Identification of Informative and Uninformative Features

Nate Olson

2017-04-11

Objective

Identification of informative and uninformative features. Informative features defined as features with observed counts in all pre-treatment samples and all titration samples and all or none of the post-treatment replicates.

Feature Categories

- Null - features not present in more than one PCR replicate for any sample of a biological replicate, and pipeline.

Informative

- Full - features present all PCR replicates for all samples (unmixed and titrations) for a biological replicate.
- Pre - present in all PCR replicates for unmixed pre-treatment samples, not observed in any PCR replicates of the unmixed post treatment samples, and present in all titration PCR replicates.
- Post - present in all PCR replicates for the unmixed post-treatment samples, not observed in any PCR replicates of the unmixed pre-treatment samples, and present all titration PCR replicates.

Characterization of low abundance features * Mix - features present in mixed samples but not observed in any of the unmixed sample PCR replicates.

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

      mrobj <- cumNorm(mrobj, p = 0.75)
      mrobj %>%
            # not sure whether or not to normalize counts prior to analysis
            MRcounts(norm = TRUE, log = FALSE, sl = 1000) %>%
            as.data.frame() %>%
            rownames_to_column(var = "feature_id") %>%
            gather("id","count", -feature_id)
}


get_rep_info <- function(count_df){
      count_replicate_df <- count_df %>%
            mutate(detect = if_else(count > 1, 1, 0)) %>%
            group_by(pipe, biosample_id, titration, t_fctr, feature_id) %>%
            summarise(total_detect = sum(detect),
                      n_replicates = n(),
                      avg_non0_count = sum(count)/total_detect) %>%
            mutate(detect_prop = total_detect/n_replicates) %>%
            select(-total_detect)

      count_replicate_df %>% ungroup() %>%
      mutate(t_fctr = paste0("T",t_fctr)) %>%
      select(pipe, biosample_id, feature_id, t_fctr, detect_prop)
}


assign_cat <- function(rep_info){
      prop_summary <- rep_info %>%
            group_by(pipe, biosample_id, feature_id) %>%
            summarise(prop_max = max(detect_prop),
                      prop_min = min(detect_prop),
                      prop_sum = sum(detect_prop))

      unmix_prop <- rep_info %>%
            filter(t_fctr %in% c("T0","T20")) %>%
            spread(t_fctr, detect_prop)

      left_join(prop_summary, unmix_prop) %>%
            mutate(cat_null = if_else(prop_max == 0, 1, 0),
                   cat_full = if_else(prop_min == 1, 1, 0),
                   cat_near_full = if_else(prop_min == 0.75, 1, 0),
                   cat_mix  = if_else(prop_max == 1 & T0 == 0 & T20 == 0, 1, 0),
                   ## Post prop 5 - expected at least three replicates for titrations 4, 5, 10, and 15
                   ## Pre prop 3 - expected at least three replicates for titrations 1, 2, 3, and 4
                   ## titration 4, is ~94% post
                   ## titration 4, is ~94% post
                   cat_pre  = if_else(T20 == 1 & T0 == 0 & prop_sum == 8, 1, 0),
                   cat_post = if_else(T0 == 1 & T20 == 0 & prop_sum == 8, 1, 0),
                   cat_none = if_else(cat_null + cat_full + cat_near_full + cat_mix + cat_pre + cat_pos
}
```

Feature Level Category Assignments

```
count_df <- mrexp %>% map_df(get_count_df, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) #%>%
      # filter(biosample_id != "NTC")

#count_df
rep_info <- get_rep_info(count_df)

#rep_info
feature_info <- assign_cat(rep_info)

feature_cat <- feature_info %>%
      select(-prop_max, -prop_min, -prop_sum, -T0, -T20) %>%
      gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
      filter(value == 1) %>% select(-value)

feature_cat %>% saveRDS("../data/feature_categories_df.rds")
```

Category Sanity Check

```
cat_check <- feature_cat %>%
      group_by(pipe, biosample_id, feature_id) %>%
      summarise(n_cat = n())
cat_check %>% filter(n_cat != 1)
```

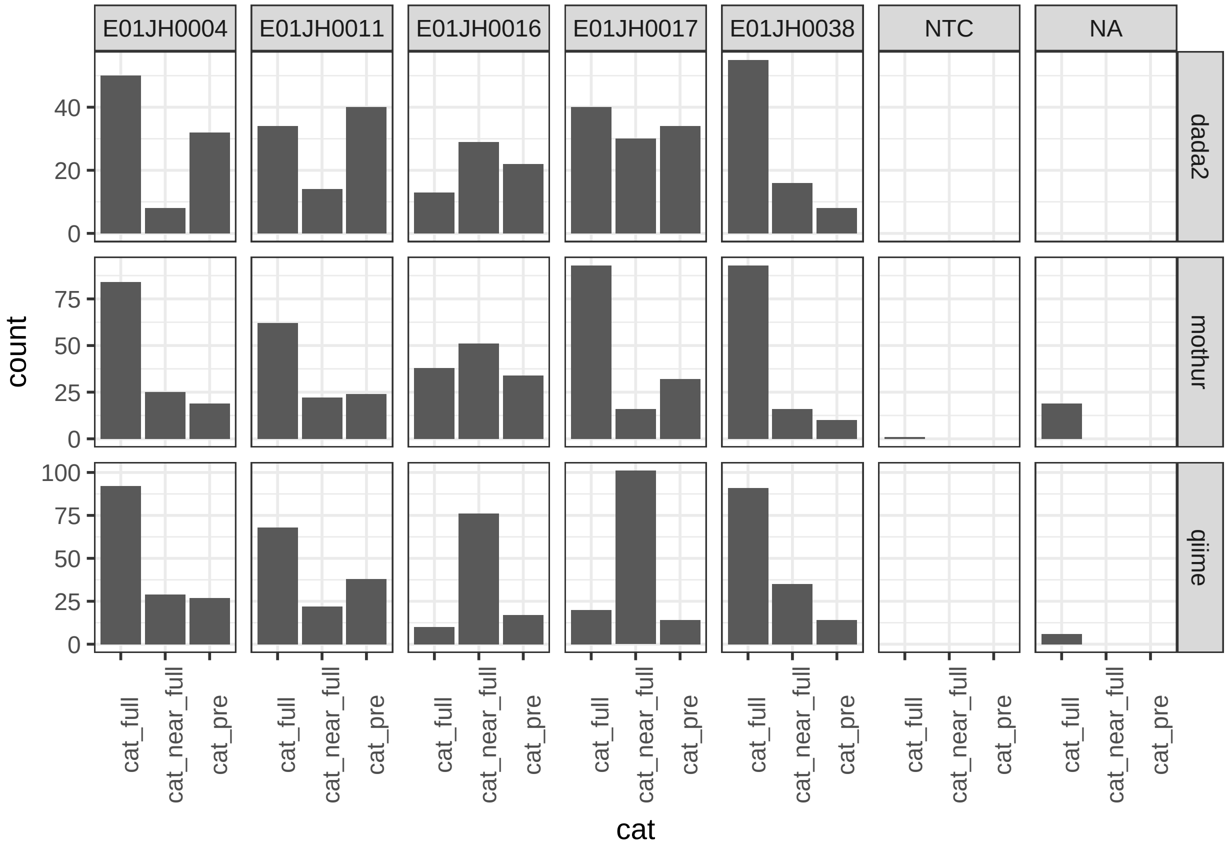
```
## Source: local data frame [0 x 4]
## Groups: pipe, biosample_id [0]
##
## # ... with 4 variables: pipe <chr>, biosample_id <chr>, feature_id <chr>,
## #   n_cat <int>
```

```
# cat_check <- feature_info %>%
#       select(-prop_max, -prop_min, -prop_sum, -T0, -T20) %>%
#       gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
#       group_by(pipe, biosample_id, feature_id) %>%
#       mutate(n_cat = sum(value)) %>%
#       filter(n_cat != 1, value != 0)
# cat_check %>% arrange(feature_id)
```

```
      prop_summary <- rep_info %>%
            group_by(pipe, biosample_id, feature_id) %>%
            summarise(prop_max = max(detect_prop),
                      prop_min = min(detect_prop),
                      prop_sum = sum(detect_prop))
```

Informative Features

```
feature_cat %>% filter(cat != "cat_null", cat != "cat_none", cat != "cat_mix") %>%
      ggplot() + geom_bar(aes(x = cat)) +
      facet_grid(pipe ~ biosample_id, scales = "free_y") +
      theme_bw() + theme(axis.text.x = element_text(angle = 90))
```



Recovering Semi-Informative Features

```
cat_none_df <- feature_cat %>% left_join(rep_info) %>% filter(cat == "cat_none")
total_prop_cat_none <- cat_none_df %>% group_by(pipe, biosample_id, feature_id) %>%
    summarise(total_prop = sum(detect_prop))
```

Most of the uncategorized features were observed in less than 4 PCR replicates

```
total_prop_cat_none %>% mutate(total_prop = floor(total_prop)) %>%
    group_by(pipe, total_prop) %>% summarise(count = n()) %>%
    spread(pipe, count) %>% knitr::kable()
```

total_prop	dada2	mothur	qiime
0	926	29239	14458
1	405	1090	2473
2	93	337	877
3	75	201	447
4	82	148	298
5	87	140	270
6	98	156	189
7	186	168	310
8	64	103	113

Assuming 4 PCR replicates for all samples. Using proportions if samples are excluded from analysis, for example samples with few reads compared to the rest of the samples.

```
cat_none_df %>% filter(!(detect_prop %in% c(0,0.25,0.5,0.75,1)))
```

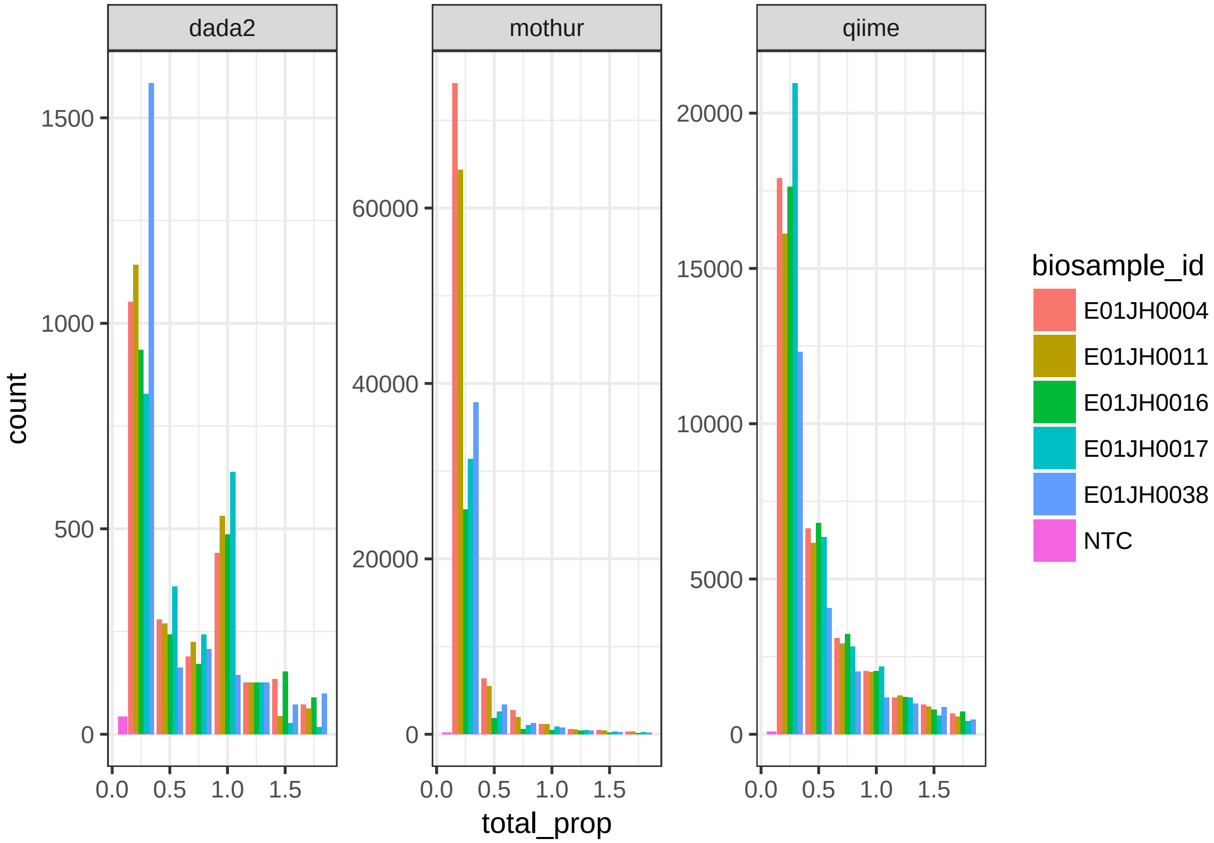
```
## Source: local data frame [410 x 6]
## Groups: pipe, biosample_id [3]
##
##     pipe biosample_id feature_id      cat t_fctr detect_prop
##    <chr>        <chr>      <chr>    <chr>  <chr>       <dbl>
## 1  dada2          NTC        SV1 cat_none    TNA  0.45454545
## 2  dada2          NTC     SV1024 cat_none    TNA  0.09090909
## 3  dada2          NTC      SV106 cat_none    TNA  0.09090909
## 4  dada2          NTC       SV11 cat_none    TNA  0.09090909
## 5  dada2          NTC      SV113 cat_none    TNA  0.09090909
## 6  dada2          NTC      SV118 cat_none    TNA  0.09090909
## 7  dada2          NTC     SV1189 cat_none    TNA  0.09090909
## 8  dada2          NTC       SV12 cat_none    TNA  0.18181818
## 9  dada2          NTC     SV1371 cat_none    TNA  0.09090909
## 10 dada2          NTC      SV141 cat_none    TNA  0.09090909
## # ... with 400 more rows
```

What is the detect_proportions for samples with total prop < 2.

```
cat_none_low <- cat_none_df %>% group_by(pipe, biosample_id, feature_id) %>%
    mutate(total_prop = sum(detect_prop)) %>% filter(total_prop > 0, total_prop < 2)
```

Most of the low total detect proportion features were only observed in one PCR replicate.

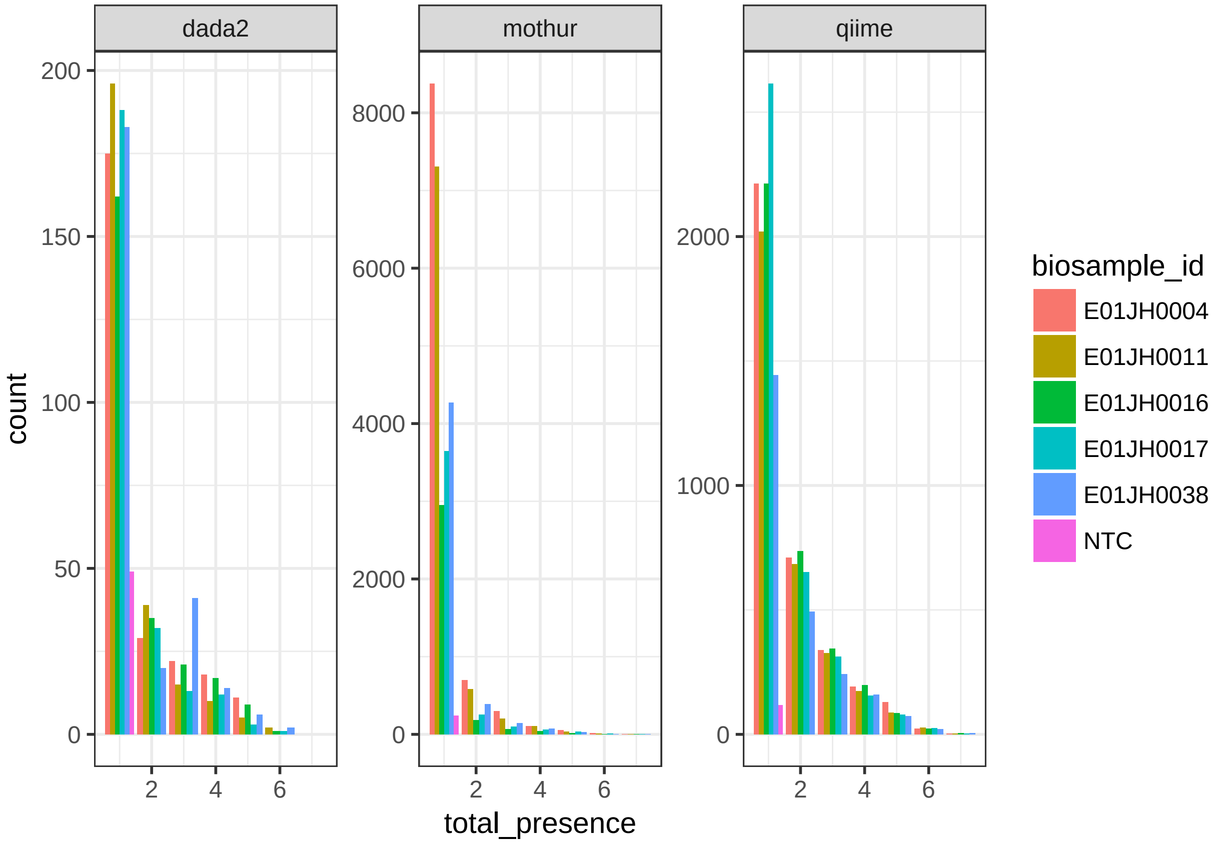
```
cat_none_low %>%
    ggplot() + geom_bar(aes(x = total_prop, fill = biosample_id),
                        position = "dodge") +
    facet_wrap(~pipe, scales = "free_y") + theme_bw()
```



```
# cat_none_low %>% filter(total_prop > 0.5) %>% spread(t_fctr, detect_prop) %>% arrange(total_prop)
```

Most of the low detect features are only present in 1 sample

```
cat_none_low %>% mutate(pa = if_else(detect_prop == 0, 0, 1)) %>% summarise(total_presence = sum(pa)) %>
    ggplot() + geom_bar(aes(x = total_presence, fill = biosample_id),
                        position = "dodge") +
    facet_wrap(~pipe, scales = "free_y") + theme_bw()
```

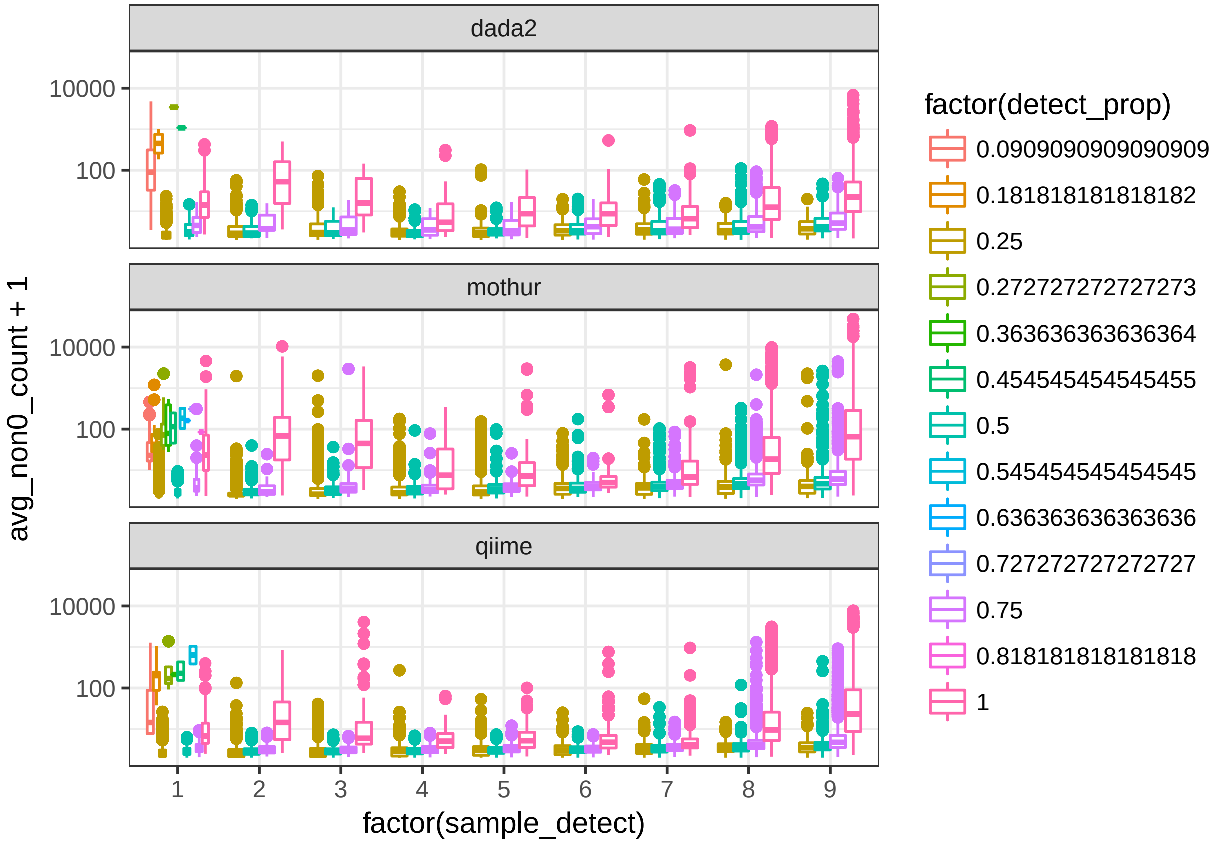


Open Question Are low detect features due to random sampling or bioinformatic/ experimental artifact. * Approach - average counts by number of PCR replicates and number of samples with observed counts.

```
count_replicate_df <- count_df %>%
            mutate(detect = if_else(count > 1, 1, 0)) %>%
            group_by(pipe, biosample_id, titration, t_fctr, feature_id) %>%
            mutate(total_detect = sum(detect),
                      n_replicates = n(),
                      avg_non0_count = sum(count)/total_detect) %>%
            mutate(detect_prop = total_detect/n_replicates) %>%
            select(-total_detect)
count_rep_pa <- count_replicate_df %>%
    group_by(pipe, biosample_id, feature_id, t_fctr) %>%
    summarise(pa = if_else(sum(detect) != 0, 1, 0)) %>%
    group_by(pipe, biosample_id, feature_id) %>%
    mutate(sample_detect = sum(pa))
count_replicate_pa_df <- count_rep_pa %>% filter(sample_detect != 0) %>% left_join(count_replicate_df)
```

X-axis: Number of samples out of the 2 unmixed and 7 titrations with at least one of the four PCR replicates with observed counts Y-axis: Mean counts for PCR replicates with non-zero count values Color: Of the sample (either unmixed or titration) proportion of PCR replicates with observed counts

```
count_replicate_pa_df %>% ggplot() +
    geom_boxplot(aes(y = avg_non0_count + 1, x = factor(sample_detect), color = factor(detect_prop))) +
    scale_y_log10() + facet_wrap(~pipe, ncol = 1) + theme_bw()
```



Extracting additional informative features

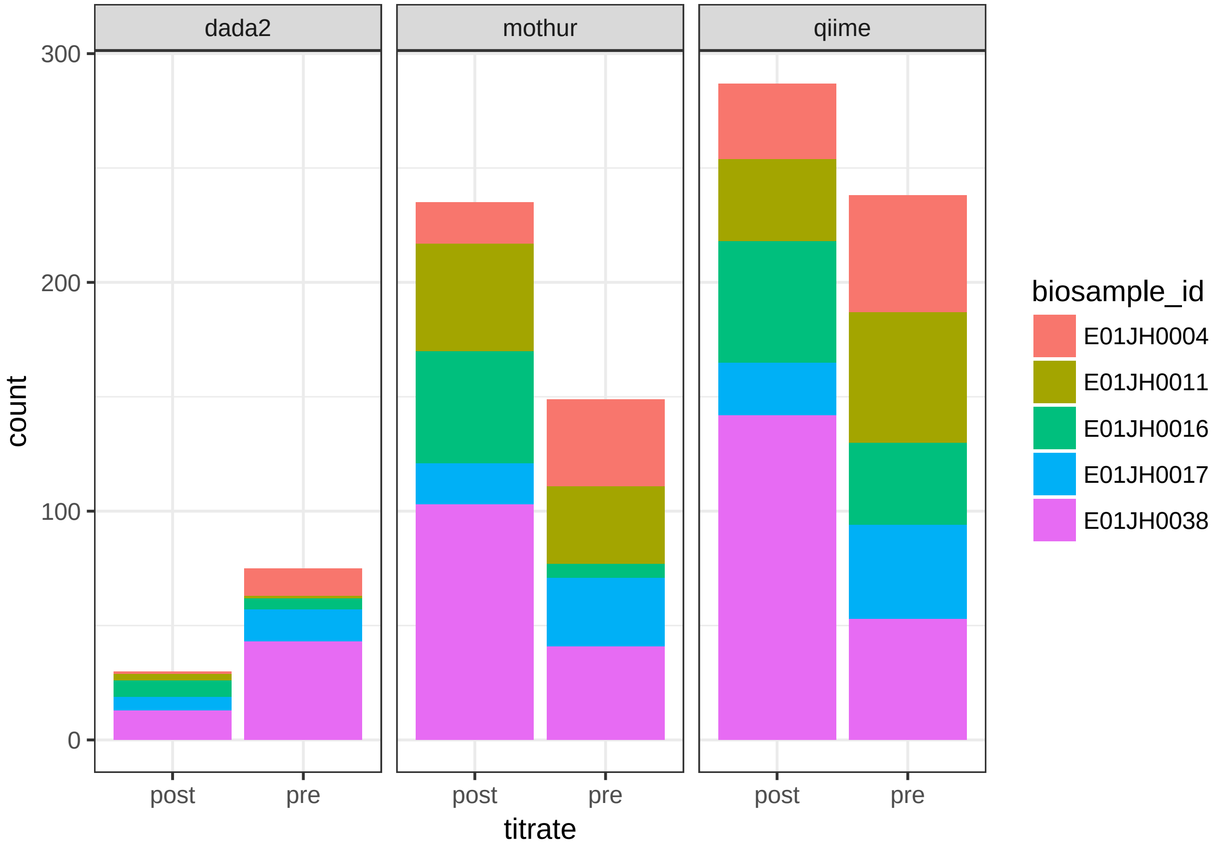
```
pre_post_titrate <- cat_none_df %>% spread(t_fctr, detect_prop) %>%
    mutate(pre_titration = if_else(T20 != 0 & T15 != 0 &
                                       T20 >= T15 & T15 >= T10 & T10 >= T5 &
                                       T5 >= T4 & T4 >= T3 & T3 >= T2  &
                                       T2 >= T1 & T1 >= T0, 1, 0),
           post_titration = if_else(T0 != 0 & T1 != 0 &
                                        T20 <= T15 & T15 <= T10 & T10 <= T5 &
                                       T5 <= T4 & T4 <= T3 & T3 <= T2  &
                                       T2 <= T1 & T1 <= T0, 1, 0)) %>%
    filter(pre_titration == 1 | post_titration == 1)
```

Feature assigned to both groups if present in the same number of PCR replicates for all samples

```
pre_post_titrate %>% mutate(t_sum = pre_titration + post_titration) %>% filter(t_sum > 1)
```

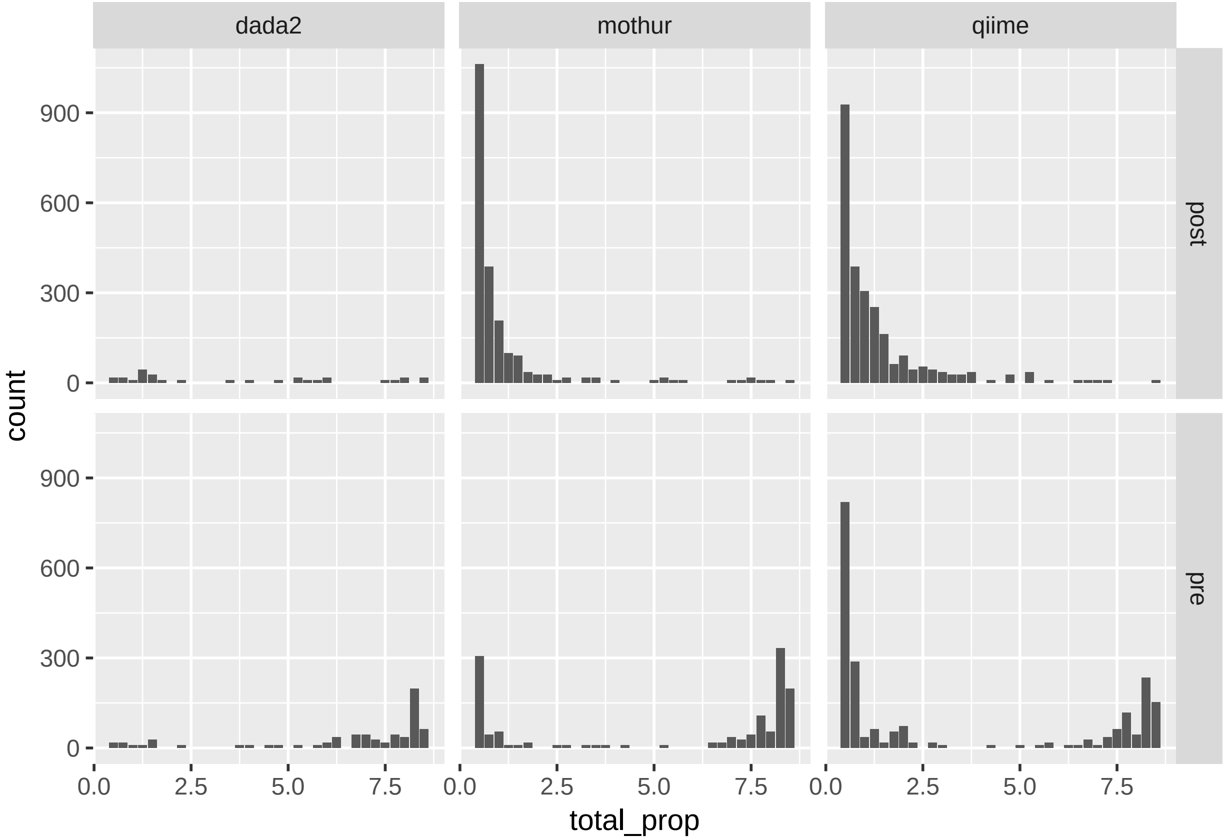
```
## Source: local data frame [0 x 17]
## Groups: pipe, biosample_id [0]
##
## # ... with 17 variables: pipe <chr>, biosample_id <chr>, feature_id <chr>,
## #   cat <chr>, T0 <dbl>, T1 <dbl>, T10 <dbl>, T15 <dbl>, T2 <dbl>,
## #   T20 <dbl>, T3 <dbl>, T4 <dbl>, T5 <dbl>, TNA <dbl>,
## #   pre_titration <dbl>, post_titration <dbl>, t_sum <dbl>
```

```
pre_post_titrate %>% mutate(t_sum = pre_titration + post_titration) %>% filter(t_sum == 1) %>%
    mutate(titrate = if_else(pre_titration == 1, "pre","post")) %>%
    ggplot() + geom_bar(aes(x = titrate, fill = biosample_id)) + facet_wrap(~pipe) + theme_bw()
```



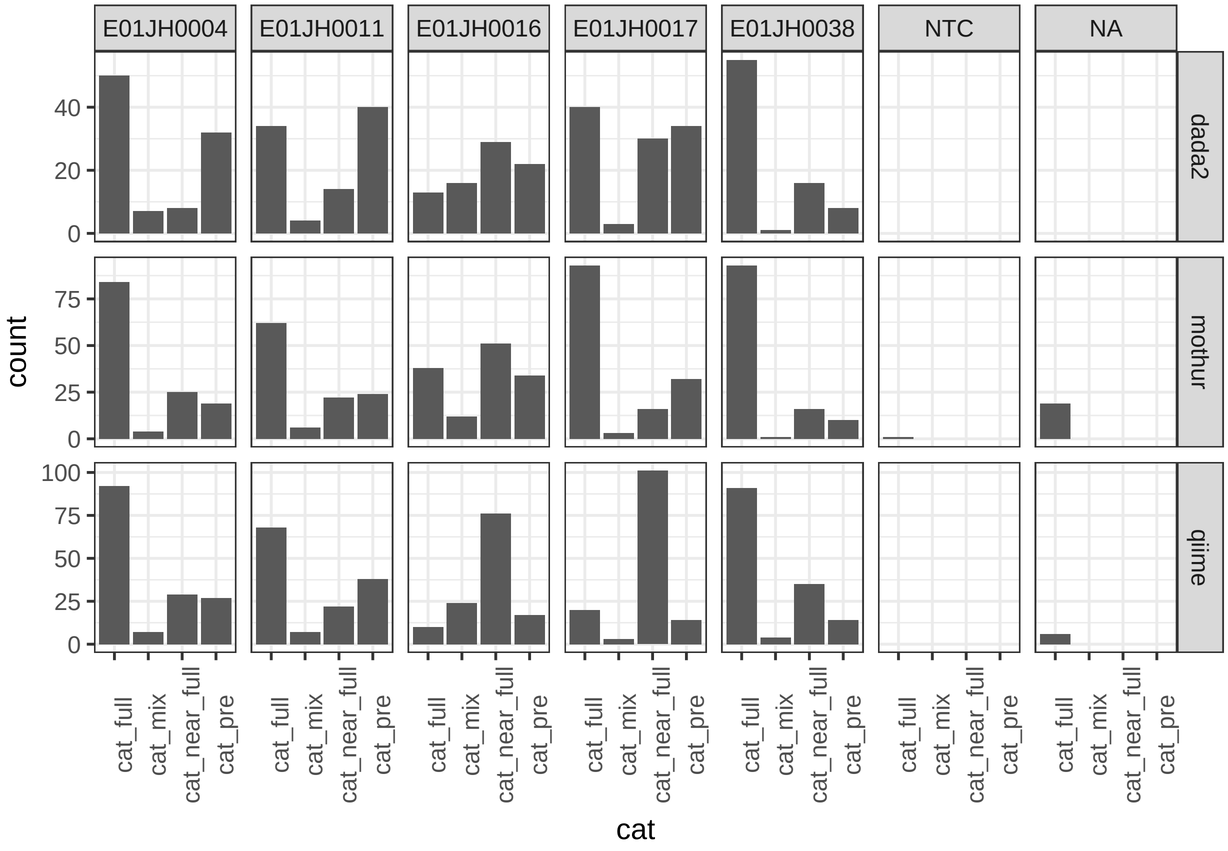
```
total_prop_titrate <- pre_post_titrate %>% mutate(t_sum = pre_titration + post_titration) %>% filter(t_
    mutate(titrate = if_else(pre_titration == 1, "pre","post")) %>%
    select(pipe, biosample_id, feature_id, titrate) %>% left_join(cat_none_df) %>%
    group_by(pipe, biosample_id, feature_id,titrate) %>% mutate(total_prop = sum(detect_prop))
```

```
total_prop_titrate %>% ggplot() + geom_bar(aes(x = total_prop)) + facet_grid(titrate~pipe)
```



Summary Figures

```
feature_cat %>% filter(cat != "cat_null", cat != "cat_none") %>%
      ggplot() + geom_bar(aes(x = cat)) +
      facet_grid(pipe ~ biosample_id, scales = "free_y") +
      theme_bw() + theme(axis.text.x = element_text(angle = 90))
```



Genus Level Category Assignments

```
count_df <- mrexp %>% map_df(get_count_df,agg_genus = TRUE, .id = "pipe") %>%
      left_join(pData(mrexp$dada2)) %>%
      filter(biosample_id != "NTC")

rep_info <- get_rep_info(count_df)
rep_info %>% saveRDS("../data/genus_rep_info_df.rds")

feature_info <- assign_cat(rep_info)
feature_info %>% saveRDS("../data/genus_info_df.rds")

feature_cat <- feature_info %>%
      select(pipe, biosample_id, feature_id,
             cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
      gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
      filter(value == 1) %>% select(-value)

feature_cat %>% saveRDS("../data/genus_categories_df.rds")
```

Category Sanity Check

```
cat_check <- feature_cat %>%
      group_by(pipe, biosample_id, feature_id) %>%
      summarise(n_cat = n())
cat_check %>% filter(n_cat != 1)
```

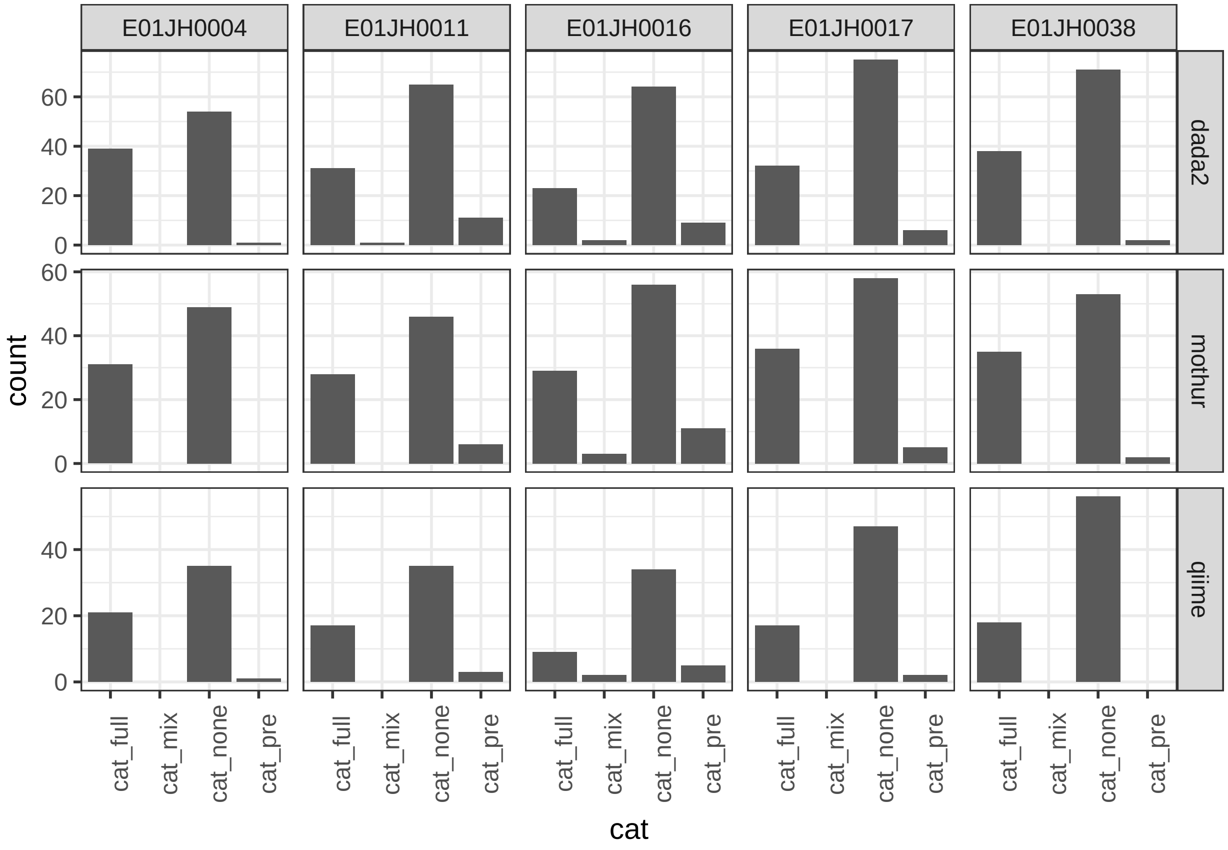
```
## Source: local data frame [0 x 4]
## Groups: pipe, biosample_id [0]
##
## # ... with 4 variables: pipe <chr>, biosample_id <chr>, feature_id <chr>,
## #   n_cat <int>
```

```
# cat_check <- feature_categories %>%
#       select(pipe, biosample_id, feature_id,
#              cat_null, cat_full, cat_mix, cat_pre, cat_post, cat_none) %>%
#       gather(cat, value, -pipe, -biosample_id, -feature_id) %>%
#       group_by(pipe, biosample_id, feature_id) %>%
#       mutate(n_cat = sum(value)) %>%
#       filter(n_cat != 1, value != 0)
# cat_check %>% arrange(feature_id)
```

Summary Figures

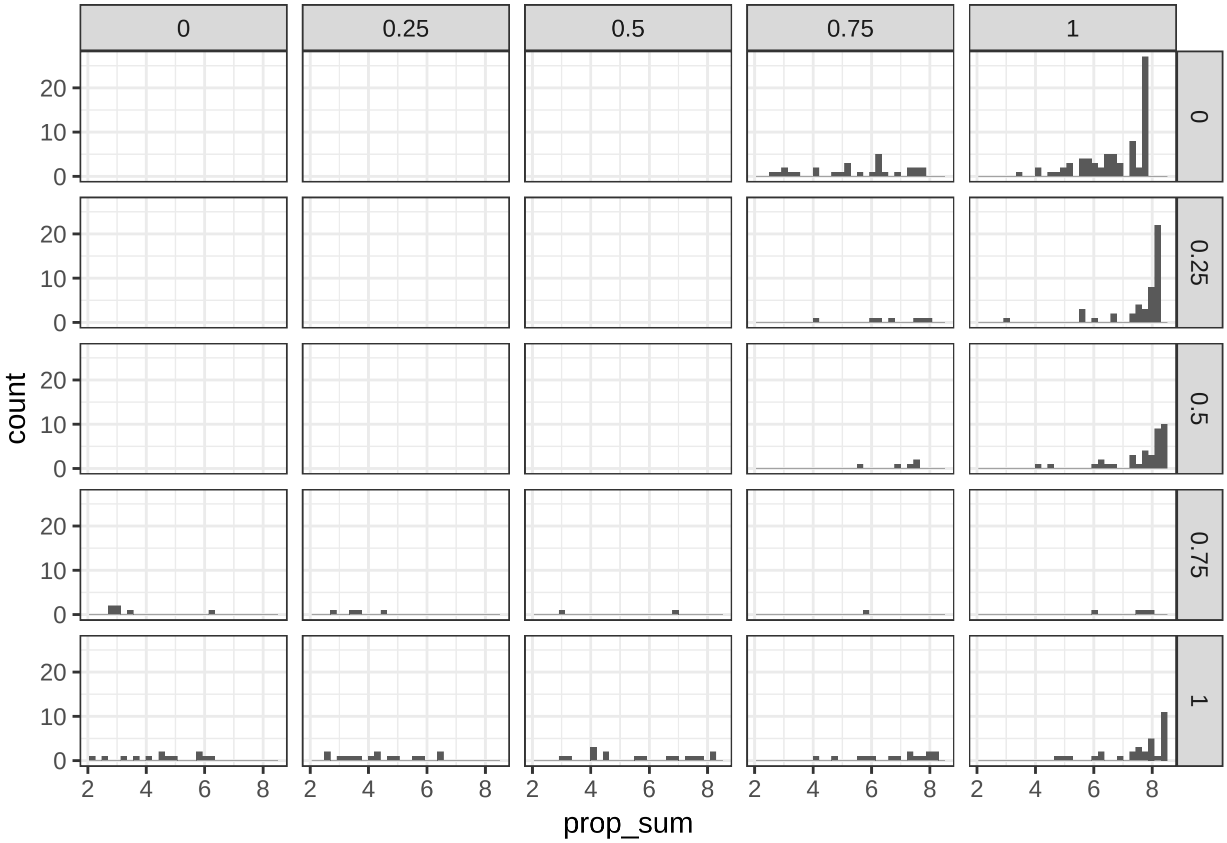
Larger proportion of full category features and fewer mix specific features when aggregating to the genus level compared to unaggregated features.

```
feature_cat %>% filter(cat != "cat_null") %>%
      ggplot() + geom_bar(aes(x = cat)) +
      facet_grid(pipe ~ biosample_id, scales = "free_y") +
      theme_bw() + theme(axis.text.x = element_text(angle = 90))
```



While there are a large number of unclassified features, few are potentially informative. Ones that stand out are features detected in 4 T0 (pre-treatment features) with prop sum value close to 8.

```
feature_info %>% filter(cat_none == 1, prop_sum > 2, T0 > 0.5 | T20 > 0.5) %>%
      ggplot() + geom_histogram(aes( x= prop_sum)) + facet_grid(T0 ~ T20) + theme_bw()
```



Alternative Characterization

```
detect_idx <- rep_info %>% mutate(detect = detect_prop * 4,
                    t_fctr = fct_relevel(t_fctr, paste0("T",c(0:5,10,15,20))),
                    idx_buffer = 10^(as.numeric(t_fctr)-1),
                    idx = detect * idx_buffer) %>%
      group_by(pipe,biosample_id,feature_id) %>% mutate(idx = sum(idx)) %>%
      filter(idx != 0)

detect_idx %>%
      group_by(idx, pipe, biosample_id, feature_id) %>%
      mutate(total_detect = sum(detect)) %>%
      filter(total_detect > 9) %>%
      group_by(idx, pipe) %>%
      summarise(count = n()) %>%
       arrange(desc(count))
```

```
## Source: local data frame [348 x 3]
## Groups: idx [313]
##
```

```
##           idx   pipe count
##         <dbl>  <chr> <int>
## 1  444444444  dada2  1467
## 2  444444444 mothur  1431
## 3  444444444  qiime   738
## 4  444444440  dada2   261
## 5  444444440 mothur   216
## 6  444344444  dada2   135
## 7  444434444  dada2   117
## 8  444444440  qiime    99
## 9  444444441  dada2    99
## 10 444344440  dada2    90
## # ... with 338 more rows
```

Appendix

Table with the total proportion of PCR replicates by biosample and pipeline.

```
total_prop_cat_none %>% mutate(total_prop = floor(total_prop)) %>%
    group_by(pipe, biosample_id, total_prop) %>% summarise(count = n()) %>%
    spread(biosample_id, count) %>% knitr::kable()
```

pipe	total_prop	E01JH0004	E01JH0011	E01JH0016	E01JH0017	E01JH0038	NTC
dada2	0	169	182	150	159	217	49
dada2	1	86	85	95	90	49	NA
dada2	2	14	18	18	20	23	NA
dada2	3	17	9	17	15	17	NA
dada2	4	15	12	17	20	18	NA
dada2	5	14	13	20	19	21	NA
dada2	6	9	12	26	19	32	NA
dada2	7	24	21	71	46	24	NA
dada2	8	13	7	9	14	21	NA
mothur	0	9266	7982	3125	3893	4729	244
mothur	1	284	270	136	218	182	NA
mothur	2	99	62	44	53	79	NA
mothur	3	51	41	25	33	51	NA
mothur	4	25	27	25	36	35	NA
mothur	5	27	24	26	22	41	NA
mothur	6	29	20	30	36	41	NA
mothur	7	24	21	57	35	31	NA
mothur	8	20	22	12	23	26	NA
qiime	0	3070	2801	3075	3351	2044	117
qiime	1	539	522	531	489	392	NA
qiime	2	174	205	181	163	154	NA
qiime	3	87	107	88	78	87	NA
qiime	4	61	70	53	59	55	NA
qiime	5	51	49	60	50	60	NA
qiime	6	42	49	36	33	29	NA
qiime	7	56	57	81	69	47	NA
qiime	8	21	29	15	22	26	NA

Session 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     2017-04-11
```

```
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	1.8.1	2016-11-07	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)
glmnet	2.0-5	2016-03-17	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	0.20-34	2016-09-06	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	1.26.1	2016-11-07	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

package	version	date	source
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)
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