Feature Inference DADA2 Unique Sequence Abundance Distribution

Nate Olson

2017-03-09

Objective

A relatively low number of inferred features per samples identified using the sequence inference pipeline (DADA2) relative to the open-reference clustering (QIIME) and de-novo clustering (Mothur) pipelines. This low number of inferred features could be due to grouping sequences that are representative of distinct biological units.

Approach

To evaluate the composition of sequences in the DADA2 features. Initial analysis of biological replicate E01JH00011, PCR replicates from half of plate 1. Characterize the distribution of sequences assigned to a feature across titrations, the assumption is that unrelated sequences will have different distributions.

Getting Data for Analysis

The `dada_to_seq_table` function generates a table with the denoised sequence, unique sequence, and the id for the input sequence from the fastq file.

```
derepFs <- file.path(dada_data_dir, "derepFs-2016-11-07.rds") %>% readRDS()
dadaFs <- file.path(dada_data_dir, "dadaFs-single-inference-2016-11-07.rds") %>%
      readRDS()
```

```
get_seqtable <- function(sam){
      sr <- ShortRead::readFastq("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/
                           pattern = pats <- paste0(sam,".*R1.*filt.fastq.gz"))
      dadaRes <- dadaFs[[sam]]
      derep <- derepFs[[sam]]
      dada2:::dada_to_seq_table(dadaRes = dadaRes, derep = derep, sr = sr)
}
```

```
## sample IDs
sams <- paste0("1-",LETTERS[1:8],"2") %>% c(.,"1-B2")
sam_list <- as.list(sams) %>% set_names(sams)
seqTable <- map_df(.x = sam_list, .f = get_seqtable, .id = "sampleID")
# saveRDS(seqTable, "E01JH00011_plate1_half1_dada_seq_tbl.rds")
## Clean-up
rm(derepFs)
rm(dadaFs)
```

seqTable data.table variable description

- sampleID - PCR (sample) the sequence is from
- seq - representative sequence, most abundant sequence in the feature

- n0 - could not find in dada2 documentation
- uniqueIndex - dada2 assigned index
- genotypeIndex - dada2 assigned index
- derepSeq - unique sequence
- id - sequence id for source read

Distribution of Dereplicated Sequences across samples.

Extracting the sequence top 4 and most abundant features.

```
top_seqs <- seqTable %>% group_by(seq) %>% summarise(count = n()) %>%
      arrange(desc(count)) %>% .$seq %>% .[1:4]
top_seqTable <- seqTable %>% filter(seq %in% top_seqs)
top_sam_seqTable <- top_seqTable %>% group_by(sampleID, seq, derepSeq) %>% summarise(count = n())
```

Sequences with abundance consistently above 10 tend to have similar distribution patterns to the most abundant supporting the hypothesis that the sequences in the feature are representatives of the sample biological units. May want to consider testing for consistencies in abundances distributions between the samples.

```
top_sam_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~seq, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```

Total Variance Feature Ranking

Based on conversation with Hector (3/1), for each feature

1. center and scale unique sequence abundance distribution across samples
2. center values by sample
3. calculate total feature variance as $\sqrt{SS}$

```
seq_abu <- seqTable %>% group_by(seq, derepSeq, sampleID) %>%
      summarise(count = n())


seq_abu_filt <- seq_abu %>%
      ## removing unique sequences only present in one sample
      ## irrelevant to distribution based analysis
      group_by(seq, derepSeq) %>% summarise(count = n()) %>%
      filter(count > 1) %>% select(-count) %>%
      ## removing feature with only one unique sequence - not working ...
      # group_by(seq) %>% summarise(count = n()) %>%
      # filter(count > 1) %>% select(-count) %>%
      left_join(seq_abu)
```

```
## Joining, by = c("seq", "derepSeq")
```

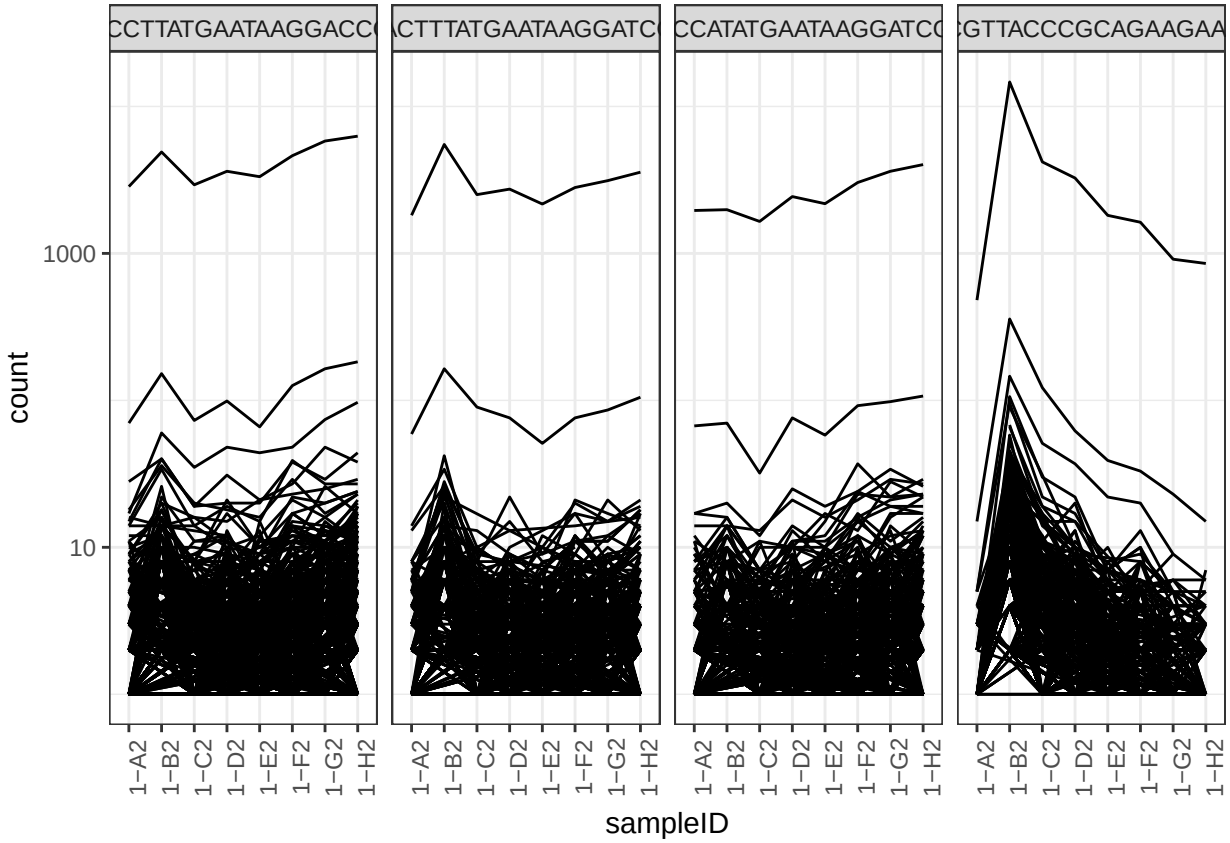
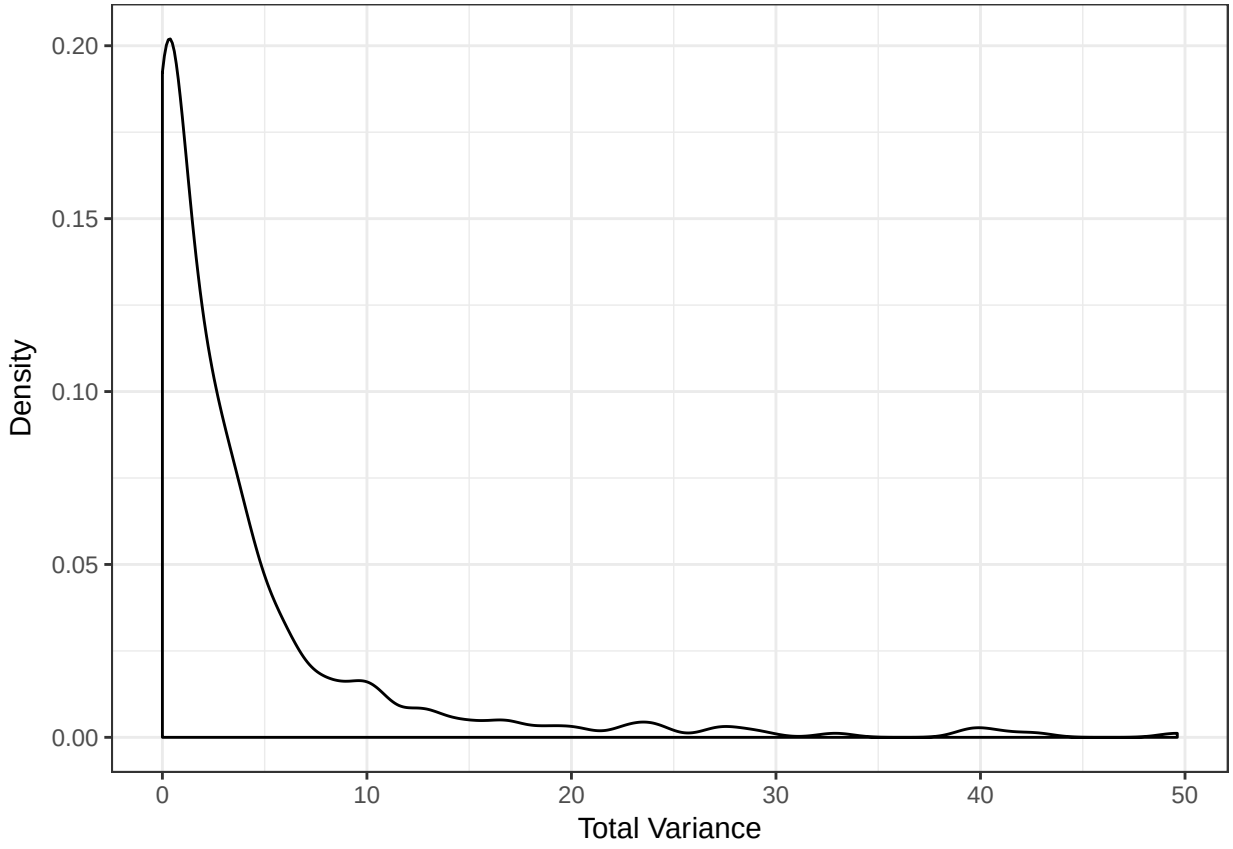


Figure 1: Distribution of unique sequences assigned to the 10 most abundant features for biological replicate 2, samples from the first half of PCR plate 1. Each line represents the abundance of a unique (dereplicated) sequence across samples.

```
seq_rank <- seq_abu_filt %>%
      group_by(seq, derepSeq) %>% mutate(mu = mean(count), stdev = sd(count)) %>%
      ## when standard deviation is 0 or NA how to scale? - replacing with 1
      ungroup() %>% mutate(stdev = if_else(stdev == 0, 1, stdev)) %>%
      ## count_cs unique sequence wise centered and scaled value
      ungroup() %>% mutate(count_cs = (count - mu)/stdev)  %>%
      group_by(seq, sampleID) %>% mutate(seq_cs_mu = mean(count_cs)) %>%
      ## count_csc sample-wise centered count_cs
      ungroup() %>% mutate(count_csc = count_cs - seq_cs_mu) %>%
      ## seq_var_ss - sequence wise square root sum of squares of count_csc
      group_by(seq) %>% summarise(seq_var_ss = sqrt(sum(count_csc^2))) %>%
      ## ranking features by variance ss
      ungroup() %>% arrange(desc(seq_var_ss)) %>% mutate(var_ss_rank = 1:n())
```

Distribution of feature level total variance.

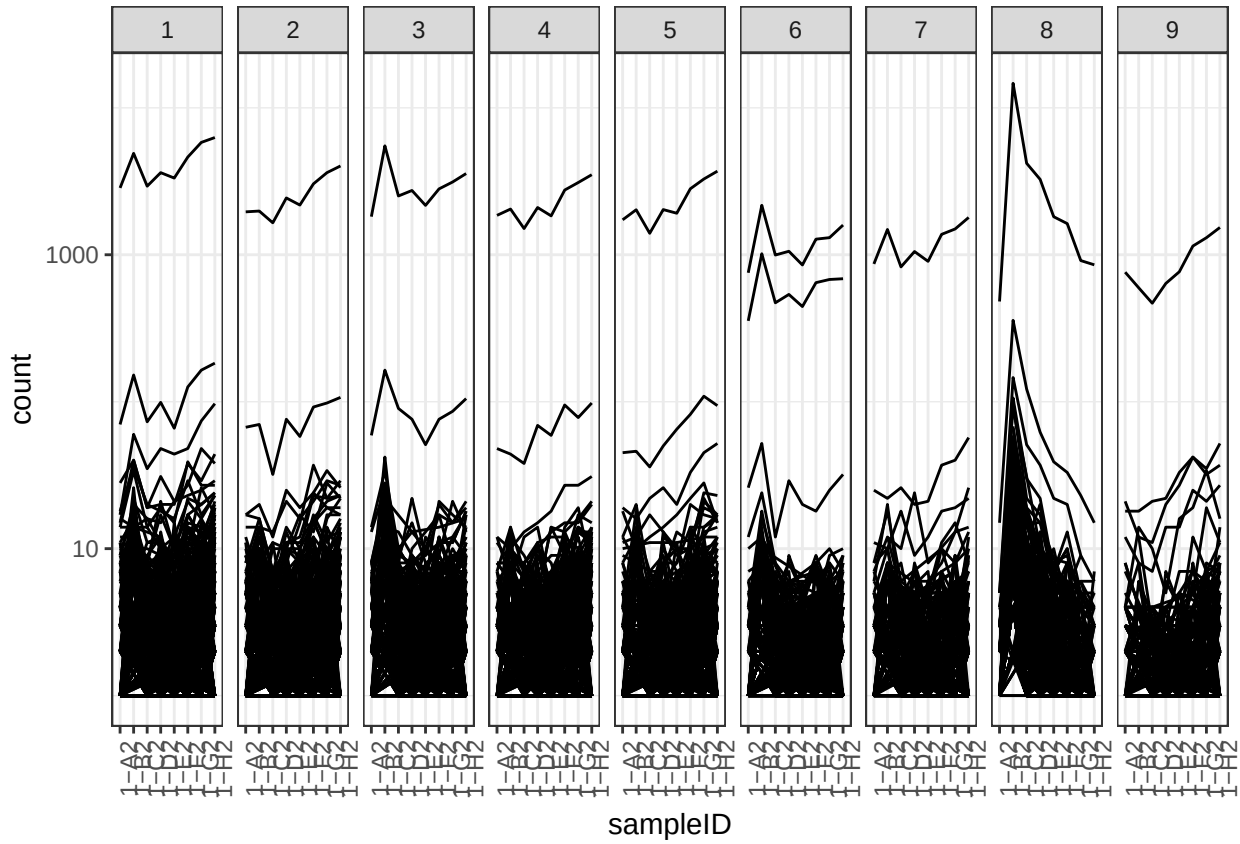
```
seq_rank %>% ggplot() + geom_density(aes(x = seq_var_ss)) + theme_bw() +
      labs(x = "Total Variance", y = "Density")
```




```
top_seqTable <- seq_rank %>% filter(var_ss_rank < 10) %>% left_join(seq_abu)
```

```
## Joining, by = "seq"
```

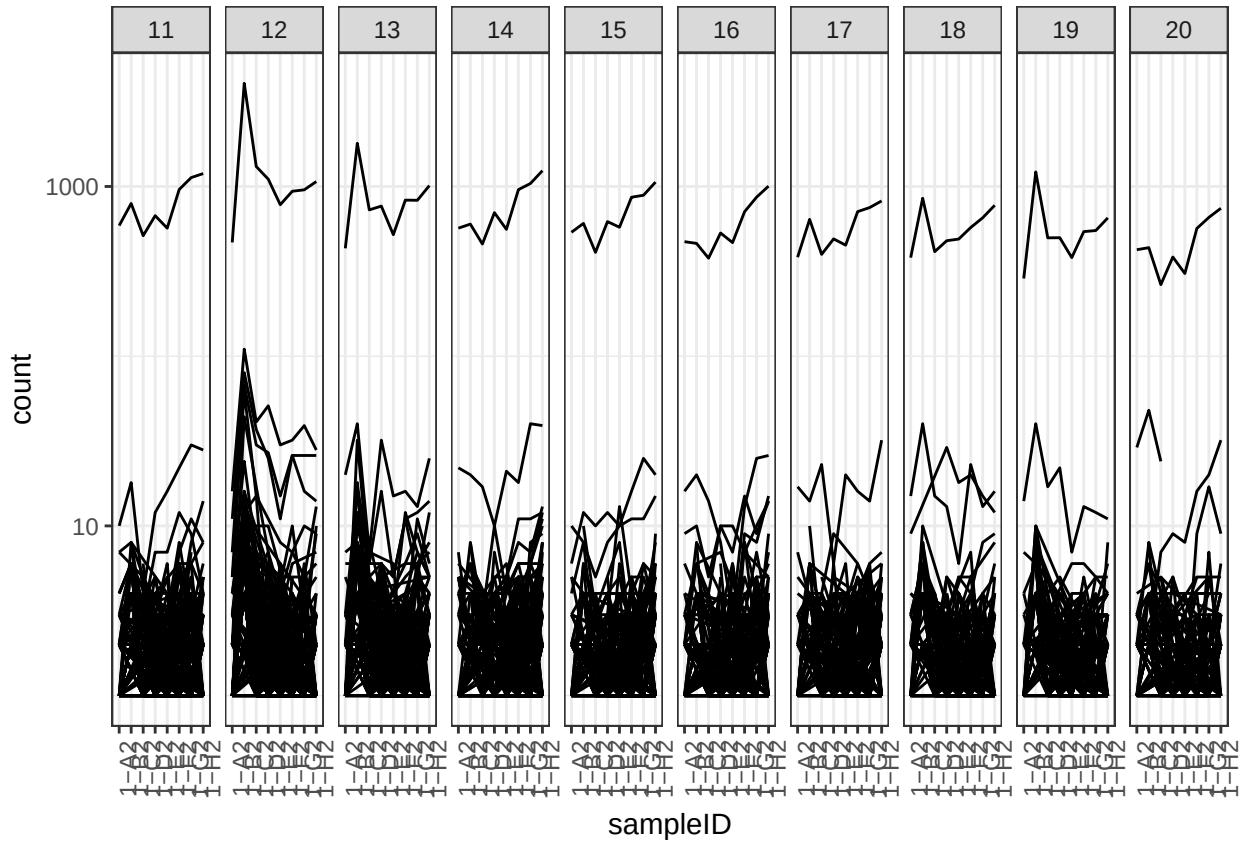
```
top_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```




```
top_seqTable <- seq_rank %>% left_join(seq_abu_filt) %>% filter(var_ss_rank %in% 11:20)
```

```
## Joining, by = "seq"
```

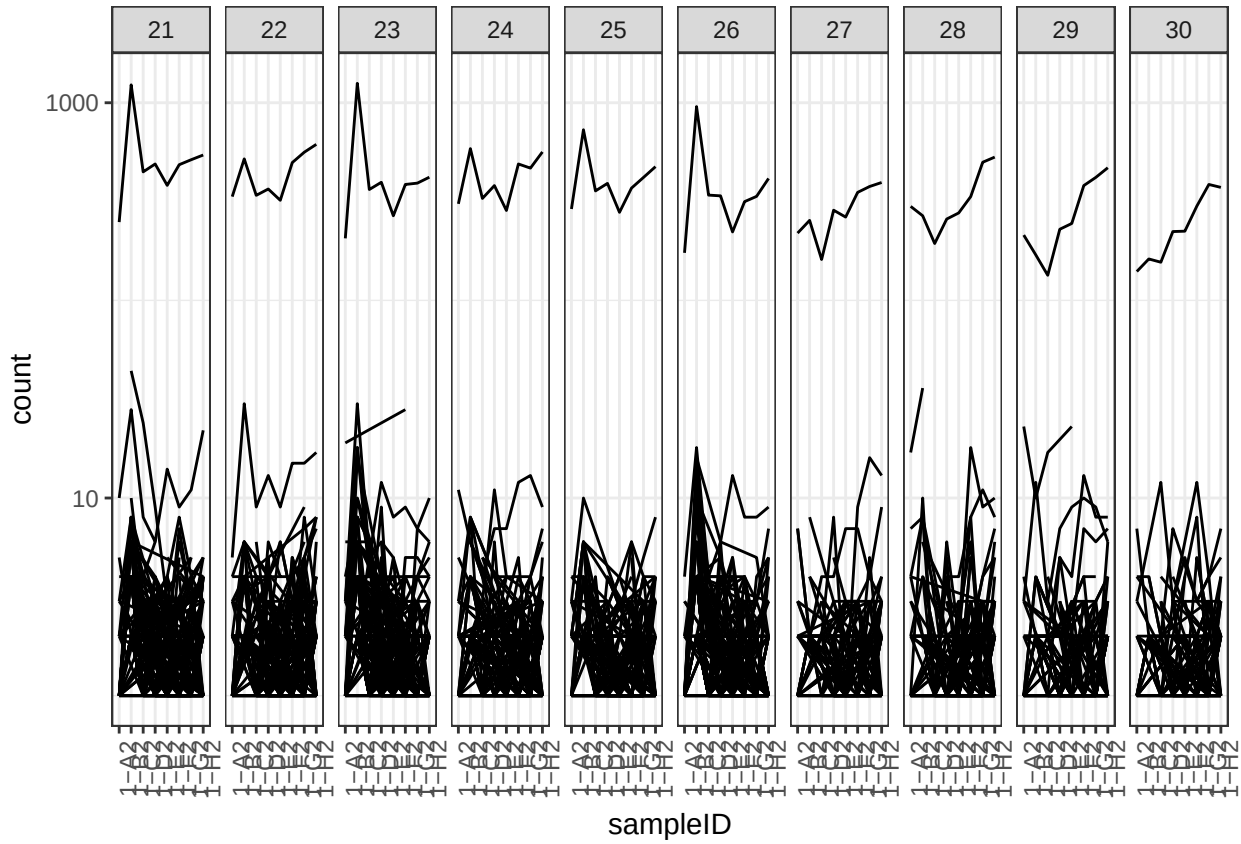
```
top_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```




```
top_seqTable <- seq_rank %>% left_join(seq_abu_filt) %>% filter(var_ss_rank %in% 21:30)
```

```
## Joining, by = "seq"
```

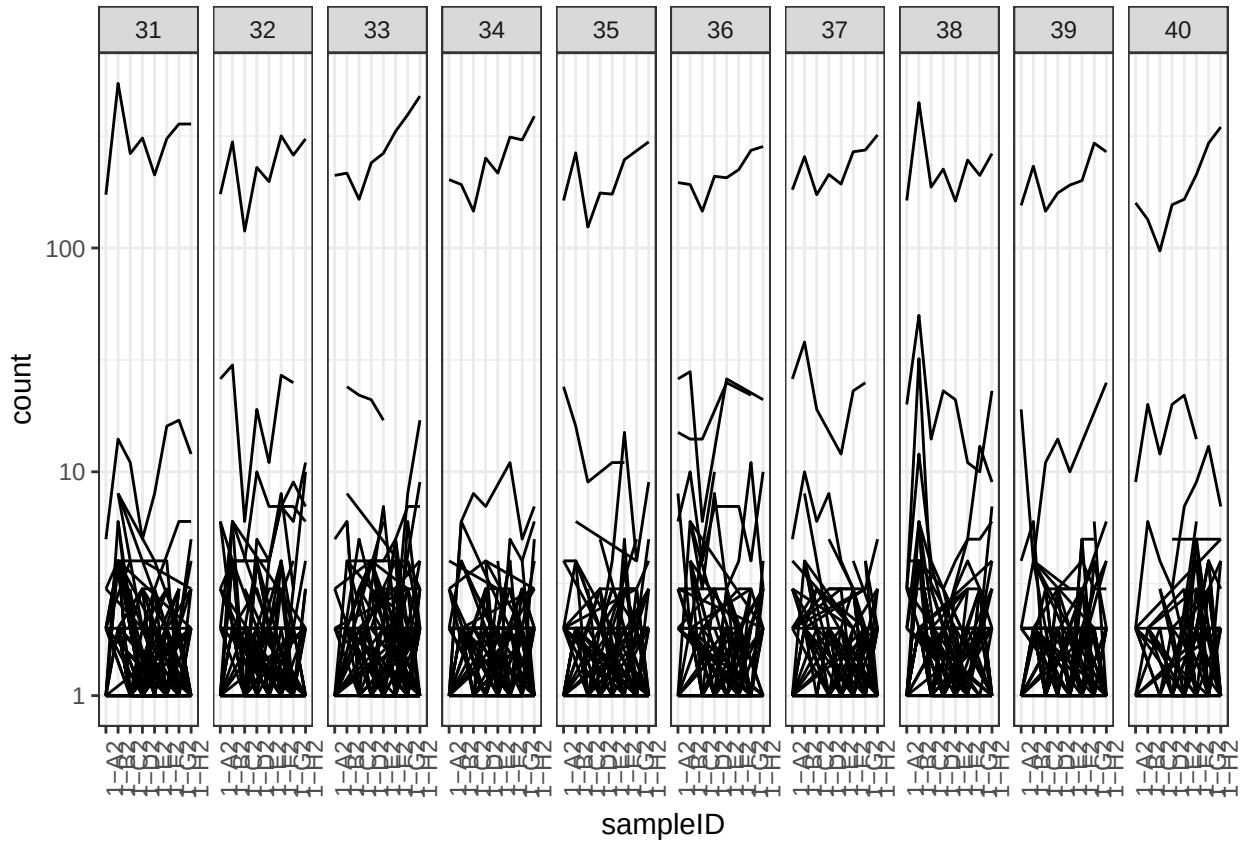
```
top_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```




```
top_seqTable <- seq_rank %>% left_join(seq_abu_filt) %>% filter(var_ss_rank %in% 31:40)
```

```
## Joining, by = "seq"
```

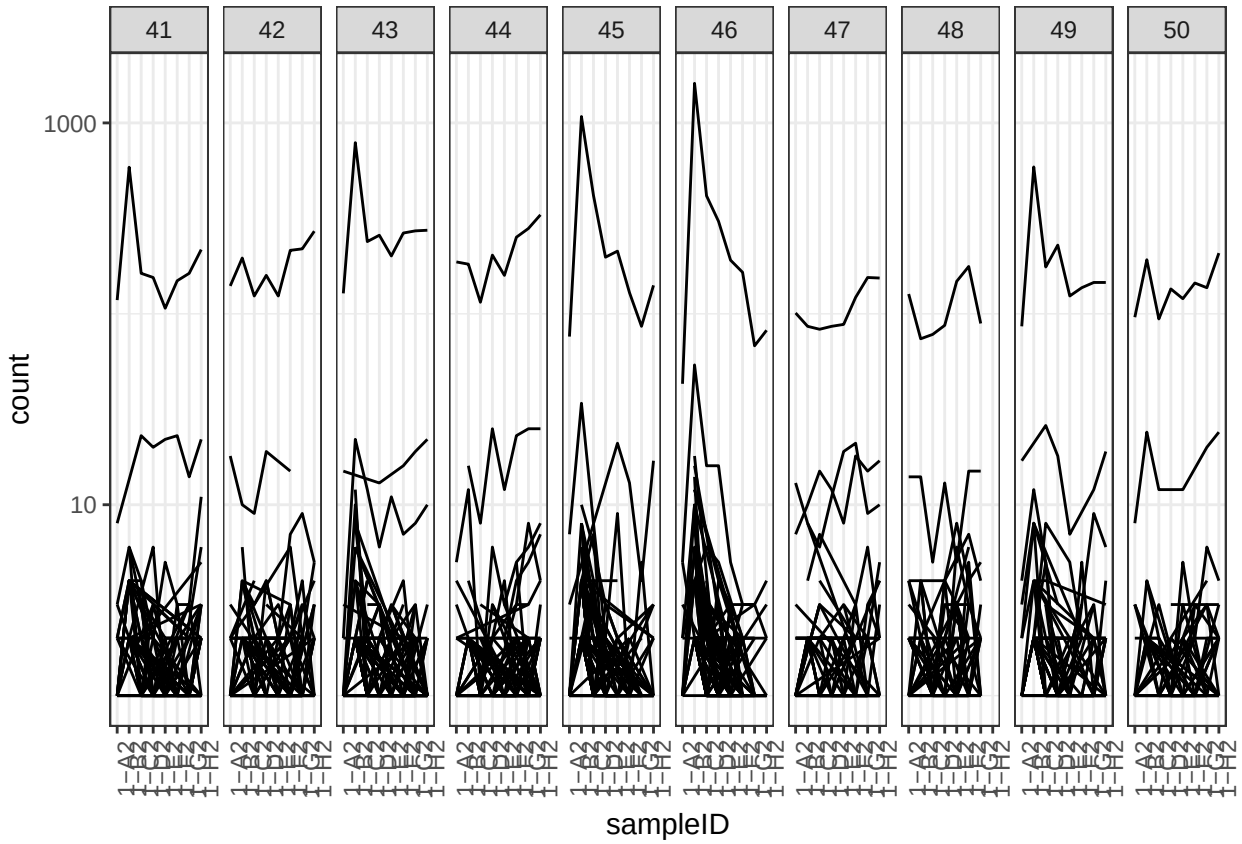
```
top_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```




```
top_seqTable <- seq_rank %>% left_join(seq_abu_filt) %>% filter(var_ss_rank %in% 41:50)
```

```
## Joining, by = "seq"
```

```
top_seqTable %>%  ggplot() +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank, nrow = 1) +
      theme(axis.text.x = element_text(angle = 90))
```



Generating feature distribution plot for all sequences.

```
# full_seqTable <- seq_rank %>% left_join(seq_abu_filt)
# full_plot <- full_seqTable %>%  ggplot() +
#       geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
#       scale_y_log10() + theme_bw() + facet_wrap(~var_ss_rank) +
#       theme(axis.text.x = element_text(angle = 90))
# ggsave(full_plot, filename = "unique_seq_dist_full.pdf", width = 48, height = 36)
```