qiime_pipeline

Nate Olson

October 26, 2016

```
library(biomformat)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ---------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(readr)
library(stringr)
library(forcats)
```

Pipeline description

- Merging paired-end reads
  - fastq-join - Erik Aronesty, 2011. ea-utils : "Command-line tools for processing biological sequencing data" (https://github.com/ExpressionAnalysis/ea-utils/blob/wiki/FastqJoin.md)
  - fastq-join uses sqr(distance)/len to anchor alignments
  - does not use Smith-Waterman alignment
  - method not suitable for sequencing technologies with high insertion and deletion rates
  - Reference describing methods http://benthamopen.com/contents/pdf/TOBIOIJ/TOBIOIJ-7-1.pdf
- Quality filtering - phred >= 3 default value, minimum base quality score
- Clustering: open reference using UCHIME and greengenes 13.8 database clustered at threshold of 0.97
  - reference based clustering of sequences
  - unclustered seqs: (1) random subset of sequences selected and clustered. (2) New cluster centers are used as references in reference based clustering. (3) Remaining sequences are de novo clustered.
  - Clusters from four steps are combined.
  - taxonomic assignments from UCLUST reference based clustering
- pyNAST refernece alignment - cluster centers and associated reads that fail to align to the reference database are removed from the analysis

Pipeline Budget

Raw sqeunces

Starting number of sequences per sample.

```
seq_meta <- readRDS("../data/seq_metadata_df.RDS")
```

```
raw_count <- seq_meta %>% filter(Read == "R1") %>% select(ill_id,reads) %>%
      dplyr::rename(id = ill_id, total = reads) %>% mutate(pipe_step = "raw")
```

Merged sequences

Number of merged sequences per sample.

```
merged_countfile <- paste0("~/Projects/16S_etec_mix_study/analysis/pipelines",
                           "/qiime/joined_pairs/fastqjoin_counts.txt")
merged_count <- read_lines(merged_countfile) %>% tibble(id = .) %>%
      separate(id, c("id","total"),sep = ":") %>%
      mutate(id = str_replace(id, "_S.*",""), total = as.numeric(total), pipe_step = "merged")
```

Quality Filter

Number of sequences per sample after quality filtering.

```
filter_countfile <- paste0("~/Projects/16S_etec_mix_study/analysis/pipelines",
                           "/qiime/split_libs/split_library_log.txt")

filter_count <- read_lines(filter_countfile) %>% tibble(id = . ) %>%
      filter(str_detect(id, "L001_R1_001\t")) %>%
      separate(id, c("id","total"),sep = "\t") %>%
      mutate(id = str_replace(id, "_S.*",""), total = as.numeric(total), pipe_step = "filter")
```

Clusters

Total number of sequences and features after open reference clustering with UCLUST.

```
cluster_df <- read_biom("~/Projects/16S_etec_mix_study/analysis/pipelines/qiime/otus_uc_fast/otu_table_
      biom_data() %>%
      SparseM::as.matrix() %>% as.data.frame() %>%
      rownames_to_column(var = "otuID") %>%
      gather("id","count", -otuID) %>% filter(count != 0)
```

```
## Warning in strsplit(msg, "\n"): input string 1 is invalid in this locale
```

```
cluster_count <- cluster_df %>% group_by(id) %>%
      summarise(total = sum(count), unique = n()) %>%
      mutate(pipe_step = "cluster")
```

Filtering pyNAST failures

Number of sequences and unique OTUs per sample after removing sequences with pynast failures.

```
pynast_df <- read_biom("~/Projects/16S_etec_mix_study/analysis/pipelines/qiime/otus_uc_fast/otu_table_m
      biom_data() %>%
      SparseM::as.matrix() %>% as.data.frame() %>%
      rownames_to_column(var = "otuID") %>%
      gather("id","count", -otuID) %>% filter(count != 0)
```

```
## Warning in strsplit(msg, "\n"): input string 1 is invalid in this locale
```

```
pynast_count <- pynast_df %>% group_by(id) %>%
      summarise(total = sum(count), unique = n()) %>%
      mutate(pipe_step = "pynast_filter")
```

Combining count results

```
count_df <- bind_rows(raw_count, merged_count, filter_count, cluster_count, pynast_count) %>%
      gather("count_type","value", -id, -pipe_step) %>%
      mutate(pipe_step = fct_relevel(pipe_step, c("raw","merged","filter","cluster","pynast")),
             step_num = as.numeric(pipe_step))
```

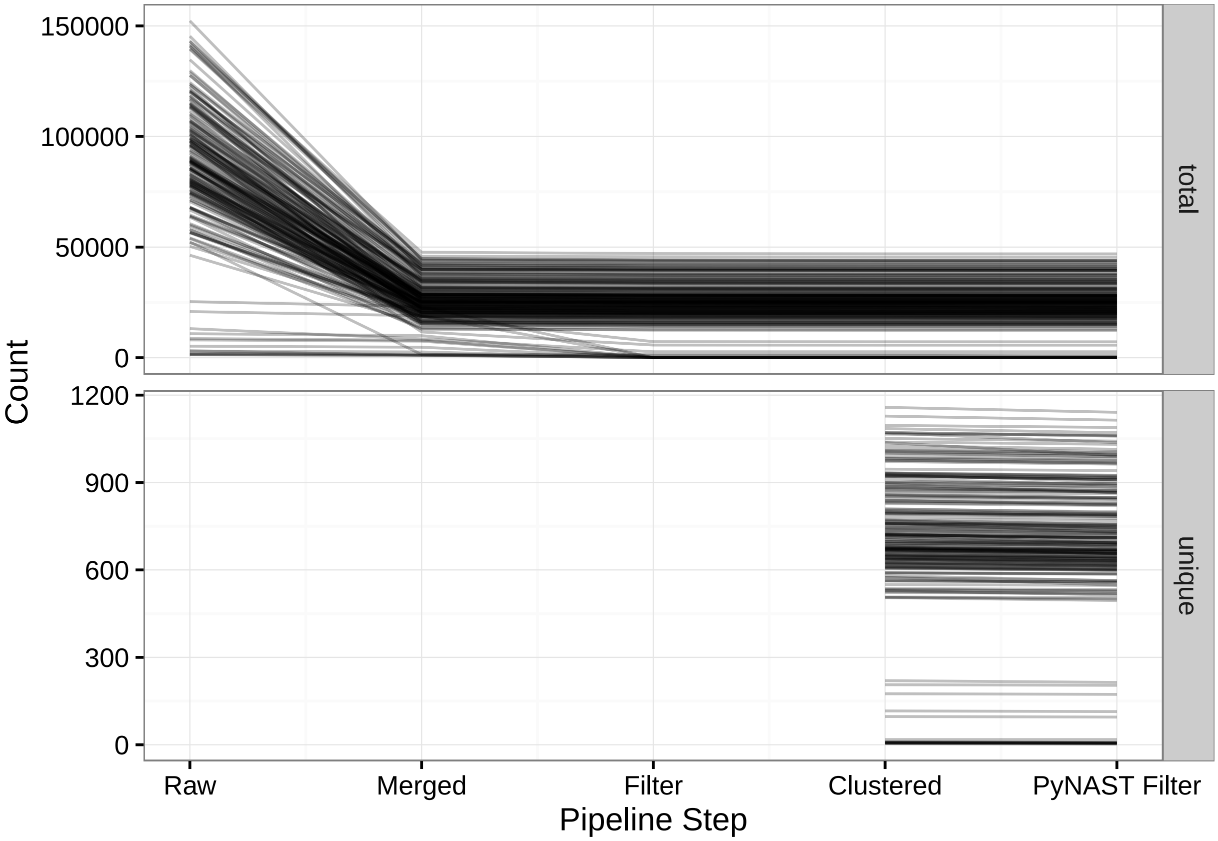
```
## Warning: Unknown levels in `f`: pynast
```

Pipeline Processing

Most of the low quality reads are removed during the initial merging of forward and reverse reads. Minimal change in the number of OTUs after filtering cluster centers that fail PyNAST alignment.

Compared to mothur, fewer reads per sample after processing but more OTUs.

```
count_df %>%
 ggplot() + geom_path(aes(x = step_num, y = value, group = id), alpha = 0.25) +
      facet_grid(count_type~.,scale = "free_y") +
      scale_x_continuous(breaks = 1:5,label = c("Raw","Merged","Filter", "Clustered","PyNAST Filter")) +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```


Count
total
unique
Raw
Merged
Filter
Clustered
PyNAST Filter
Pipeline Step