Relative Abundance Normalization Method Comparison

Nate Olson

2017-12-04

Comparison of relative abundance error rate for different normalization methods

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(ggridges)
```

```
norm_count_df <- readRDS("~/Desktop/norm_count_df.RDS")
```

Mean variance relationship by normalization method - not sure if the difference is due to scaling or normalization method.

```
filtered_norm <- norm_count_df %>%
    filter(mean_count != 0, var_count > 1e-10)

ggplot(filtered_norm) +
    geom_hex(aes(x = mean_count, y = var_count)) +
    geom_smooth(aes(x = mean_count, y = var_count)) +
    geom_abline(aes(intercept = 0, slope = 1 ), color = "darkorange") +
    facet_wrap(~norm_method) +
    theme_bw() + scale_y_log10() + scale_x_log10() +
    labs(x = "Mean", y = "Variance") +
    # coord_equal() +
    theme(legend.position = "bottom", axis.text.x = element_text(angle = 315))
```

```
## `geom_smooth()` using method = 'gam'
```

Calculating Error Rate

```
pa_summary_anno_df <- readRDS("~/Desktop/to_file/mgtst_RDS/pa_summary_anno_df.RDS")
theta_est <- readRDS("~/Desktop/to_file/mgtst_RDS/bootstrap_theta_estimates.rds")

pre_post_prop <- norm_count_df %>%
      ungroup() %>%
      filter(t_fctr %in% c(0,20)) %>%
      mutate(end_point = if_else(t_fctr == 0 , "post", "pre")) %>%
      select(-t_fctr, -var_count) %>%
      ## setting values to 0 when one or more of the PCR replicates are 0 for titration end-points
      spread(end_point,mean_count, fill = 0)
```

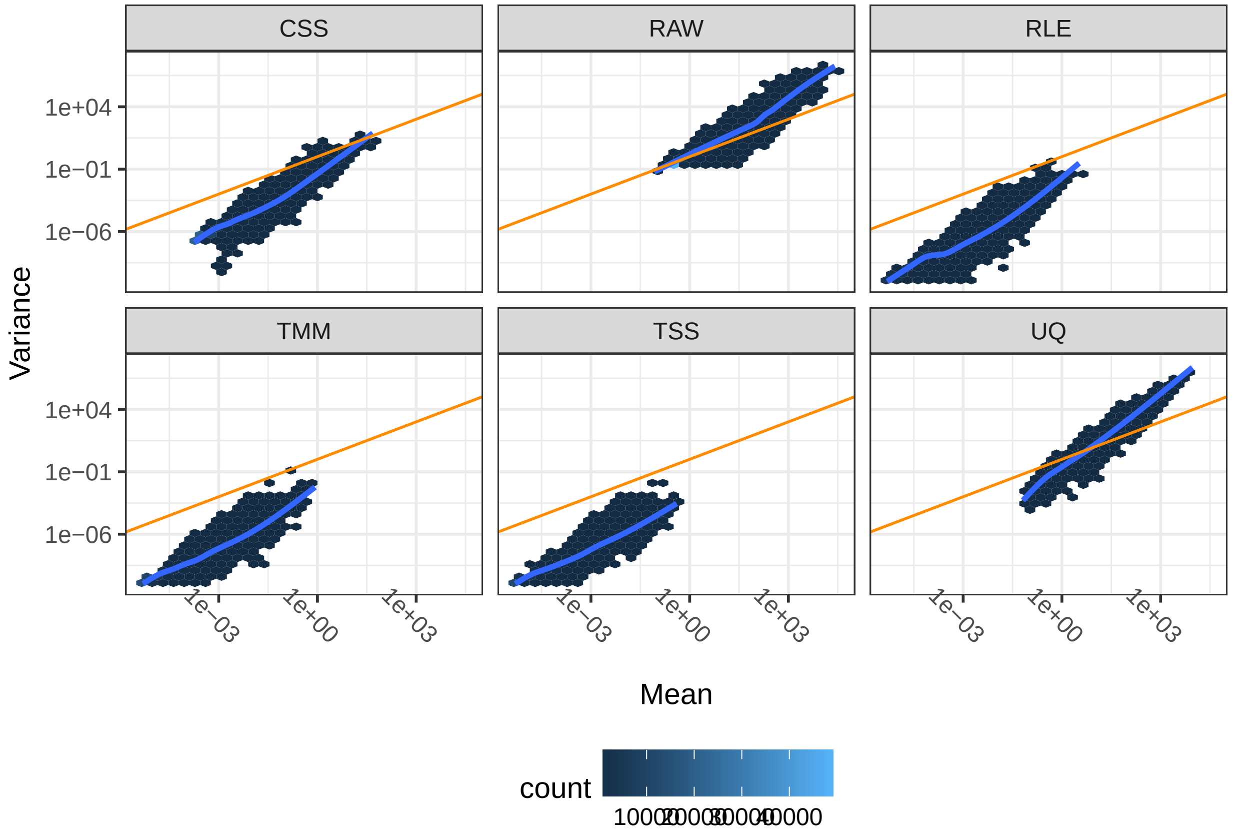


Figure 1: Comparison of relative abundance mean and variance relationship for PCR replicates across normalization methods. RLE - relative log expression, TMM - weighted trim mean of M-values, RAW - unnormalized, CSS - cumulative sum scaling, TSS - total sum scaling, UQ - upperquartile. Points with means of zero and variance < 1e-10 were excluded from the plot

```
prop_inferred <- theta_est %>%
      filter(pipe == "unclustered") %>%
      ungroup() %>%
      mutate(t_fctr = factor(t_fctr, levels = c(0:5, 10, 15, 20))) %>%
      select(biosample_id, theta_hat_mean, t_fctr) %>%
      right_join(norm_count_df) %>%
    right_join(pre_post_prop) %>%
      filter(t_fctr %in% c(1:5,10,15)) %>%
      ## Using inferred theta estimates to calculate expected values
      mutate(inferred_prop = post * theta_hat_mean + pre * (1 - theta_hat_mean))
```

```
## Joining, by = c("biosample_id", "t_fctr")
```

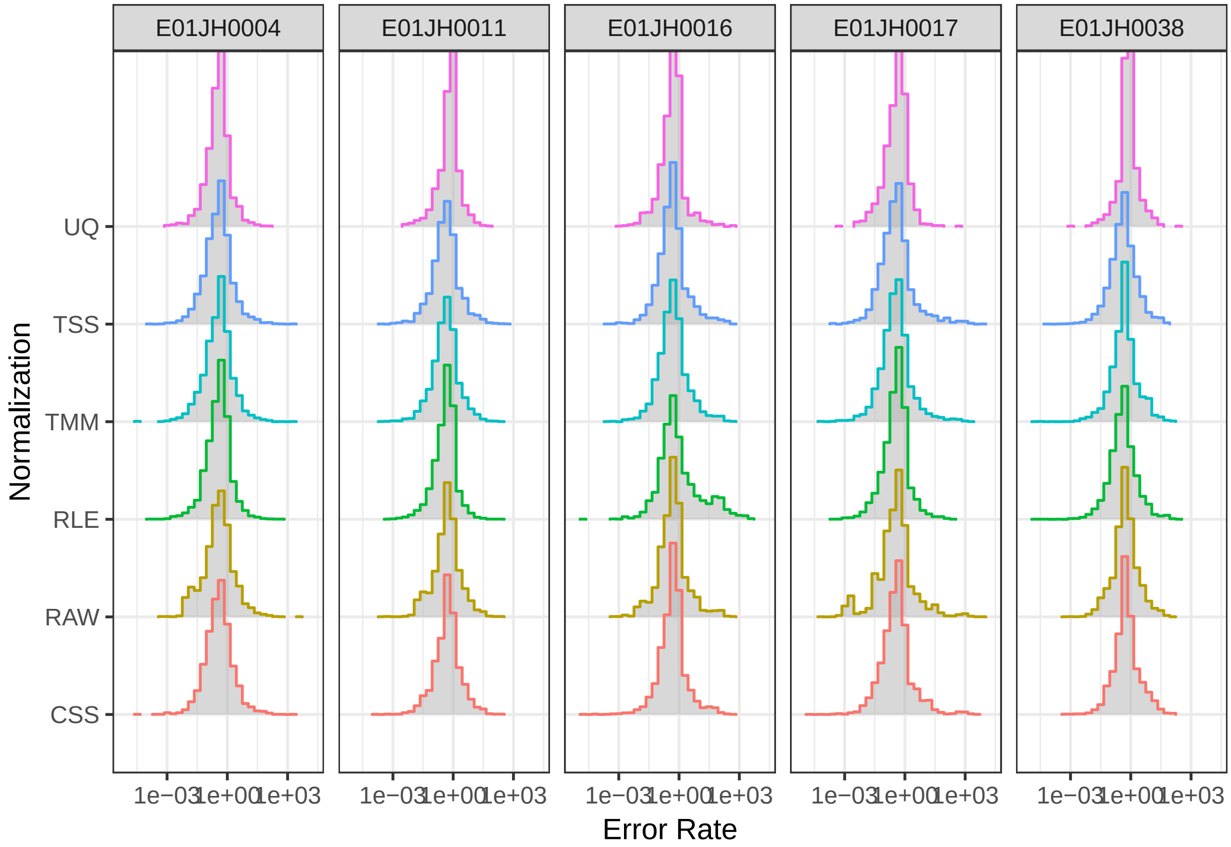
```
## Warning: Column `t_fctr` joining factors with different levels, coercing to
## character vector
```

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

```
## Excluding mix and unmix specific features
## Only including features observed in all or none of the four pre- post- PCR replicates
## Features with relative abundance estimates less than 1e-7, these are features that we would not expe
pa_filter <- pa_summary_anno_df %>%
      filter(pa_specific == "unspecific") %>%
      select(biosample_id, pipe, feature_id, full_pre, T00, T20, pa_mixed) %>%
      filter(T00 %in% c(0,4), T20 %in% c(04))

# prop_inferred <- prop_inferred %>%
#       right_join(pa_filter) %>%
#       filter(nb_prop > 1e-7)


#### Error Rate Calculations
rel_abu_error <- prop_inferred %>%
      mutate(t_fctr = factor(t_fctr, levels = c(1:5, 10, 15))) %>%
      mutate(inferred_error = abs(mean_count - inferred_prop),
             inferred_error_rate = inferred_error/inferred_prop)
```

```
rel_abu_ridge_df <- rel_abu_error %>%
    mutate(inferred_error_rate = if_else(inferred_error_rate < 1e-10, 0, inferred_error_rate)) %>%
    filter(inferred_error_rate != 0 & mean_count > 1e-10) %>%
    mutate(inferred_error_rate = if_else(inferred_prop == 0, NaN, inferred_error_rate))

rel_abu_med <- rel_abu_ridge_df %>%
      group_by(biosample_id, norm_method) %>%
      summarise(med_error = median( inferred_error_rate,na.rm = TRUE))

rel_abu_ridge_df %>%
      ggplot() +
      geom_density_ridges(aes(x =  inferred_error_rate, y = norm_method, color = norm_method),
                          alpha = 0.5, stat = "binline", bins = 30, draw_baseline = FALSE)  +
      # geom_text(data = rel_abu_med,
      #           aes(x = 100, y = norm_method, label = round(med_error,2)), nudge_y = 0.1) +
      facet_wrap(~biosample_id, nrow = 1) + theme_bw() +
    scale_x_log10() +
      labs(x = "Error Rate", y = "Normalization", color = "Normalization") +
    theme(legend.position = "none")
```

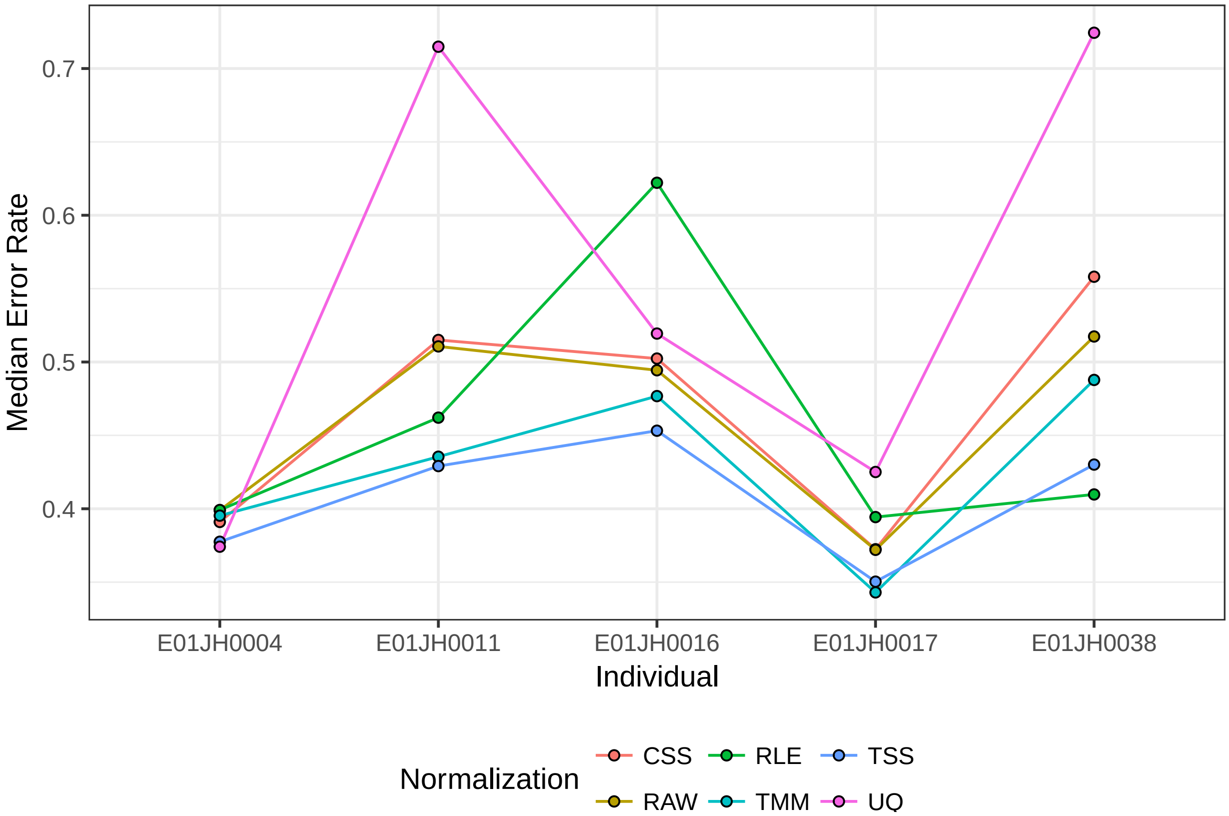
```
## Warning: Removed 189663 rows containing non-finite values (stat_binline).
```



```
rel_abu_med %>% spread(norm_method, med_error) %>% knitr::kable(digits = 3)
```

biosample_id	CSS	RAW	RLE	TMM	TSS	UQ
E01JH0004	0.391	0.399	0.399	0.395	0.378	0.374
E01JH0011	0.515	0.511	0.462	0.435	0.429	0.715
E01JH0016	0.502	0.494	0.622	0.477	0.453	0.519
E01JH0017	0.372	0.372	0.394	0.343	0.350	0.425
E01JH0038	0.558	0.517	0.410	0.488	0.430	0.724

```
rel_abu_med %>% ungroup() %>%
    mutate(biosample_id = factor(biosample_id)) %>%
    ggplot(aes(x = biosample_id, y = med_error)) +
    geom_blank() +
    geom_path(aes(x = as.numeric(biosample_id), y = med_error, color = norm_method)) +
    geom_point(aes(x = biosample_id, y = med_error, fill = norm_method), shape = 21) +
    theme_bw() +
    labs(x = "Individual", y = "Median Error Rate", fill = "Normalization", color = "Normalization") +
    theme(legend.position = "bottom")
```



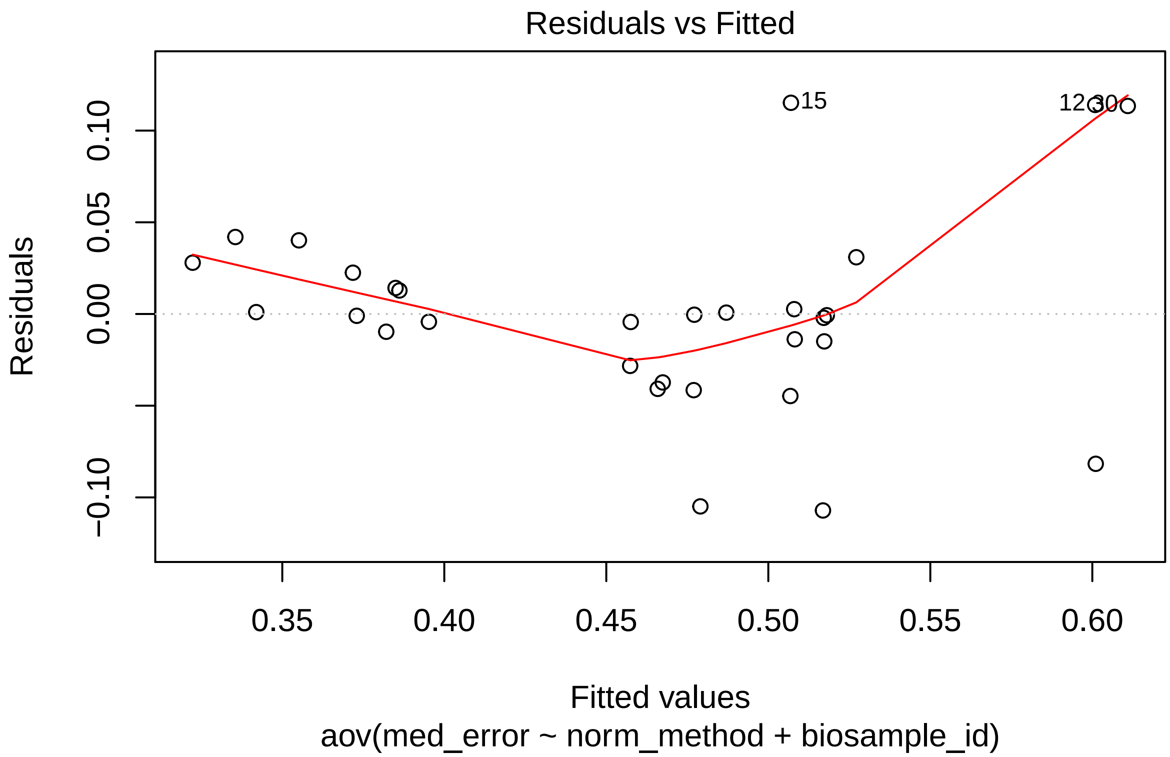
```
fit <- aov(med_error ~ norm_method + biosample_id, data = rel_abu_med)
```

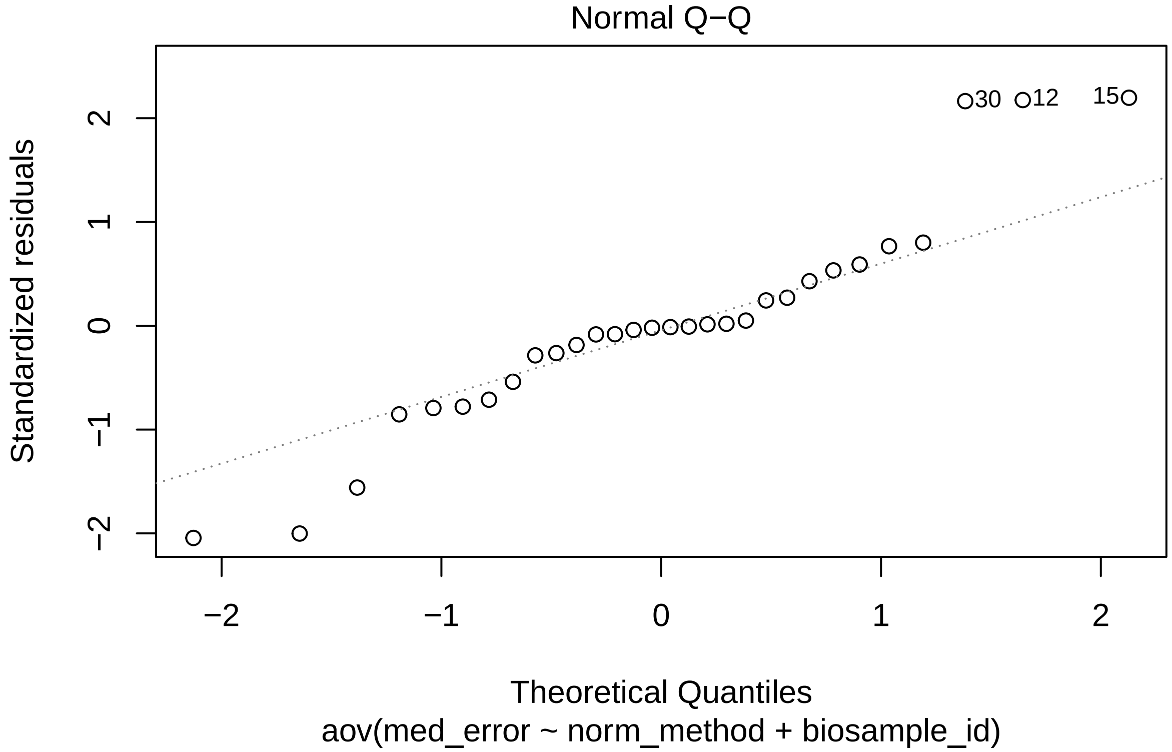
Significant difference between normalization methods when including biosample in the model. Need to used a mixed effects model to account for

```
summary(fit)
```

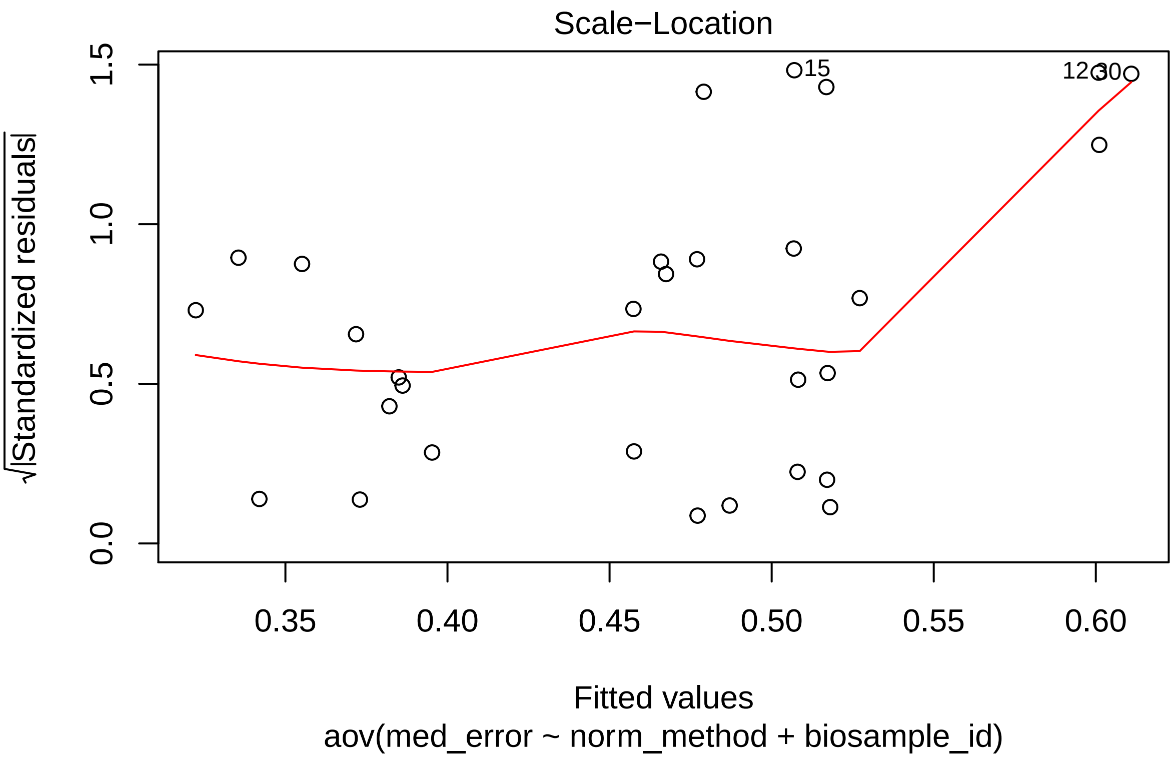
```
##              Df  Sum Sq Mean Sq F value   Pr(>F)    
## norm_method   5 0.06089 0.01218   2.956 0.037128 *  
## biosample_id  4 0.12607 0.03152   7.650 0.000656 ***
## Residuals    20 0.08240 0.00412                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(fit)
```





```
## hat values (leverages) are all = 0.3333333
##  and there are no factor predictors; no plot no. 5
```



Only two pairs normalization methods are significantly different from each other. Upper quartile is significantly different from TMM and TSS.

```
TukeyHSD(fit)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = med_error ~ norm_method + biosample_id, data = rel_abu_med)
##
## $norm_method
##                 diff          lwr        upr     p adj
## RAW-CSS -0.009103820 -0.136707088 0.11849945 0.9999085
## RLE-CSS -0.010287103 -0.137890371 0.11731616 0.9998331
## TMM-CSS -0.040116754 -0.167720023 0.08748651 0.9163956
## TSS-CSS -0.059738967 -0.187342236 0.06786430 0.6850444
## UQ-CSS   0.083786623 -0.043816646 0.21138989 0.3439791
## RLE-RAW -0.001183283 -0.128786551 0.12641999 1.0000000
## TMM-RAW -0.031012934 -0.158616202 0.09659033 0.9705377
## TSS-RAW -0.050635147 -0.178238415 0.07696812 0.8087322
## UQ-RAW   0.092890443 -0.034712826 0.22049371 0.2442483
## TMM-RLE -0.029829651 -0.157432919 0.09777362 0.9750577
## TSS-RLE -0.049451864 -0.177055132 0.07815140 0.8230580
## UQ-RLE   0.094073726 -0.033529542 0.22167699 0.2329747
## TSS-TMM -0.019622213 -0.147225481 0.10798106 0.9962439
## UQ-TMM   0.123903377 -0.003699891 0.25150665 0.0601562
## UQ-TSS   0.143525590  0.015922322 0.27112886 0.0219373
##
## $biosample_id
##                              diff         lwr         upr     p adj
## E01JH0011-E01JH0004  0.1218466191  0.01095283  0.23274041 0.0269479
## E01JH0016-E01JH0004  0.1220207352  0.01112694  0.23291453 0.0266792
## E01JH0017-E01JH0004 -0.0131667180 -0.12406051  0.09772708 0.9962975
## E01JH0038-E01JH0004  0.1319069014  0.02101311  0.24280070 0.0149831
## E01JH0016-E01JH0011  0.0001741161 -0.11071968  0.11106791 1.0000000
## E01JH0017-E01JH0011 -0.1350133371 -0.24590713 -0.02411954 0.0124641
## E01JH0038-E01JH0011  0.0100602823 -0.10083351  0.12095408 0.9987010
## E01JH0017-E01JH0016 -0.1351874532 -0.24608125 -0.02429366 0.0123357
## E01JH0038-E01JH0016  0.0098861662 -0.10100763  0.12077996 0.9987869
## E01JH0038-E01JH0017  0.1450736194  0.03417983  0.25596741 0.0068211
```