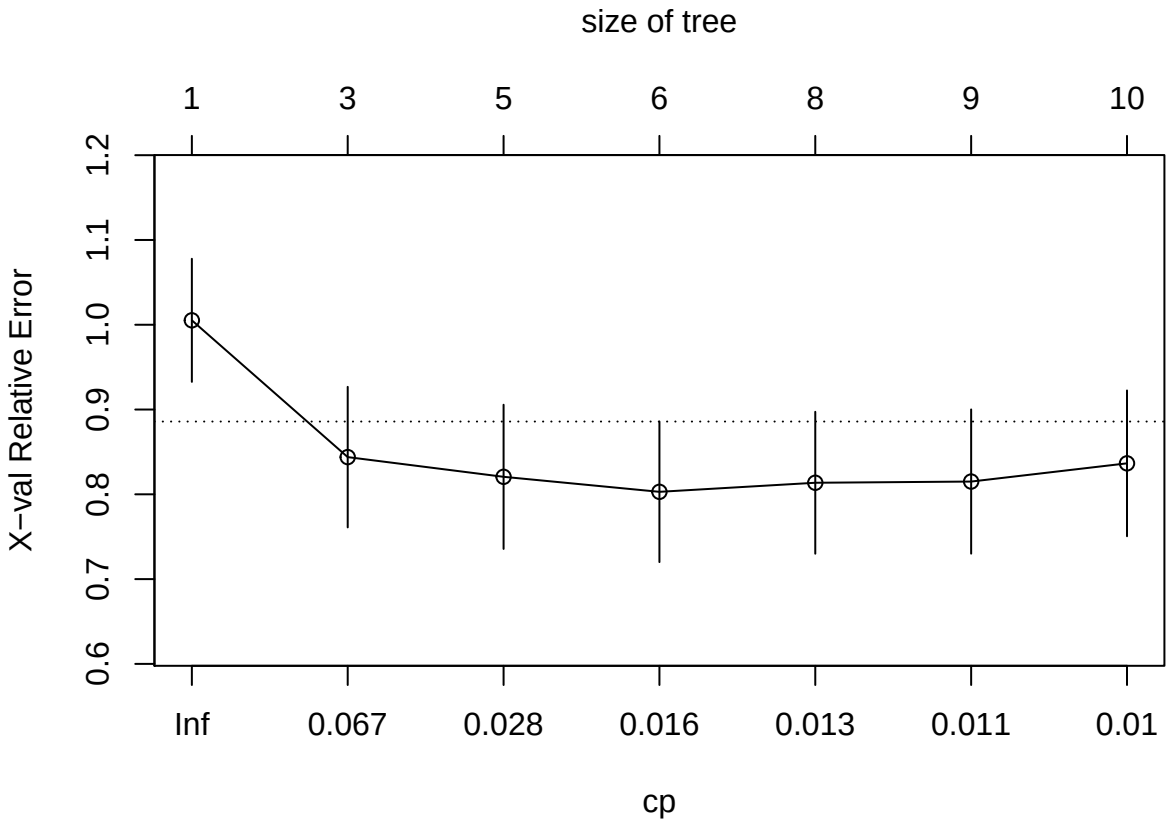
```
library(rpart)
fit <- rpart(estimate ~ prepost_logFC + prepost_logCPM + pa_mixed,
             data=logFC_lm_anno %>% filter(estimate < 0))
```

```
printcp(fit)
```

```
##
## Regression tree:
## rpart(formula = estimate ~ prepost_logFC + prepost_logCPM + pa_mixed,
##     data = logFC_lm_anno %>% filter(estimate < 0))
##
## Variables actually used in tree construction:
## [1] pa_mixed       prepost_logCPM prepost_logFC
##
## Root node error: 14.198/410 = 0.034629
##
## n= 410
```

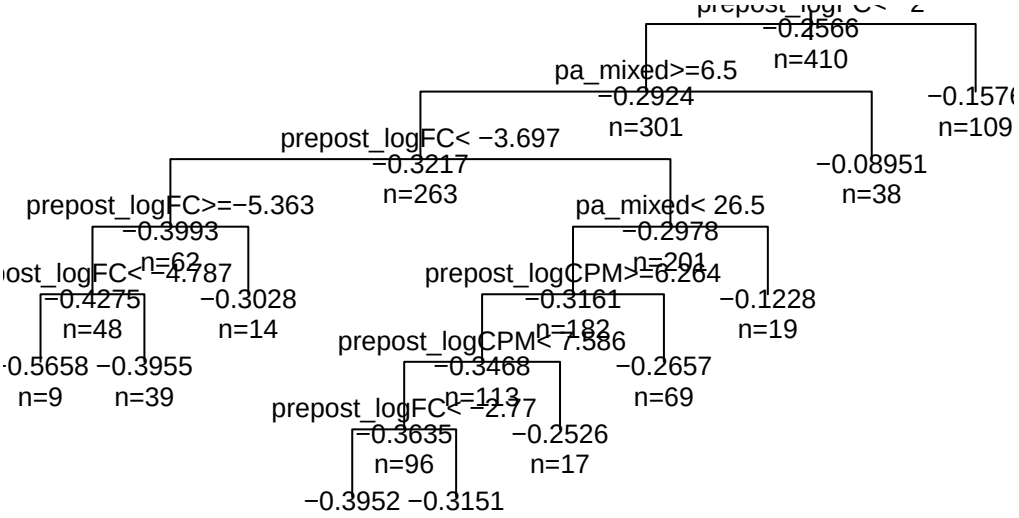
```
##
##         CP nsplit rel error  xerror     xstd
## 1 0.114235      0   1.00000 1.00523 0.072598
## 2 0.039824      2   0.77153 0.84389 0.082897
## 3 0.019852      4   0.69188 0.82064 0.085031
## 4 0.013396      5   0.67203 0.80295 0.082907
## 5 0.012512      7   0.64524 0.81364 0.083691
## 6 0.010373      8   0.63272 0.81506 0.085065
## 7 0.010000      9   0.62235 0.83652 0.085999
```

```
plotcp(fit)
```



Not sure what to make of results.
Seems as though prepost_logFC is the primary driver of the groups.

```
plot(fit, uniform=TRUE); text(fit, use.n=TRUE, all=TRUE, cex=.8)
```



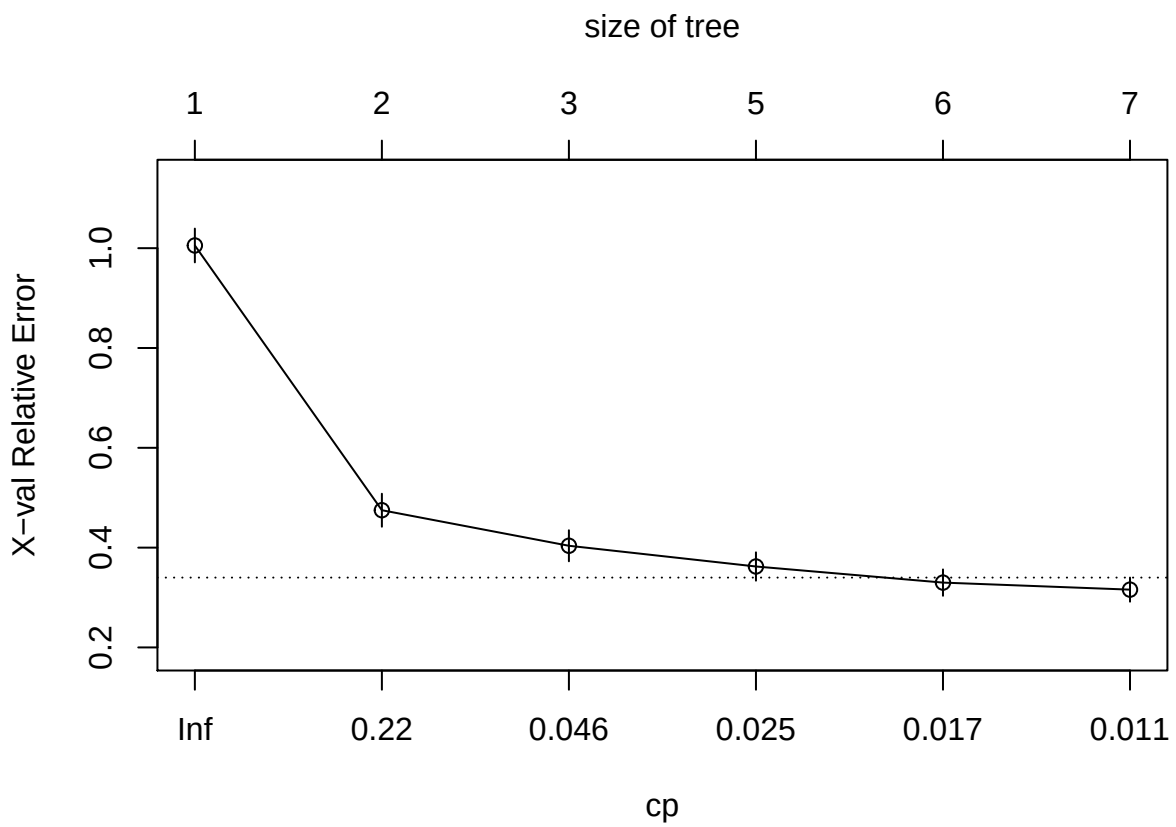
Features with positive slope estimates

```
library(rpart)
fit <- rpart(adj.r.squared ~ estimate + prepost_logFC + prepost_logCPM + pa_mixed,
             data=logFC_lm_anno %>% filter(estimate > 0))
```

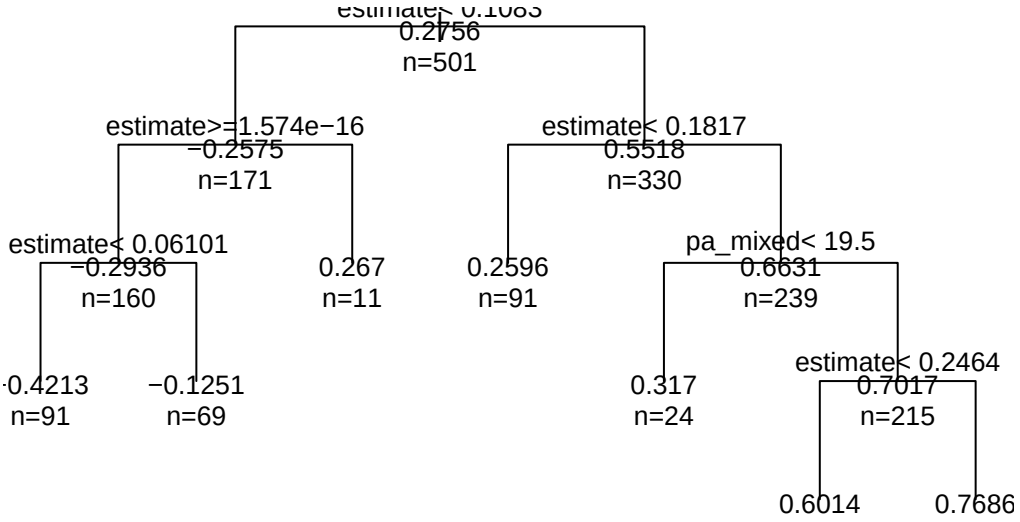
```
printcp(fit)
```

```
##
## Regression tree:
## rpart(formula = adj.r.squared ~ estimate + prepost_logFC + prepost_logCPM +
##     pa_mixed, data = logFC_lm_anno %>% filter(estimate > 0))
##
## Variables actually used in tree construction:
## [1] estimate pa_mixed
##
## Root node error: 129.37/501 = 0.25823
##
## n= 501
##
##         CP nsplit rel error  xerror     xstd
## 1 0.570321      0   1.00000 1.00540 0.033843
## 2 0.082935      1   0.42968 0.47492 0.033028
## 3 0.025809      2   0.34674 0.40372 0.031087
## 4 0.024707      4   0.29513 0.36217 0.028392
## 5 0.011144      5   0.27042 0.32990 0.026433
## 6 0.010000      6   0.25928 0.31582 0.024172
```

```
plotcp(fit)
```




```
plot(fit, uniform=TRUE); text(fit, use.n=TRUE, all=TRUE, cex=.8)
```



0.6.1 Next Steps

- Determine the relationship between feature characteristics and linear model fit. Characteristics defined as presence/ absence and pre-post logFC How to present these results? Can/Differentiating between poor logFC results due to pipelines and wet lab sample processing.

0.7 im-proptu meeting with Hector

- Look at samples where theta estimates are generally consistent with expectation
- Compare logFC change between titrations 1 and 3 for the unclustered dataset to expected value of -2 (logFC error)
- Relate logFC error to logCPM
- Expand to compare logFC error for other titration comparisons

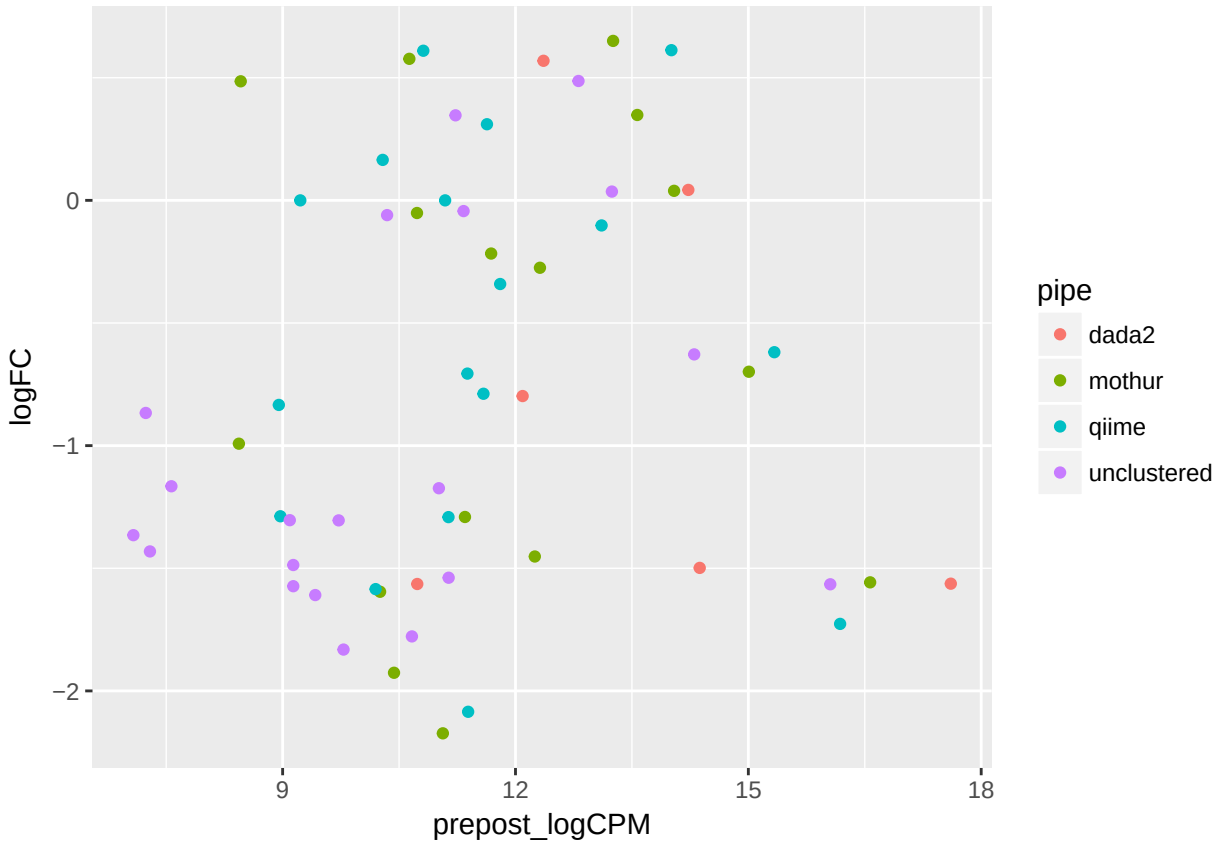
```
logFC_biosam_11_T13 <- logFC_biosam_11 %>%
      filter(T1 == 1, T2 == 3)

logFC_biosam11_T13 <- logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20, logFC < -4) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC)  %>%
      rename(prepost_logFC = logFC, prepost_logCPM = logCPM) %>%
      left_join(logFC_biosam_11_T13)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```

```
logFC_biosam11_T13 %>%
      filter(pa_mixed > 8) %>%
      group_by(pipe) %>% summarise(count = n())
```

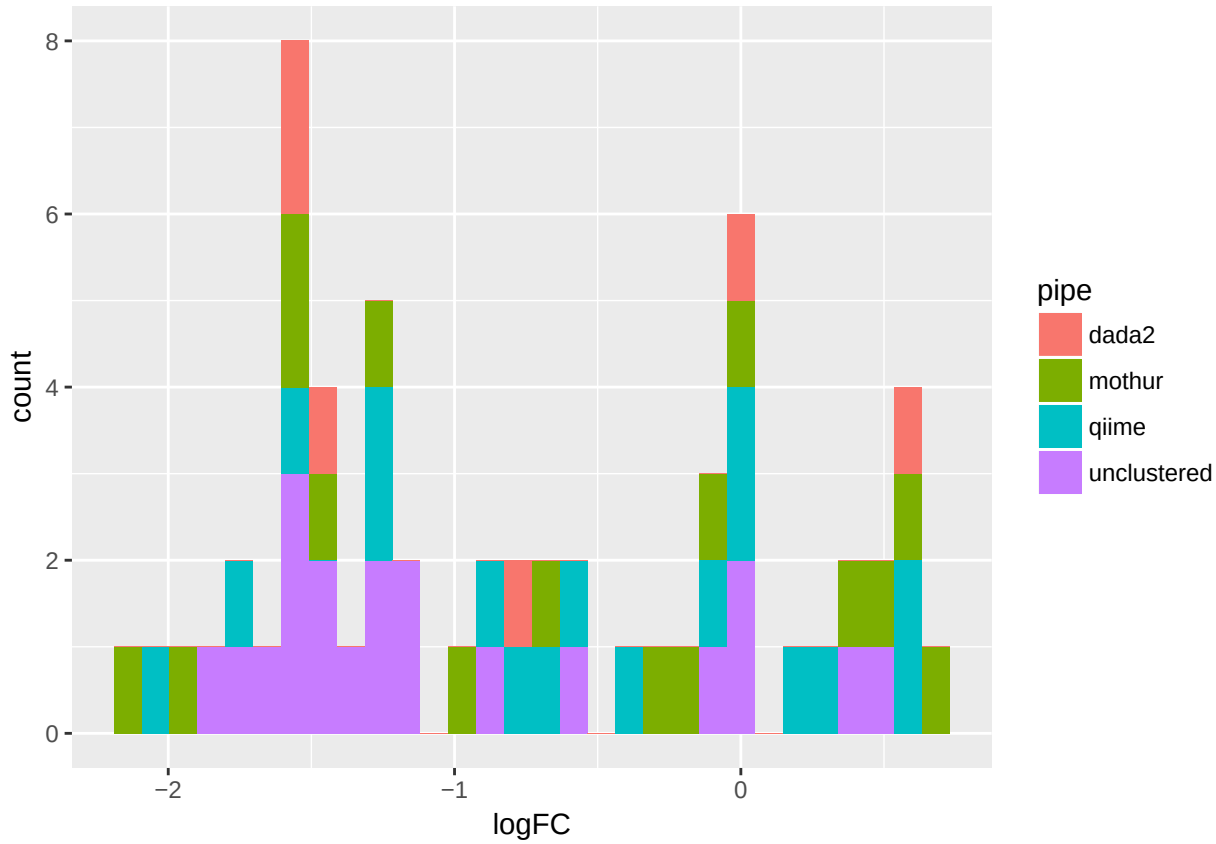
```
## # A tibble: 4 x 2
##          pipe count
##         <chr> <int>
## 1       dada2     8
## 2      mothur    16
## 3       qiime    17
## 4 unclustered    21
```

```
logFC_biosam11_T13 %>%
      filter(pa_mixed > 12) %>%
      ggplot() + geom_point(aes(x = prepost_logCPM, y = logFC, color = pipe))
```




```
logFC_biosam11_T13 %>%
      filter(pa_mixed > 12) %>%
      ggplot() + geom_histogram(aes(x = logFC, fill = pipe))
```

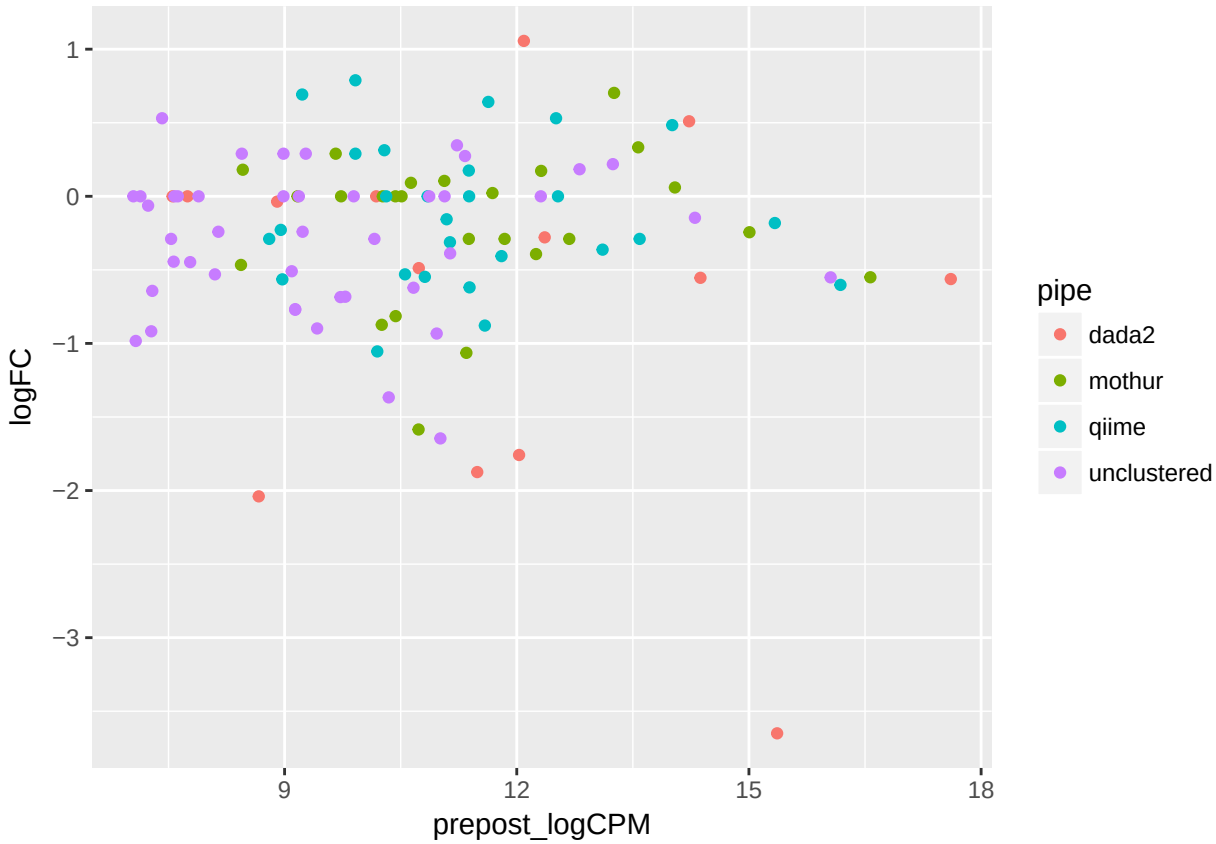
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
logFC_biosam_11_T12 <- logFC_biosam_11 %>%
      filter(T1 == 1, T2 == 2)

logFC_biosam11_T12 <- logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20, logFC < -4, pa_mixed != 0) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC)  %>%
      rename(prepost_logFC = logFC, prepost_logCPM = logCPM) %>%
      left_join(logFC_biosam_11_T12)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```

```
logFC_biosam11_T12 %>%
      ggplot() + geom_point(aes(x = prepost_logCPM, y = logFC, color = pipe))
```




```
logFC_biosam_04 <- readRDS("~/Desktop/logFC_edgeR_df.rds") %>%
      filter(biosample_id == "E01JH0004") %>%
      rename(feature_id = OTUname) %>%
      left_join(pa_summary_anno_df)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```

```
logFC_biosam_04_T13 <- logFC_biosam_11 %>%
      filter(T1 == 1, T2 == 3)

logFC_biosam04_T13 <- logFC_biosam_11 %>%
      filter(T1 == 0, T2 == 20, logFC < -4) %>%
      ungroup() %>%
      select(pipe, biosample_id, feature_id, logCPM, logFC)  %>%
      rename(prepost_logFC = logFC, prepost_logCPM = logCPM) %>%
      left_join(logFC_biosam_11_T13)
```

```
## Joining, by = c("pipe", "biosample_id", "feature_id")
```

```
logFC_biosam04_T13 %>%
      filter(pa_mixed > 8) %>%
      group_by(pipe) %>% summarise(count = n())
```

```
## # A tibble: 4 x 2
##          pipe count
##         <chr> <int>
## 1       dada2     8
## 2      mothur    16
## 3       qiime    17
```

```
## 4 unclustered    21
```

```
logFC_biosam04_T13 %>% filter(pa_mixed > 8) %>%
      ggplot() + geom_point(aes(x = prepost_logCPM, y = logFC, color = pipe))
```