Results Section: Supplementary subsampling

Read In The Data

```
library(ggplot2)
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
```

```
options(scipen=999)

make_plot <- function (df, ylab) {
    p <- ggplot(data=df, aes(x=x, y=y)) +
        ylab(ylab) +
        xlab("Coverage") +
        geom_point(aes(color=color)) +
        scale_x_continuous(breaks = seq(min(df$x), max(df$x), by = 20)) +
        theme_bw() +
        theme(axis.text=element_text(size=12),
              axis.title=element_text(size=14,face="bold"))
    return(p)
}

results <- read.table(
    "../data/supplementary-subsample/subsample-summary.txt",
    header = TRUE,
    sep = "\t"
)
results <- results[results$mutations < 10000 & results$simulation != 'EF',]
colnames(results)
```

```
##  [1] "sample"               "runtime"              "price"
##  [4] "mutations"            "simulation"           "coverage"
##  [7] "total_contig_200bp"   "total_gene"           "total_contig"
## [10] "total_contig_length"  "max_contig_length"    "mean_contig_length"
## [13] "median_contig_length" "min_contig_length"    "n50_contig_length"
## [16] "coverage_cleanup"     "coverage_original"    "total_kmer"
## [19] "total_singleton"      "total_variant"        "total_snps"
## [22] "total_indel"
```

Overview

We wanted to determine if at what level of coverage diminishing returns were observed in our analysis. This is important because high coverage sequences require more compuational resources (mostly in the form of memory) and take longer to process. Because we used Cancer Genomics Cloud (http://www.cancergenomicscloud.org/) to process each project, this was also important to reduce overall costs. We used runtime, assembly metrics and singleton kmer counts to determine a cutoff for coverage at which further coverage did not improve the results.

Simulating Sequencing

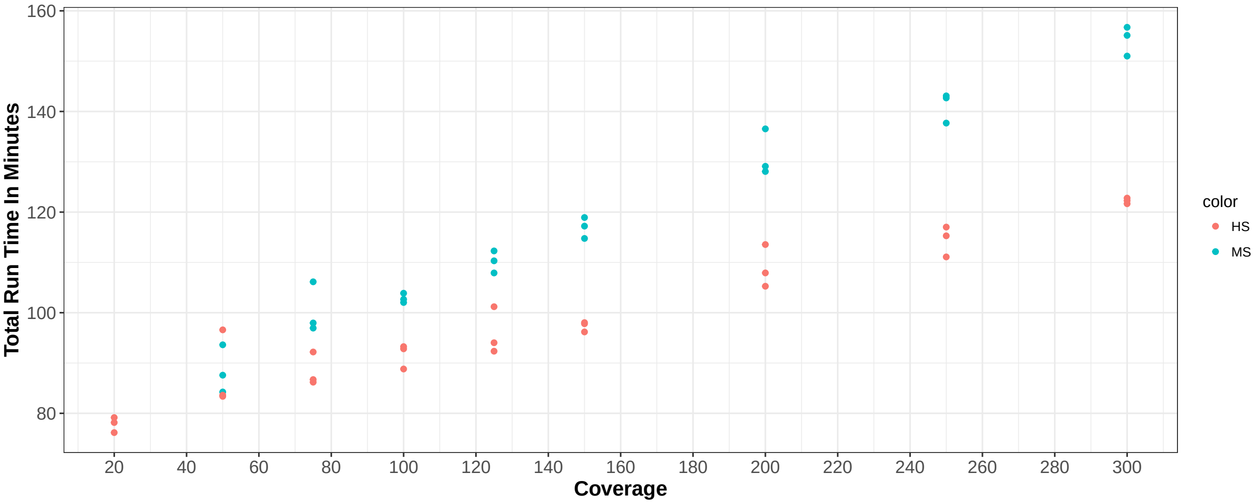
We simulated HiSeq and MiSeq sequencing of the *S. aureus* N315 (NC_00274) reference genome with ART (Huang, W., Li, L., Myers, J.R., Marth, G.T., 2012. ART: a next-generation sequencing read simulator. Bioinformatics 28, 593–594.). Multiple coverages were simulated (see below) and processed through the Staphopia analysis pipeline on CGC.

We simulated multiple coverages:

```r
sort(unique(results$coverage))
```

```
## [1]  20  50  75 100 125 150 200 250 300
```

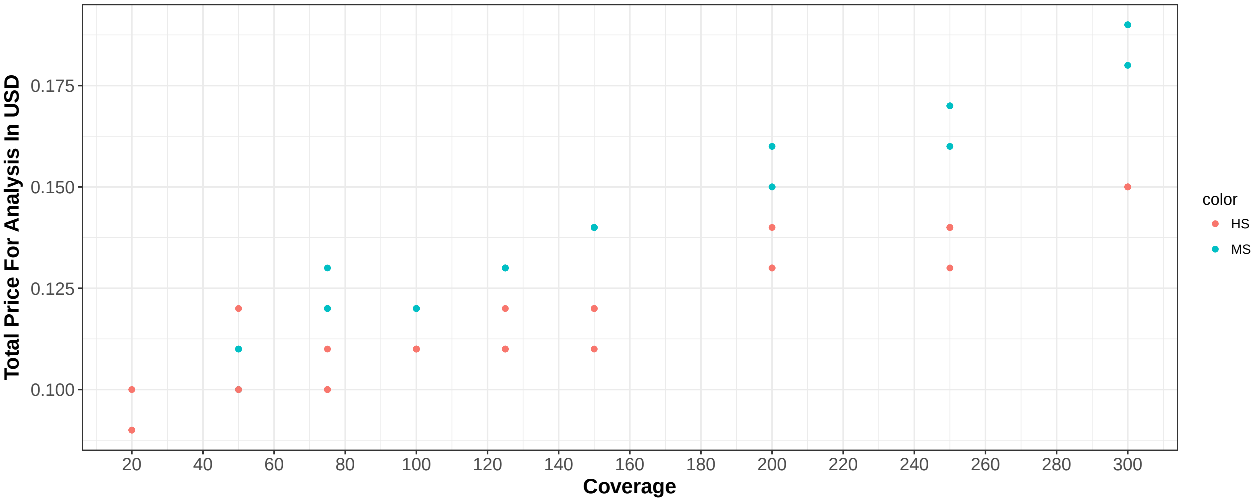
How does coverage affect run time?



In the plot above, there is evidence that increasing coverage leads to longer runtimes. Based on simulations MiSeq (MS) seqeuncing tended to take longer to process than HiSeq (HS).

How does coverage affect costs?

```r
p <- make_plot(
    data.frame(x=results$coverage, y=results$price, color=results$simulation),
    "Total Price For Analysis In USD"
)
p
```

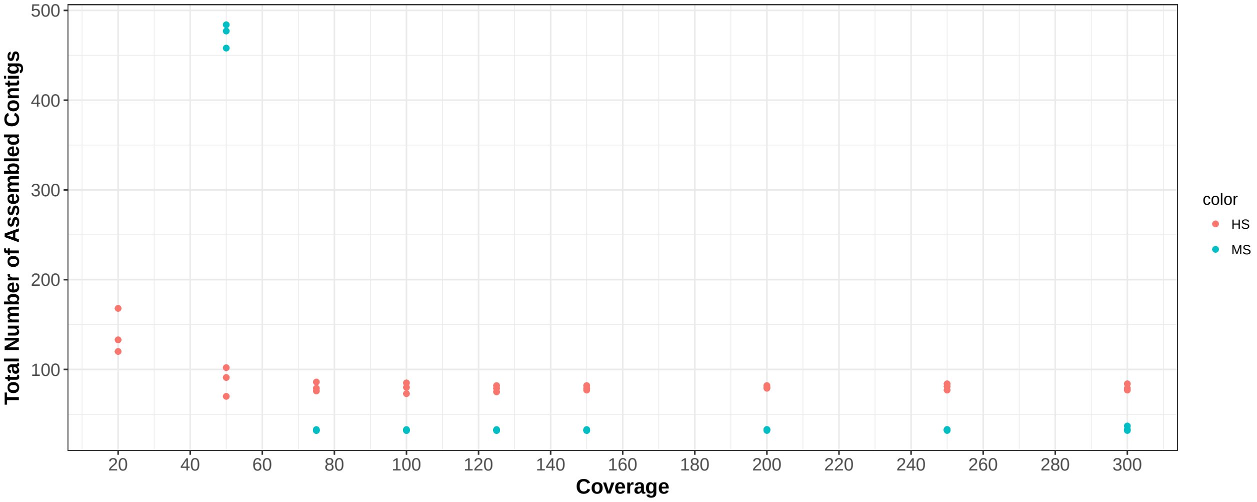


The job cost on CGC is dependent on overall runtime. In the plot above, there is not much of a difference between 75x and 125x (~$0.125), but at 300x (~$0.175) it is about a $0.05 difference. For 44,000 genomes, it costs ~$5,500 to process genomes at a 75-125x coverage cutoff and ~$7,700 to process genomes at a 300x coverage cutoff. This is roughly a $2,200 difference in price.

How does coverage affect assembly?

Total number of contigs

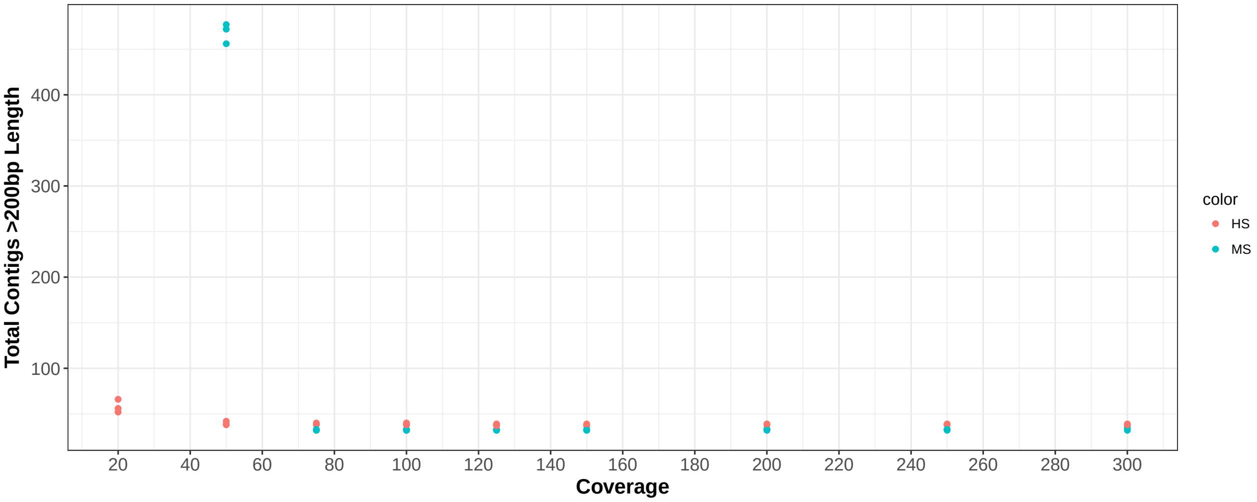
```
p <- make_plot(
    data.frame(x=results$coverage, y=results$total_contig, color=results$simulation),
    "Total Number of Assembled Contigs"
)
p
```



In the plot above, at 20x and 50x coverages there are more contigs that at >75x coverage, suggesting these coverages may not produce the best assembly. At 75x coverage and onwards, the total number of contigs does not change much. At >200x, there looks like a slight increase in total contigs.

Total number of contigs greater than 200bp

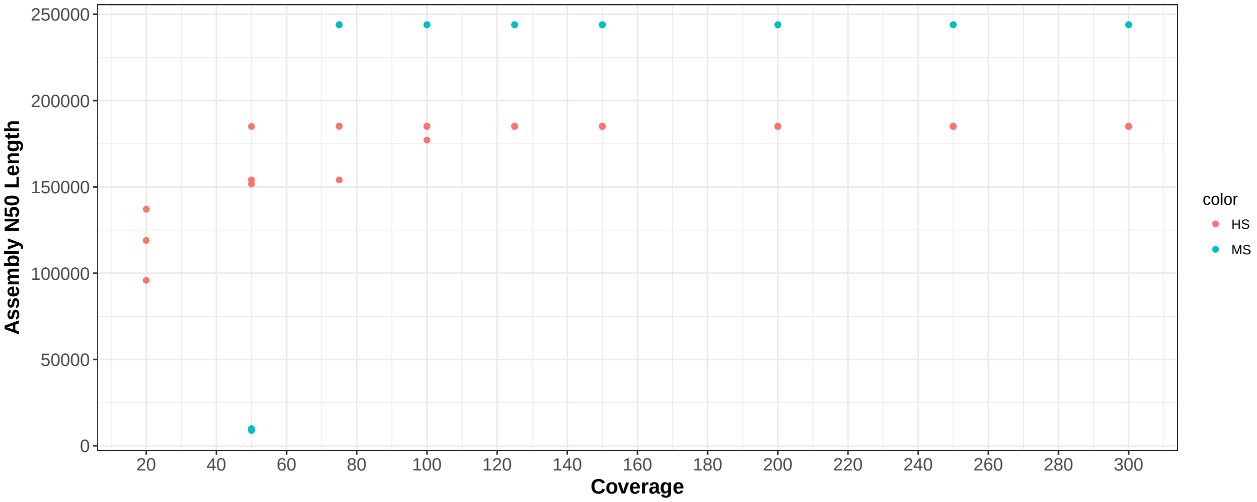
```r
p <- make_plot(
    data.frame(x=results$coverage, y=results$total_contig_200bp, color=results$simulation),
    "Total Contigs >200bp Length"
)
p
```



Looking at the total number of contigs greater than 200bp, a similar pattern is oberseved. Again 20x and 50x may not produce the best assembly, and 75x coverage an onwards produce similar numbers.

N50 contig length

```r
p <- make_plot(
    data.frame(x=results$coverage, y=results$n50_contig_length, color=results$simulation),
    "Assembly N50 Length"
)
p
```

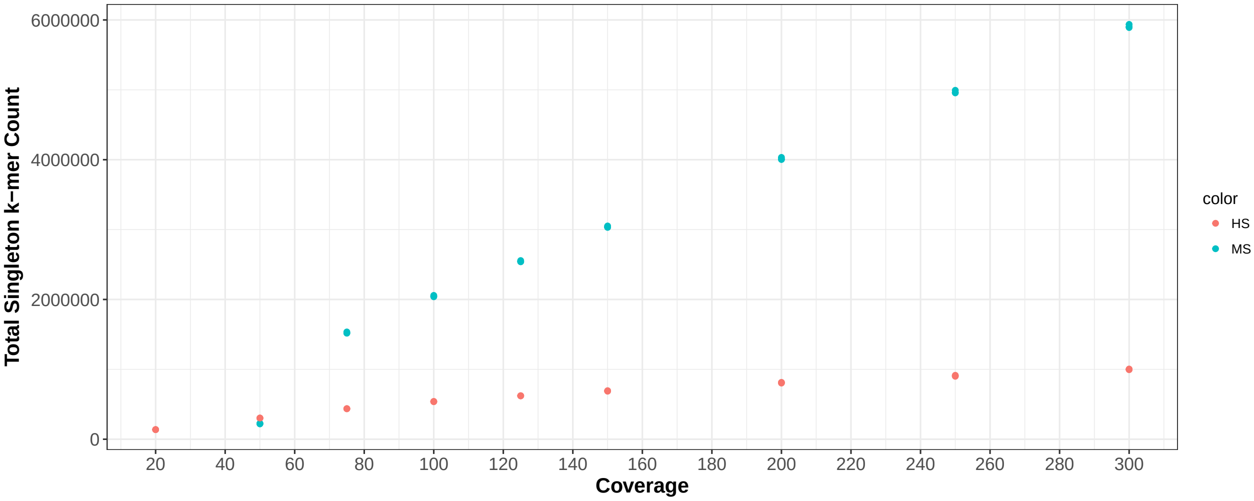


Again, similar to the above to plots. THe exception is the N50 appears to level off around 100x instead of

75x.

How does coverage affect total number of singleton kmers?

```
p <- make_plot(
    data.frame(x=results$coverage, y=results$total_singleton, color=results$simulation),
    "Total Singleton k-mer Count"
)
p
```



In the plot above, it appears that further sequencing depth increases the number of observed singleton kmers. While there may be true singletons in the sequencing, many of these can be assumed to be due to sequencing errors. This suggests that at higher seqeuncing depths, there is greater need to correct erroneous reads that can affect analysis results. The effect is much greater in MiSeq than HiSeq, most likely due to the difference in error profiles used by ART.

Conclusions

Overall using the metrics described above it appears coverage cutoff can be used without affecting the results of an analysis. Although, samples with 20x and 50x coverage had the lowest runtimes they produced assemblies that could be improved by further coverage. For samples with 150x or greater coverage produced assemblies similar to 75x-125x coverage, but took longer to process (increased costs) and also had more singleton kmers. This leaves 75x, 100x, and 125x coverages that were each very similar in runtime, costs, assemblies and kmers. Out of these three coverages, we arbitrary selected 100x as the default coverage cutoff for Staphopia analysis.