

# PopPsiSeq Dev1

Charlie Soeder

11/14/2018

16 Nov 2018

experimental data:

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

Table 1: Sequenced Experimental Samples

name	paired	experimental	source
SRR5860570	TRUE	control	NCBI
SRR303333	FALSE	selection	EarlyJones2011
10A	TRUE	selection	EarlyJones2013

Population-wide sample count by species:

Table 2: Number of Sequenced Samples per Species

species	sample_count
drosophila sechellia	3
drosophila simulans	3

load & discuss FASTP summary

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_character(),
##   X4 = col_double()
## )
```

prefilt:

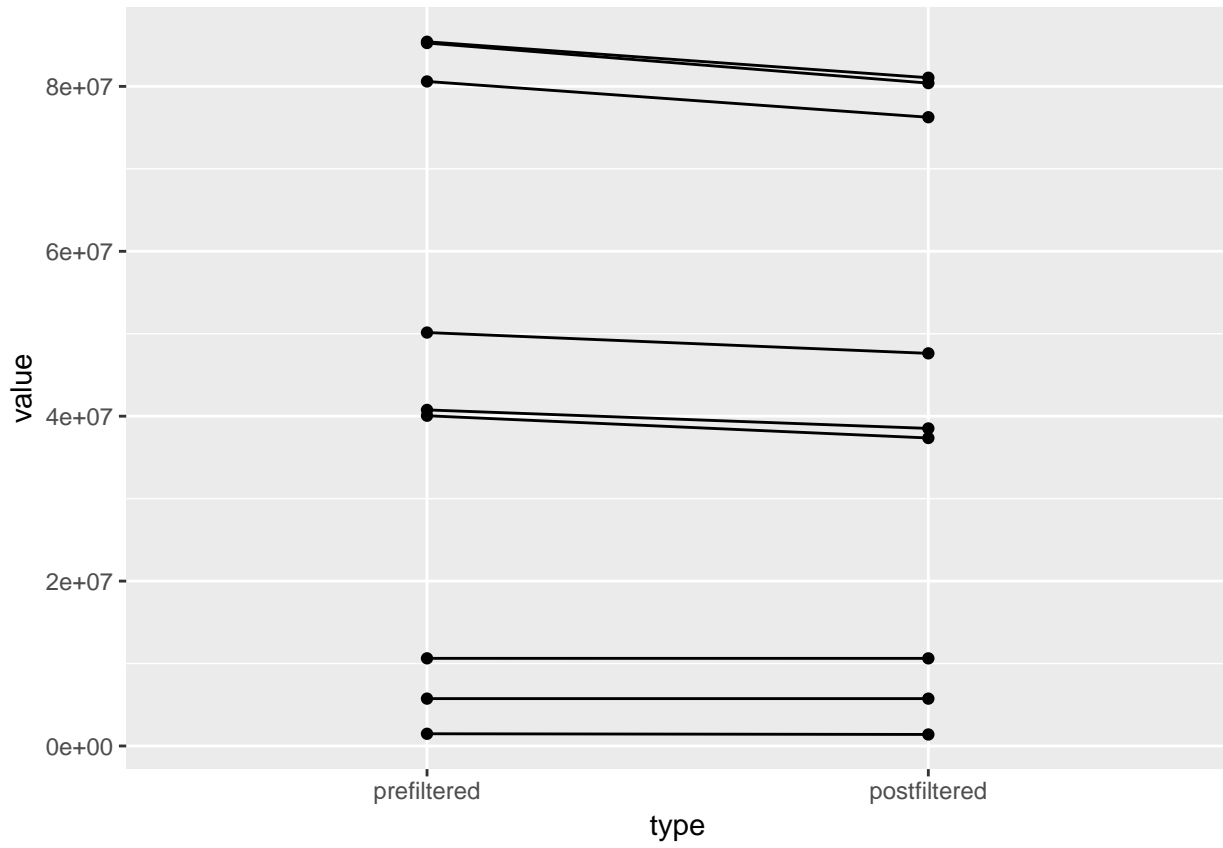
```
## Warning: Column `name` joining factors with different levels, coercing to
## character vector
```

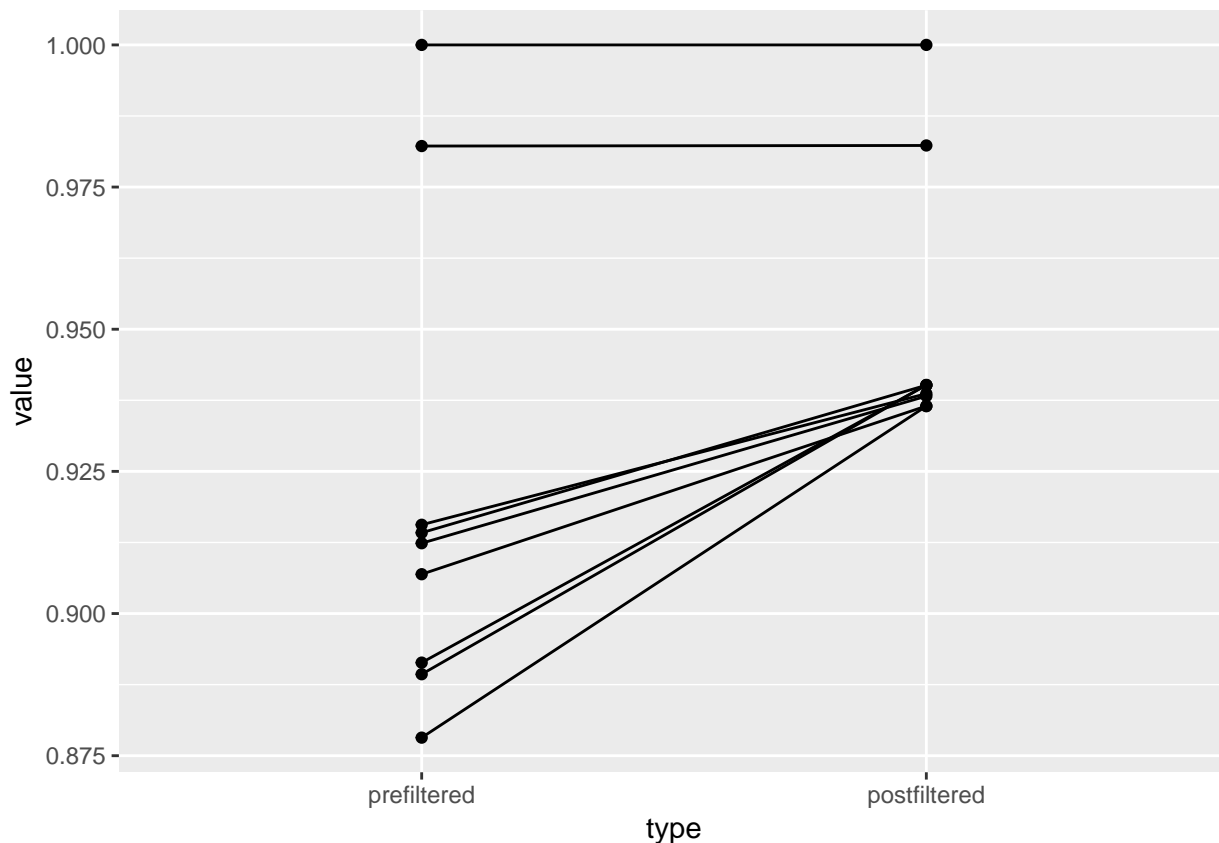
```
## # A tibble: 1 x 3
##   minimum average maximum
##   <dbl>     <dbl>   <dbl>
## 1 1481482 44455123. 85417202
```

```
## Warning: Column `name` joining factors with different levels, coercing to
## character vector
```

type	minimum	average	maximum
prefiltered	1.481482e+06	4.445512e+07	85417202
postfiltered	1.397152e+06	4.210807e+07	81052256

type	minimum	average	maximum
percent retention	9.326349e+01	9.564348e+01	100





```
## Warning: Column `name` joining factors with different levels, coercing to
## character vector
```

19 Nov 2018

load and discuss bam summary

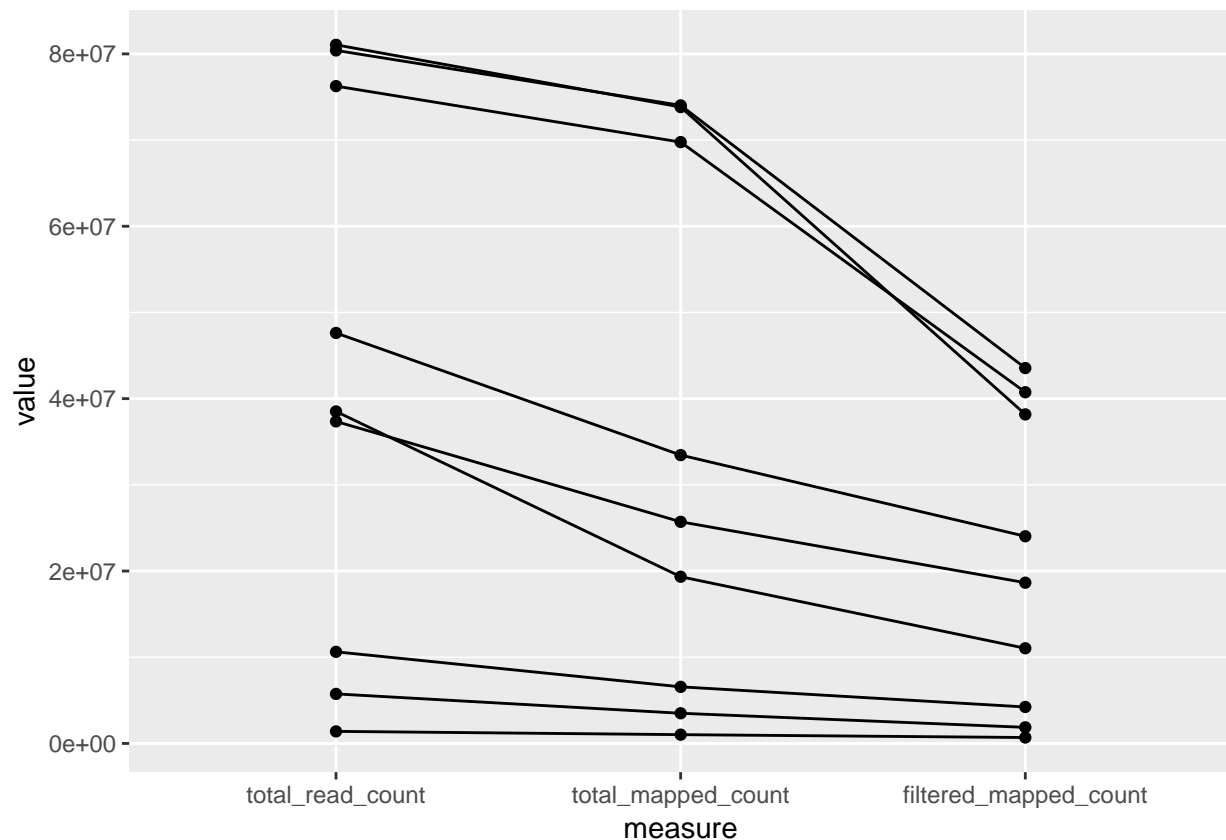
depth of coverage is effed???

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_double()
## )
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_double()
## )

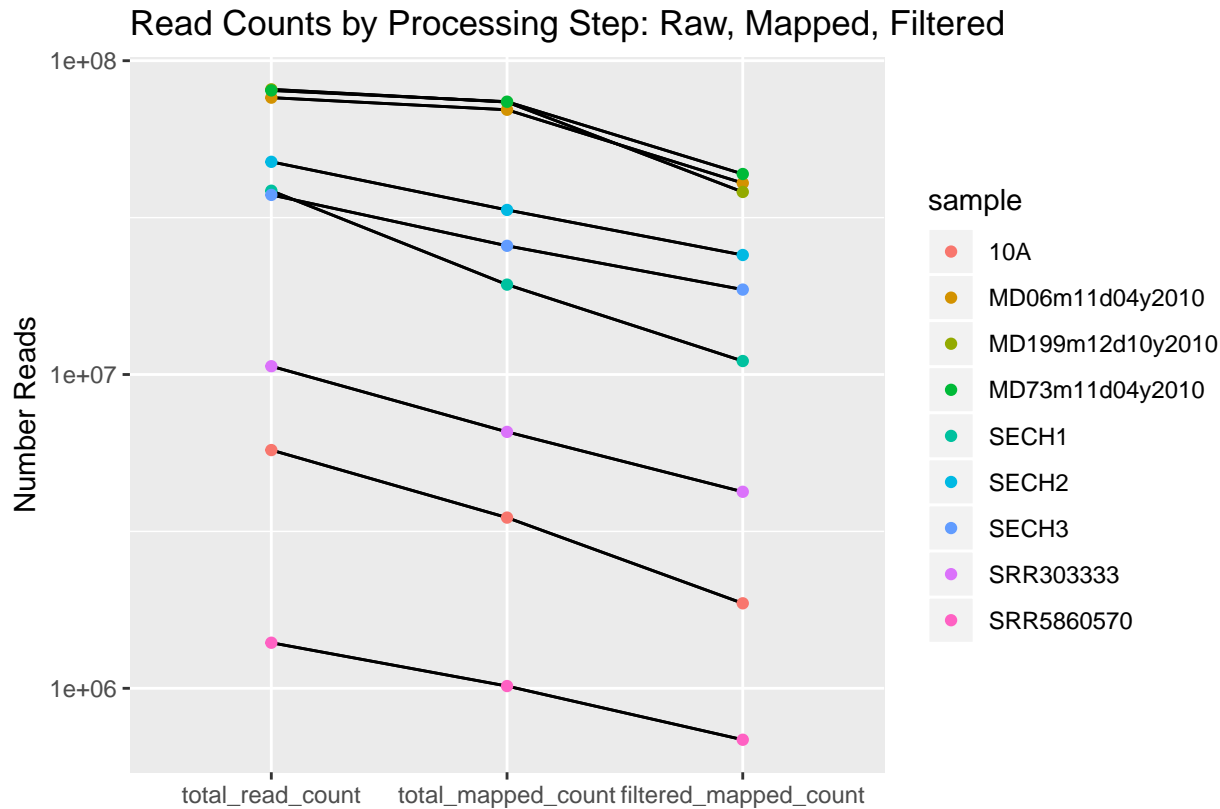
## # A tibble: 18 x 4
##   sample      measure      value aligner
##   <fct>      <fct>      <dbl> <fct>
## 1 10A        total_read_count  5743832 bwa
## 2 10A        total_mapped_count 3499415 bwa
## 3 MD06m11d04y2010 total_read_count  76259772 bwa
```

```
## 4 MD06m11d04y2010 total_mapped_count 69765684 bwa
## 5 MD199m12d10y2010 total_read_count 81052256 bwa
## 6 MD199m12d10y2010 total_mapped_count 73828982 bwa
## 7 MD73m11d04y2010 total_read_count 80400246 bwa
## 8 MD73m11d04y2010 total_mapped_count 74027424 bwa
## 9 SECH1 total_read_count 38516580 bwa
## 10 SECH1 total_mapped_count 19340203 bwa
## 11 SECH2 total_read_count 47620576 bwa
## 12 SECH2 total_mapped_count 33453423 bwa
## 13 SECH3 total_read_count 37356234 bwa
## 14 SECH3 total_mapped_count 25711919 bwa
## 15 SRR303333 total_read_count 10625978 bwa
## 16 SRR303333 total_mapped_count 6563682 bwa
## 17 SRR5860570 total_read_count 1397152 bwa
## 18 SRR5860570 total_mapped_count 1016591 bwa

## # A tibble: 9 x 4
## sample measure value aligner
## <fct> <chr> <dbl> <fct>
## 1 10A filtered_mapped_count 1865642 bwaUniq
## 2 MD06m11d04y2010 filtered_mapped_count 40746133 bwaUniq
## 3 MD199m12d10y2010 filtered_mapped_count 38171055 bwaUniq
## 4 MD73m11d04y2010 filtered_mapped_count 43547846 bwaUniq
## 5 SECH1 filtered_mapped_count 11038965 bwaUniq
## 6 SECH2 filtered_mapped_count 24033588 bwaUniq
## 7 SECH3 filtered_mapped_count 18649404 bwaUniq
## 8 SRR303333 filtered_mapped_count 4229353 bwaUniq
## 9 SRR5860570 filtered_mapped_count 685616 bwaUniq
```



```
## # A tibble: 3 x 5
##   measure      minimum  average  median  maximum
##   <chr>      <dbl>    <dbl>   <dbl>   <dbl>
## 1 filtered_mapped_count 685616 20329734. 18649404 43547846
## 2 total_mapped_count  1016591 34134147 25711919 74027424
## 3 total_read_count    1397152 42108070. 38516580 81052256
```



20 Nov 2018

Depth of coverage:

Table 4: Average Depth of Coverage for Raw and Filtered Alignments

step	minimum	average	median	maximum
pre-filtration depth	0.8886830	18.2754737	22.8901000	38.1817000
post-filtration depth	0.6161020	11.5527824	15.4427000	21.3541000
depth retention	0.5332057	0.6371191	0.6142139	0.7491623

```
## Warning: Column `sample`/`name` joining factors with different levels,
## coercing to character vector

## # A tibble: 3 x 6
##   species      step      minimum average median maximum
##   <fct>      <chr>      <dbl>    <dbl>  <dbl>   <dbl>
## 1 drosophila sechellia pre-filtration depth 16.2     22.6  22.9    28.8
## 2 drosophila simulans pre-filtration depth 25.8     30.5  27.4    38.2
```

```
## 3 <NA>                pre-filtration depth  0.889    1.73    2.07    2.23
## Warning: Column `sample`/`name` joining factors with different levels,
## coercing to character vector
```

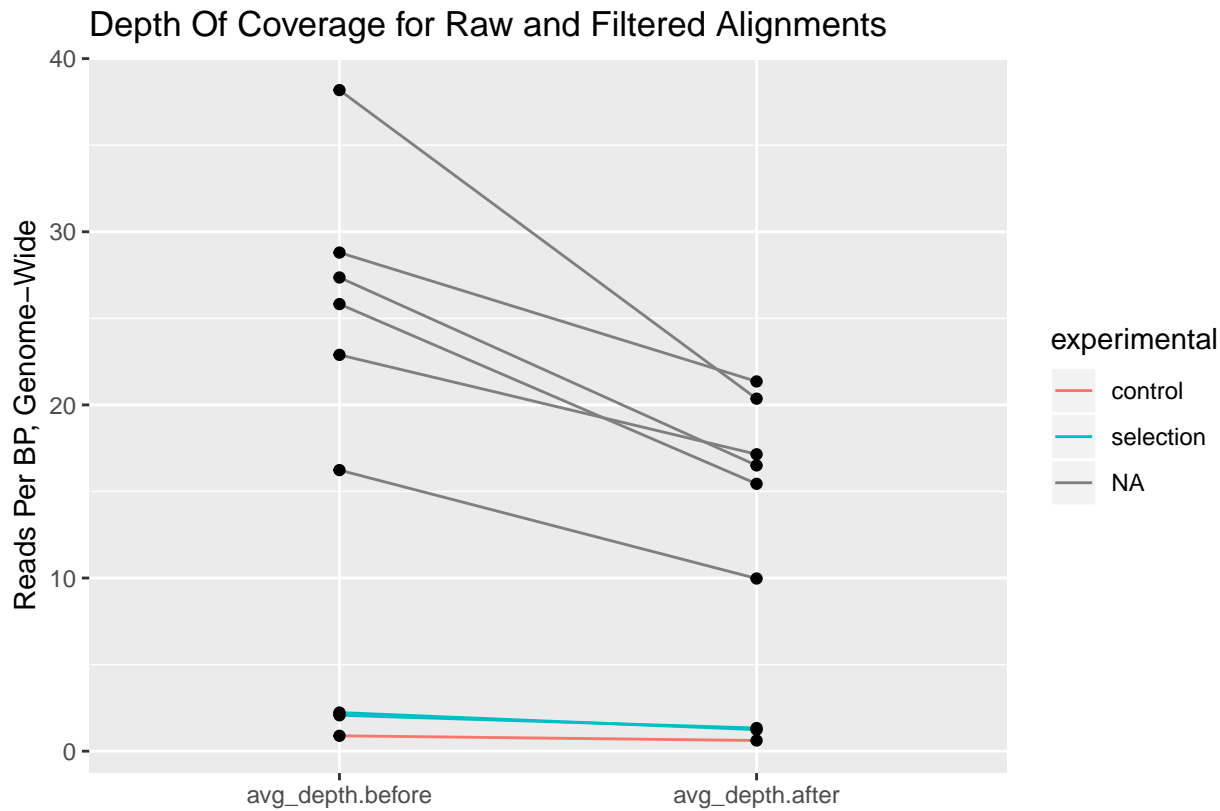
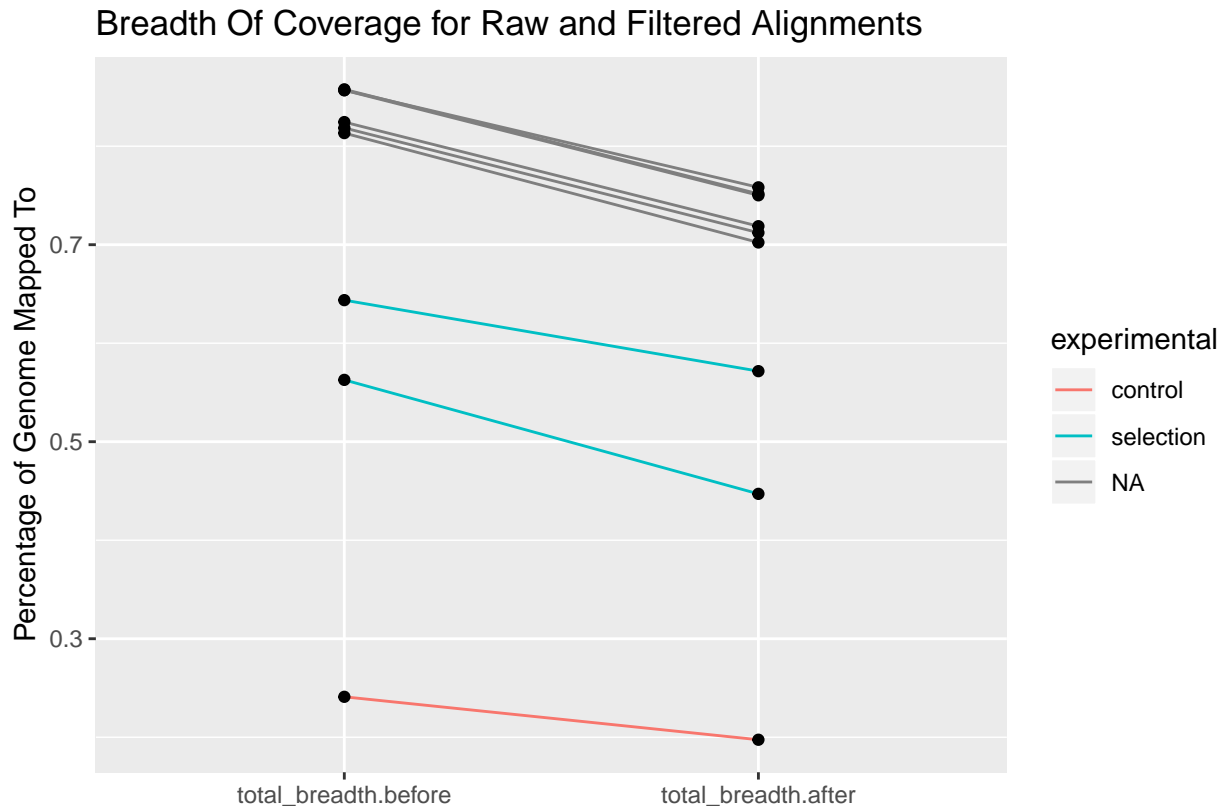


Table 5: Breadth of Coverage Statistics for Raw and Filtered Alignments

step	minimum	average	median	maximum
pre-filtration breadth	24.1	71.9	81.9	85.8
post-filtration breadth	19.7	62.3	71.2	75.8
breadth retention	79.4	86.0	87.2	88.8



27 Nov 2018

better kable-tables with `prettyNum()` and `sitools::f2si`

<https://stackoverflow.com/questions/3245862/format-numbers-to-significant-figures-nicely-in-r>

sitools: <https://stackoverflow.com/questions/11340444/is-there-an-r-function-to-format-number-using-unit-prefix>

Table 6: Read Counts by Sample

type	minimum	average	maximum
prefiltered	1.48 M	44.5 M	85.4 M
postfiltered	1.4 M	42.1 M	81.1 M
percent retention	93.3	95.6	100

Table 7: Breadth of Coverage Statistics for Raw and Filtered Alignments

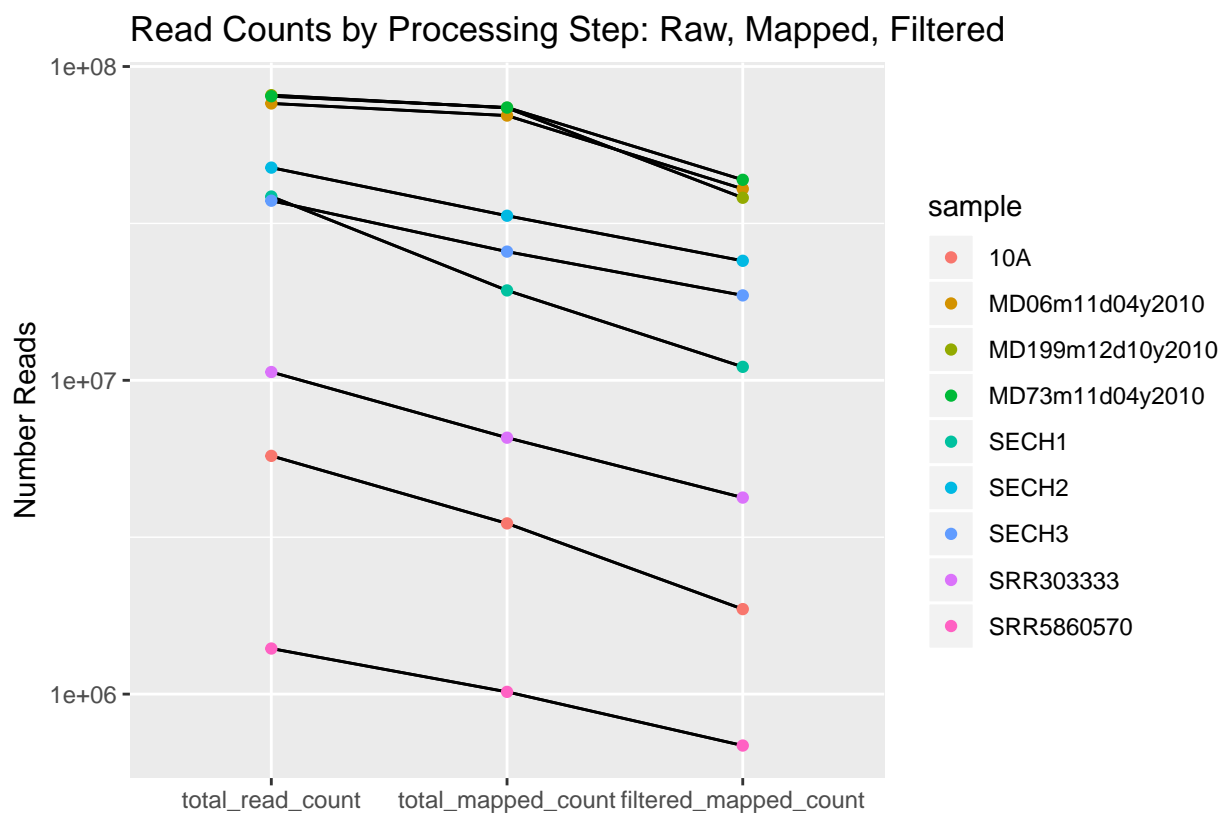
step	minimum	average	median	maximum
pre-filtration breadth	24.1	71.9	81.8532	85.8
post-filtration breadth	19.7	62.3	71.2195	75.8
breadth retention	79.4	86	87.1791	88.8

Also, need to add panels by reference genome.

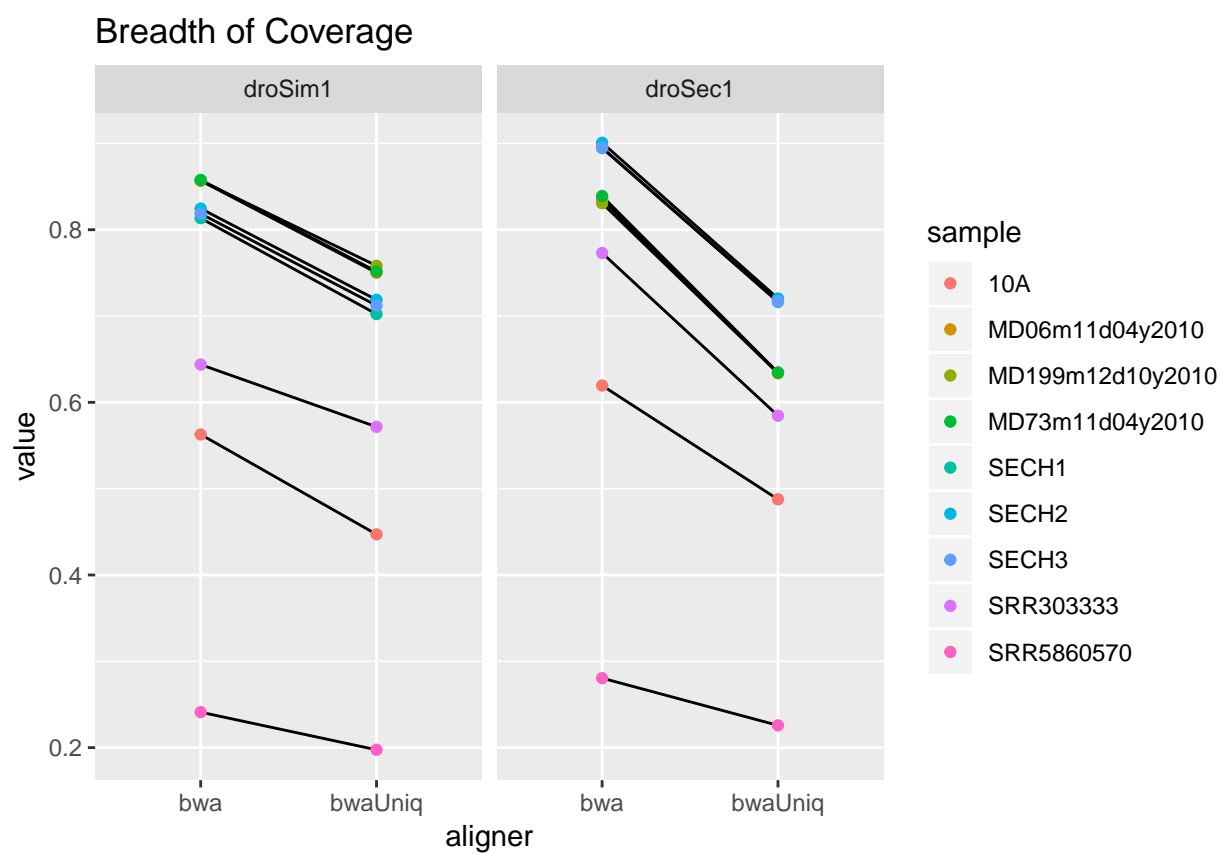
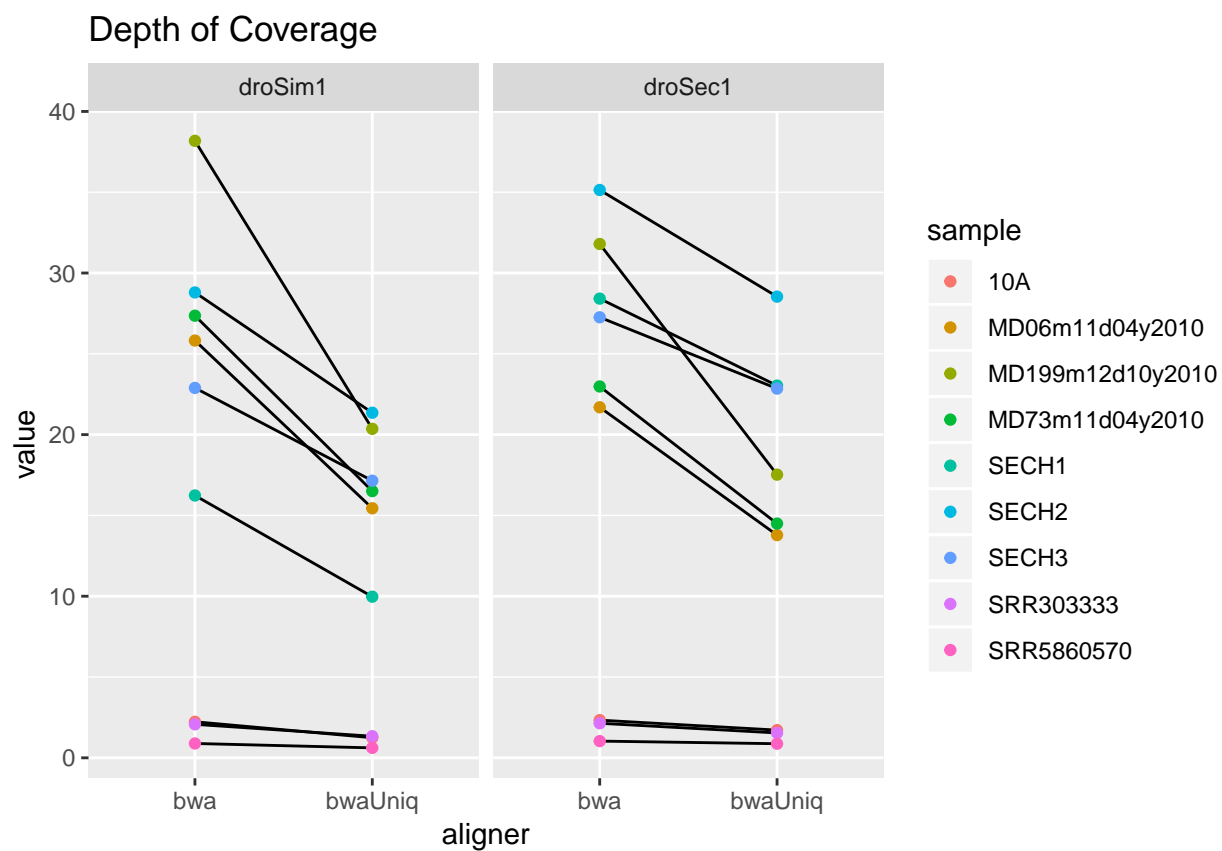
Also, some mention of reference genomes in the summary, with stats?

First, clean up the summarizers with a loading wrapper function

Previous stuff still works:







28 Nov 2018

Retooling some diagrams and pipes

## Read Counts by Processing Step: Raw, Mapped, Filtered

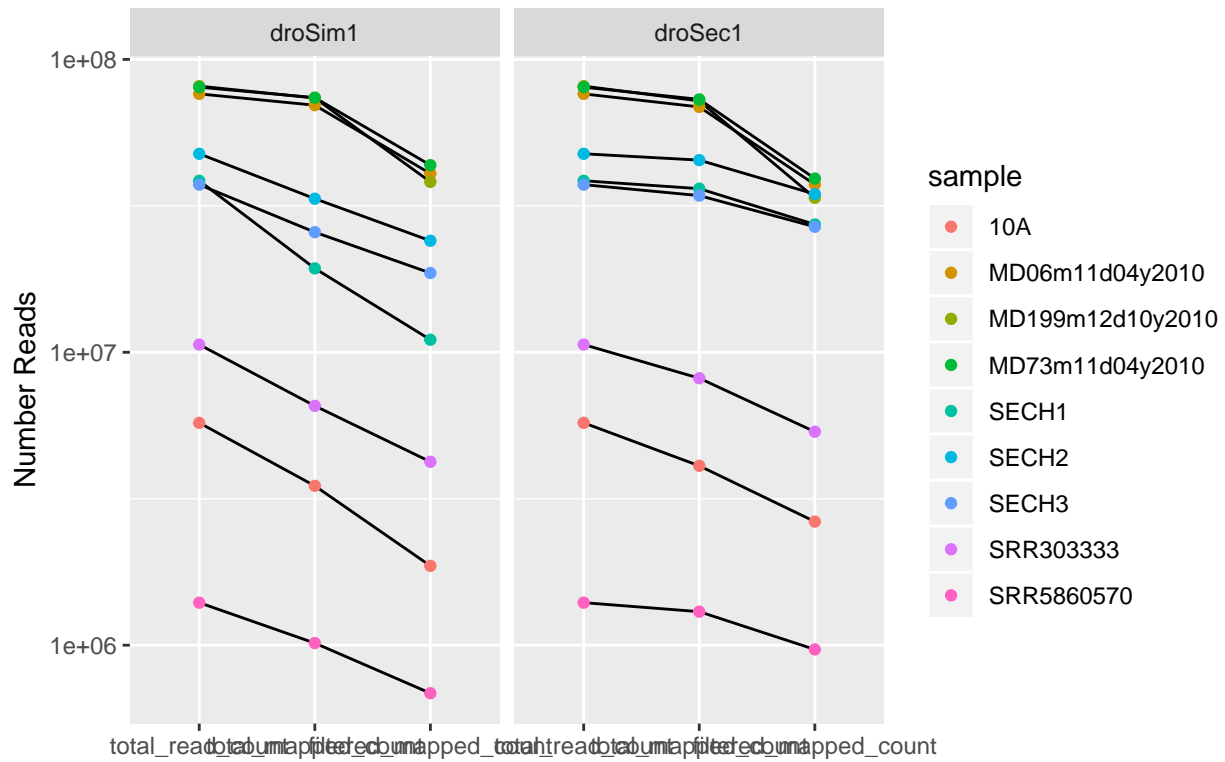


Table 8: Read Counts During Alignment & Filtration

measure	minimum	average	median	maximum
filtered_mapped_count	686 k	21.7 M	25.4 M	43.5 M
total_mapped_count	1.02 M	36.1 M	33.9 M	74 M
total_read_count	1.4 M	42.1 M	38.5 M	81.1 M

We can easily break down the table further with a second grouping:

Table 9: Read Counts During Alignment & Filtration

measure	reference	minimum	average	median	maximum
filtered_mapped_count	droSim1	686 k	20.3 M	18.6 M	43.5 M
filtered_mapped_count	droSec1	967 k	23.1 M	27.3 M	39.2 M
total_mapped_count	droSim1	1.02 M	34.1 M	25.7 M	74 M
total_mapped_count	droSec1	1.3 M	38.1 M	36.2 M	73 M
total_read_count	droSim1	1.4 M	42.1 M	38.5 M	81.1 M
total_read_count	droSec1	1.4 M	42.1 M	38.5 M	81.1 M

using spread and gather to clean up this mess:

Again, just group\_by() for a more detailed breakdown:

29 Nov 2018

do this to include script contents eg in the methods

```
cat scripts/bam_summarizer.py
```

```
import argparse

parser = argparse.ArgumentParser()
parser.add_argument("-f", "--flagstat_in", help="samtools flagstat report")
parser.add_argument("-i", "--idxstat_in", help="samtools idxstat report")
parser.add_argument("-g", "--genomecov_in", help="bedtools genomecov report")
parser.add_argument("-d", "--depthstats_in", help="samtools depth report")
#parser.add_argument("stat_in", help="samtools stats report")
parser.add_argument("-o", "--flat_out", help="flatfile summary")
parser.add_argument("-t", "--tag", help="line-name for the flatfile", default=None)
args = parser.parse_args()

summary_dict={}

flagstat = open(args.flagstat_in, 'r')
flagstat_lines = flagstat.readlines()
flagstat.close()

idxstat = open(args.idxstat_in, 'r')
idxstat_lines = idxstat.readlines()[:-1]
idxstat.close()

gencov = open(args.genomecov_in, 'r')
gencov_lines = gencov.readlines()
gencov.close()

dpth = open(args.depthstats_in, 'r')
dpth_lines = dpth.readlines()
dpth.close()

summary_dict['total_read_count'] = int(flagstat_lines[0].split(" ")[0])
summary_dict['total_mapped_count'] = int(flagstat_lines[4].split(" ")[0])
summary_dict['properly_paired_count'] = int(flagstat_lines[0].split(" ")[0])
#summary_dict['avg_depth'] = sum([float(p.split('\t')[2]) for p in idxstat_lines ])/sum([int(q.split('\t')[0]) for q in idxstat_lines ])
summary_dict['total_breadth'] = float(gencov_lines[-1].split()[-1])
summary_dict['avg_depth'] = float(dpth_lines[0].split("\t")[1])
summary_dict['std_depth'] = float(dpth_lines[1].split("\t")[1])

phial_out = open(args.flat_out, 'w')

keys = ['total_read_count', 'total_mapped_count', 'properly_paired_count', 'avg_depth', 'std_depth', 'total_breadth']

lines2write = [ [k, summary_dict[k]] for k in keys]
if args.tag:
    [ ell.insert(0, args.tag) for ell in lines2write ]
```

```
for preline in lines2write:
    field_count = len(preline)
    line = ("%s" + "\t%s"*(field_count-1) + "\n") % tuple(preline)
    phial_out.write(line)
```

```
phial_out.close()
```

yikes, looks like i might need to run a pep8 check LOL