Feature Inference Discrepancies - DADA2

Nate Olson

2017-02-28

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.

1. Investigate the within feature pairwise sequence distances.

2. 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 <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/derepFs-2016-
dadaFs <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/dadaFs-single-
```

```
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")
```

Extracting the sequence top 10 and most abundant features.

```
top_seqs <- seqTable %>% group_by(seq) %>% summarise(count = n()) %>%
      arrange(desc(count)) %>% .$seq %>% .[1:10]
top_seqTable <- seqTable %>% filter(seq %in% top_seqs)
top_sam_seqTable <- top_seqTable %>% group_by(sampleID, seq, derepSeq) %>% summarise(count = n())
```

Distribution of dereplicated sequences across samples. Sequences with the abundances consistently above 10 tend to have similar distribution patterns to the most abundant supporting the hypothesis that the sequences

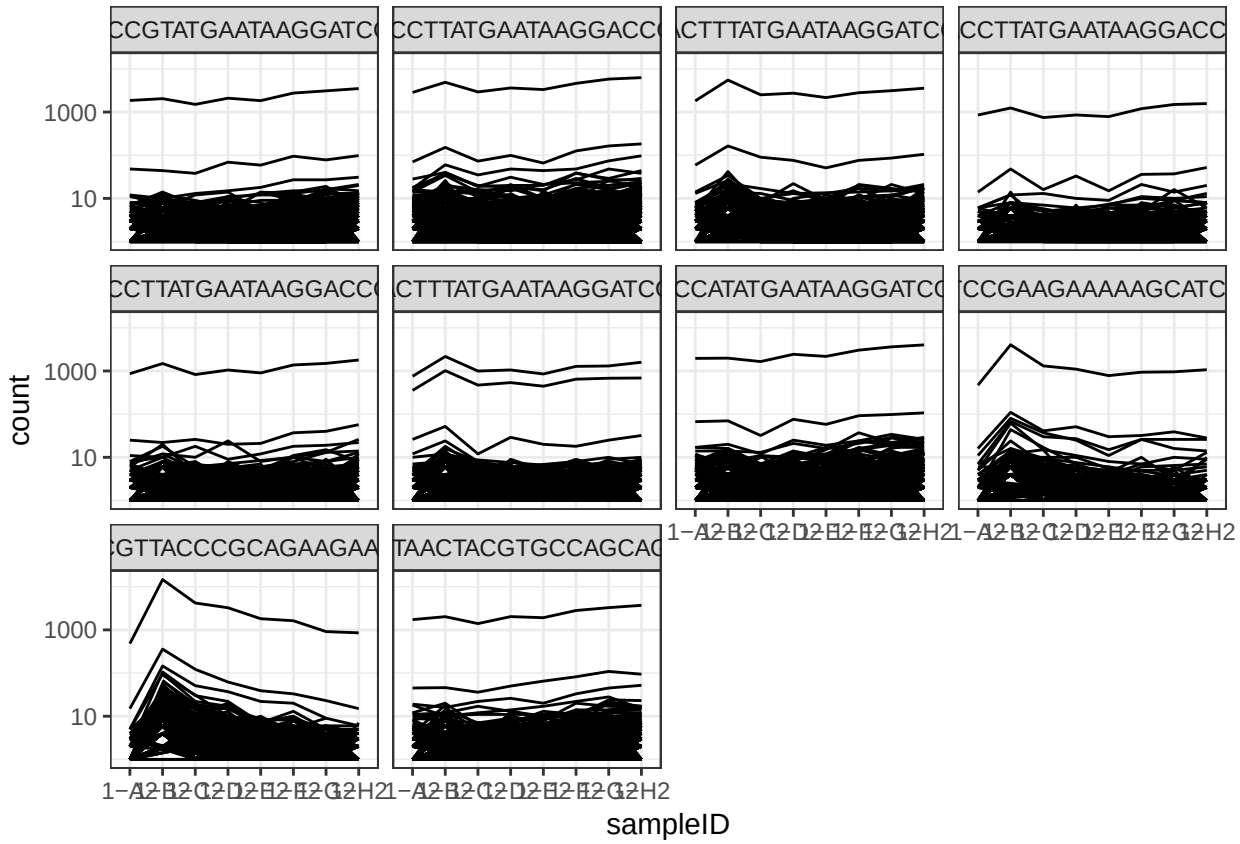

Figure 1: Distribution of unique sequences assigned to the 10 most abundant features for biological replicate 2, samples from the first half of PCR plate 1. Each line represents the abundance of a unique (dereplicated) sequence across samples.

in the feature are representatives of the sample biological units. May want to consider testing for consistencies in abundances distributions between the samples.

```
ggplot(top_sam_seqTable) +
      geom_path(aes(x = sampleID, y = count, group = derepSeq)) +
      scale_y_log10() + theme_bw() + facet_wrap(~seq)
```

Oligotype Analysis for most abundant feature

Extracting sequences for the most abundant feature and saving as fasta.

```
abu_seq <- seqTable %>% group_by(seq) %>% summarise(count = n()) %>%
      arrange(desc(count)) %>% .$seq %>% .[1]
abu_seqTable <- seqTable %>% filter(seq == abu_seq)
seqs <- abu_seqTable$derepSeq
names(seqs) <- paste0(abu_seqTable$sampleID,"_", abu_seqTable$id)

## saving sequences to file
DNAStringSet(seqs) %>% writeXStringSet("~/Desktop/test.fasta")
```

Performing oligotype analysis

Oligotype analysis performed on the command line

```
entropy-analysis test.fasta
```

Issues with the oligotyping based approach - high entropy values even after splitting into different oligotypes.

```
knitr::include_graphics("~/Desktop/test.fasta-ENTROPY.png")
```

Shannon entropy by sequence position for all sequences assigned to the most abundant feature.

After entropy decomposition (oligotyping) using a single base position the most abundant oligotype has low entropy values(~0.2). These values are defined as the noise level for Illumina sequencing data in the Oligotyping article. For the other oligotypes the entropy level is higher. Inspection of the sequences indicates that the sequences have different starting positions which resulted in high entropy values.

```
# knitr::include_graphics(c("~/Desktop/test-c1-s1-a0.0-A0-M4/OLIGO-REPRESENTATIVES/00000_C-ENTROPY.png"
#                           "~/Desktop/test-c1-s1-a0.0-A0-M4/OLIGO-REPRESENTATIVES/00001_G-ENTROPY.png"
#                           "~/Desktop/test-c1-s1-a0.0-A0-M4/OLIGO-REPRESENTATIVES/00002_A-ENTROPY.png"
#                           "~/Desktop/test-c1-s1-a0.0-A0-M4/OLIGO-REPRESENTATIVES/00003_T-ENTROPY.png"
```

Issue relating to different starting position for dereplicated sequences performing multiple sequence alignment to see if that clears up the issue Using mothur align.seqs with SILVA referencde alignment to address multiple starting positions issue. Issue with long and short sequence, removing using screen.seqs. Using filter.seqs to remove gap positions, may want to filter end positions with "." as well.

```
align.seqs(fasta=test.fasta, reference=/Users/nolson/Projects/16S_etec_mix_study/analysis/pipelines/moth
screen.seqs(fasta=test.align, minlength=270, maxlength=279)
filter.seqs(fasta=test.good.align, vertical=T, trump=-)
```

Next steps - look into entropy for aligned seqs

Within Feature Pairwise Seq Distances

1. Obtain unique sequences assigned to individual features
2. Calculate pairwise distances on unique sequences - may need to do on forward and reverse reads individually
    1. calculate pairwise distances using ..... Mothur with sequence alignment?