EO Metric

Nate Olson

2017-04-14

Objective

Develop a metric for characterizing how well a observed features agree with expectation.

EO Metric

$$\frac{expected - observed}{expected + observed}$$

Values range from -1 to 1 with values of;

- 1 for observed counts of 0 and non-zero expected counts,
- -1 when expected count is 0 and observed counts are non-zero,
- 0 represents agreement between observed and expected counts.

Expected Count Values

Negative Binomal for Weighted Count Estimates

Calculating proportion of pre and post counts using negative binomial.

- $q_{i,j,k}$ is the proportion of feature i in PCR k of sample j where a sample is defined as an individual unmixed or mixed samples for a biological replicate.

- $p_{j,k}$ is the total feature abundance for sample j, sum of all feature counts not the number of sequences generated for the sample.

- $v_{i,j,k}$ is the variance of feature i in PCR replicate j of sample k.

$$v_{i,j,k} = \frac{q_{i,j,k}(1 - q_{i,j,k})}{p_j, k}$$

- $w_{i,j,k}$ is the weight function

$$w_{i,j,k} = \frac{v_{i,j,k}^{-1}}{\sum_{k \in j} v_{i,j,k}^{-1}}$$

- $q_{i,j}$ - the weighted count estimate for feature i, k

$$q_{i,j} = \sum_{k \in j} w_{i,j,k} q_{i,j,k}$$

Loading Data and Calculating Expected Values

```
## Extracting a tidy dataframe with count values from MRexpiment objects
get_count_df <- function(mrobj, agg_genus = FALSE, css = TRUE){
      if(agg_genus){
            mrobj <- aggregateByTaxonomy(mrobj, lvl = "Rank6",
                                         norm = FALSE, log = FALSE, sl = 1)
      }

      if(css == TRUE){
            mrobj <- cumNorm(mrobj, p = 0.75)
            count_mat <- MRcounts(mrobj, norm = TRUE, log = FALSE, sl = 1000)
      }else{
            count_mat <- MRcounts(mrobj, norm = FALSE, log = FALSE, sl = 1)
      }
       count_mat %>%
            as.data.frame() %>%
            rownames_to_column(var = "feature_id") %>%
            gather("id","count", -feature_id)
}

count_df <- mrexp %>% map_df(get_count_df, css = FALSE, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC", id != "1-F9") %>%
      select(pipe, biosample_id, id, pcr_rep, feature_id, t_fctr, count)

count_df <- count_df %>% group_by(id) %>% mutate(total_abu = sum(count))
```

Subsetting `count_df`

```
count_df <- count_df %>% filter(feature_id %in% c(paste0("SV", 1:3), paste0("Otu0000",1:3)))
```

Estimating q_i for pre and post

```
nb_est <- count_df %>% filter(t_fctr %in% c(0, 20)) %>%
      mutate(prop = count/total_abu,
             prop_var = (prop * (1 - prop))/total_abu,
             inv_var = 1/prop_var) %>%
      group_by(pipe, biosample_id, t_fctr, feature_id) %>%
      mutate(weight = inv_var / sum(inv_var)) %>%
      summarise(prop_est = sum(weight*prop))

# Reformatting data
pre_post_prop <- nb_est %>% ungroup() %>%
      mutate(treat = if_else(t_fctr == "20", "pre","post")) %>%
      select(-t_fctr) %>%
      mutate(prop_est = if_else(is.nan(prop_est), 0, prop_est)) %>%
      spread(treat, prop_est)
```

Calculating expected counts using proportion estimates

```
calc_expected_prop <- function(pre_post_prop){
      titration_list <- data_frame(titration = c(1:5,10,15)) %>%
            mutate(post_prop = 2^-titration) %>%
            list() %>% rep(nrow(pre_post_prop))

      pre_post_prop %>% ungroup() %>%
            add_column(titration = titration_list) %>% unnest() %>%
```

```
            mutate(exp_prop = post * post_prop + pre * (1-post_prop)) %>%
            mutate(t_fctr = factor(titration)) %>%
            select(-post_prop)
}

exp_prop_df <- calc_expected_prop(pre_post_prop)
```

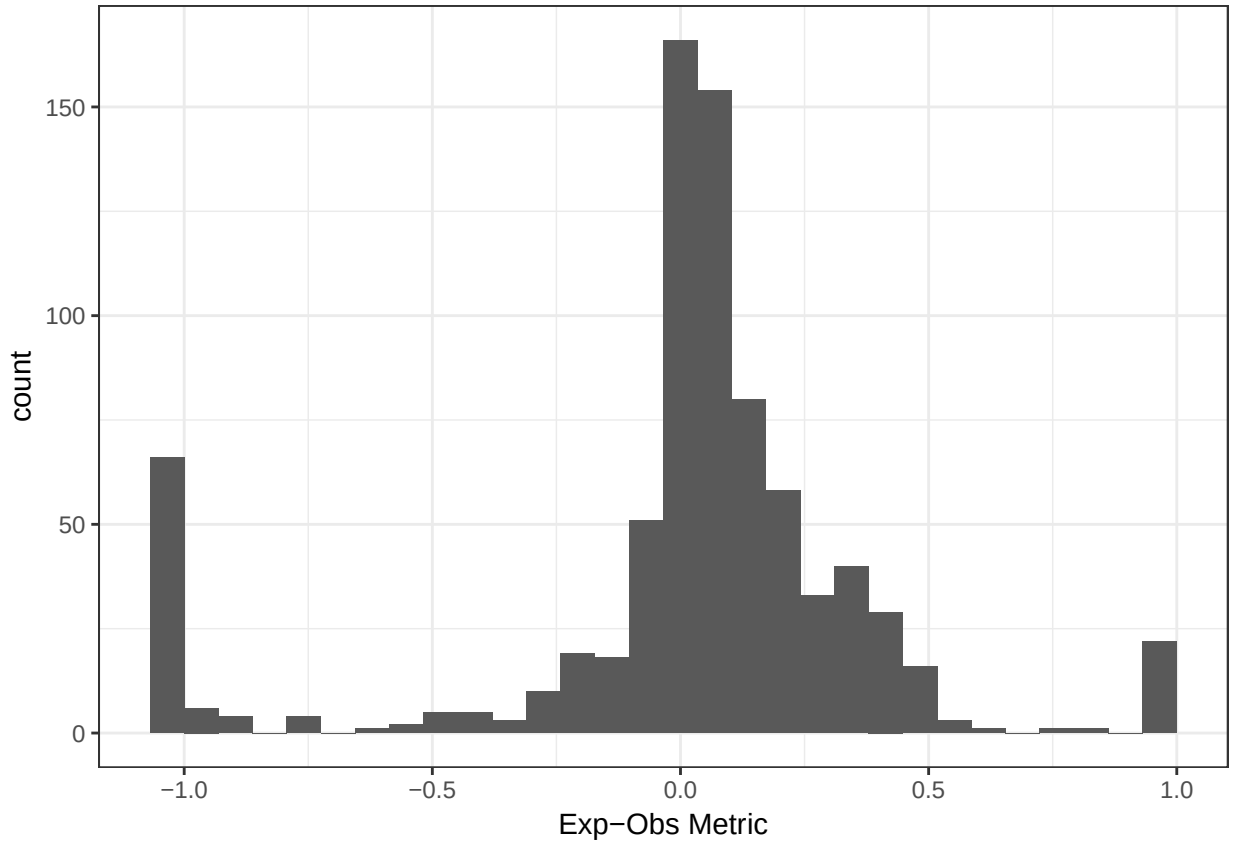
```
exp_count_df <- count_df %>%
      filter(t_fctr %in% c(1:5, 10, 15)) %>%
      left_join(exp_prop_df) %>%
      mutate(exp_count = total_abu * exp_prop) %>%
      filter(!(pre == 0 & post == 0 & count == 0))
```

EO Metric

```
eo_metric_df <- exp_count_df %>%
      mutate(eo_metric = (count - exp_count)/(count + exp_count))
```

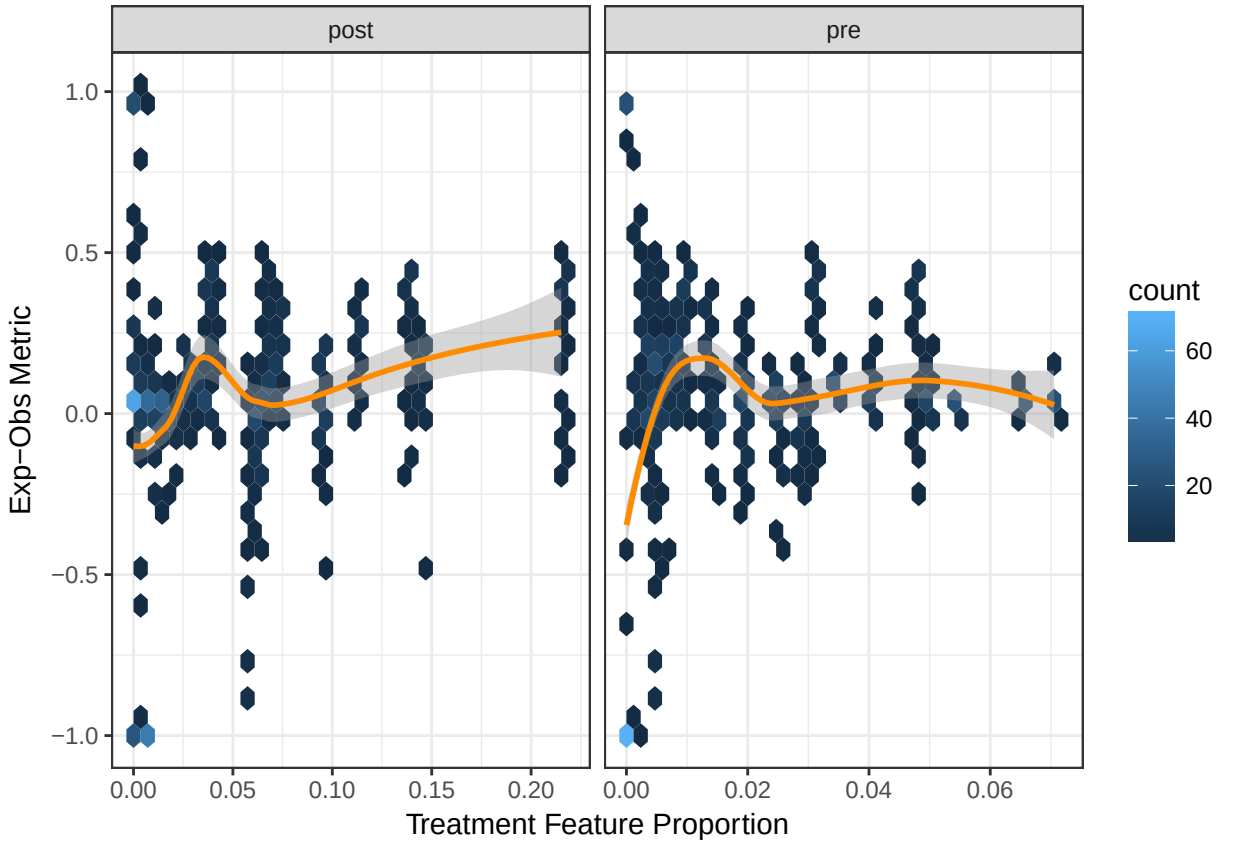
Overall the metric distribution skews to the right indicating that the expected values tend to be greater than the observed values with a medain around 0. There are more PCR replicates with observed counts of but 0 expected counts than expected counts with 0 observed counts.

```
eo_metric_df %>% ggplot() +
      geom_histogram(aes(x = eo_metric)) + theme_bw() +
      labs(x = "Exp-Obs Metric")
```



The EO metric tends to center around 0 with increasing Pre-Treament feature proportion but increases with Post-treatment feature proportion. Low pre and post treatment feature proportion have higher EO metrics indicating expected counts are overestimates for lower proportions. This is potentially due to a limit of detection, can look at the EO metric for feature-pcr reps with expected count values < 1 and $!= 0$ (see table following plot).

```
eo_metric_df %>%
      select(pipe, biosample_id, id, feature_id, t_fctr, pre, post, eo_metric) %>%
      gather("unmix","q",-eo_metric, -pipe, -biosample_id, -id, -feature_id,-t_fctr) %>%
      ggplot() +
      geom_hex(aes(x = q, y = eo_metric)) +
      geom_smooth(aes(x = q, y = eo_metric), color = "darkorange") +
      facet_wrap(~unmix, scale = "free_x") +
      theme_bw() +
      labs(x = "Treatment Feature Proportion", y = "Exp-Obs Metric")
```



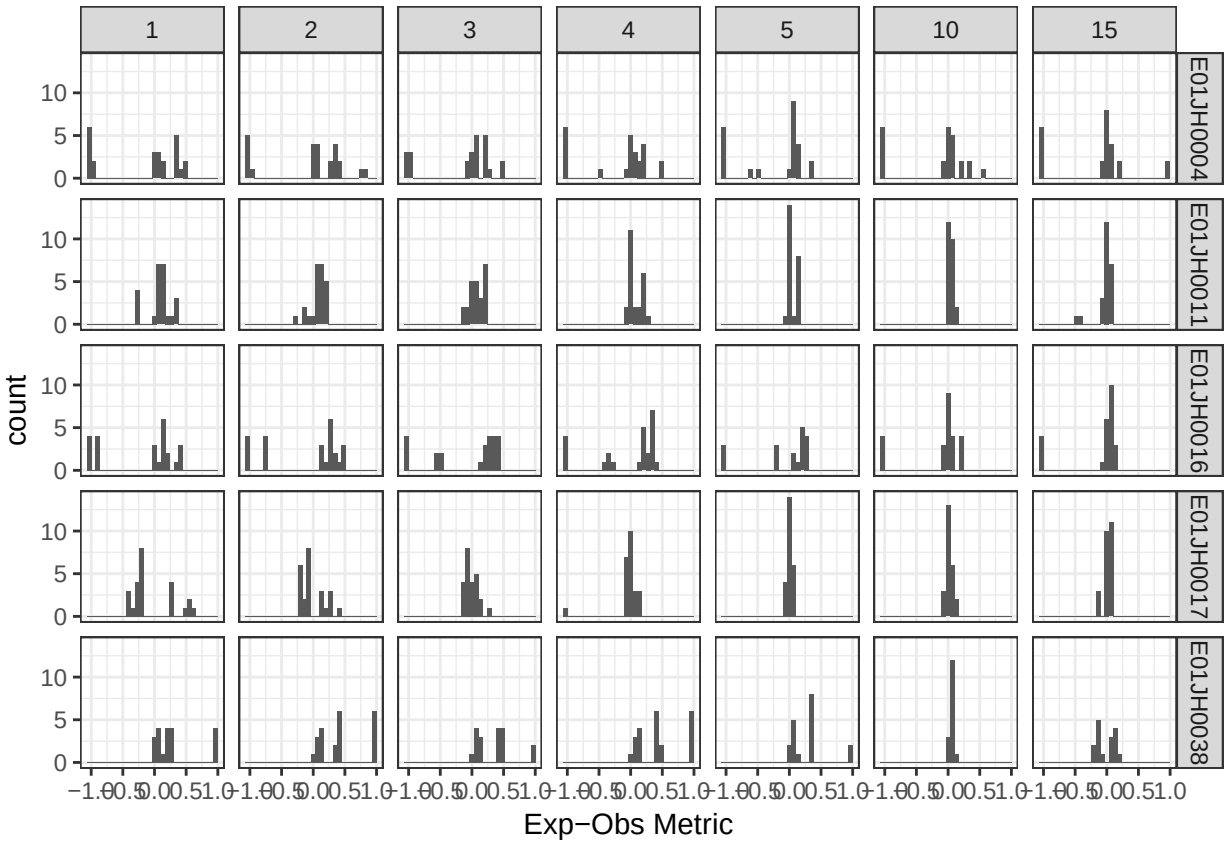

```
eo_metric_df %>%
      filter(exp_count < 1, exp_count != 0) %>%
      select(total_abu, count, exp_count, eo_metric) %>% arrange(eo_metric) %>%
      knitr::kable()
```

id	total_abu	count	exp_count	eo_metric
1-G1	231978	0	0.6264601	-1.0000000
1-G3	155717	0	0.7326357	-1.0000000
1-G7	151518	0	0.4091767	-1.0000000
1-G9	139316	0	0.6554704	-1.0000000
1-H1	228423	0	0.0192769	-1.0000000

id	total_abu	count	exp_count	eo_metric
1-H3	185713	0	0.0273051	-1.0000000
1-H7	170291	0	0.0143710	-1.0000000
1-H9	133843	0	0.0196788	-1.0000000
2-G1	235573	0	0.6361685	-1.0000000
2-G3	161371	0	0.7592373	-1.0000000
2-G7	198512	0	0.5360847	-1.0000000
2-G9	152135	0	0.7157827	-1.0000000
2-H1	137035	0	0.0115645	-1.0000000
2-H3	120043	0	0.0176498	-1.0000000
2-H7	187604	0	0.0158321	-1.0000000
2-H9	120492	0	0.0177158	-1.0000000
1-G7	151518	0	0.6624329	-1.0000000
1-H7	170291	0	0.0232659	-1.0000000
2-H1	137035	0	0.0187223	-1.0000000
2-G7	198512	3	0.8678894	0.5512336
2-H7	187604	3	0.0256312	0.9830573
1-H1	228423	8	0.0312081	0.9922283

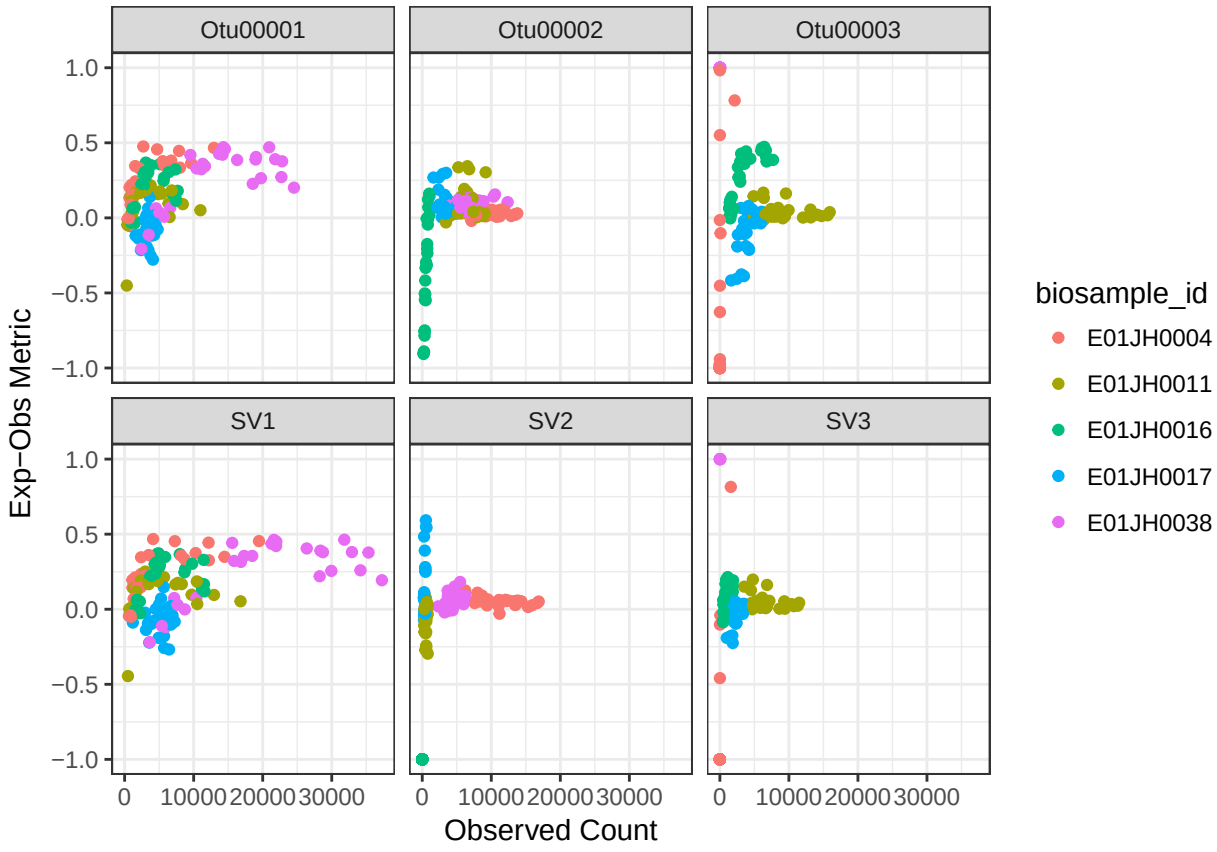
For the example subset of the metric skew and -1 and 1 peaks differ by pipeline, biosample, and titration. Will want to look at larger set of features before drawing any conclusions.

```
eo_metric_df %>%
      mutate(t_fctr = factor(t_fctr, levels = c(0:5, 10, 15, 20))) %>%
      ggplot() +
      geom_histogram(aes(x = eo_metric)) +
      facet_grid(biosample_id~t_fctr) + theme_bw() +
      labs(x = "Exp-Obs Metric")
```



The EO-metric distribution also tend to vary by feature, again a larger set of features is needed before drawing any conclusions.

```
eo_metric_df %>% ggplot() +
      geom_point(aes(x = count, y = eo_metric, color = biosample_id)) +
      facet_wrap(~feature_id) +
      theme_bw() +
      labs(x = "Observed Count", y = "Exp-Obs Metric")
```



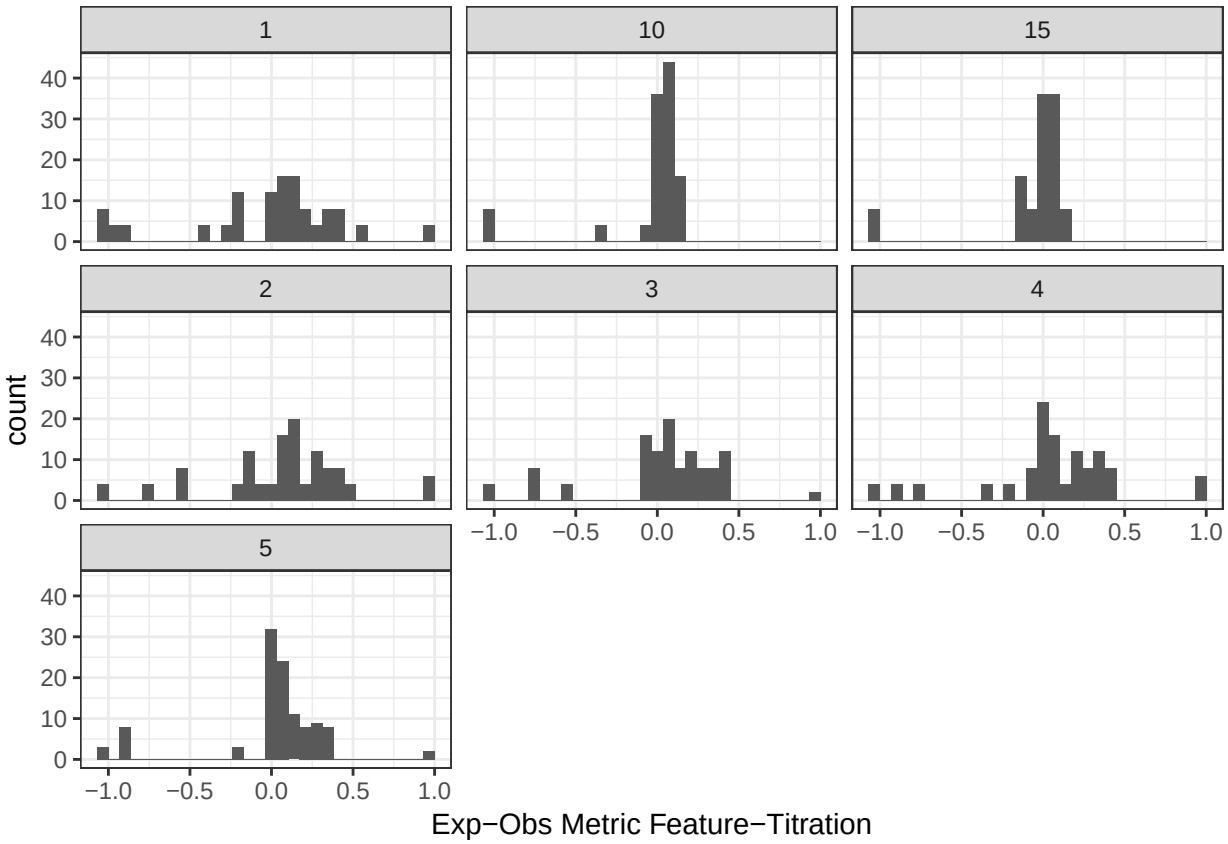
Summarizing by Titration and Feature

To evaluate performance for a set of PCR replicates and across a feature can use the mean or median as summary metrics.

```
eo_summary <- eo_metric_df %>%
      group_by(pipe, biosample_id, t_fctr, feature_id) %>%
      mutate(eot_mean = mean(eo_metric),
             eot_median = median(eo_metric)) %>%
      group_by(pipe, biosample_id, feature_id) %>%
      mutate(eof_mean = mean(eo_metric),
             eof_median = median(eo_metric))
```

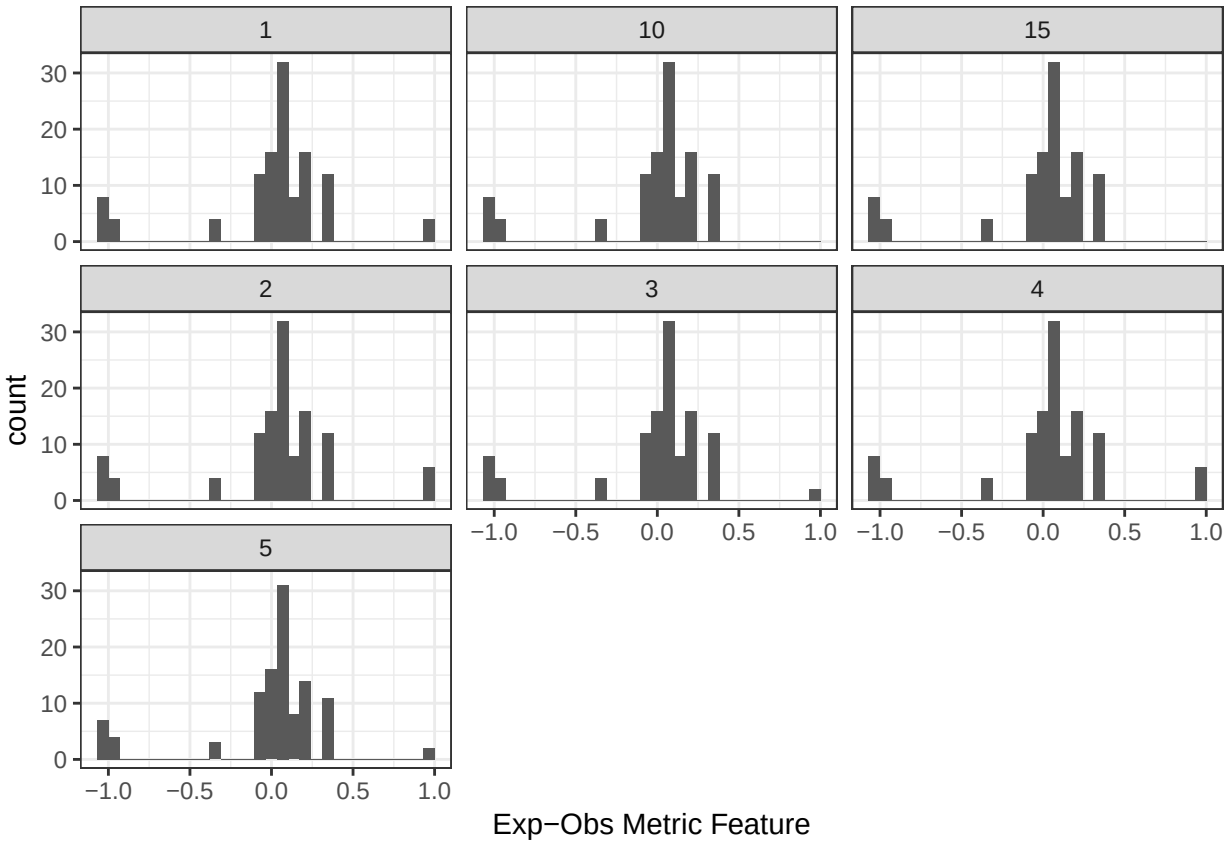
Mean EO metric for a feature titration PCR replicates.

```
eo_summary %>%
      ggplot() + geom_histogram(aes(x = eot_mean)) +
      facet_wrap(~t_fctr) + theme_bw() +
      labs(x = "Exp-Obs Metric Feature-Titration")
```



Mean EO metric across all PCR replicates and titrations for a feature.

```
eo_summary %>%
      ggplot() + geom_histogram(aes(x = eof_median)) +
      facet_wrap(~t_fctr) + theme_bw() +
      labs(x = "Exp-Obs Metric Feature")
```



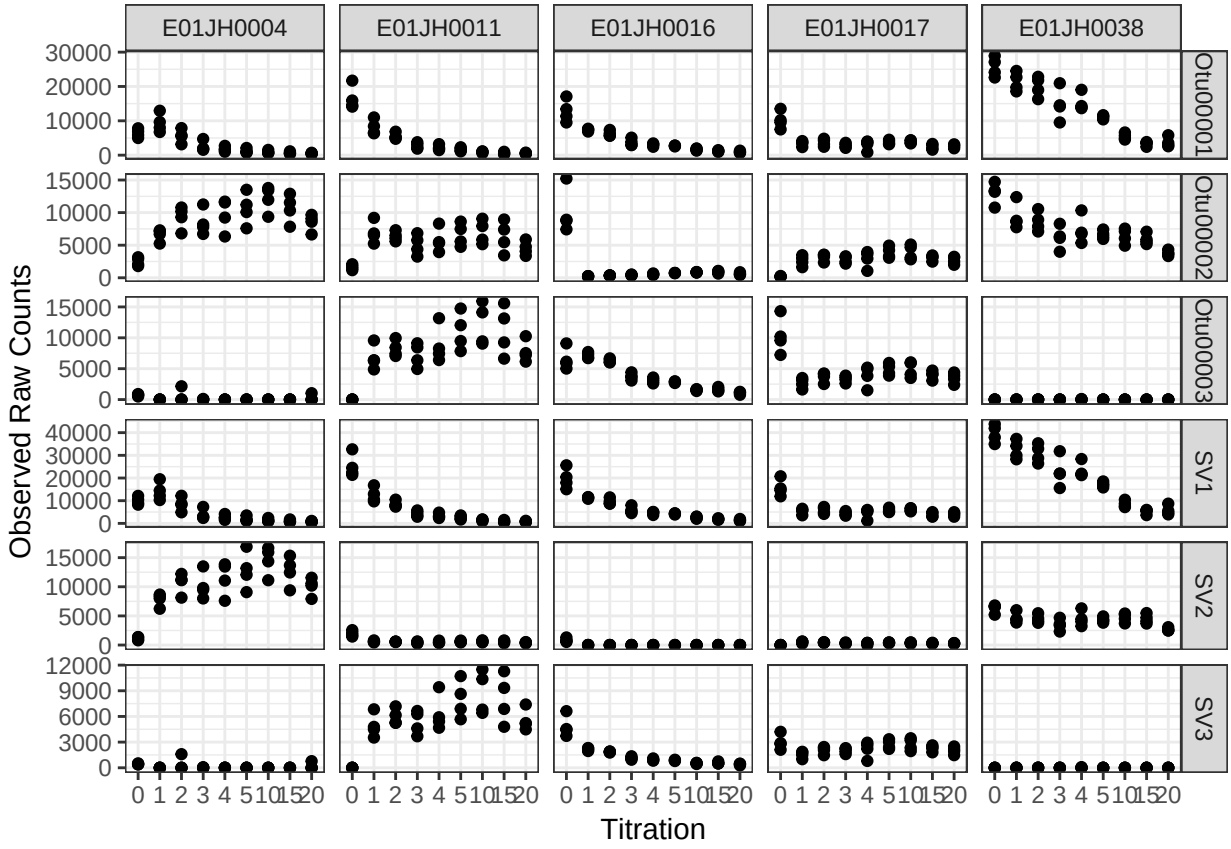
Metric and Scatter Plots

Table of feature level OE metrics and corresponding scatter plots of observed counts by titration. OE-metric is NA when all counts for all samples and replicates is 0.

```
eo_summary %>% ungroup() %>%
      select(biosample_id, feature_id, eof_median) %>% unique() %>%
      spread(biosample_id, eof_median) %>%
      knitr::kable()
```

feature_id	E01JH0004	E01JH0011	E01JH0016	E01JH0017	E01JH0038
Otu00001	0.2292027	0.1600426	0.2255008	-0.0525340	0.3363921
Otu00002	0.0302735	0.0506682	-0.3331750	0.0738099	0.1032761
Otu00003	-0.9957389	0.0242968	0.3744502	-0.0301966	1.0000000
SV1	0.2235329	0.1624606	0.2232204	-0.0532788	0.3385021
SV2	0.0536454	-0.0420589	-1.0000000	0.0728065	0.0611647
SV3	-1.0000000	0.0354836	0.1016067	-0.0046678	1.0000000

```
count_df %>%
      mutate(t_fctr = factor(t_fctr, levels = c(0:5, 10, 15, 20))) %>%
      ggplot() +
      geom_point(aes(x = t_fctr, y = count)) +
      facet_grid(feature_id~biosample_id, scales = "free") +
      theme_bw() +
      labs(x = "Titration", y = "Observed Raw Counts")
```



Session information

Git repo commit information

```
library(git2r)
repo <- repository(path = "../")
last_commit <- commits(repo)[[1]]
```

The current git commit of this file is 7477b2bd5a692b10cb61b4b43db0d0476729fb60, which is on the master branch and was made by nate-d-olson on 2017-04-13 23:15:14. The current commit message is new metric relating error and observed counts. The repository is online at https://github.com/nate-d-olson/mgtst-pub

Platform Information

```
s_info <- devtools::session_info()
print(s_info$platform)
```

```
##  setting  value
##  version  R version 3.3.3 (2017-03-06)
##  system   x86_64, darwin15.6.0
##  ui       unknown
##  language (EN)
```

```
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2017-04-14
```

Package Versions

```
s_info$packages %>% filter(`*` == "*") %>% select(-`*`) %>%
      knitr::kable()
```

package	version	date	source
bbmle	1.0.18	2016-02-11	CRAN (R 3.3.2)
Biobase	2.34.0	2016-11-07	Bioconductor
BiocGenerics	0.20.0	2016-11-07	Bioconductor
BiocParallel	1.8.2	2017-04-12	Bioconductor
Biostrings	2.42.1	2016-12-19	Bioconductor
DESeq	1.26.0	2016-11-28	Bioconductor
DESeq2	1.15.28	2017-02-02	bioc (readonly/DESeq2@125913)
dplyr	0.5.0	2016-06-24	CRAN (R 3.3.2)
edgeR	3.16.5	2017-02-02	Bioconductor
forcats	0.2.0	2017-01-23	CRAN (R 3.3.2)
foreach	1.4.3	2015-10-13	CRAN (R 3.3.1)
GenomeInfoDb	1.10.3	2017-03-28	Bioconductor
GenomicAlignments	1.10.1	2017-03-28	Bioconductor
GenomicRanges	1.26.4	2017-03-28	Bioconductor
ggplot2	2.2.1	2016-12-30	CRAN (R 3.3.2)
git2r	0.18.0	2017-01-01	CRAN (R 3.3.2)
glmnet	2.0-5	2016-03-17	CRAN (R 3.3.1)
hexbin	1.27.1	2015-08-19	CRAN (R 3.3.1)
IRanges	2.8.2	2017-03-28	Bioconductor
knitr	1.15.1	2016-11-22	CRAN (R 3.3.2)
lattice	0.20-35	2017-03-25	CRAN (R 3.3.3)
limma	3.30.13	2017-03-28	Bioconductor
locfit	1.5-9.1	2013-04-20	CRAN (R 3.3.1)
Matrix	1.2-8	2017-01-20	CRAN (R 3.3.3)
metagenomeSeq	1.16.0	2016-11-07	Bioconductor
modelr	0.1.0	2016-08-31	cran (@0.1.0)
permute	0.9-4	2016-09-09	CRAN (R 3.3.1)
phyloseq	1.19.1	2017-01-04	Bioconductor
ProjectTemplate	0.7	2016-08-11	CRAN (R 3.3.1)
purrr	0.2.2	2016-06-18	CRAN (R 3.3.1)
RColorBrewer	1.1-2	2014-12-07	CRAN (R 3.3.1)
readr	1.1.0	2017-03-22	CRAN (R 3.3.2)
readxl	0.1.1	2016-03-28	cran (@0.1.1)
Rqc	1.8.0	2016-11-07	Bioconductor
Rsamtools	1.26.2	2017-04-12	Bioconductor
S4Vectors	0.12.2	2017-03-28	Bioconductor
sads	0.3.1	2016-05-13	CRAN (R 3.3.2)
savR	1.12.0	2016-11-07	Bioconductor
ShortRead	1.32.1	2017-03-28	Bioconductor
stringr	1.2.0	2017-02-18	CRAN (R 3.3.2)
SummarizedExperiment	1.4.0	2016-11-07	Bioconductor
tibble	1.3.0	2017-04-01	CRAN (R 3.3.3)

package	version	date	source
tidyr	0.6.1	2017-01-10	CRAN (R 3.3.2)
tidyverse	1.1.1	2017-01-27	CRAN (R 3.3.2)
vegan	2.4-3	2017-04-07	CRAN (R 3.3.3)
XVector	0.14.1	2017-03-28	Bioconductor