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
17B	TRUE	selection	EarlyJones2013
17A	TRUE	selection	EarlyJones2013
10B	TRUE	selection	EarlyJones2013
10A	TRUE	selection	EarlyJones2013

Population-wide sample count by species:

Table 2: Number of Sequenced Samples per Species

species	sample_count
drosophila sechellia	4
drosophila simulans	4

load & discuss FASTP summary

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

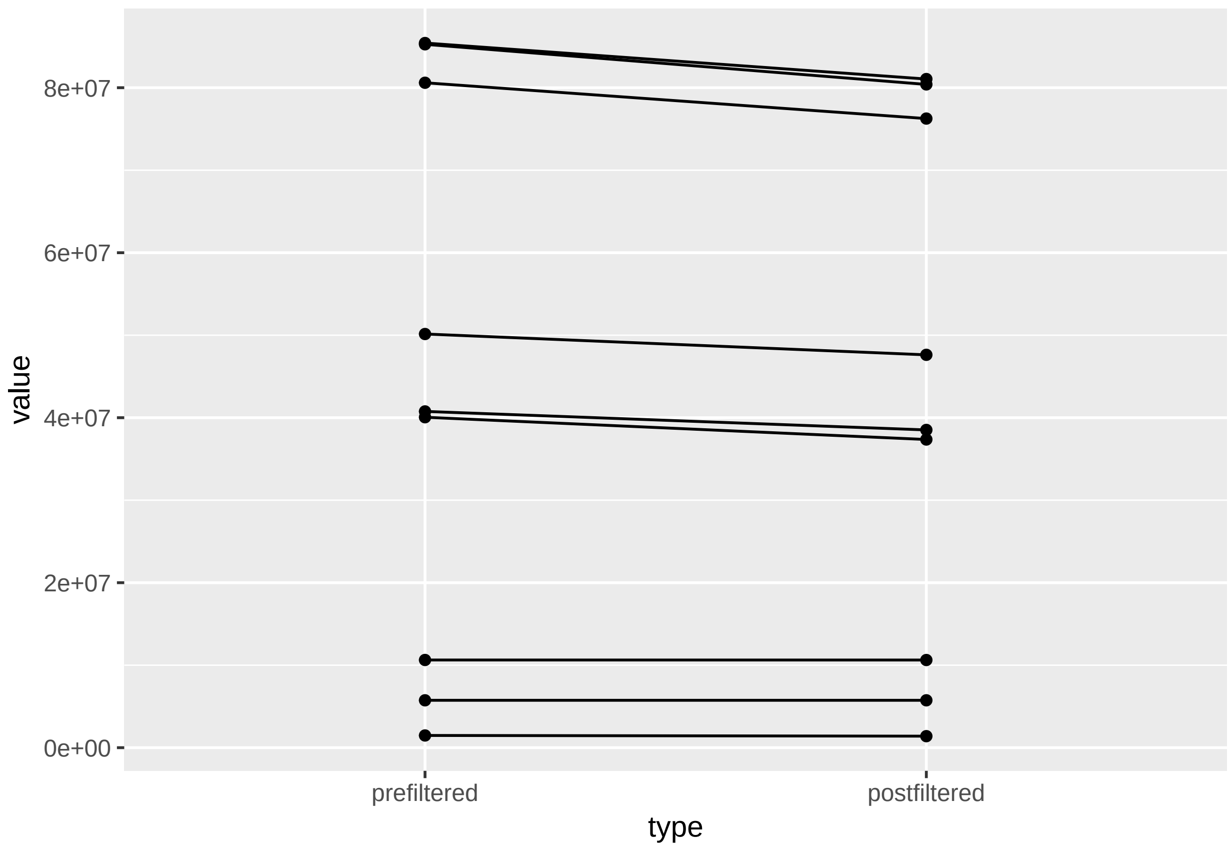
prefilt:

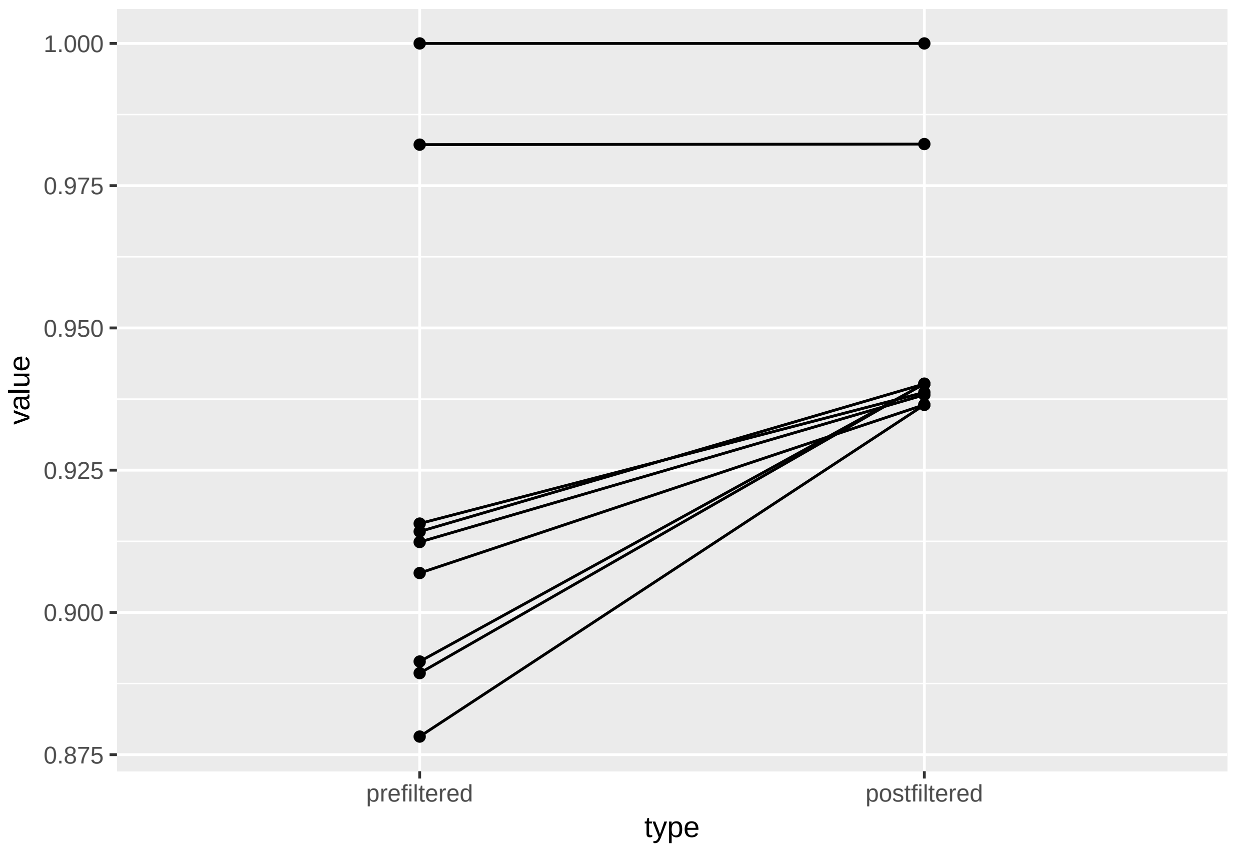
```
## Warning: Column `name` joining factors with different levels, coercing to
## character vector
```

```
## # 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
percent retention	9.326349e+01	9.564348e+01	100





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

19 Nov 2018

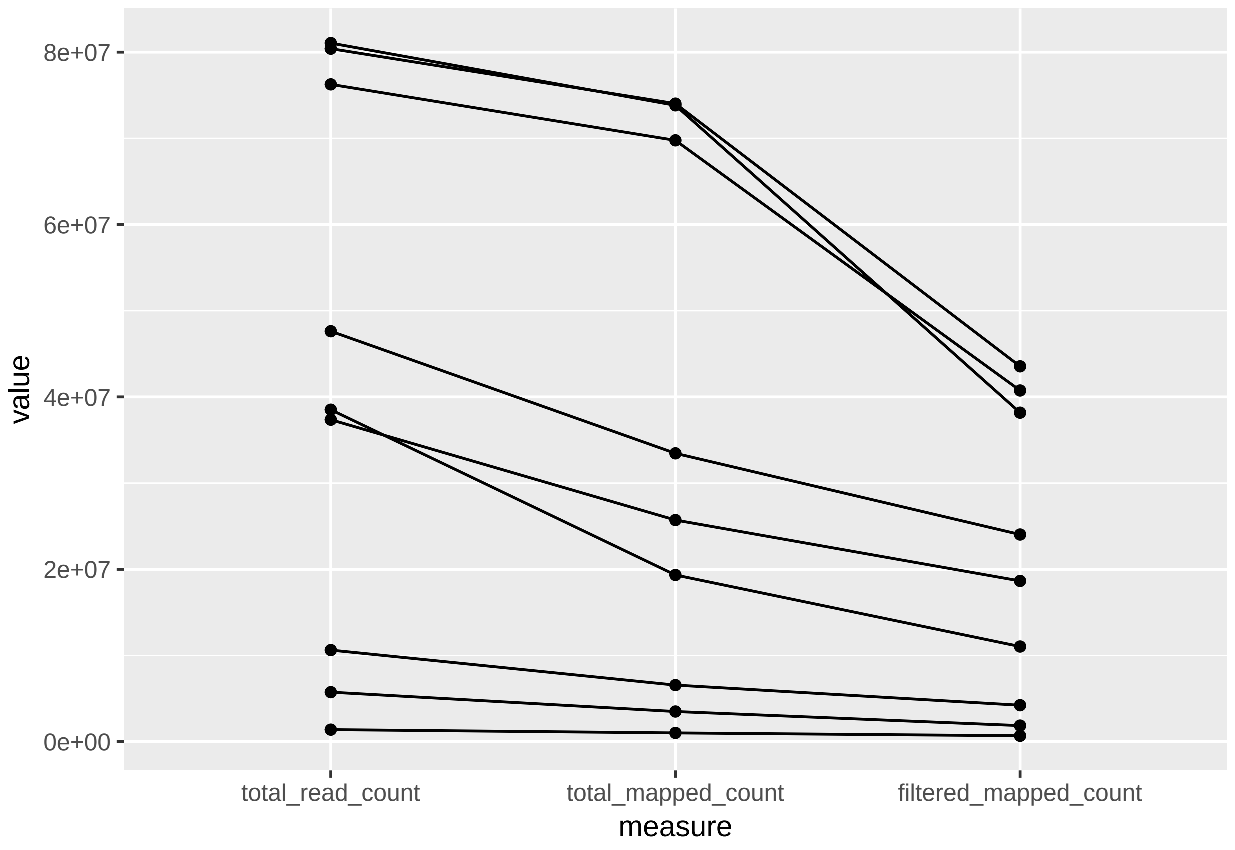
load and discuss bam summary

depth of coverage is effed???

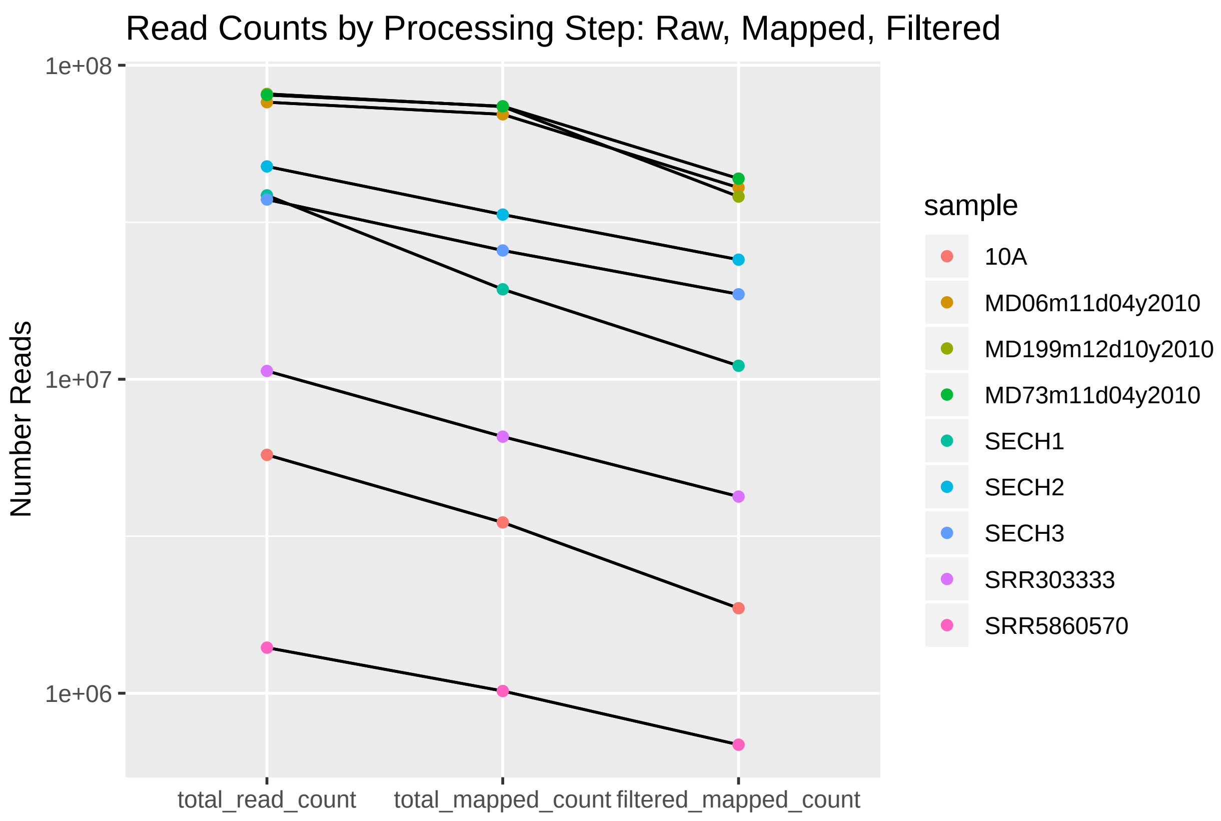
```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_double()
## )
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_double()
## )
```

```
## # 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
```

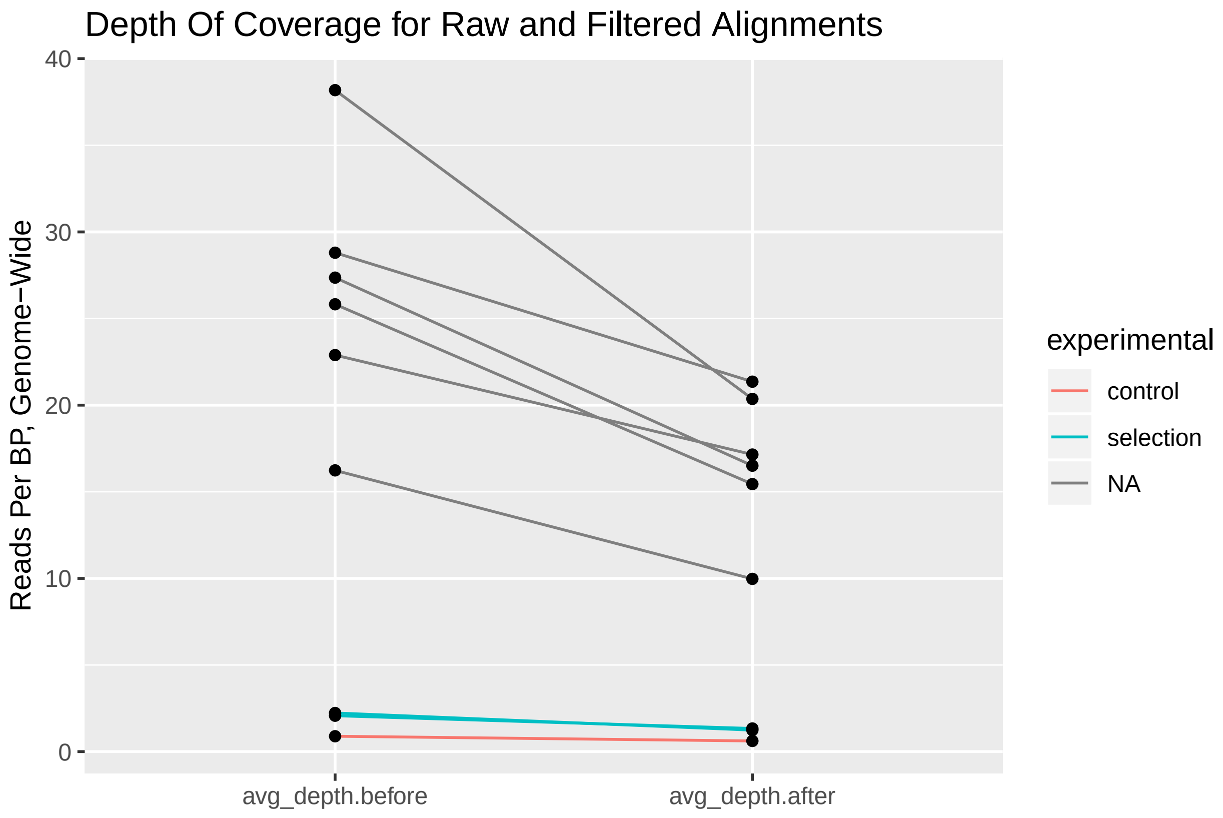
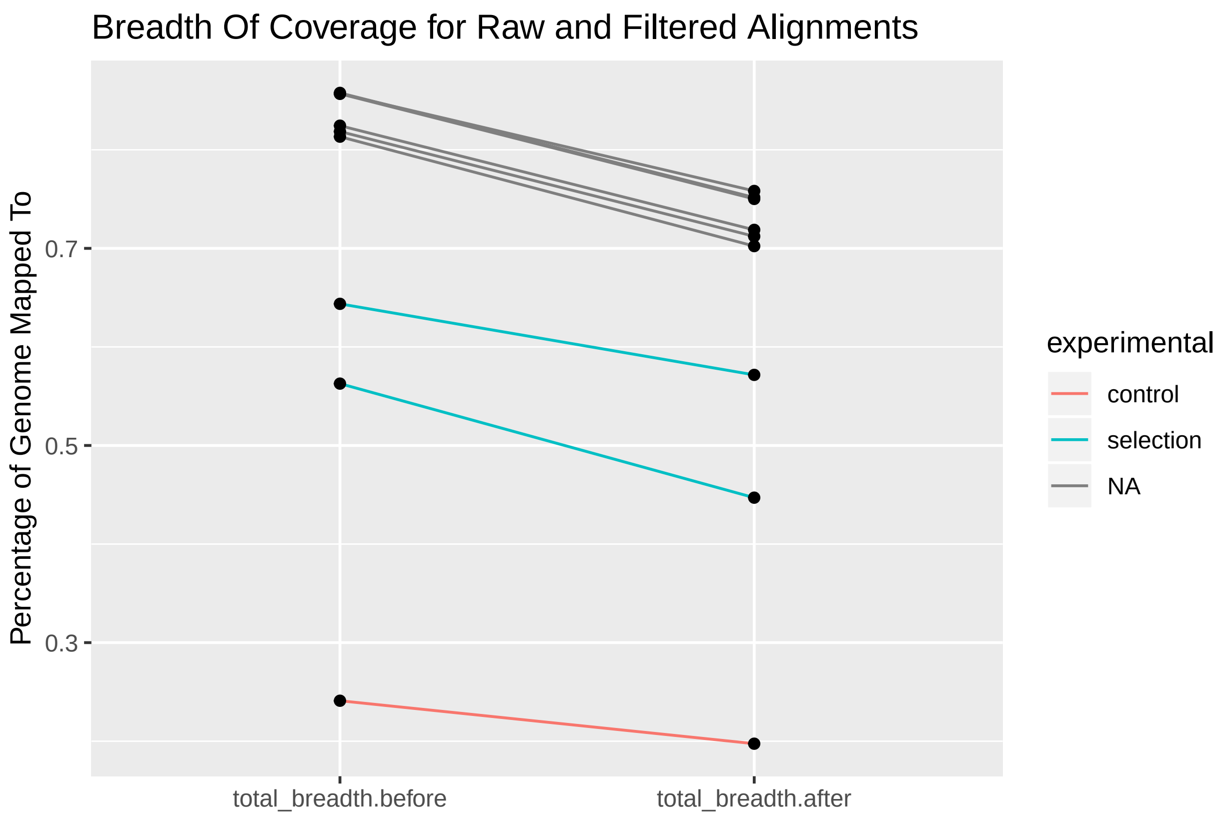


Table 5: Breadth of Coverage Statistics for Raw and Filtered Alignments

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

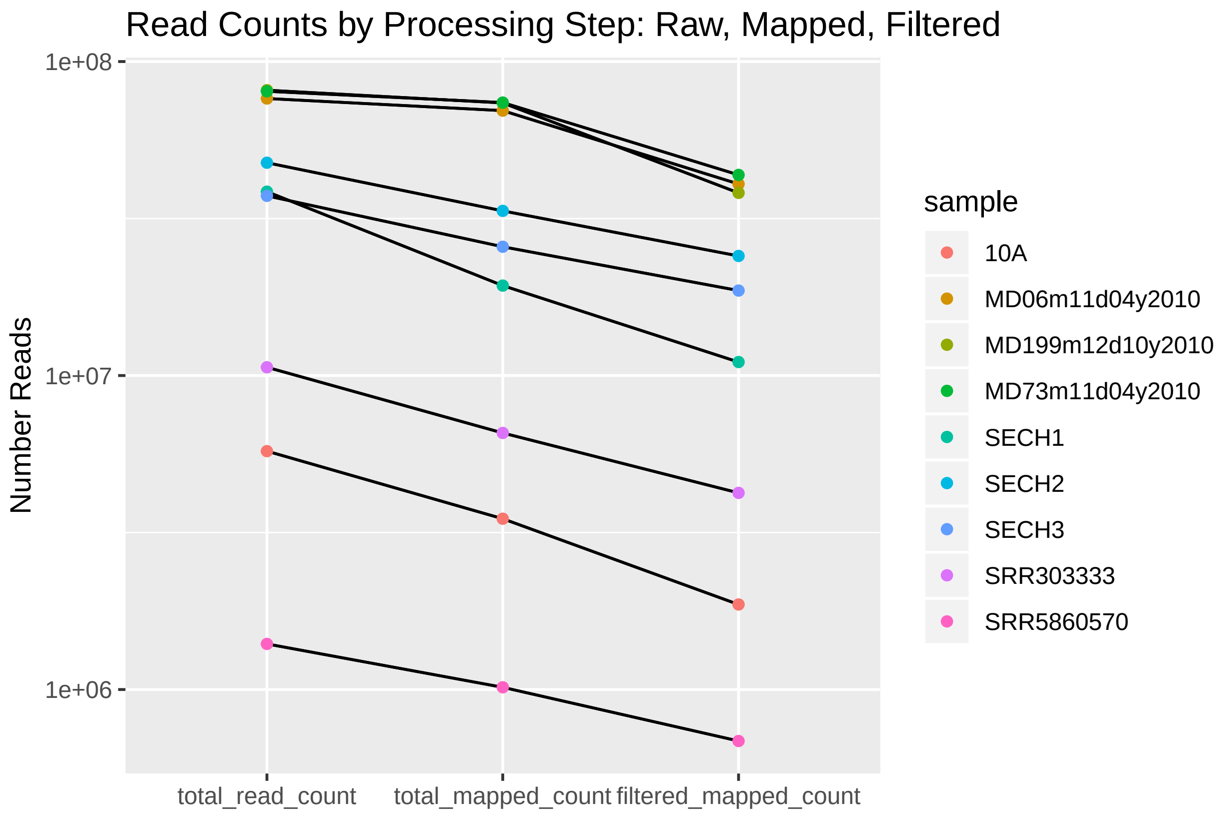
step	minimum	average	median	maximum
pre-filtration breadth	24.1	71.9	81.8532	85.8
post-filtration breadth	19.7	62.3	71.2195	75.8
breadth retention	79.4	86	87.1791	88.8

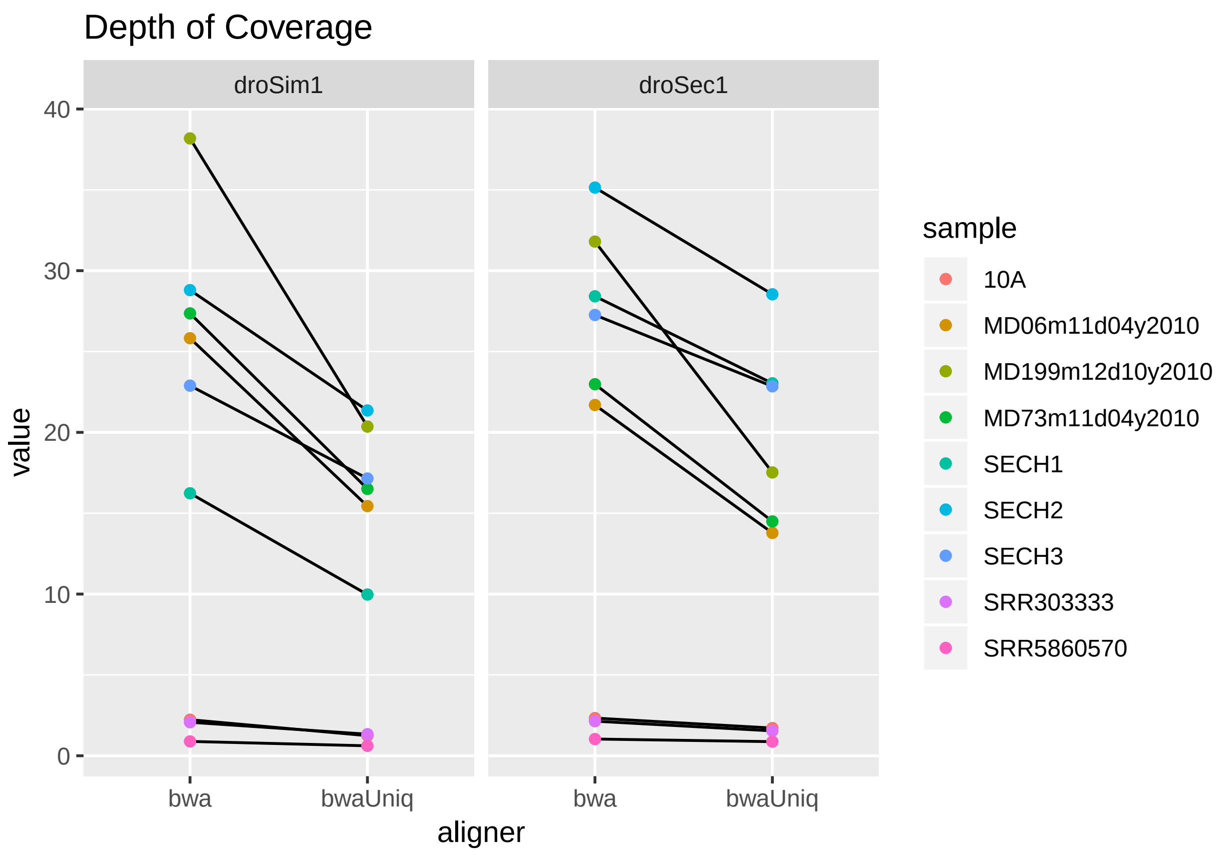
Also, need to add panels by reference genome.

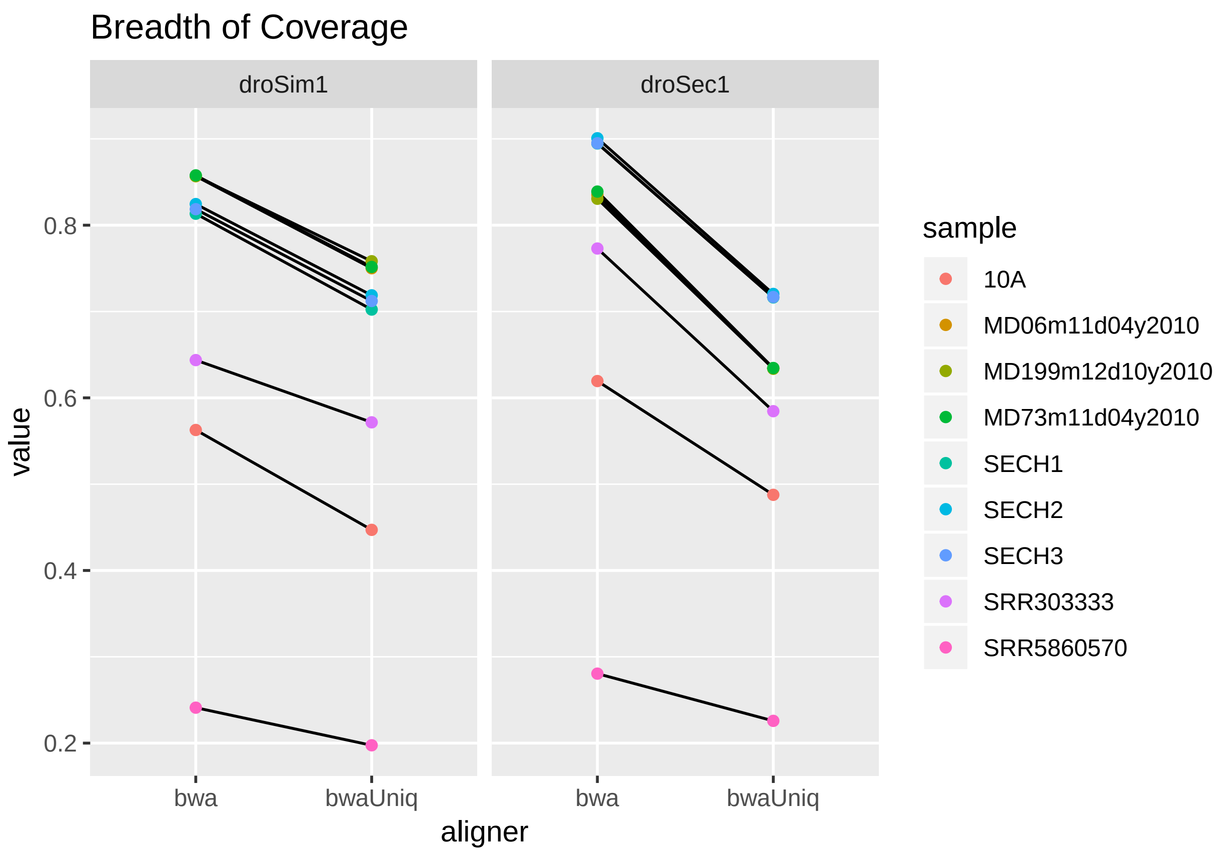
Also, some mention of reference genomes in the summary, with stats?

First, clean up the summarizers with a loading wrapper function

Previous stuff still works:




Depth of Coverage
droSim1
droSec1
value
aligner
bwa
bwaUniq
0
10
20
30
40
sample
10A
MD06m11d04y2010
MD199m12d10y2010
MD73m11d04y2010
SECH1
SECH2
SECH3
SRR303333
SRR5860570


Breadth of Coverage
droSim1
droSec1
value
aligner
bwa
bwaUniq
0.2
0.4
0.6
0.8
sample
10A
MD06m11d04y2010
MD199m12d10y2010
MD73m11d04y2010
SECH1
SECH2
SECH3
SRR303333
SRR5860570

28 Nov 2018

Retooling some diagrams and pipes

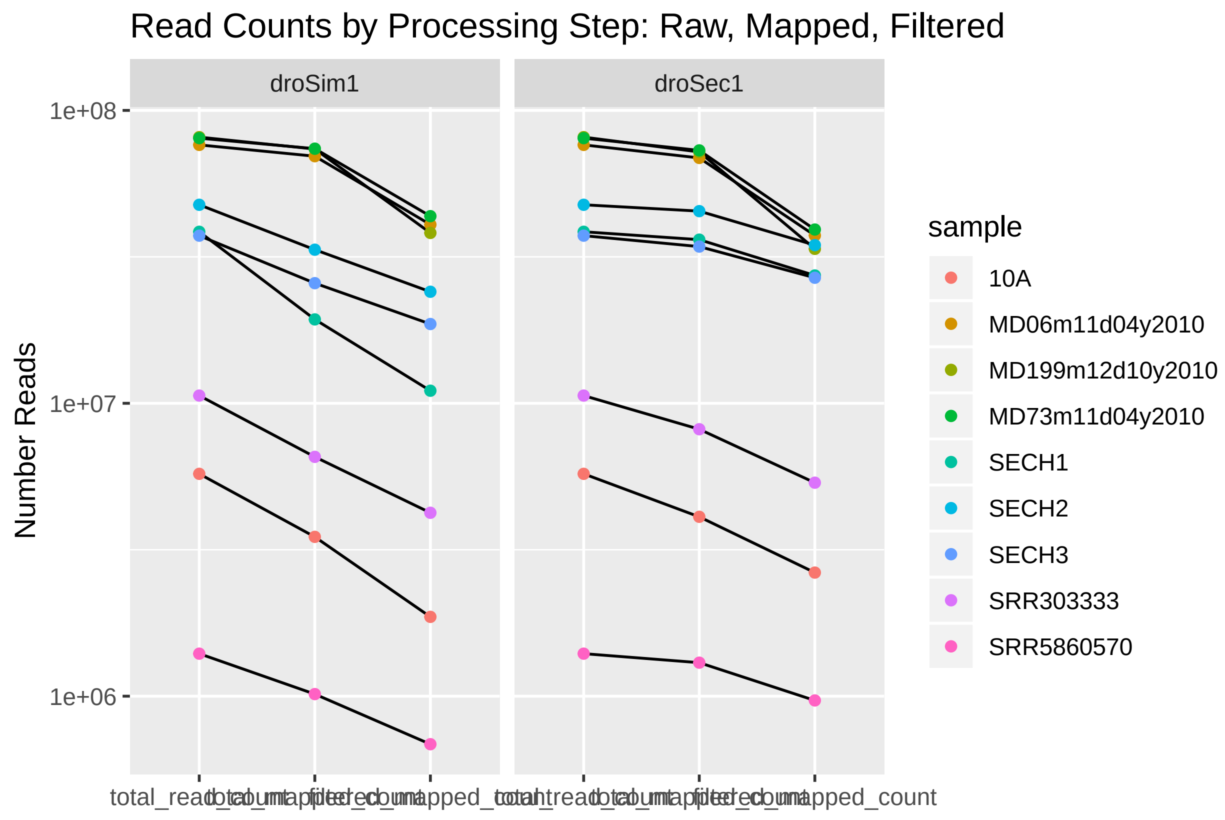


Table 8: Read Counts During Alignment & Filtration

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:

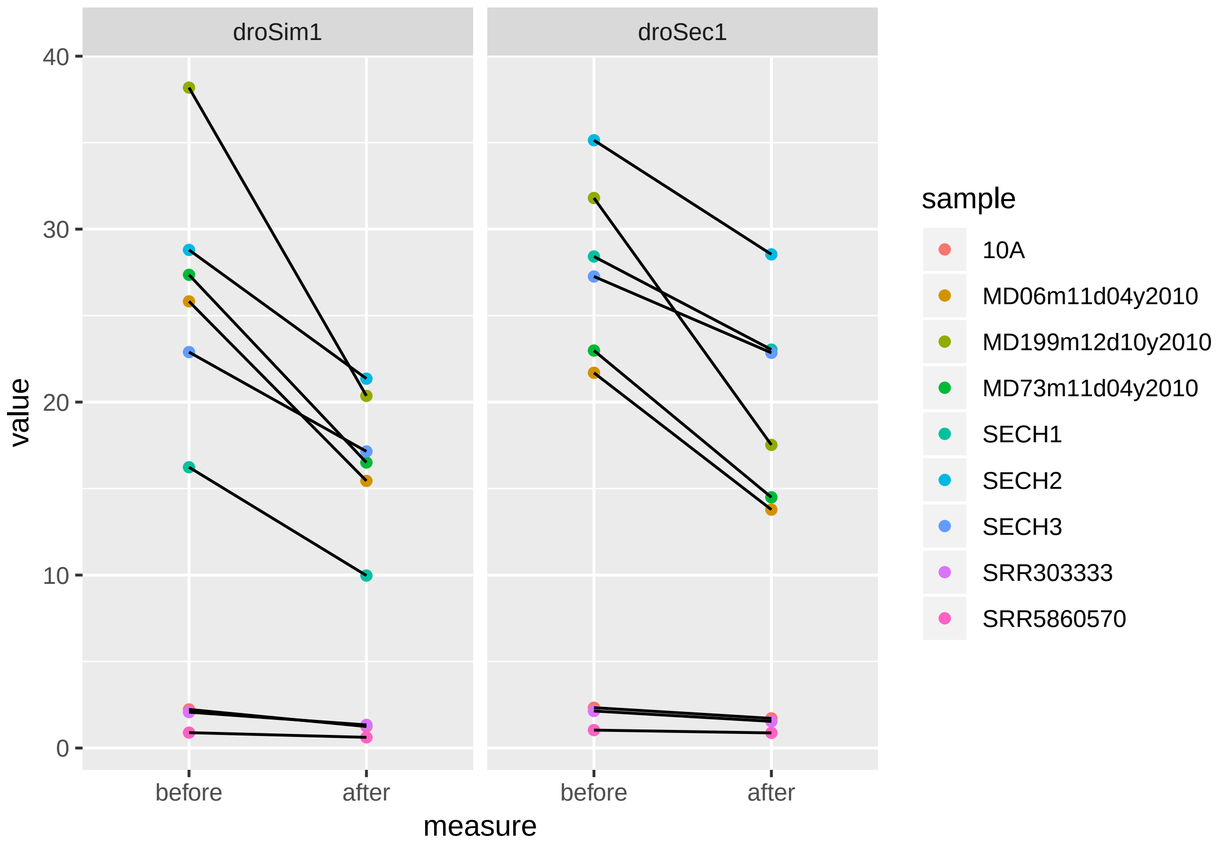
Table 9: Read Counts During Alignment & Filtration

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:

Table 10: Depth of Coverage Statistics for Raw and Filtered Alignments

step	minimum	average	median	maximum
pre-filtration depth	0.9	18.7	22.9	38.2
post-filtration depth	0.6	12.7	15.0	28.5
depth retention percent	53.3	68.4	66.9	84.4



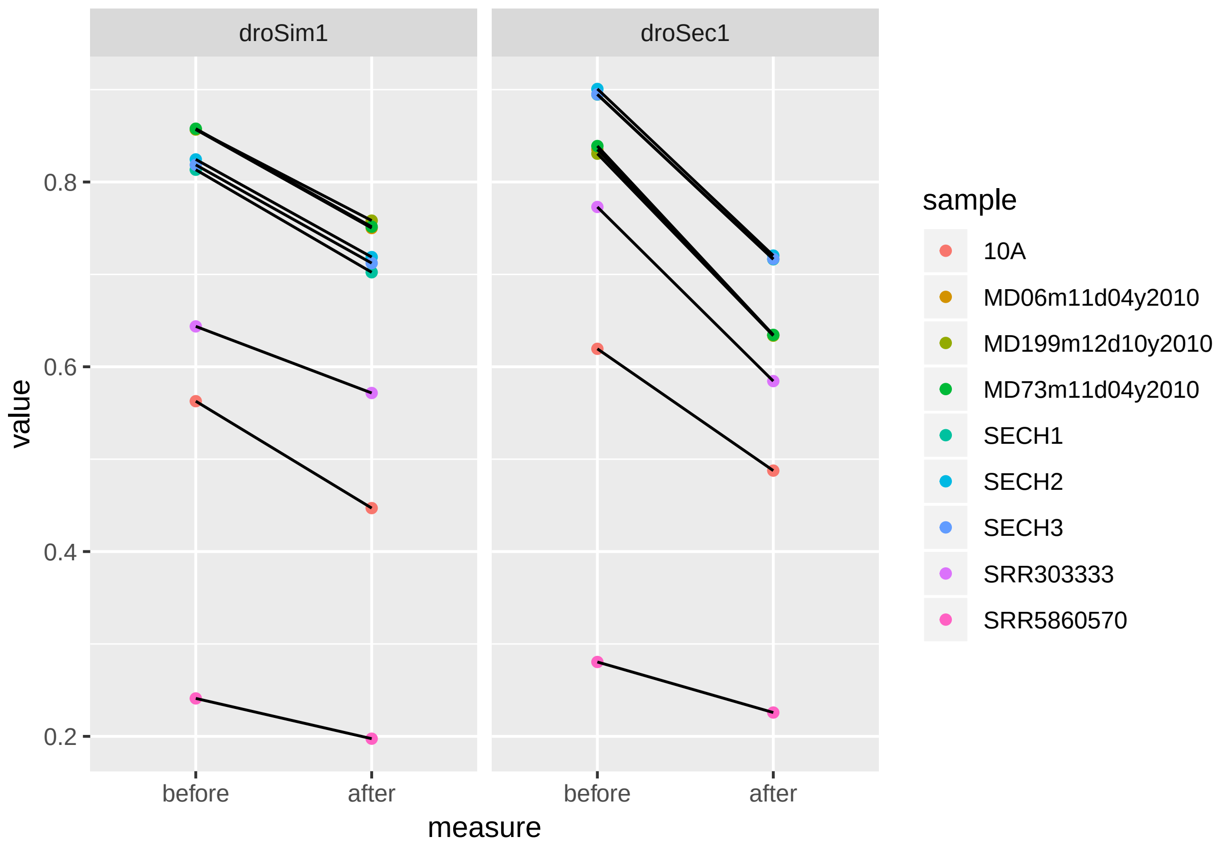
Again, just group_by() for a more detailed breakdown:

```
## # A tibble: 2 x 6
##   reference step                 minimum average median maximum
##   <fct>     <chr>                  <dbl>   <dbl>  <dbl>   <dbl>
## 1 droSim1   pre-filtration depth   0.889    18.3   22.9    38.2
## 2 droSec1   pre-filtration depth   1.03     19.2   23.0    35.1
```

29 Nov 2018

Table 11: Breadth of Coverage Statistics for Raw and Filtered Alignments

step	minimum	average	median	maximum
pre-filtration breadth	24.1	74.1	82.8	90.1
post-filtration breadth	19.7	60.9	66.8	75.8
breadth retention percent	75.6	82.1	80.3	88.8



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('\
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', 'tota

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

VCFs are done building:

cat all_samples.vs_droSim1.bwaUniq.vcf | head -n 1000 > all_samples.vs_droSim1.bwaUniq.vcf.subset

30 Nov 2018

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

Table 12: Size and Consolidation of Reference Genomes

Reference Genome:	dm6	droSec1	droSim1
number_bases	144 M	167 M	142 M
number_contigs	1.87 k	14.7 k	18

Shored up the command-line PDF generation to build the output in a designated path (ie, the PopPsiSeq head)

https://stackoverflow.com/questions/31463143/pass-parameters-from-command-line-into-r-markdown-document https://github.com/yihui/knitr/issues/913

Starting basic stats on the VCFs. . . .

total SNP count & rate:

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

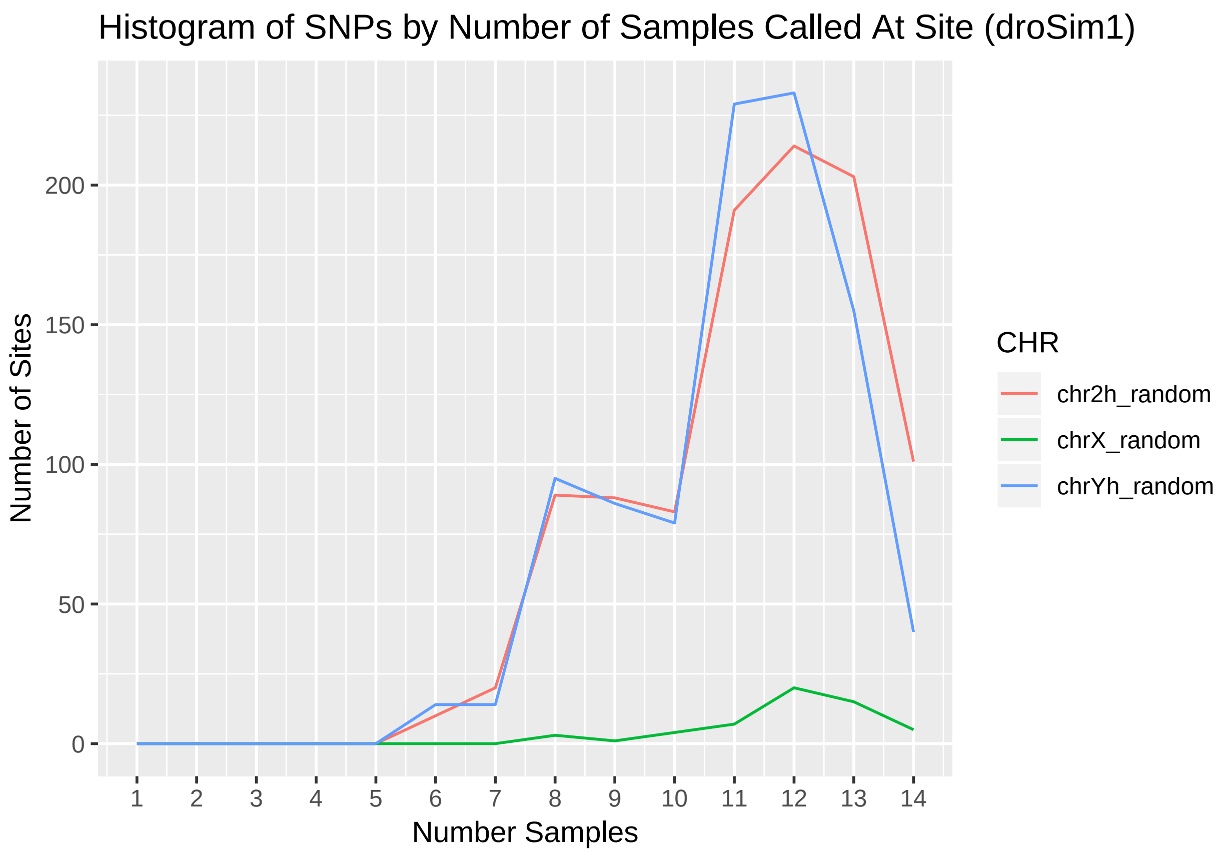
Table 13: SNP count and per-KB SNP rate across all samples

reference genome	Genome size (bp)	total SNP count	SNPs per kB
droSec1	167 M	2.44 M	14.66585
droSim1	142 M	2.7 M	18.93993

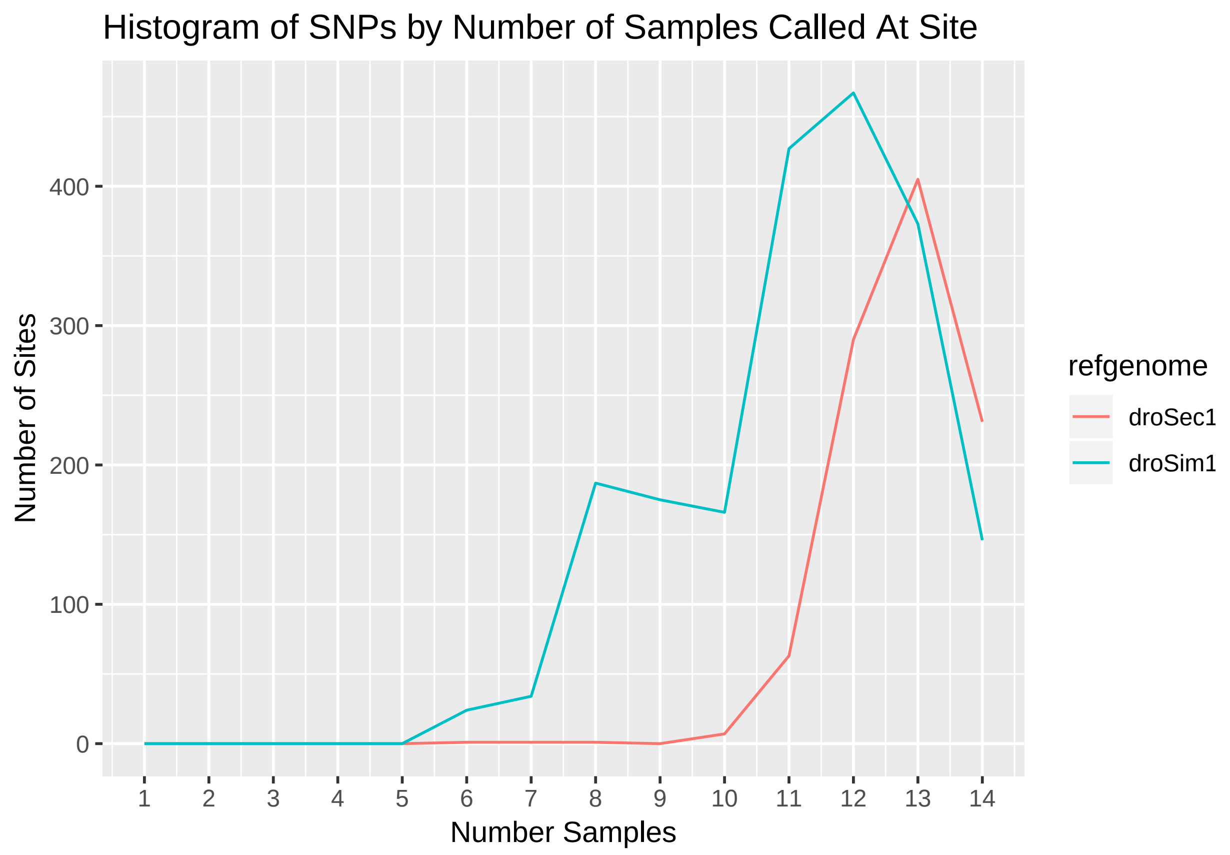
sample calls by site:

```
## Parsed with column specification:
## cols(
##   CHR = col_character(),
##   POS = col_integer(),
##   N_DATA = col_integer(),
##   N_GENOTYPE_FILTERED = col_integer(),
##   N_MISS = col_integer(),
##   F_MISS = col_double()
## )
## Parsed with column specification:
## cols(
##   CHR = col_character(),
##   POS = col_integer(),
##   N_DATA = col_integer(),
##   N_GENOTYPE_FILTERED = col_integer(),
##   N_MISS = col_integer(),
##   F_MISS = col_double()
## )
```

```
## Warning: Removed 6 rows containing missing values (geom_path).
```



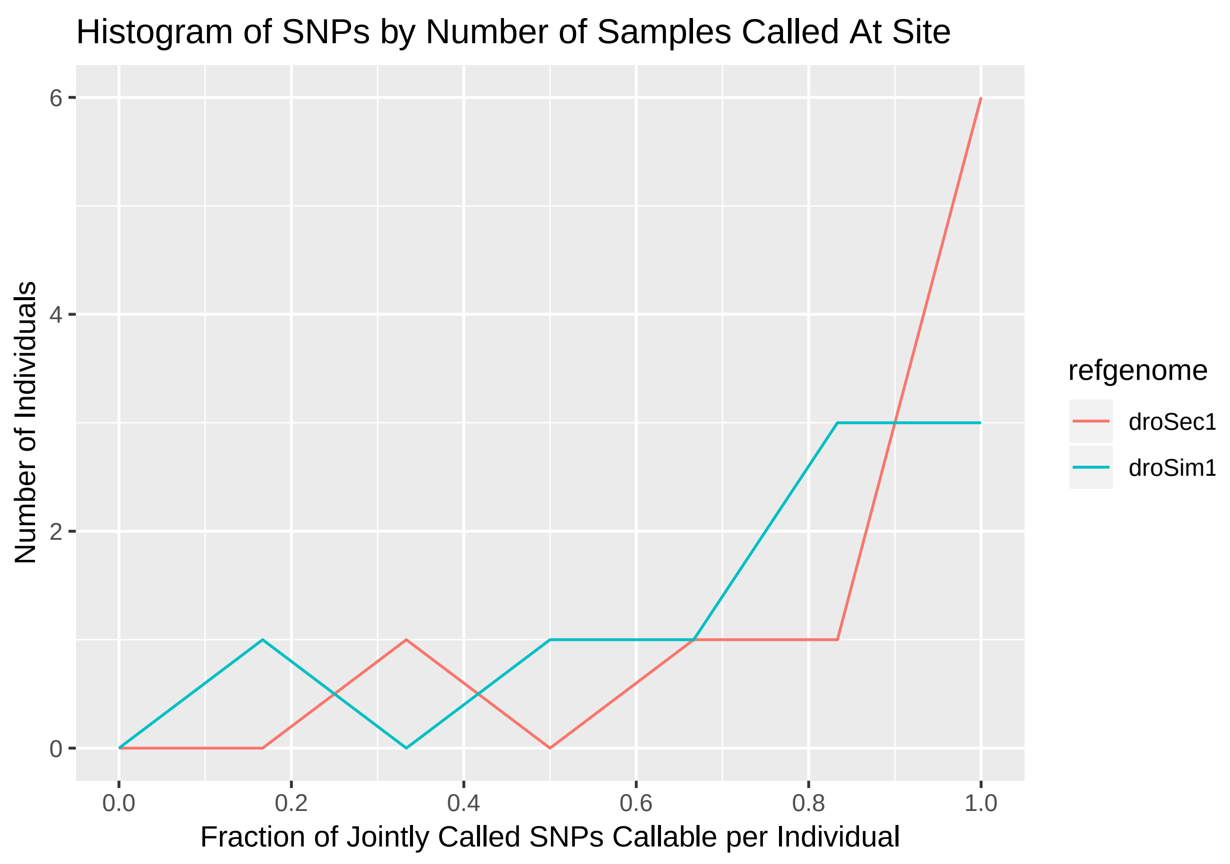
```
## Warning: Removed 4 rows containing missing values (geom_path).
```



uncalled sites by sample:

```
## Parsed with column specification:
## cols(
##   INDV = col_character(),
##   N_DATA = col_integer(),
##   N_GENOTYPES_FILTERED = col_integer(),
##   N_MISS = col_integer(),
##   F_MISS = col_double()
## )
## Parsed with column specification:
## cols(
##   INDV = col_character(),
##   N_DATA = col_integer(),
##   N_GENOTYPES_FILTERED = col_integer(),
##   N_MISS = col_integer(),
##   F_MISS = col_double()
## )
```

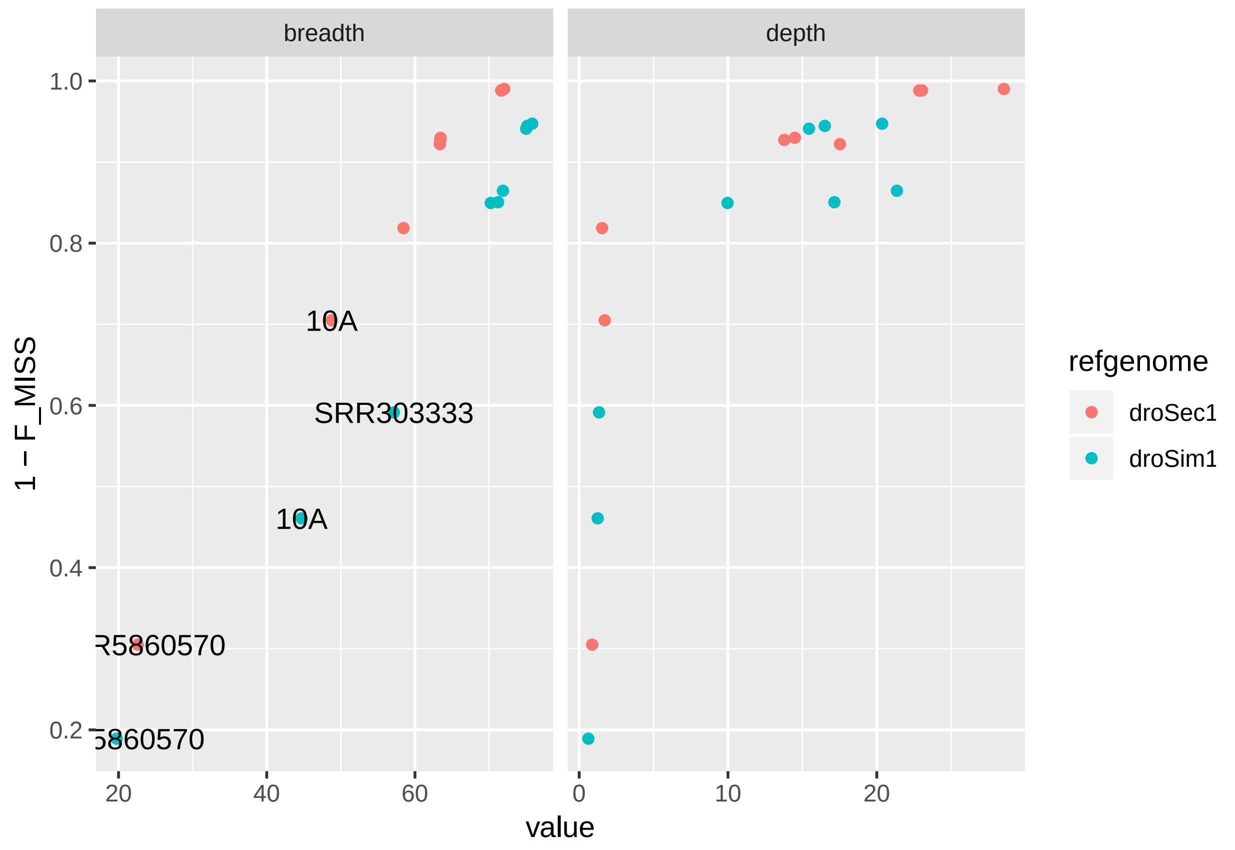
```
## Warning: Removed 4 rows containing missing values (geom_path).
```



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

```
## Warning: Column `refgenome`/`reference` joining character vector and
## factor, coercing into character vector
```

https://stackoverflow.com/questions/15015356/how-to-do-selective-labeling-with-ggplot-geom-point



3 Dec 2018

working on some of the analytics, using the vcftools standalone commands:

```
vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c SECH1,SECH2,SECH3 | head -n 200000 | vcfto

vcftools  --vcf PopSech.vs_droSim1.bwaUniq.vcf --out potato --freq


vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c MD06m11d04y2010,MD73m11d04y2010,MD199m12d1

vcftools  --vcf PopSim.vs_droSim1.bwaUniq.vcf --out PopSim --freq

vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c SRR5860570,10A | head -n 200000 | vcftools

vcftools  --vcf experimental.vs_droSim1.bwaUniq.vcf --out experimental --freq
paste PopSech.frq PopSim.frq experimental.frq | awk '{if($3<3)print;}' | awk '{if($9<3)print;}' | awk '
```

Add a group: PopSec,All tag or something in the config.yaml so that the -c string is callable

-u to keep uncalled sites

Maybe add filtering for sites (eg, AC > thresh, AF>thresh)

Filter down to biallelic sites? –min-alleles 2 –max-alleles 2

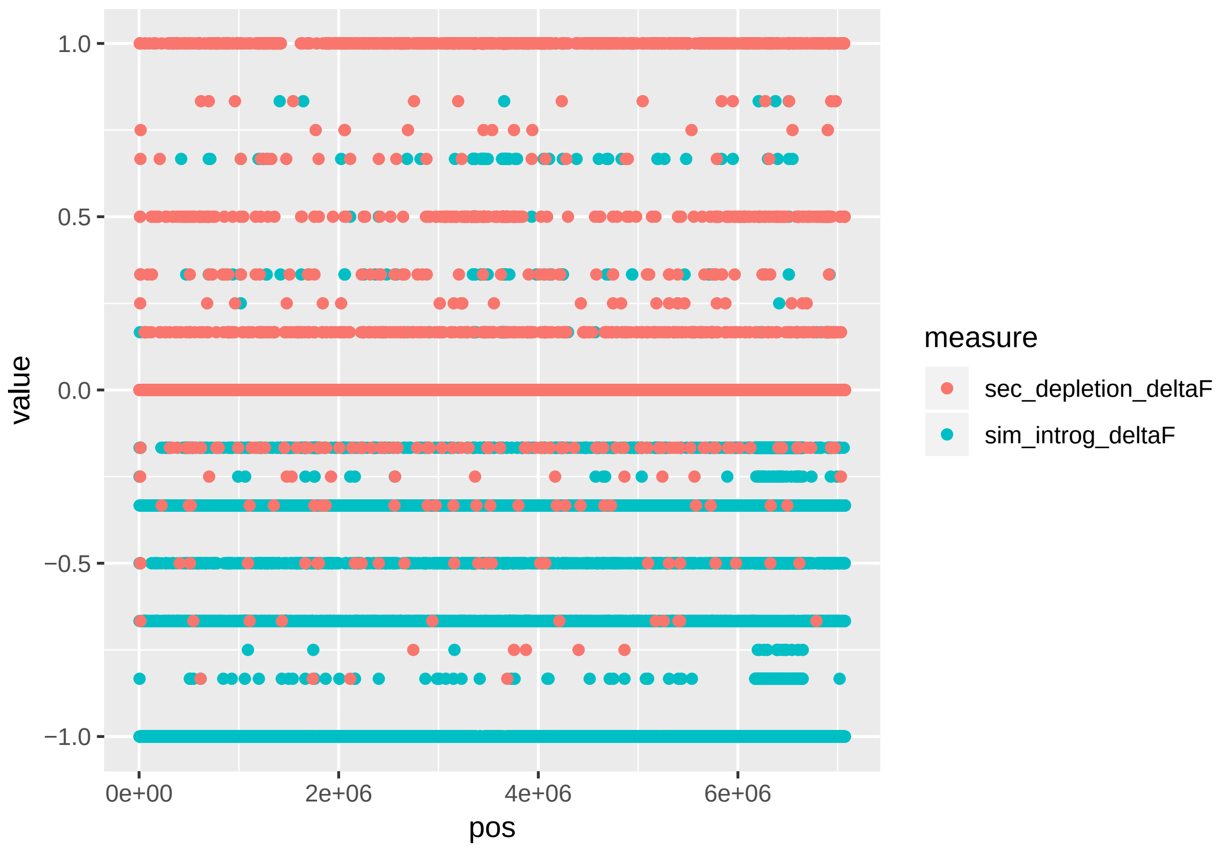
4 Dec 2018

```
vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c SECH1,SECH2,SECH3 | head -n 200000 | vcfto

vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c MD06m11d04y2010,MD73m11d04y2010,MD199m12d1

vcf-subset variants/all_samples.vs_droSim1.bwaUniq.vcf -u -c 10A | head -n 200000 | vcftools --min-allel

vcftools  --vcf PopSech.vs_droSim1.bwaUniq.vcf --out PopSech --freq
vcftools  --vcf PopSim.vs_droSim1.bwaUniq.vcf --out PopSim --freq
vcftools  --vcf experimental.vs_droSim1.bwaUniq.vcf --out experimental --freq
```

–max-missing-count 1 sets the number of uncalled samples allowed per site (see also –max-missing [float] for fraction)

```
bedtools intersect -wa -wb -a <(cat PopSech.frq | tail -n +2 | awk '{print $1,$2,$2+1,$4,$5,$6}' | tr "
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_integer(),
##   X3 = col_character(),
##   X4 = col_character(),
##   X5 = col_integer(),
##   X6 = col_double(),
##   X7 = col_integer(),
##   X8 = col_double(),
##   X9 = col_integer(),
##   X10 = col_double()
## )
```



Plotting two values here: change in AF towards the AF in simulans (sim_introg_deltaF) and change away from the AF in sech (sec_depletion_deltaF)