BINF 8960 Capstone Project (EXAMPLE)

Jason G. Wallace

Overview

This report is an example of what you could have put together for your BINF 8960 capstone project. Everyone's report will look slightly different, so the important thing is to make sure the key components match.

This project aimed to do basic quality control and variant calling on a set of *E. coli* sequencing data following samples from a long-term evolution experiment. Details of the long-term experiment are available in Tenaillon et al. 2016.

The pipeline to carry out this analysis is available at https://github.com/jgwall/binf8960_capstone/settings.

NOTE: I will probably delete this repo before I teach class next, so if you want to check it out, do so soon.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

# Load stats and display top
stats=read.csv("results/summary_stats.csv")
head(stats)
```

```
##       sample     step   count
## 1 SRR2584863      raw 1553259
## 2 SRR2584863  trimmed 1485960
## 3 SRR2584863  aligned 2929074
## 4 SRR2584863 variants      30
## 5 SRR2584866      raw 2768398
## 6 SRR2584866  trimmed 2709672
```

Step 1 Quality control

The initial set of reads was quality-controlled using FASTQC and MultiQC to check read quality, and TRIMMOMATIC to remove adapters and cut out low-quality segments of the genome.

The TRIMMOMATIC trim parameter were: - ILLUMINACLIP:data/raw_fastq/NexteraPE-PE.fa:2:30:10:5:True - SLIDINGWINDOW:4:20

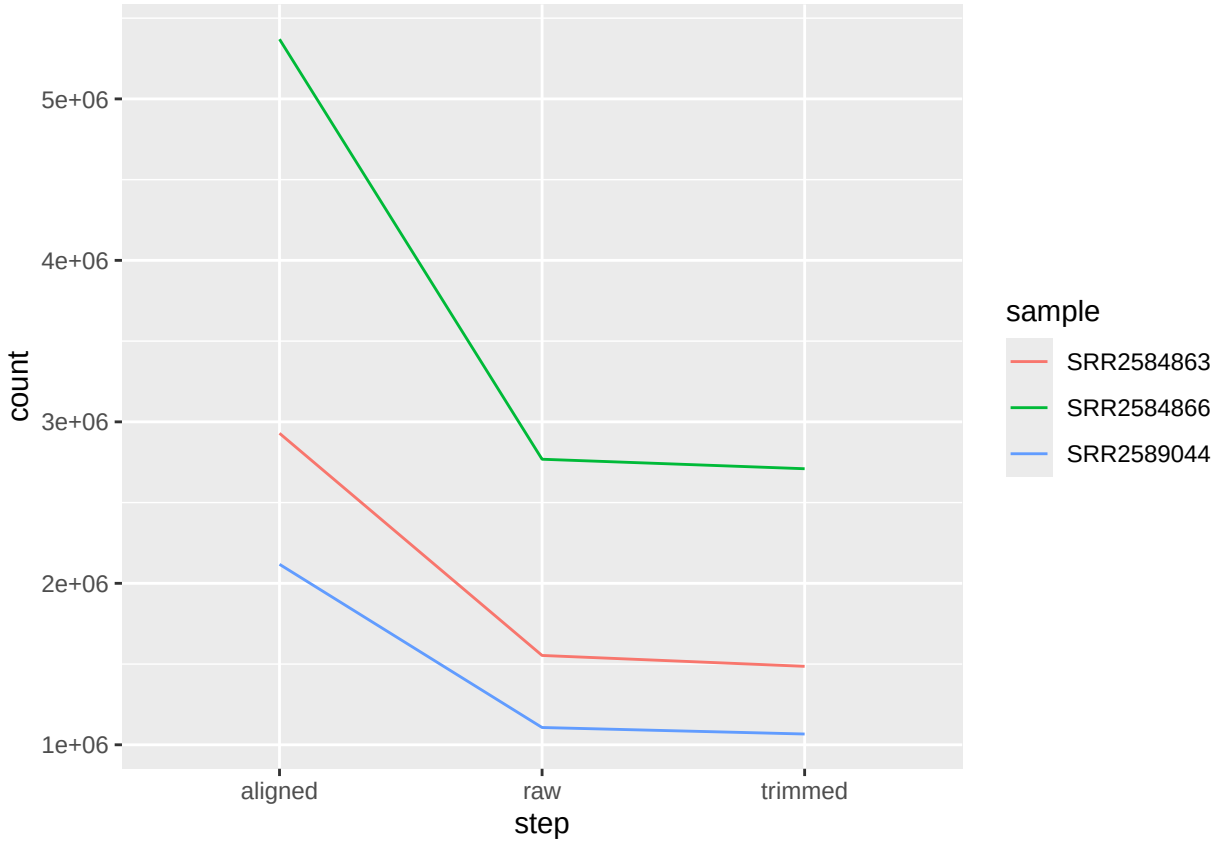
(The first part of ILLUMINACLIP is the path to a FASTA file with the Illumina adapter sequences.)

Step 2 - Alignment

After trimming, reads were aligned to the E. coli REL606 genome (NCBI Accession CP000819.1) using BWA-MEM with the default settings.

The below graph shows the number of reads retained at each step (raw, trimmed, and aligned):

```
# Filter stats for just the ones I want
reads = stats %>%
  filter(step %in% c("raw", "trimmed", "aligned"))
ggplot(reads) +
  aes(x=step, y=count, color=sample, group=sample) +
  geom_line()
```



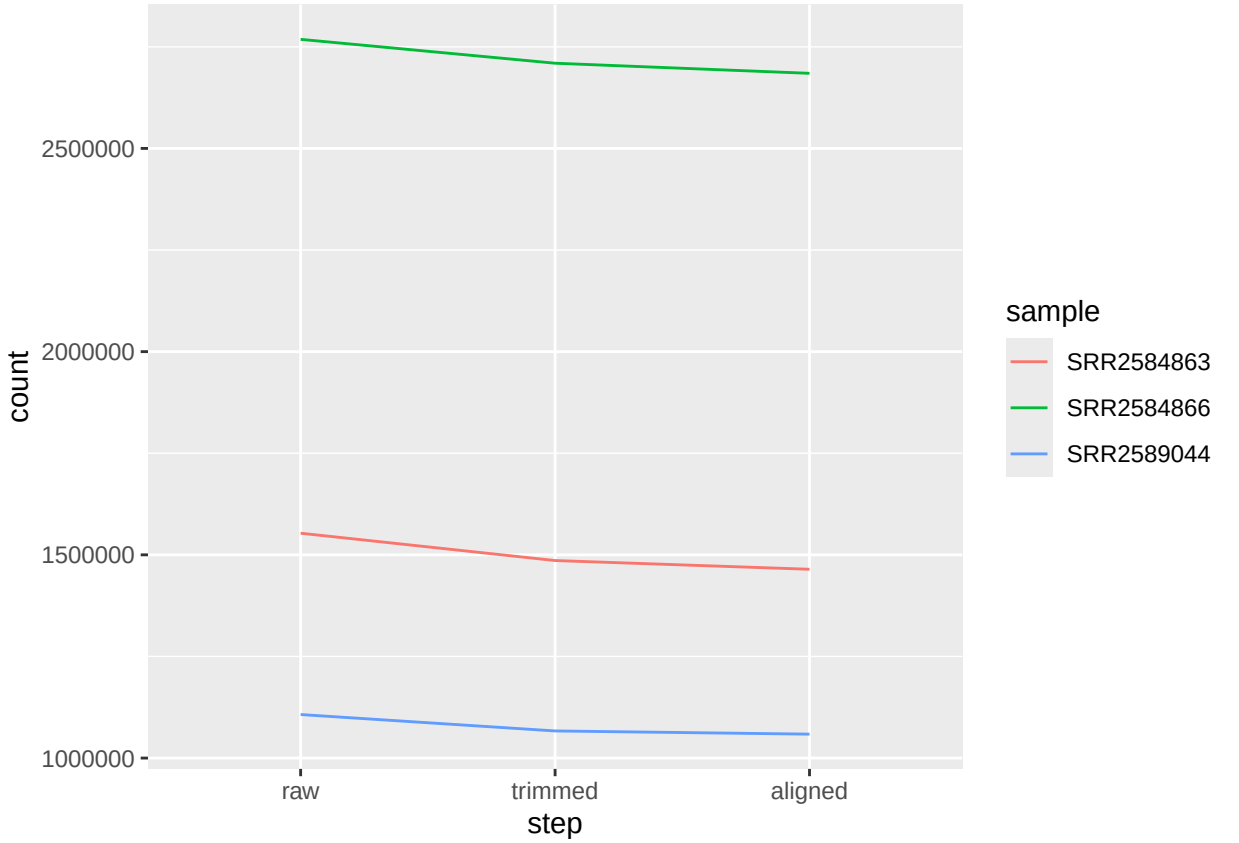
Step 2a (optional) - Improved graph

The above graph is clear, but would be better putting the X values in a different order and fixing the fact that when I counted reads, I only counted forward ones for raw and trimmed but all for aligned. Changing the order uses factors (which we didn't cover in class), while adjusting the counts requires some boolean selection:

```
# Change order
fix_reads = reads %>%
  mutate(step = factor(step, levels=c("raw", "trimmed", "aligned")))
```

```
# Fix aligned reads by selecting only them and dividing by 2
is_alignment = fix_reads$step=="aligned"
fix_reads$count[is_alignment] = fix_reads$count[is_alignment]/2

# Plot better graph
ggplot(fix_reads) +
  aes(x=step, y=count, color=sample, group=sample) +
  geom_line()
```



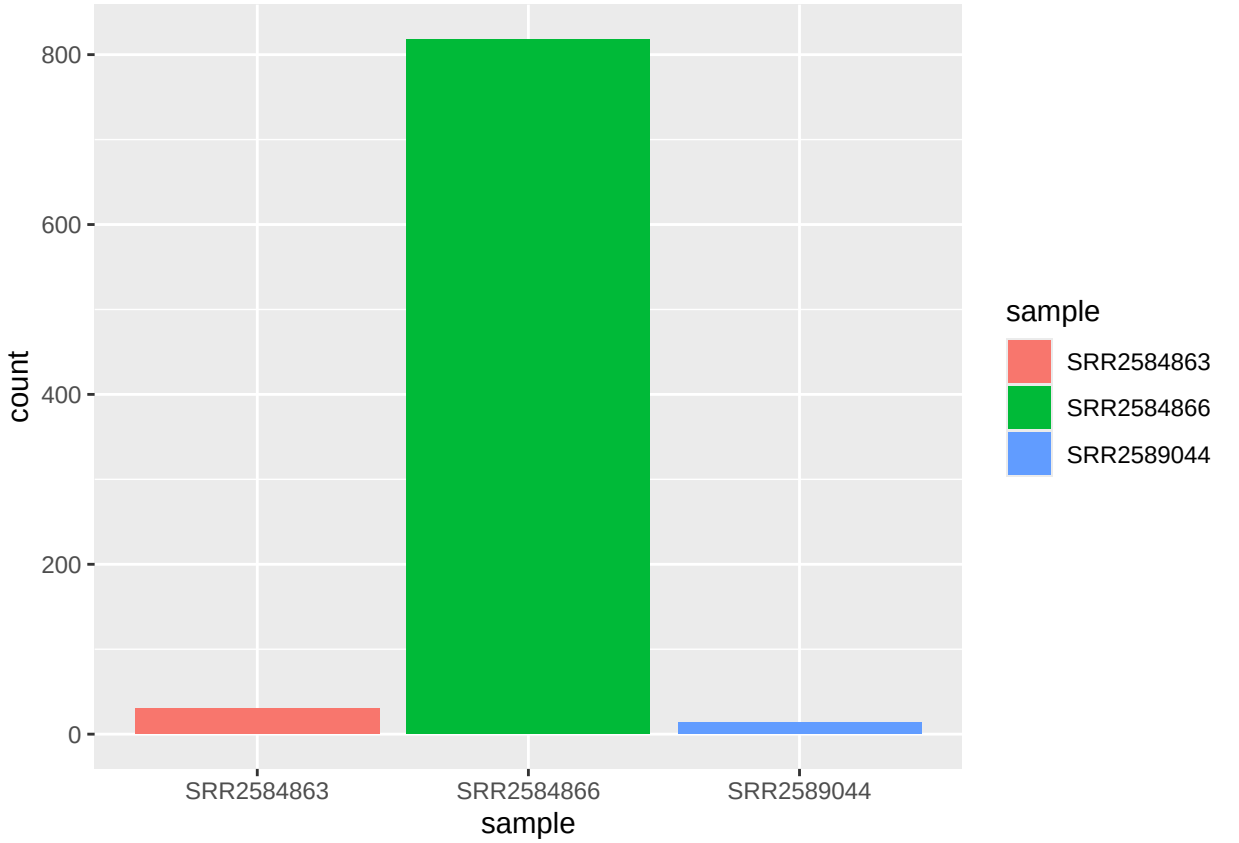
Now this looks more like what we would expect: A slight loss of reads at each step, but most being retained.

Step 3 - Variant Calling

Finally, variants were called using BCFtools, first by running the "mpileup" command (default options) and then the "call" command (options "–ploidy 1" and "-m" since *E. coli* is haploid and we wanted multi-alleleic calling).

This graph shows how many variants were identified in each sample:

```
# Pull out just the variants
vars = stats %>%
  filter(step == "variants")
ggplot(vars) +
  aes(x=sample, y=count, fill=sample) +
  geom_col()
```



From this we see that the middle sample (SRR2584866) has many, many more variants than the other two, probably because it comes from a later generation that has accumulated many more mutations. (We would need to check against our metadata file to see if that is actually the case)

And that's all your project required. This is actually a minimal report; for a real project, I would put more things in here, like parts of the MultiQC output, and then subsequent steps like filtering the variants and doing quality control on them. Good luck!