Mothur Pipeline

Nate Olson

October 26, 2016

```
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(forcats)

mothur_dir <- file.path("~/Projects/16S_etec_mix_study/analysis/pipelines/mothur/data/process")
```

Pipeline Summary

- merging paired end reads using needleman and filtering merged contigs
- aligns sequences to reference alignment - SILVA
  - find closet match in reference multiple sequence alignment using a k-mer based method then aligns to closest match using needleman
- Removes duplicates and pre.clustering to reduce the number of sequences clustered
  - Pre.cluster - pseudo-single linkage algorithm
    - Ranks sequences in order of abundance - clusters less abundant sequences within the specified edit distance from the more abundant sequences
- Chimera filtering using UCHIME
  - looks for chimeras using more abundant sequences within a sample as reference
- Classify sequences using RDP with 80% threshold
- performs average neighbor clustering - distance threshold 0.03 after splitting sequences based on taxonomy
  - level 4- Order
- Classifies OTUs based on the consensus of the sequence classifications for the sequences assigned to the OTU

Pipeline Budget

NOTE Current work does not include total number of unique in the dataset

- number of raw sequences

```
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")
```

number of successfully merged pairs

- total number and unique merged pairs passing initial filter by sample, based on length (500 bp), number of ambiguous bases (0), and maximum homopolymer length (8)

```
merged_count <- file.path(mothur_dir, "mgtst.trim.contigs.good.good.count_table") %>%
      read_tsv() %>%
      gather("id","count",-Representative_Sequence,-total) %>%  filter(count != 0) %>%
      group_by(id) %>% summarise(total = sum(count), unique = n()) %>% mutate(pipe_step = "merged")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   Representative_Sequence = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

number of sequences successfully aligned to reference alignment

```
aligned_countfile <- "mgtst.trim.contigs.good.unique.good.filter.count_table"
aligned_count <- file.path(mothur_dir, aligned_countfile) %>%
      read_tsv() %>%
      gather("id","count",-Representative_Sequence,-total) %>%  filter(count != 0) %>%
      group_by(id) %>% summarise(total = sum(count), unique = n()) %>% mutate(pipe_step = "aligned")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   Representative_Sequence = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

number of features after pre-clustering

```
precluster_countfile <- paste0("mgtst.trim.contigs.good.unique.good.filter.",
                               "unique.precluster.count_table")
precluster_count <- file.path(mothur_dir, precluster_countfile) %>%
      read_tsv() %>%
      gather("id","count",-Representative_Sequence,-total) %>%  filter(count != 0) %>%
      group_by(id) %>% summarise(total = sum(count), unique = n()) %>% mutate(pipe_step = "precluster")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   Representative_Sequence = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

number of features and sequences after chimera filtering

```
chimera_countfile <- paste0("mgtst.trim.contigs.good.unique.good.filter.",
                            "unique.precluster.denovo.uchime.pick.count_table")
chimera_count <- file.path(mothur_dir, chimera_countfile) %>%
      read_tsv() %>%
      gather("id","count",-Representative_Sequence,-total) %>%  filter(count != 0) %>%
      group_by(id) %>% summarise(total = sum(count), unique = n()) %>% mutate(pipe_step = "chimera")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   Representative_Sequence = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

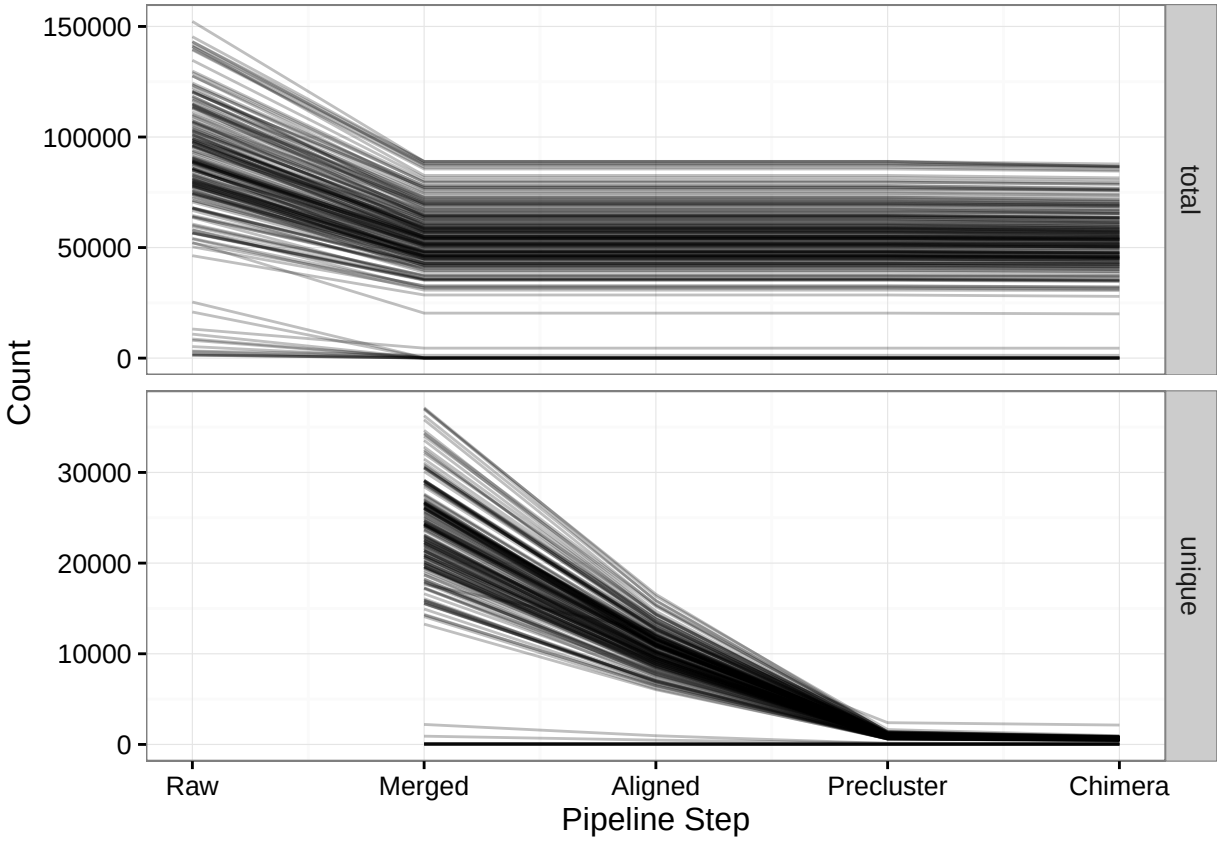
Combining count results

```
count_df <- bind_rows(raw_count, merged_count, aligned_count, precluster_count,chimera_count) %>%
      gather("count_type","value", -id, -pipe_step) %>%
      mutate(pipe_step = fct_relevel(pipe_step, c("raw","merged","aligned","precluster","chimera")),
             step_num = as.numeric(pipe_step))
```

Pipeline Processing

Few sequences filtered from the dataset after the initial quality filtering.
Sequences are either merged with others for new features with only singletons being excluded from the dataset.
Individual lines indicate the total number of seuqences and number of unique sequences or features at each stage in the pipeline.
Might want to futher characterize loss at each step as the total number and features being filtered.

```
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","Aligned","Precluster","Chimera")) +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```


150000
100000
50000
0
total
30000
20000
10000
0
unique
Count
Raw
Merged
Aligned
Precluster
Chimera
Pipeline Step