Pipeline QA

Nate Olson

2016-10-05

Sequence Processing

Loading sequencing data

```
mrexp_filenames <- list(mothur = "../data/mrexp_mothur.RDS",
      qiime_denovo_chimerafilt = "../data/mrexp_qiime_denovo_chimera_filt.RDS",
      qiime_denovo_nochimerafilt = "../data/mrexp_qiime_denovo_nochimera.RDS",
      qiime_openref_chimerafilt = "../data/mrexp_qiime_refclus_chimera_filt.RDS",
      qiime_openref_nochimerafilt = "../data/mrexp_qiime_refclus_nochimera.RDS",
      dada = "../data/mrexp_dada2.RDS")

mrexp_obj <- mrexp_filenames %>% map(readRDS)
fvarLabels(mrexp_obj$qiime_openref_chimerafilt) <- paste0("taxonomy",1:7)
```

```
## Loading required package: metagenomeSeq
## Loading required package: limma
##
## Attaching package: 'limma'
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
## Loading required package: glmnet
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:S4Vectors':
##
##     expand
## The following object is masked from 'package:tidyr':
##
##     expand
## Loading required package: foreach
##
## Attaching package: 'foreach'
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
## Loaded glmnet 2.0-5
## Loading required package: RColorBrewer
```

Rename qiime samples for consistent set of ids

```
get_new_ids <- function(mr_qiime, sample_sheet){

      qiime_id_set <- pData(mr_qiime) %>% rownames()

      id_fix_df <- sample_sheet %>%
            filter(seq_lab == "JHU",barcode_lab == "JHU") %>%
            select(id, pcr_16S_plate, pos) %>%
            mutate(pos = str_replace(pos, "_",""),
                   qiime_id = str_c(pcr_16S_plate, pos, sep = "-")) %>%
            filter(qiime_id %in% qiime_id_set) %>%
            group_by(id) %>%
            mutate(id2 = if_else(grepl(x = id, pattern = "B0_M0"),
                                 paste(id,1:n(),sep = "_"),id))
      id_fix_df$id2[match(id_fix_df$qiime_id, qiime_id_set)]
}


id_set <- get_new_ids(mrexp_obj$qiime_denovo_chimerafilt, sample_sheet)
rownames(pData(mrexp_obj$qiime_denovo_chimerafilt)) <- id_set
colnames(assayData(mrexp_obj$qiime_denovo_chimerafilt)$counts) <- id_set

id_set <- get_new_ids(mrexp_obj$qiime_denovo_nochimerafilt, sample_sheet)
rownames(pData(mrexp_obj$qiime_denovo_nochimerafilt)) <- id_set
colnames(assayData(mrexp_obj$qiime_denovo_nochimerafilt)$counts) <- id_set

id_set <- get_new_ids(mrexp_obj$qiime_openref_chimerafilt, sample_sheet)
rownames(pData(mrexp_obj$qiime_openref_chimerafilt)) <- id_set
colnames(assayData(mrexp_obj$qiime_openref_chimerafilt)$counts) <- id_set

id_set <- get_new_ids(mrexp_obj$qiime_openref_nochimerafilt, sample_sheet)
rownames(pData(mrexp_obj$qiime_openref_nochimerafilt)) <- id_set
colnames(assayData(mrexp_obj$qiime_openref_nochimerafilt)$counts) <- id_set
```

Pipeline characteristics

- Section objectives
  - make non-quantitative statements
  - capturing differences in quality across samples
- Characterization of different pipelines
  - number of clusters
  - different taxonomic assignments
- Statements/ Figures showing how datasets behave
- number of assigned vs. non-assigned
- **TODO** difference in richness
  - need to figure out how I want to normalize/ transform the data prior to calculating diversity values
- number of features found across samples and replicates
- **TODO** Table - pipeline sequence budget
  - number of reads filtered due to low quality
  - number of reads merged
  - number of chimeras

Developing Code for characterizing pipeline results

Number of OTUs

```
mrexp_obj %>% map(nrow)
```

```
## $mothur
## Features
##    25739
##
## $qiime_denovo_chimerafilt
## Features
##    14326
##
## $qiime_denovo_nochimerafilt
## Features
##    24617
##
## $qiime_openref_chimerafilt
## Features
##     2832
##
## $qiime_openref_nochimerafilt
## Features
##    11381
##
## $dada
## Features
##     3691
```

Feature Count Distributions

```
extract_count_df <- function(mrexp){
     mrexp@assayData$counts %>% as_data_frame() %>%
      rownames_to_column(var = "otu") %>%
      gather("sample_name","count",-otu) %>%
      separate(sample_name,into = c("bio_rep","titration","plate","lib","sq"), sep = "_")
}

count_df <- mrexp_obj %>% map_df(extract_count_df, .id = "pipeline")
```

```
## Warning: Too many values at 171912 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9,
## 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

```
## Warning: Too many values at 295404 locations: 147703, 147704, 147705,
## 147706, 147707, 147708, 147709, 147710, 147711, 147712, 147713, 147714,
## 147715, 147716, 147717, 147718, 147719, 147720, 147721, 147722, ...
```

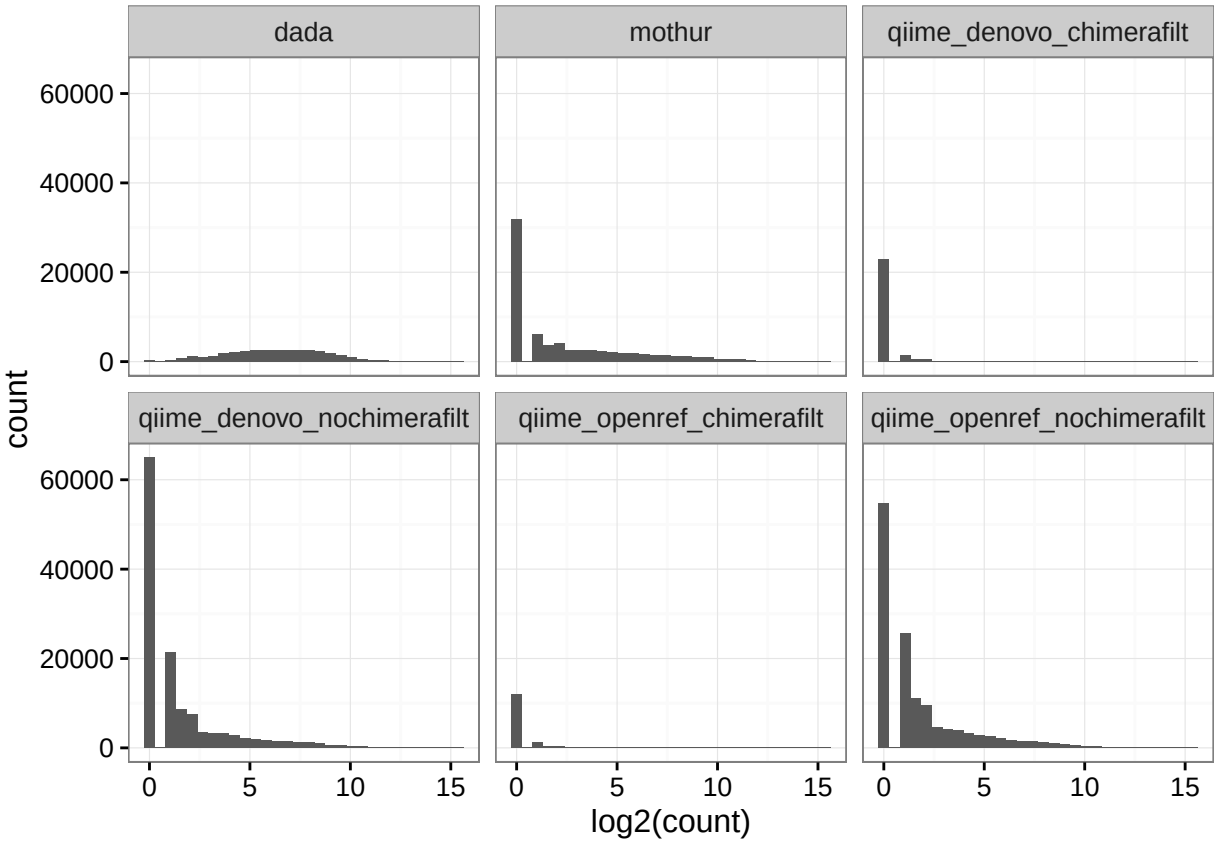
```
## Warning: Too many values at 19824 locations: 118945, 118946, 118947,
## 118948, 118949, 118950, 118951, 118952, 118953, 118954, 118955, 118956,
## 118957, 118958, 118959, 118960, 118961, 118962, 118963, 118964, ...
```

```
## Warning: Too many values at 136572 locations: 819433, 819434, 819435,
## 819436, 819437, 819438, 819439, 819440, 819441, 819442, 819443, 819444,
## 819445, 819446, 819447, 819448, 819449, 819450, 819451, 819452, ...
```

```
## Warning: Too few values at 704981 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
## 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

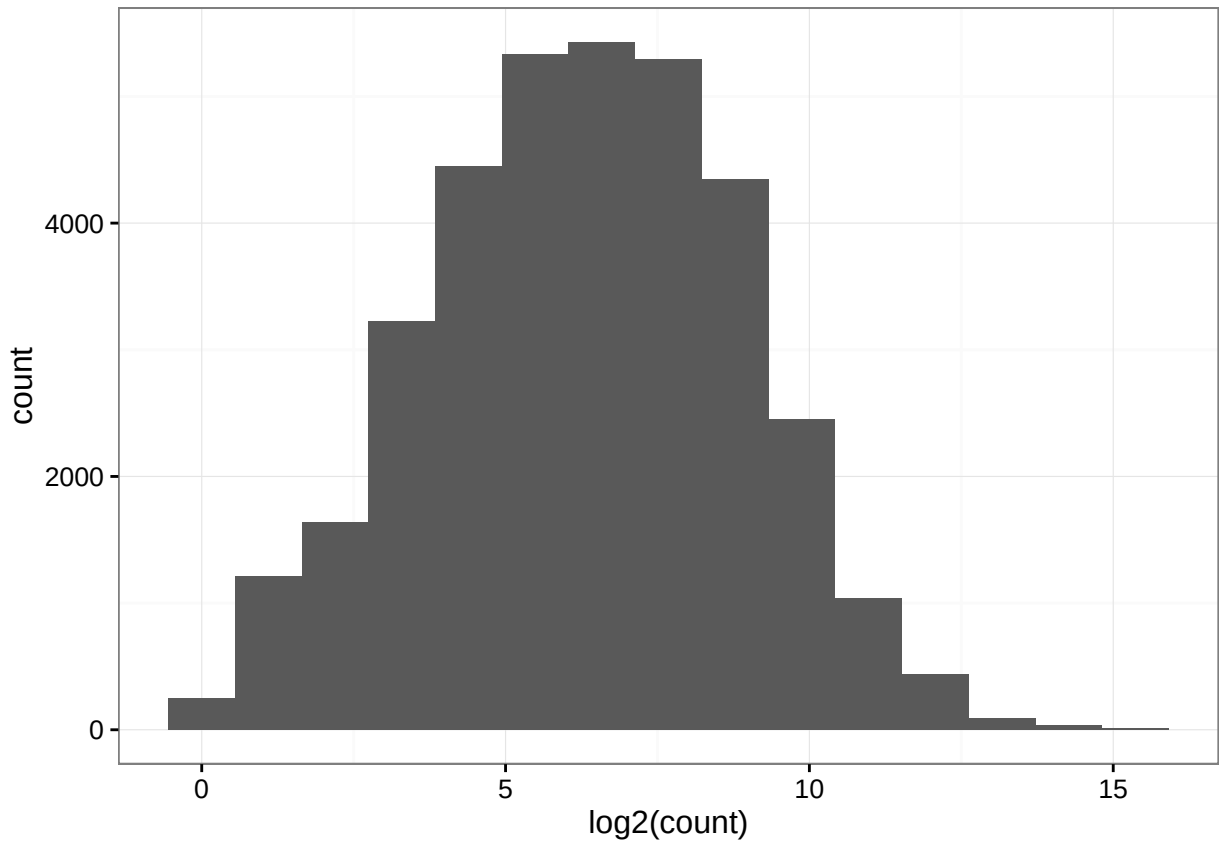
```
count_df %>% filter(count != 0) %>%  ggplot() + geom_histogram(aes(x = log2(count))) + facet_wrap(~pipel
```

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



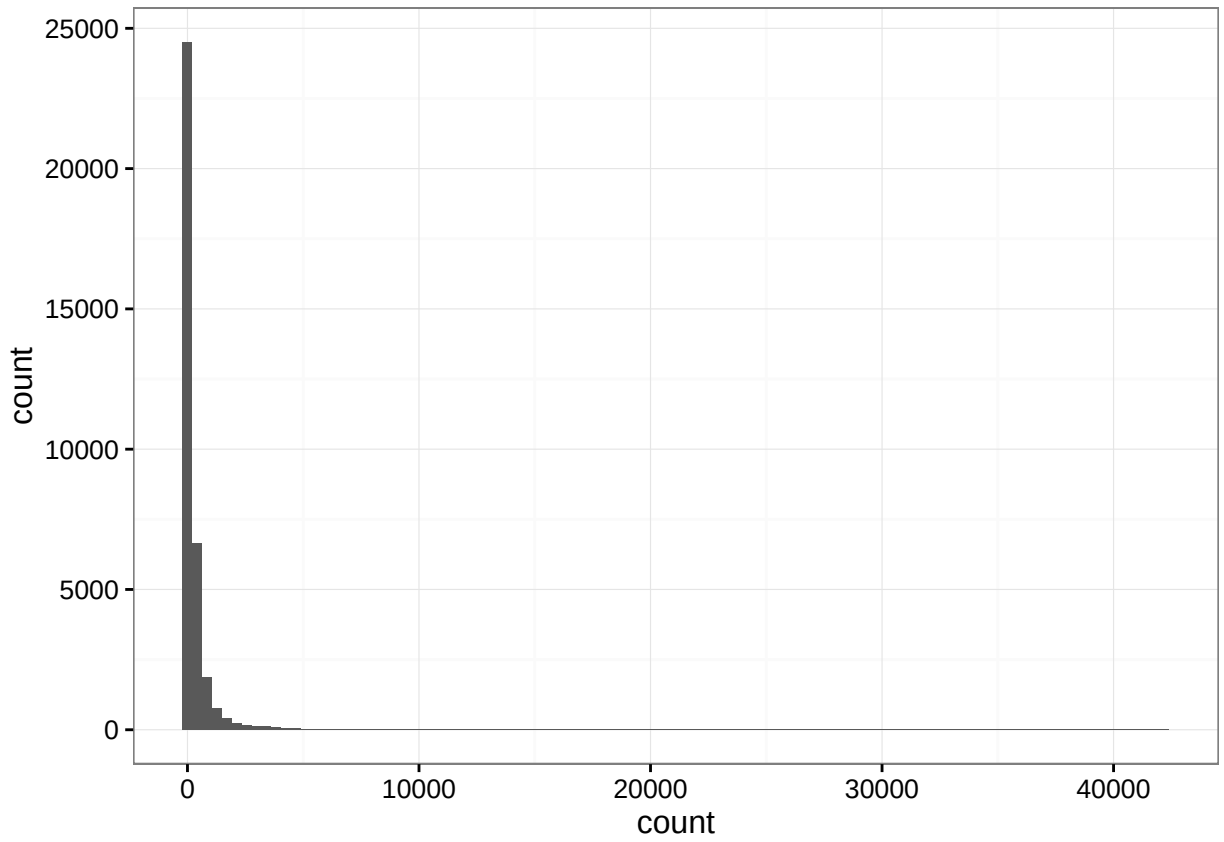
Feature count distribution for the dada2 pipeline is distinct from feature abundances from other pipelines with a bimodal distribution at 2^7 (128).

```
count_df %>% filter(count != 0, pipeline == "dada") %>%
      ggplot() + geom_histogram(aes(x = log2(count)),bins = 15) + theme_bw()
```



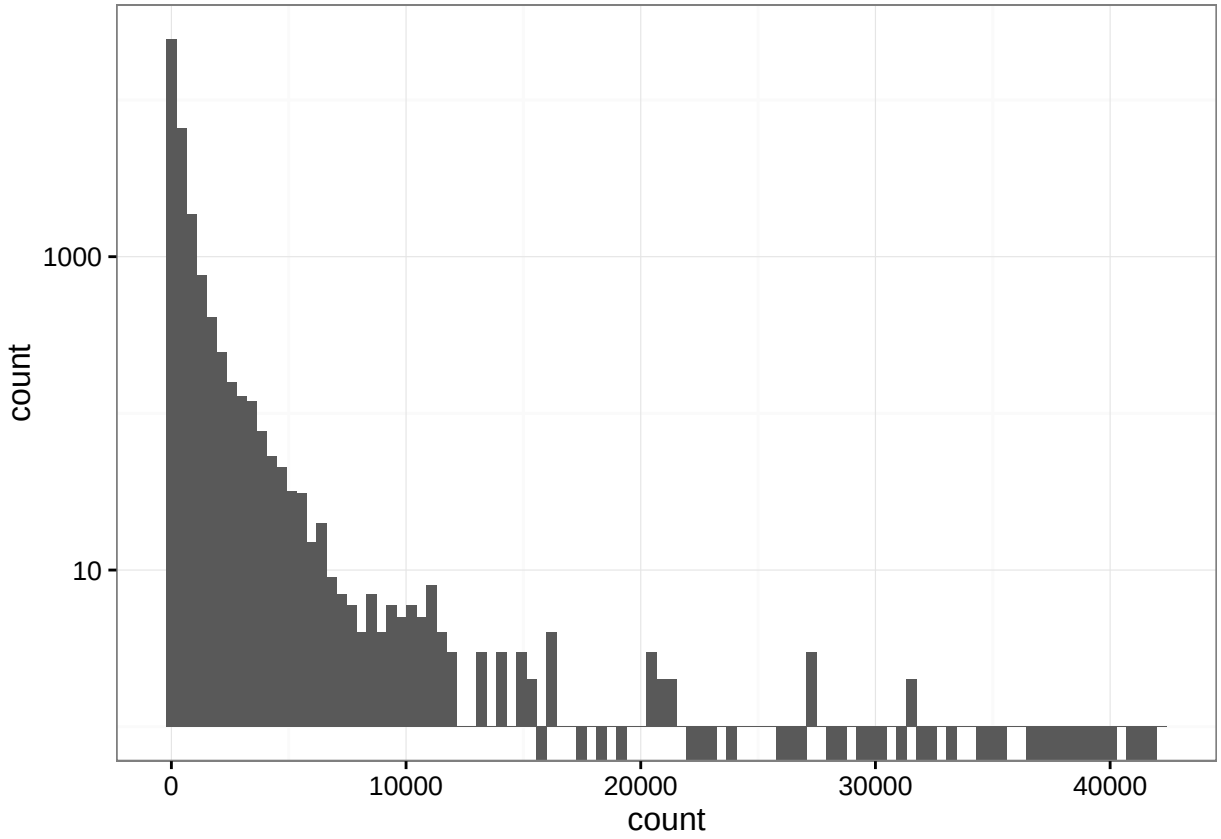
Non-log transformed feature count distribution, peak close to 0.

```
count_df %>% filter(count != 0, pipeline == "dada") %>%
      ggplot() + geom_histogram(aes(x = count),bins = 100) + theme_bw()
```




```
count_df %>% filter(count != 0, pipeline == "dada") %>%
      ggplot() + geom_histogram(aes(x = count),bins = 100) + theme_bw() + scale_y_log10()
```

```
## Warning: Stacking not well defined when ymin != 0
```


1000
10
count
0
10000
20000
30000
40000
count