End Joining Signatures - dev2

Charlie Soeder

3/25/2019

25 March 2019

From Talia:

> If Danny's progeny are from two types of parental crosses, I guess I would just do analysis on one of those crosses, just to keep it easier. Find Indels in those progeny, then compare to mcm5a7 progeny.

Ok, going to restrict to w3 and w4.

Maybe redo the samples summary to reflect this?

> Males were numbered based on whether their father was homozygous w1118 or Canton-S and the number of their het- erozygous mother. For example, male cs12.3 had a Canton-S father, its mother was female number 12, and it was the third male selected for DNA extraction. Sibling numbers may not be continuous, as males with low DNA concentrations after DNA extraction were not selected for sequencing.

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [1,
## 2].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [2].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

Table 1: Sequenced Samples from Control Cross

name	source	pedigree	sex	mating	offspring_id	father_type	mother_id
w1118	danny	parent	M	NA	NA	NA	NA
CantonS	danny	parent	F	NA	NA	NA	NA
w4_4	danny	child	M	w4	4	w1118	4
w4_3	danny	child	M	w4	3	w1118	4
w4_2	danny	child	M	w4	2	w1118	4
w4_17	danny	child	M	w4	17	w1118	4
w4_16	danny	child	M	w4	16	w1118	4
w4_15	danny	child	M	w4	15	w1118	4
w4_13	danny	child	M	w4	13	w1118	4
w4_12	danny	child	M	w4	12	w1118	4
w4_11	danny	child	M	w4	11	w1118	4
w4_1	danny	child	M	w4	1	w1118	4
w3_9	danny	child	M	w3	9	w1118	3
w3_8	danny	child	M	w3	8	w1118	3
w3_6	danny	child	M	w3	6	w1118	3
w3_5	danny	child	M	w3	5	w1118	3
w3_4	danny	child	M	w3	4	w1118	3
w3_26	danny	child	M	w3	26	w1118	3
w3_25	danny	child	M	w3	25	w1118	3
w3_24	danny	child	M	w3	24	w1118	3

name	source	pedigree	sex	mating	offspring_id	father_type	mother_id
w3_21	danny	child	M	w3	21	w1118	3
w3_18	danny	child	M	w3	18	w1118	3
w3_17	danny	child	M	w3	17	w1118	3
w3_16	danny	child	M	w3	16	w1118	3
w3_15	danny	child	M	w3	15	w1118	3
w3_14	danny	child	M	w3	14	w1118	3
w3_13	danny	child	M	w3	13	w1118	3
w3_12	danny	child	M	w3	12	w1118	3
w3_11	danny	child	M	w3	11	w1118	3
w3_1	danny	child	M	w3	1	w1118	3

Table 2: Control Cross Samples by Cross Type

father_type	count
w1118	28

Table 3: Number of Male Offsping Sequenced, by Cross Type and Female ID

father_type	mother_id	count
w1118	3	18
w1118	4	10

It might be good to go ahead and do a YAML for all the flies, with the ones being ignored in a null subgroup. Write a rule to download based on provided SRAs.

28 March 2019

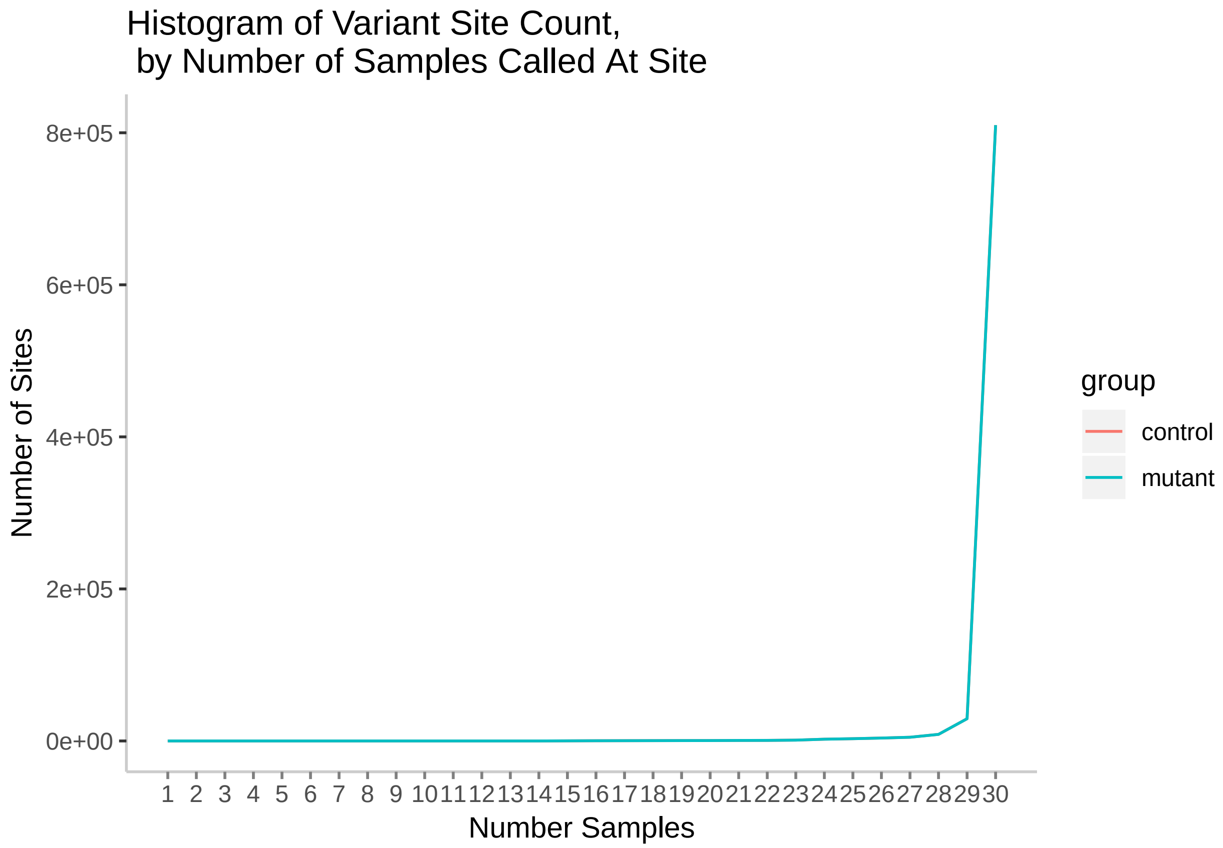
Ok cool I finally got some slots on Longleaf and the VCFs are building. using temporaries right now (variants called to date).

It occurs to me I may have to go back and rework some reporting rules to reflect the comparison of two different subrgoup variants..........

```
$ pwd
/Users/csoeder/Research/EJgrepper/dev/meta/VCFs
$ prefix=control
$ cat control.vs_dm6.bwaUniq.summary | sed -e 's/^/'$prefix'\t/g'> ../all_groups.vs_dm6.bwaUniq.calledVa
$ prefix=mutant
$ cat mutant.vs_dm6.bwaUniq.summary | sed -e 's/^/'$prefix'\t/g' >> ../all_groups.vs_dm6.bwaUniq.calledV
```

This is a little different than before; uses the callable chromosome number instead of the sample count (since the two VCFs could conceivably have different sample numbers) and rounds down.

```
## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min

## Need specific help about ggbio? try mailing
##  the maintainer or visit http://tengfei.github.com/ggbio/

##
## Attaching package: 'ggbio'

## The following objects are masked from 'package:ggplot2':
##
##     geom_bar, geom_rect, geom_segment, ggsave, stat_bin,
##     stat_identity, xlim

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



29 March 2019

Ok cool VCFs built and summarized.

Ugh, geom_text + filter apparently fails with a " Aesthetics must either be length one, or the same length as the data" when the filter results are empty

```
mutant.calledVariants.imiss <- read_delim("meta/VCFs/mutant.vs_dm6.bwaUniq.summary.imiss", "\t", escape
mutant.calledVariants.imiss$group <- "mutant"

control.calledVariants.imiss <- read_delim("meta/VCFs/control.vs_dm6.bwaUniq.summary.imiss", "\t", escap
control.calledVariants.imiss$group <- "control"


allGroups.calledVariants.imiss <- rbind(control.calledVariants.imiss, mutant.calledVariants.imiss)  %>%
```

```
allGroups.imiss.augmented <- inner_join(allGroups.calledVariants.imiss, all_alignments %>%  filter(measu

allGroups.imiss.augmented <-inner_join(allGroups.imiss.augmented, all_alignments %>%  filter(measure=='

allGroups.imiss.augmented <- allGroups.imiss.augmented %>%  gather(breadth:depth, key="measure", value=
```

```
ggplot(allGroups.imiss.augmented) + geom_point(aes(x= value, y=1-F_MISS, color=group, shape=group)) + fa
```

So with the reporting out of the way, onto analysis of the variants. . . . . .

Rewriting the Winnower rule to preserve the filtered VCF, then run allele count, such that the VCF can then be used for a Novelist rule to remove sites that are variable in the parents.

oh cool adventures in purr and broom! use map_df to specify a data frame output so that the columns of the glanced stat test can be manipluated dplyr-style:

```
insert_truth.Tbl <- filteredTbl.biallele %>% mutate(ins=delta_bp>0) %>% group_by(ins,group) %>% summari

insert_truth.Tbl <- cbind(insert_truth.Tbl, map2(insert_truth.Tbl$del, insert_truth.Tbl$del+insert_trut
```

1 Apr 2019

```
ggplot(filteredTbl.biallele) + geom_freqpoly(aes(x=mac, color=group),bins=31) + facet_wrap(chrom~.) + t
```

Adding the qualification to “Novel Singeltons” that parent-derived alleles must get scrubbed.

Currently adding Site Ids by VCF surgery; maybe add them earlier??

Current approach: subset VCF to parent, select sites with parent having >0 alt alleles, kick those out. Q1: what if this rule needs to be expanded to multiallelic sites?

```
grep "1|1\|0|1"
```

vs

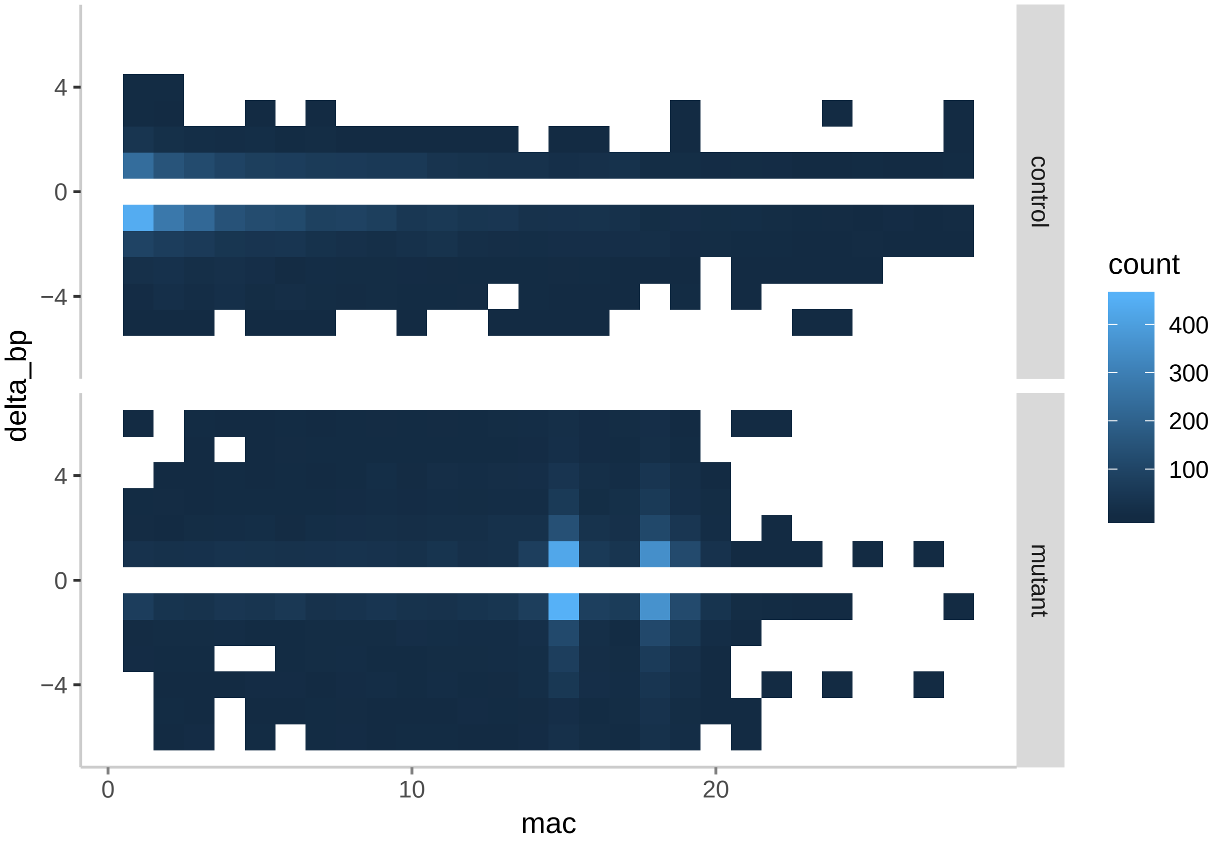
```
grep "[1-9]|[1-9]\|0|[1-9]"
```

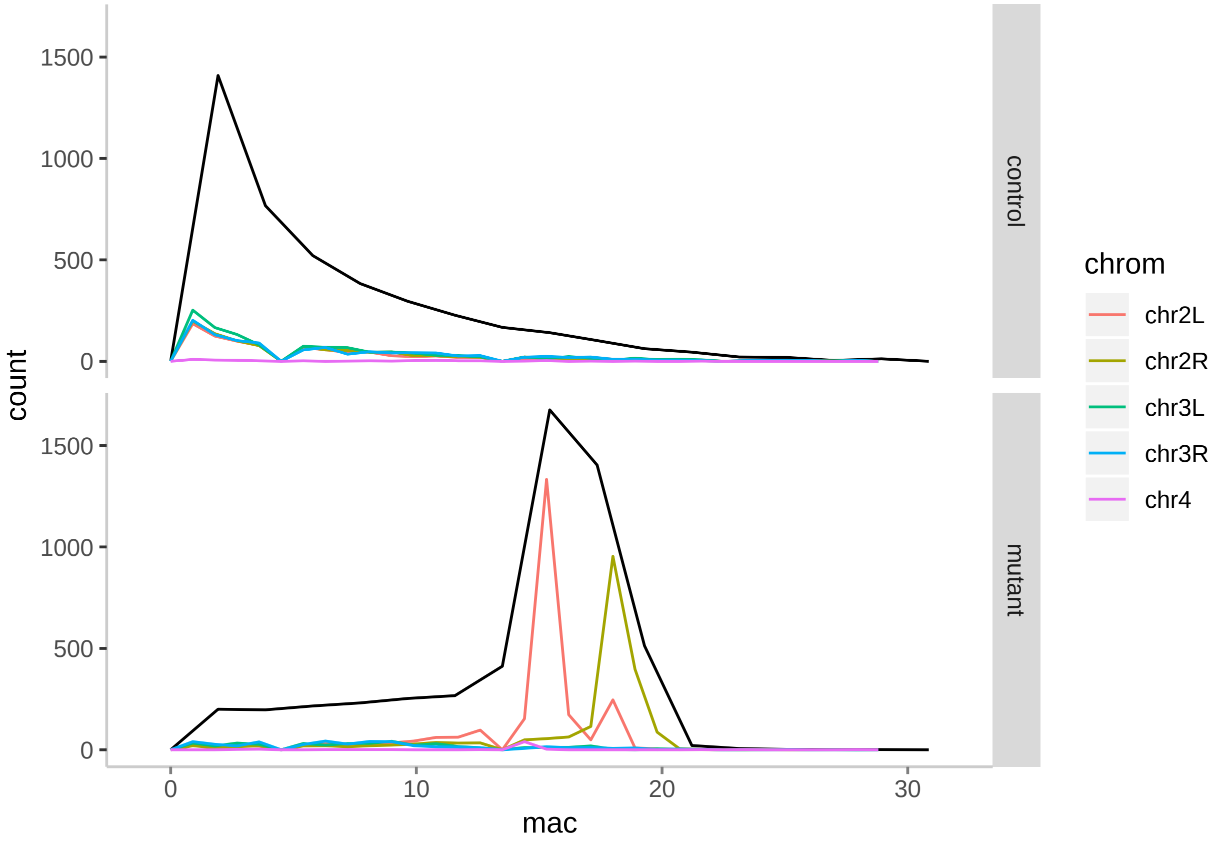
?

Q2: what if the parent allele is A and the mutant allele is R? (weird back-mutation special case) These are r

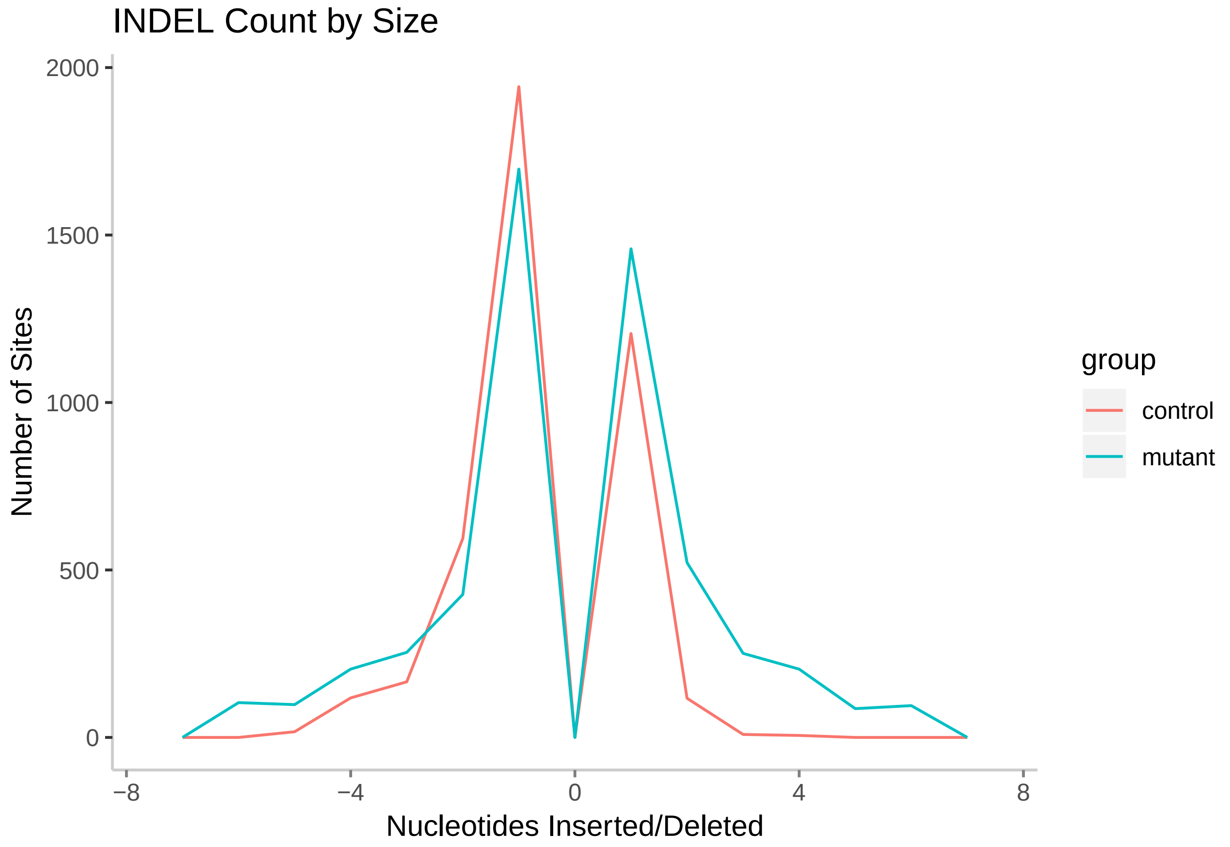
uh, the vcftools –snp/snps option doesn’t seem to be working? I guess just use grep?

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```


4
0
−4
4
0
−4
delta_bp
0
10
20
mac
control
mutant
count
400
300
200
100


1500
1000
500
0
1500
1000
500
0
count
0
10
20
30
mac
control
mutant
chrom
chr2L
chr2R
chr3L
chr3R
chr4

```
biallele.novel.byChrom <- inner_join(biallele.novel.counts %>% group_by(group, chrom) %>% summarise(nove

biallele.novel.byChrom %>%  mutate(percent=human_readable_croncher( 100*novel/total)) %>%  kable(captio
```



```
novel.insert_truth.Tbl <- biallele.novel.counts %>% mutate(ins=delta_bp>0) %>% group_by(ins,group) %>% s
novel.insert_truth.Tbl <- cbind(novel.insert_truth.Tbl, map2(novel.insert_truth.Tbl$del, novel.insert_tr

kable(novel.insert_truth.Tbl, caption = "Deletion bias in mcm5 Mutants  (Hereditarily Novel)")
```

```
## # A tibble: 9 x 3
## # Groups:   group [?]
##   group   chrom novel_singleton
##   <chr>   <fct>           <int>
## 1 control chr2L             185
## 2 control chr2R             198
## 3 control chr3L             252
## 4 control chr3R             202
## 5 control chr4                9
## 6 mutant  chr2L              27
## 7 mutant  chr2R              20
## 8 mutant  chr3L              33
## 9 mutant  chr3R              40
```

```
full_join(biallele.novel.byChrom, biallele.novel.counts.singleton %>% group_by(group, chrom) %>% summari
```

Fortifying the VCF_Novelist rule

```
vcftools --vcf {input.vcf_in}.anc.tmp {par_string} --recode --recode-INFO-all --stdout | grep -v "#" |



vcftools --remove-indv DfMcm5 --remove-indv Mcm5-A7 --vcf variants/mutant.vs_dm6.bwaUniq.alleleCounts.s
```

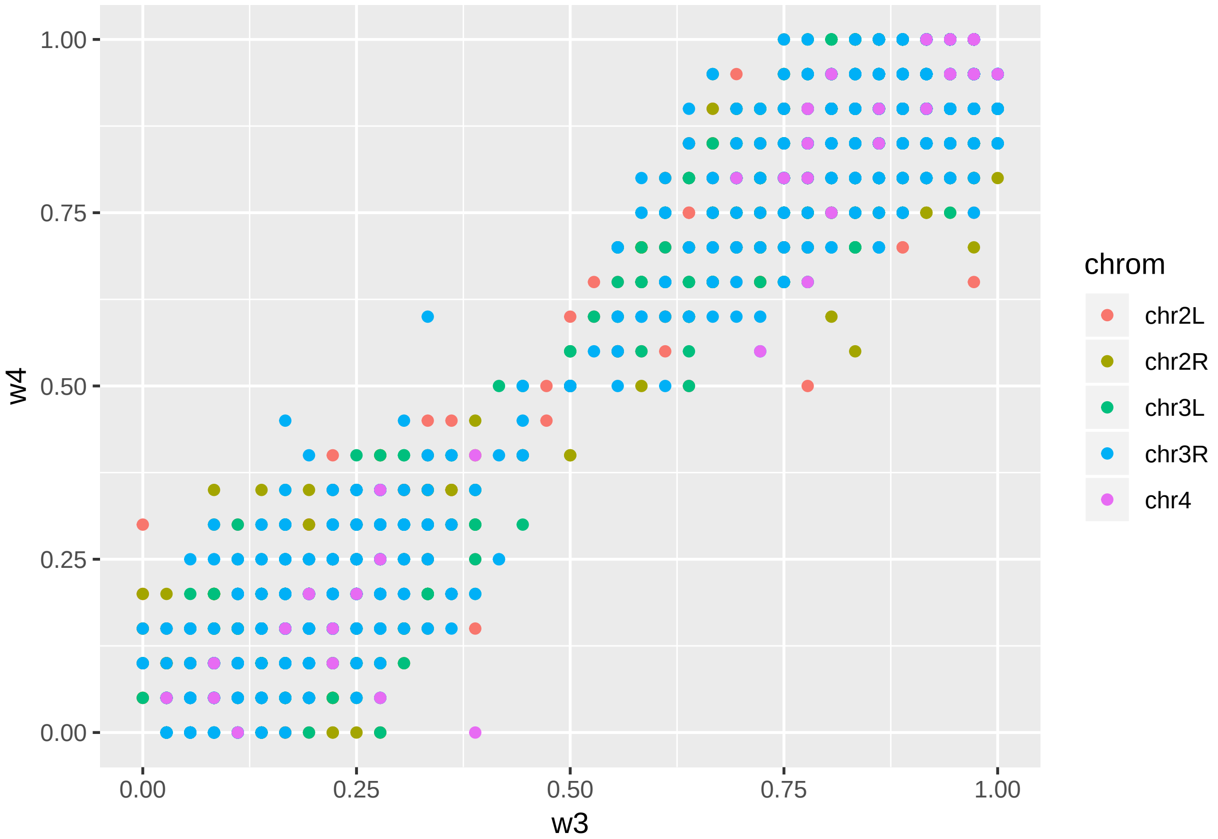
2 April 2019

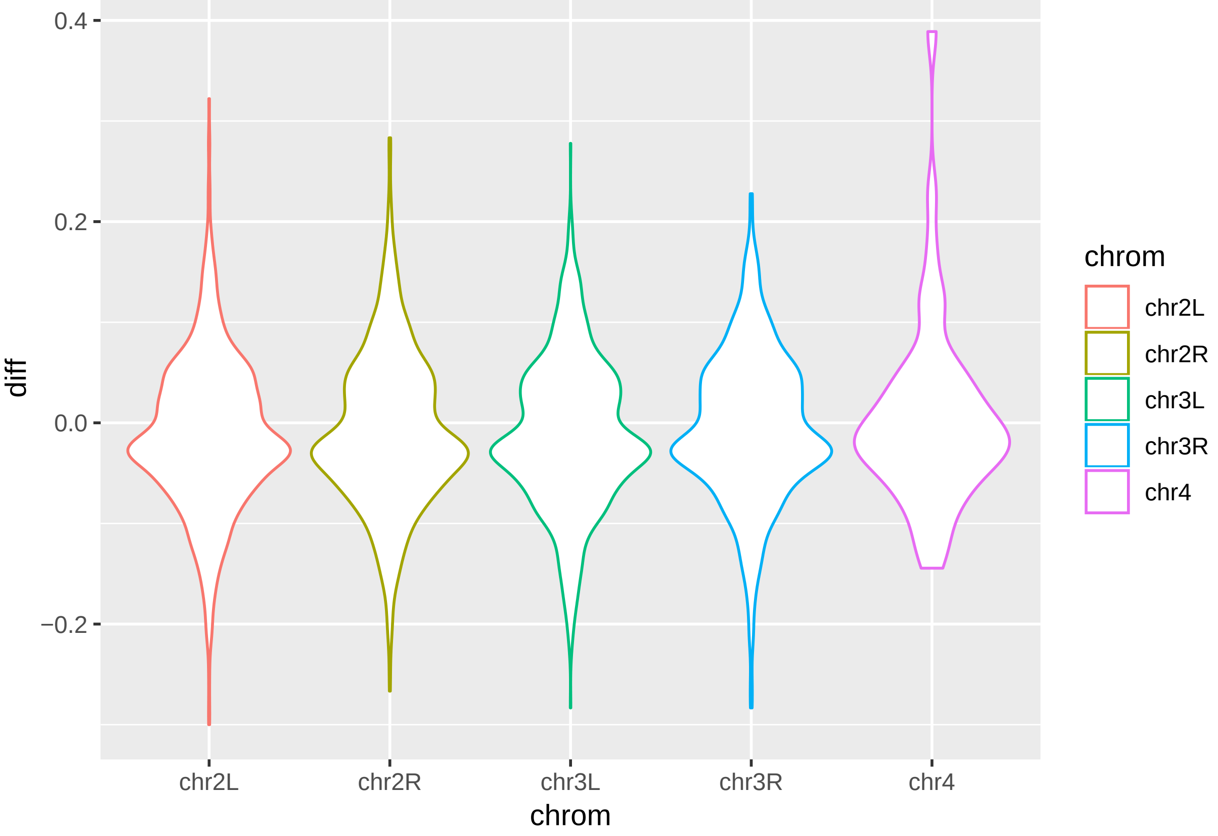
Hmm, looks like adding the rare ref alleles has dramatically changed the results: Many more novel variants and novel-singleton variants in the control now. (?!)

One possibility: there is residual difference between w3, w4, and w1118. Checking this my comparing the frequencies at novel sites: w3 and w4 would be expected to have little overlap if novel sites are due to 3 vs 4 differences

```
vcftools --vcf control.vs_dm6.bwaUniq.alleleCounts.simpleIndels.dpthFilt.biallelic.universal.novel.vcf


vcftools --vcf control.vs_dm6.bwaUniq.alleleCounts.simpleIndels.dpthFilt.biallelic.universal.novel.vcf

paste control.w[34].frq | cut -f 1,2,4 >control.diff.frq
```

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





Hmm, no looks like those AFs are pretty well correllated.

3 April 2019

Okay to compromise let's just split the two into a novel and a backmutation file. . .