20180821_mapping_stats

August 21, 2018

1 Polymarker for WheatCap

After designing the kasp markers with `polymarker-0.9.5` I want to have some general stats about how the markers whent.

The command to design the markers was:

```
#!/bin/bash
#SBATCH --mem=20Gb
#SBATCH -p jic-medium,RG-Cristobal-Uauy,nbi-medium
#SBATCH -J polymarker_WheatCap
#SBATCH -n 1
#SBATCH -o log_scratch4/polymarker_%A_%a.out
#SBATCH --array=0-2217
#SBATCH --time=2-00:00:00

source polymarker-0.9.5
chunks=`ls by_chunks/WheatCap_chunks.*`
read -r -a array <<< $chunks

i=$SLURM_ARRAY_TASK_ID
marker=${array[$i]}
echo $marker

filename=$(basename "$marker")
extension="${filename##*.}"
filename="${filename%.*}"
ref="/usr/users/ga002/ramirezr/Cristobal-Uauy/WGAv1.0/161010_Chinese_Spring_v1.0_pseudomolecules
cmd="polymarker.rb
--contigs $ref
-g 3
-m $marker
-a nrgene
--aligner blast
--max_hits 21
--output  "out_by_chunks_max_hits_21/${i}_${extension}"
echo $cmd
```

```
$cmd
echo "DONE"
```

I then merged the markers with the following command:

```
cat */primers.csv | grep -v "^Marker" > 20180820_primers_wheat_cap_max_hits_21.csv
```

and added the header again (with `vim`) to the merged file.
The analysis bellow has a general description of the primers

```
In [1]: library(ggplot2)
```

```
In [2]: library(sqldf)
```

```
Loading required package: gsubfn
Loading required package: proto
Loading required package: RSQLite
```

```
In [3]: zz=gzfile('20180821_primers_wheat_cap_max_hits_21.csv.gz','rt')
        dat=read.csv(zz,header=T)
```

```
Warning message in read.table(file = file, header = header, sep = sep, quote = quote, :
seek on a gzfile connection returned an internal errorWarning message in read.table(file = file,
seek on a gzfile connection returned an internal error
```

1.1 Histogram of number of hits

While running polymarker, I found some markers that where taking too long to run. I'm looking at the distribution of the markers. Since the markers are comning from coordinates in the reference, all of them are included, so the histogram starts in 1 (For 1 hit).

We have `1,108,355` markers, of those `23,939 (2.16%)` have more tha 30 hits and `32,466 (2.93%)` have more than 21 hits. To reduce the computation neded, markers where only designed for SNPs with 21 or less hits in the genome

```
In [14]: nrow(dat)
```

1108355

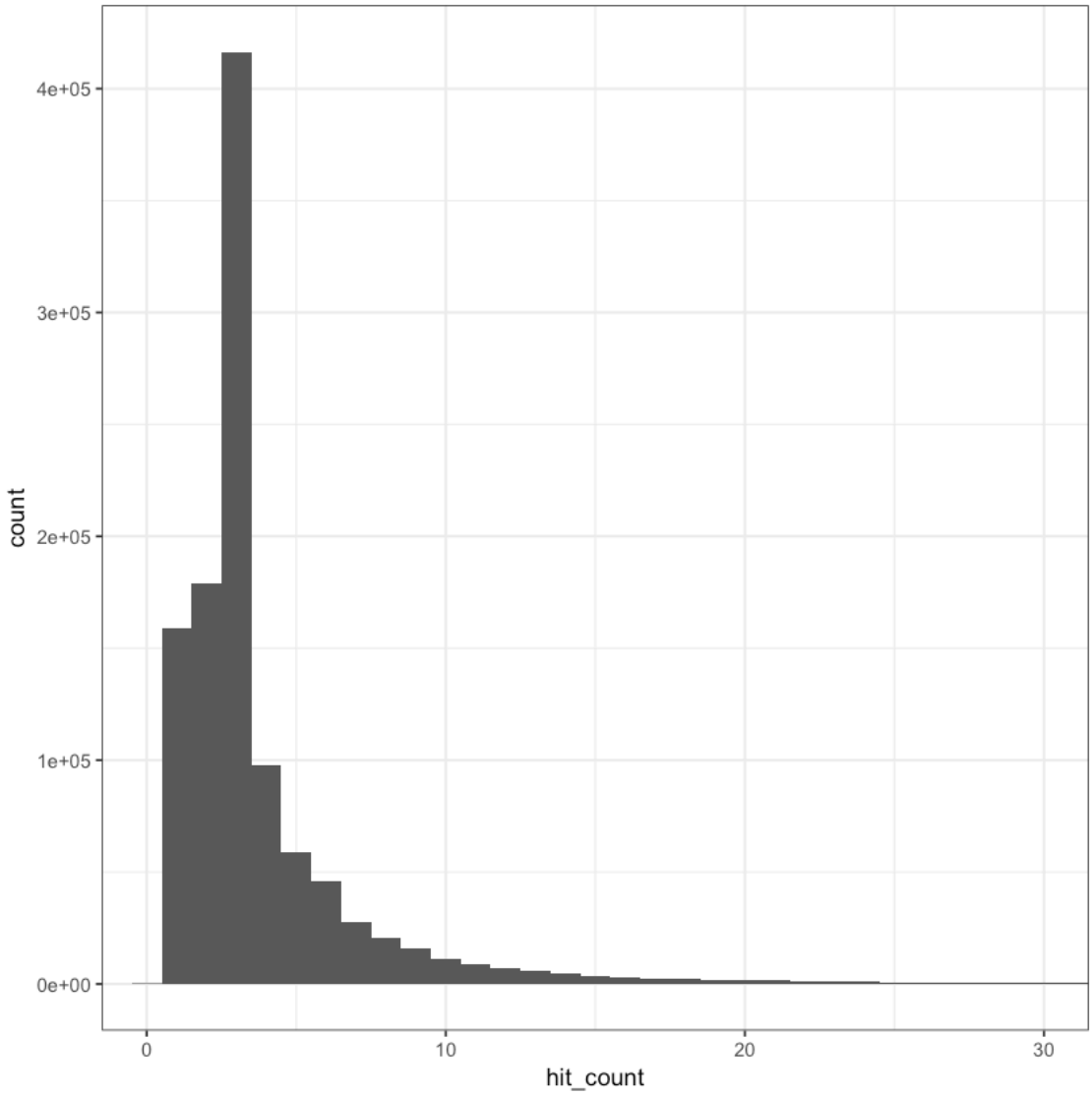
```
In [10]: sqldf("SELECT count(*) as count FROM dat WHERE  hit_count > 30")
```

count
24405

```
In [11]: sqldf("SELECT count(*) as count FROM dat WHERE  hit_count > 21")
```

count
32466

```
In [5]: ggplot(dat, aes(x=hit_count)) +
        geom_histogram(binwidth=1) +
        coord_cartesian(xlim = c(0,30)) +
        theme_bw()
```



1.2 Number of primers on the different categories

Now, we want to see how many of the SNPs are designed on each category. To interpret the CSV this are the columns:

- **Marker** The name of the marker
- **SNP** The position and change in the SNP
- **RegionSize** The size of the aligned sequence

- **chromosome** The target chromosme
- **total_contigs** The total number of chromosomes where the marker hits.
- **contig_regions** The regions where the contig is found
- **SNP_type**
- `homoeologous` when the SNP is a variation you naturally find across genomes.
- `non-homoeologous` are more likely to be varietal SNPs, as the base is consitent across the copies
- **A** Primer for first allele
- **B** Primer for second allele
- **common** The common primer
- **primer_type**
- `chromosome_specific` The marker amplifies only the target chromosome
- `chromosome_semispecific` The marker manages to exclude at least one of the homoeolog chromosomes.
- `chromosome_nonspecific` The marker amplifies all the chromosome
- **orientation** Orientation of the first primer with respect of the reference
- **A_TM** Melting temperature of first primer
- **B_TM** Melting temperature of second primer
- **common_TM** Melting temperature of common primer
- **selected_from** For debuging purposes, the primers are desgined for both alleles, but sometimes only one of them is "stable" according to primer3, so that is the selected primer
- **product_size** Size of the region to amplify
- **errors** Primer 3 and polymaker conditions that prevented the primer to be designed
- **is_repetitive** `true` if the region amplifies more than the `max_hits` variable.
- **hit_count** On how many regions the marker maps. This is the total number of hits, as opposed to the `contig_regions` that only counts the number of chromosomes.

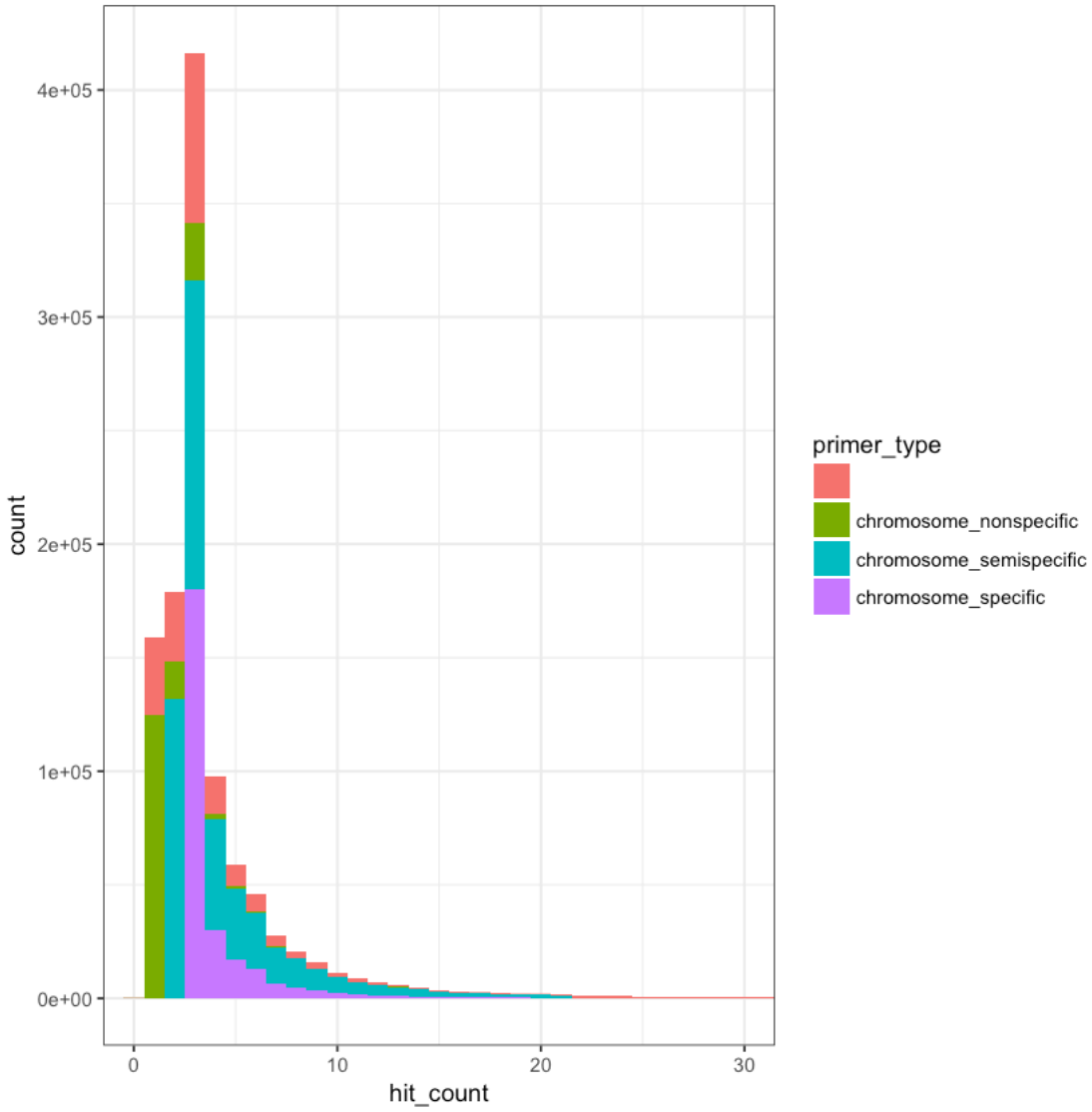
```
In [12]: sqldf("SELECT SNP_type,
         primer_type, count(*) as total,
         100.0 * count(*) / 1108355 as percentage
         FROM dat   GROUP BY SNP_type, primer_type  ")
```

SNP_type	primer_type	total	percentage
		32466	2.92920590
homoeologous		3778	0.34086552
homoeologous	chromosome_nonspecific	616	0.05557786
homoeologous	chromosome_semispecific	10575	0.95411669
homoeologous	chromosome_specific	5202	0.46934421
non-homoeologous		189269	17.07656843
non-homoeologous	chromosome_nonspecific	170574	15.38983448
non-homoeologous	chromosome_semispecific	436538	39.38611726
non-homoeologous	chromosome_specific	259337	23.39836966

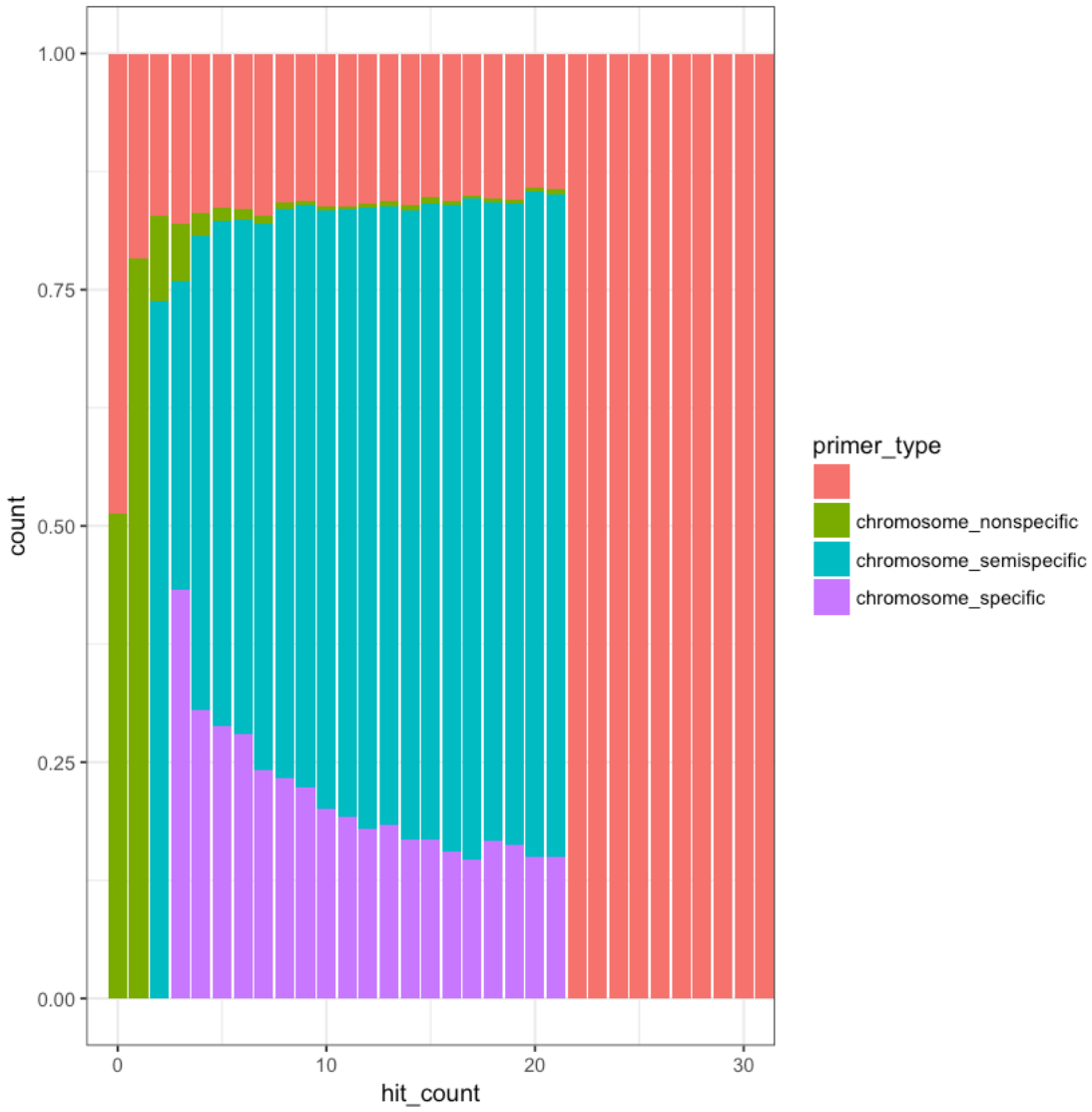
1.3 Primer types across `hit_count`

To explore how the number of hits affect the type of primer, I'm plotting the distribution of different hit counts with their type of primers. I'm also plotting the normalised values as a percentage. The hypothesis is that the more repetitive the region is, less genome-specific primers can be found (as a percentage of the total). The blank label means that the primer was not produced by primer3.

In [13]:
```
ggplot(dat, aes(x=hit_count, fill=primer_type)) +
geom_histogram(binwidth=1) +
coord_cartesian(xlim = c(0,30)) +
theme_bw()
```



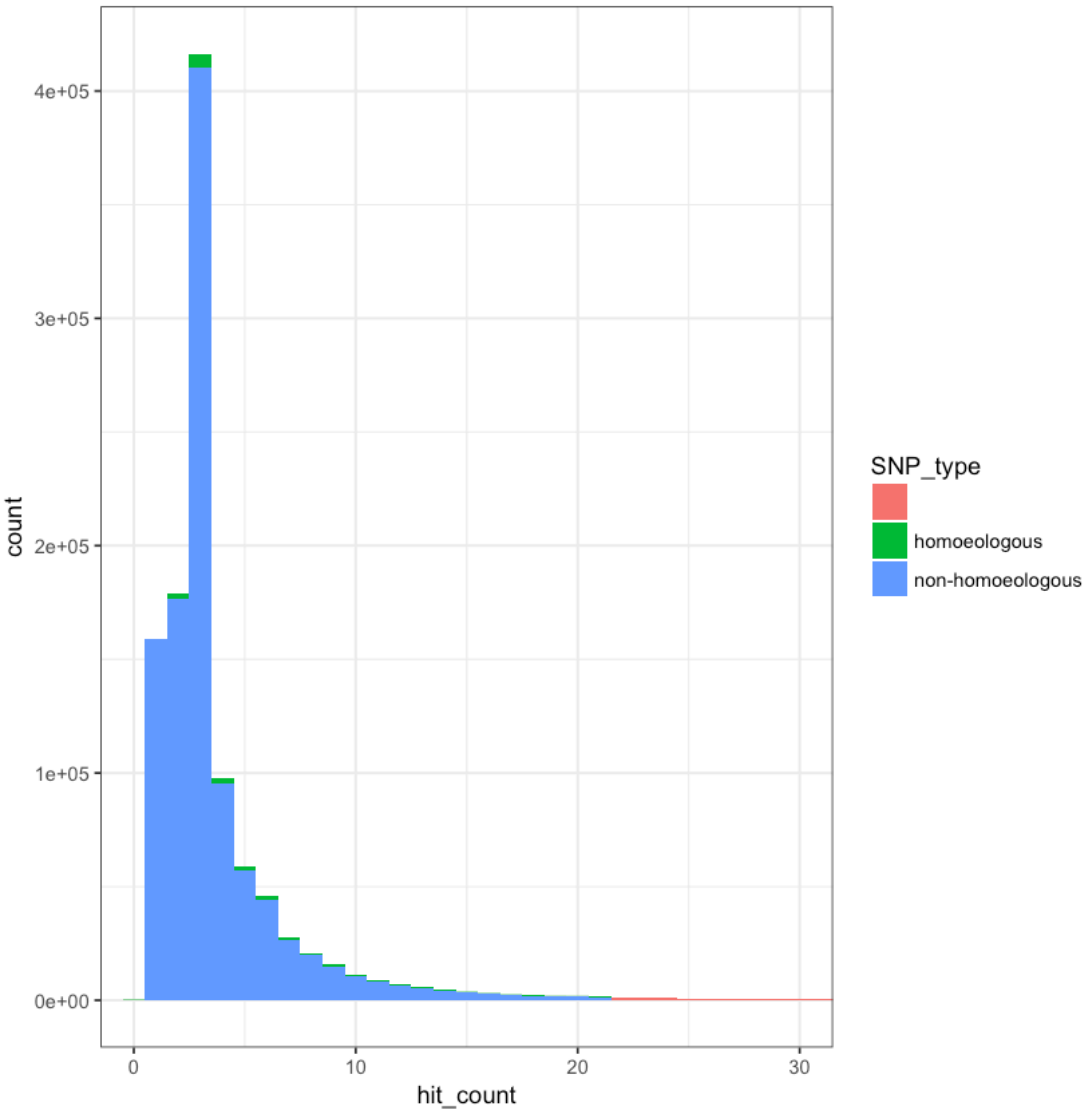
In [7]:
```
ggplot(dat, aes(x = hit_count, fill = primer_type)) +
  geom_bar(position="fill") +
coord_cartesian(xlim = c(0,30)) +
theme_bw()
```



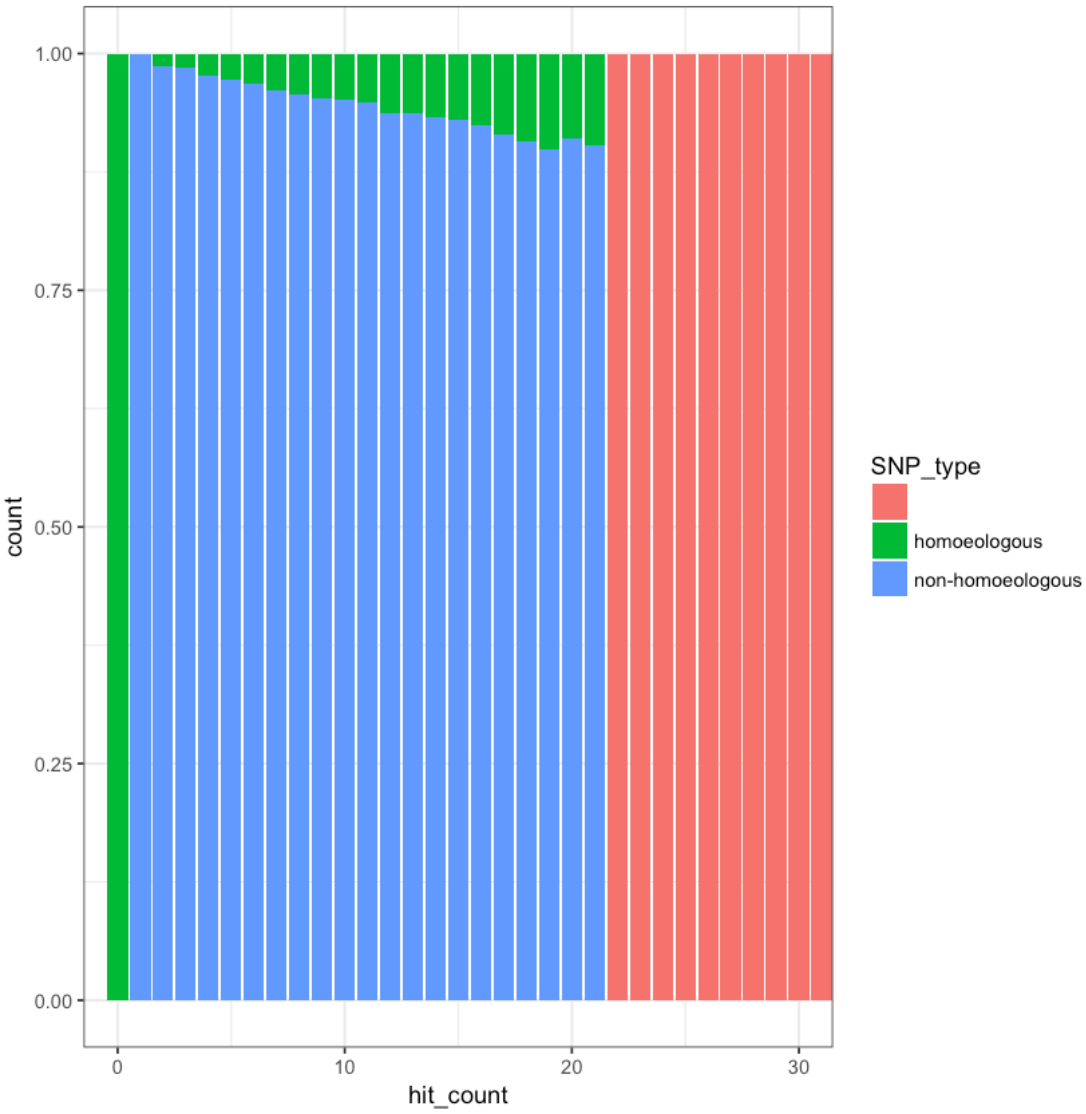
1.4 Varietal vs non varietal SNPs

I was also curious to find if there is a relationship between how repetitve the region is and the likelihood that the SNPs is in reality a homoeologous variation. In this particular dataset, we can see that effect.

```
In [8]: ggplot(dat, aes(x=hit_count, fill=SNP_type)) +
        geom_histogram(binwidth=1) +
        coord_cartesian(xlim = c(0,30)) +
        theme_bw()
```



In [9]:
```
ggplot(dat, aes(x = hit_count, fill = SNP_type)) +
  geom_bar(position="fill") +
coord_cartesian(xlim = c(0,30)) +
theme_bw()
```


1.00
0.75
0.50
0.25
0.00
count
0
10
20
30
hit_count
SNP_type
homoeologous
non-homoeologous