dada2_pipeline

Nate Olson

November 7, 2016

```
library(dada2)
```

```
## Loading required package: Rcpp
```

```
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

1. Quality filter read pairs, remove reads with ambiguous bases (Ns) or more than 4 expected errors, trim ends of reads with quality score of 2 or to 290 bp and 220 bp for forward and reverse reads respectively, and the first 10 bases are trimmed.
    - Number of expected errors calculated based on quality scores, `EE = sum(10^(-Q/10))`.
    - Quality score of 2 used by Illumina to indicate end of good sequencing data.
2. The forward and reverse reads are dereplicated. A consensus (average) quality score is assigned for each position.
3. Sequence inference, denoising - error correction step.
4. Merging forward and reverse read pairs. Uses global ends-free alignment and requires exact overlap for merging.
5. Chimera removal filters sequences with complementary regions matching more abundant sequences. Chimeras are identified when Needleman-Wunsch global alignments between a sequence and all more abundance sequences result in perfect matches for left and right partitions of the sequence to two different parent sequences.

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")
```

Quality Filter

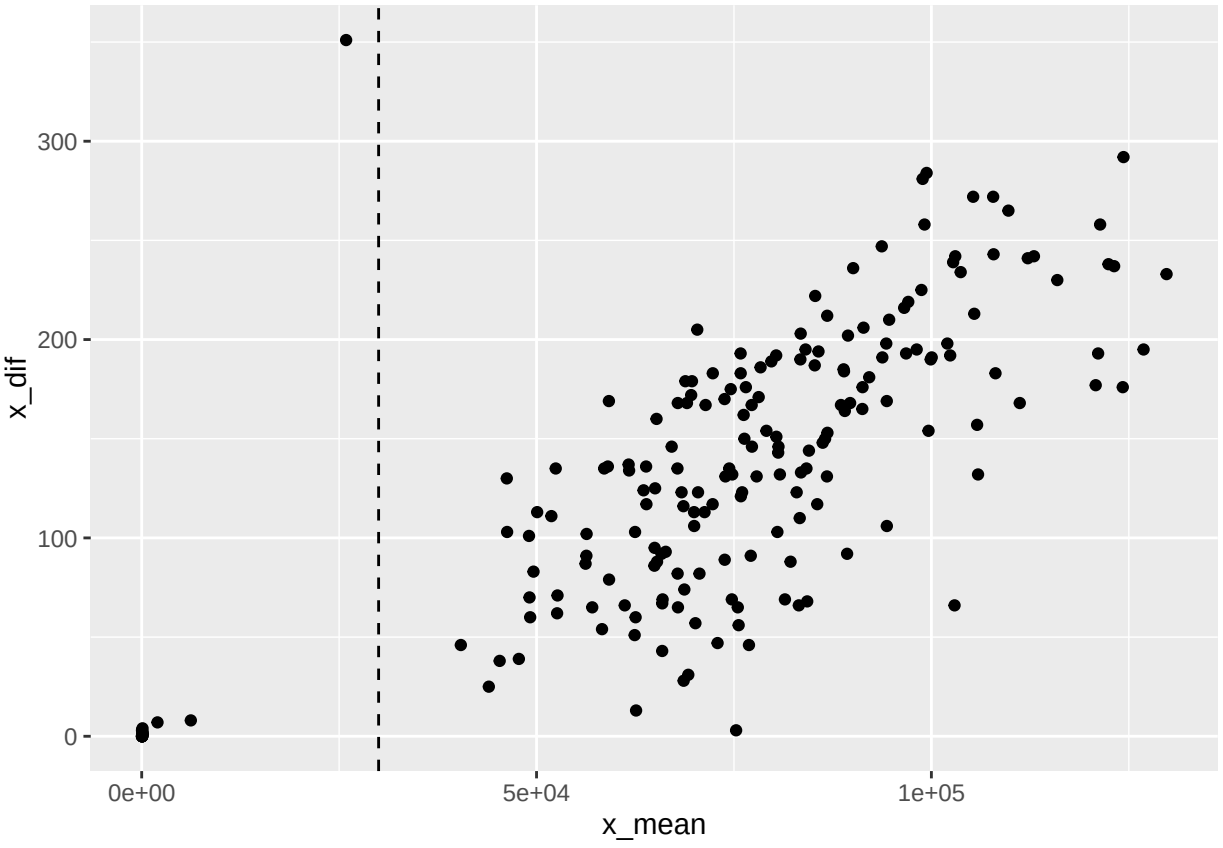
The following bash one liner was used to count the number of reads in the filtered datasets.
`zgrep -c '^@' *.fastq.gz > filter_readcount.txt`

```
filter_countfile <- paste0("~/Projects/16S_etec_mix_study/analysis/pipelines",
                           "/dada2/processed_data/filter_readcount.txt")
filter_count <- read_lines(filter_countfile) %>% tibble(id = .) %>%
      separate(id, c("id","total"),sep = ":") %>%
      mutate(read_dir = if_else(grepl("R1",id),"F","R"),
             id = str_replace(id, "_S.*",""),
             total = as.numeric(total),
             pipe_step = "filter")
```

Difference in number of forward and reverse reads passing filter
The difference in the number of forward and reverse reads passing the quality filter is correlated but less than 0.5% of the mean number of reads excluding no template control samples.

```
filter_dir_check <- filter_count %>% spread(read_dir, total) %>% mutate(x_dif = abs(`F` - R), x_mean =
```

```
filter_dir_check %>%
      ggplot(aes(x = x_mean,y = x_dif)) + geom_point() + geom_vline(xintercept = 30000, linetype = 2)
```



```
filter_dir_check %>% mutate(percent_dif = x_dif/x_mean * 100) %>% filter(x_mean > 30000) %>%
      .$percent_dif %>% summary()
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.003986 0.137500 0.182300 0.176800 0.225300 0.291400
```

Samples 1-E4 and 1-F9 are real samples, the rest of the samples with less than 30,000 sequences are no template controls.

```
filter_dir_check %>% mutate(percent_dif = x_dif/x_mean * 100) %>% filter(x_mean < 30000) %>% knitr::kabl
```

id	pipe_step	F	R	x_dif	x_mean	percent_dif
1-A12	filter	146	147	1	146.5	0.6825939
1-A6	filter	32	32	0	32.0	0.0000000
1-D12	filter	76	76	0	76.0	0.0000000
1-D6	filter	36	36	0	36.0	0.0000000
1-E4	filter	25712	26063	351	25887.5	1.3558667
1-F9	filter	2004	1997	7	2000.5	0.3499125
1-H12	filter	6225	6217	8	6221.0	0.1285967
1-H6	filter	54	53	1	53.5	1.8691589
2-A12	filter	48	48	0	48.0	0.0000000
2-A6	filter	102	106	4	104.0	3.8461538
2-D12	filter	41	44	3	42.5	7.0588235
2-D6	filter	56	56	0	56.0	0.0000000
2-H12	filter	49	49	0	49.0	0.0000000
2-H6	filter	126	128	2	127.0	1.5748031

```
## count_df from dada2 seqtab
get_count_df <- function(seqtab, pipe_step){
      tab_total <- seqtab %>% rowSums()
      tab_unique <- as.numeric(seqtab > 0) %>%  matrix(nrow = 192) %>% rowSums()

      data_frame(id = names(tab_total),
                 total = unname(tab_total),
                 unique = unname(tab_unique),
                 pipe_step = pipe_step)
}
```

Dereplicated Sequences

```
derepF_counts_file <- "derepF_counts.rds"
if(!(file.exists(derepF_counts_file))){
    derepFs <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/derepFs-20
      derepF_counts <- makeSequenceTable(derepFs)
      saveRDS(derepF_counts, derepF_counts_file)
      rm(derepFs)
}else{
      derepF_counts <- readRDS(derepF_counts_file)
}

derepF_count_df <- get_count_df(derepF_counts,"derep") %>% mutate(read_dir = "F")
rm(derepF_counts)
```

```
derepR_counts_file <- "derepR_counts.rds"
if(!(file.exists(derepR_counts_file))){
    derepRs <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/derepRs-20
      derepR_counts <- makeSequenceTable(derepRs)
      saveRDS(derepR_counts, derepR_counts_file)
      rm(derepRs)
}else{
      derepR_counts <- readRDS(derepR_counts_file)
}

derepR_count_df <- get_count_df(derepR_counts,"derep") %>% mutate(read_dir = "R")
rm(derepR_counts)
```

Denoising

```
dadaF_counts_file <- "dadaF_counts.rds"
if(!(file.exists(dadaF_counts_file))){
    dadaFs <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/dadaFs-sing
      dadaF_counts <- makeSequenceTable(dadaFs)
      saveRDS(dadaF_counts, dadaF_counts_file)
      rm(dadaFs)
}else{
      dadaF_counts <- readRDS(dadaF_counts_file)
}

dadaF_count_df <- get_count_df(dadaF_counts,"denoise") %>% mutate(read_dir = "F")
rm(dadaF_counts)
```

```
dadaR_counts_file <- "dadaR_counts.rds"
if(!(file.exists(dadaR_counts_file))){
    dadaRs <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/dadaRs-sing
      dadaR_counts <- makeSequenceTable(dadaRs)
      saveRDS(dadaR_counts, dadaR_counts_file)
      rm(dadaRs)
}else{
      dadaR_counts <- readRDS(dadaR_counts_file)
}

dadaR_count_df <- get_count_df(dadaR_counts,"denoise") %>% mutate(read_dir = "R")
rm(dadaR_counts)
```

Merging

```
merger_count_df <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/seqtab
      get_count_df("merger")
```

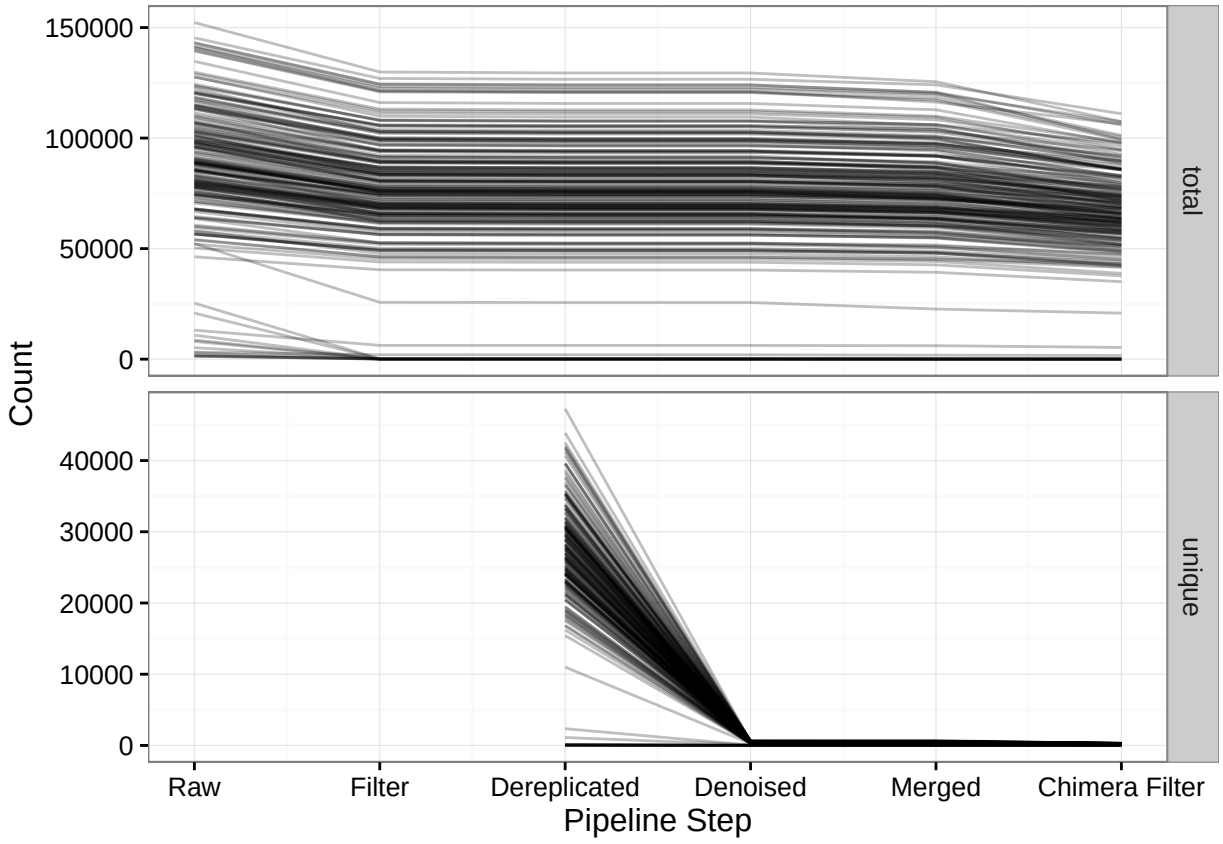
Chimera filter

```
nochimera_count_df <- readRDS("~/Projects/16S_etec_mix_study/analysis/pipelines/dada2/processed_data/se
      get_count_df("nochimera")
```

Combining Step Count Data

```
step_order <- c("raw","filter","derep","denoise","merger","nochimera")
count_df <- bind_rows(raw_count, filter_count, derepF_count_df,
                      derepR_count_df, dadaF_count_df, dadaR_count_df,
                      merger_count_df, nochimera_count_df) %>%
      gather("count_type","value", -id, -pipe_step, -read_dir) %>%
      mutate(pipe_step = fct_relevel(pipe_step, step_order),
             step_num = as.numeric(pipe_step)) %>%
      mutate(read_dir = if_else(is.na(read_dir),"M",read_dir))
```

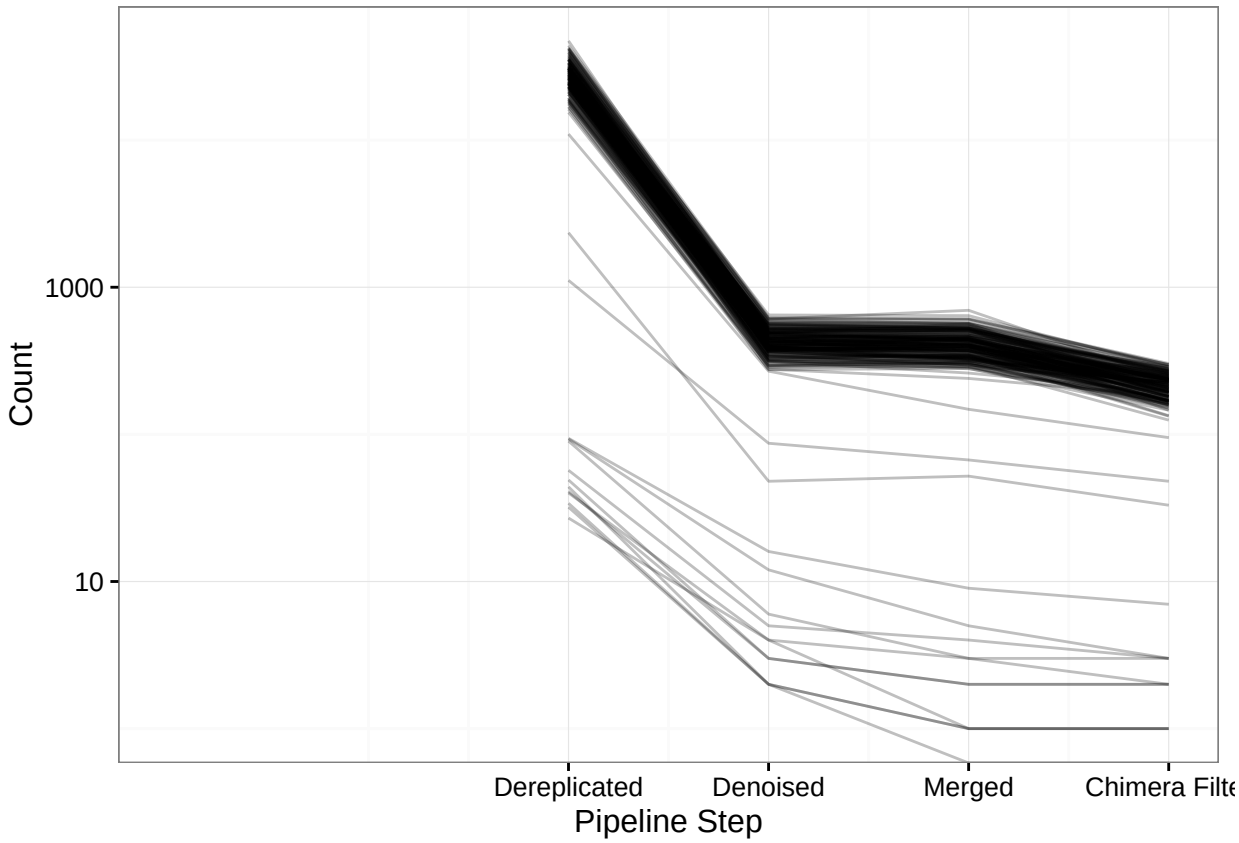
```
count_df %>% filter(read_dir != 'R') %>%
 ggplot() + geom_path(aes(x = step_num, y = value, group = id), alpha = 0.25) +
      facet_grid(count_type~.,scale = "free_y") +
      scale_x_continuous(breaks = 1:6,
                         label = c("Raw", "Filter","Dereplicated",
                                   "Denoised","Merged","Chimera Filter")) +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```




```
count_df %>% filter(read_dir != 'R', count_type == "unique") %>%
 ggplot() + geom_path(aes(x = step_num, y = value, group = id), alpha = 0.25) +
      scale_x_continuous(breaks = 3:6,
```

```
                          label = c("Dereplicated",
                                    "Denoised","Merged","Chimera Filter")) +
      scale_y_log10() +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```

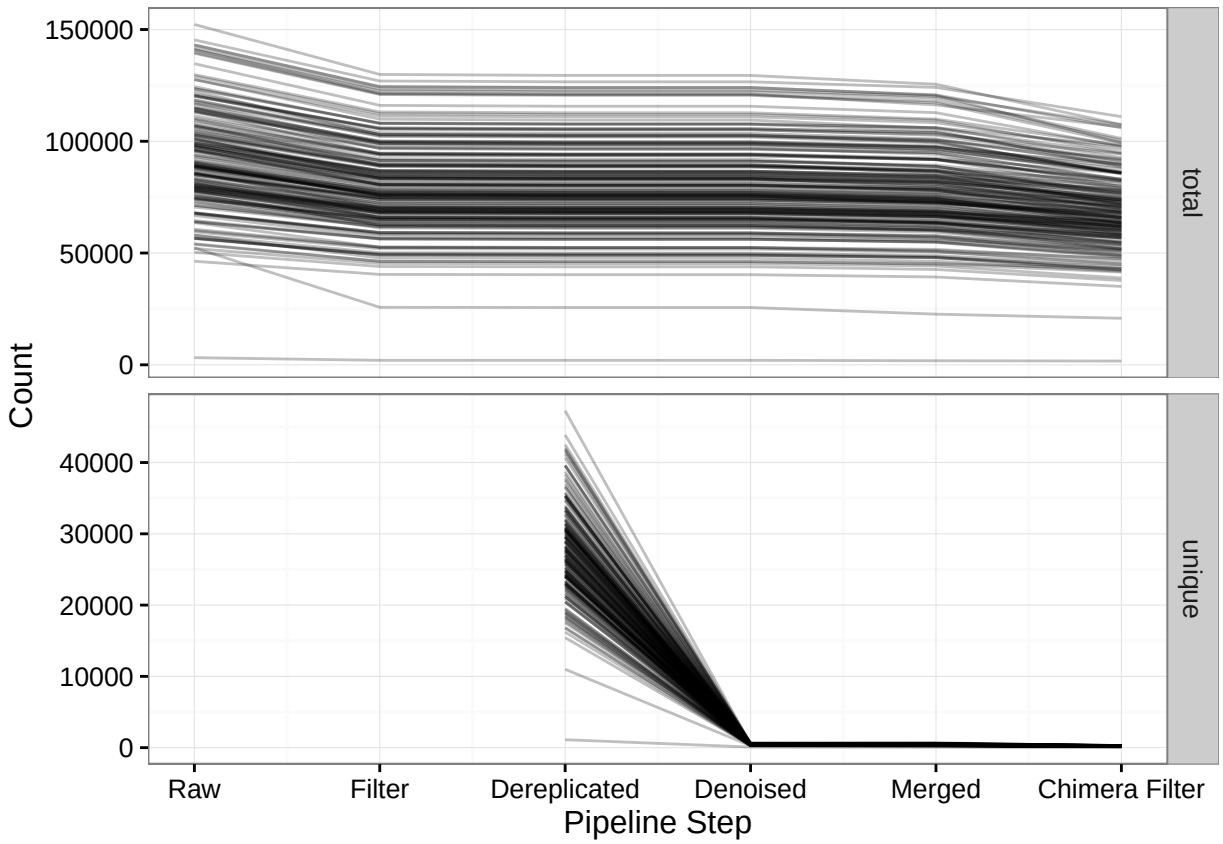
```
## Warning: Removed 384 rows containing missing values (geom_path).
```



Plots excluding no template controls

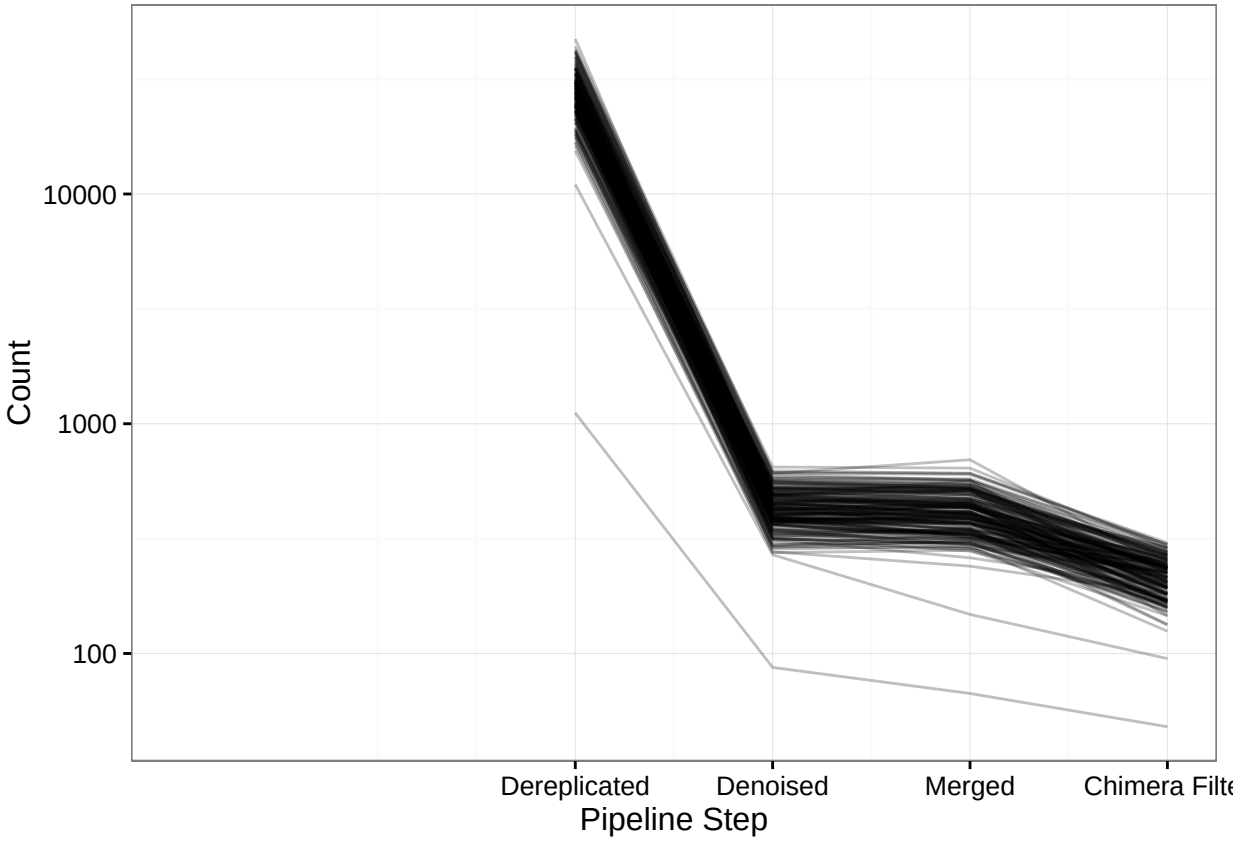
```
ntc <- paste(rep(c(1,2), each = 6),paste0(rep(c("A","D","H"),each = 2),rep(c(6,12),3)), sep = "-")
```

```
count_df %>% filter(read_dir != 'R', !(id %in% ntc)) %>%
 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 = 1:6,
                         label = c("Raw", "Filter","Dereplicated",
                                   "Denoised","Merged","Chimera Filter")) +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```




```
count_df %>% filter(read_dir != 'R', count_type == "unique", !(id %in% ntc)) %>%
 ggplot() + geom_path(aes(x = step_num, y = value, group = id), alpha = 0.25) +
      scale_x_continuous(breaks = 3:6,
                         label = c("Dereplicated",
                                   "Denoised","Merged","Chimera Filter")) +
      scale_y_log10() +
      theme_bw() + labs(x = "Pipeline Step",y = "Count")
```

```
## Warning: Removed 360 rows containing missing values (geom_path).
```


10000
1000
100
Count
Dereplicated
Denoised
Merged
Chimera Filte
Pipeline Step